applied sciences

Special Issue Reprint

AI Empowered Sentiment Analysis

Edited by
Xiangjie Kong, Wei Wang and Han Liu

mdpi.com/journal/applsci

MDPI

AI Empowered Sentiment Analysis

AI Empowered Sentiment Analysis

Editors

Xiangjie Kong
Wei Wang
Han Liu

MDPI

Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Editors

Xiangjie Kong
College of Computer Science and Technology
Zhejiang University of Technology
Hangzhou
China

Wei Wang
Artificial Intelligence Research Institute
Shenzhen MSU-BIT University
Shenzhen
China

Han Liu
School of Software
Dalian University of Technology
Dalian
China

Editorial Office
MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of articles from the Special Issue published online in the open access journal *Applied Sciences* (ISSN 2076-3417) (available at: https://www.mdpi.com/journal/applsci/special_issues/1ZF98Q5X6B).

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, A.A.; Lastname, B.B. Article Title. *Journal Name* **Year**, *Volume Number*, Page Range.

ISBN 978-3-7258-1823-5 (Hbk)
ISBN 978-3-7258-1824-2 (PDF)
doi.org/10.3390/books978-3-7258-1824-2

Contents

About the Editors

Xiangjie Kong

Xiangjie Kong is currently a Full Professor and the Associate Dean in the College of Computer Science and Technology, Zhejiang University of Technology (ZJUT), China. He received the B.Sc. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2004 and 2009 respectively. Previously, he was an Associate Professor in the School of Software, Dalian University of Technology (DUT), China, where he was the Head of the Department of Cyber Engineering. He is the Founding Director of City Science of Social Computing Lab (The CSSC Lab). He has been on the Editorial Boards of six international journals. He has served as the General Chair or Program Chair of more than 10 conferences. Dr. Kong has also authored/co-authored over 200 scientific papers in international journals and conferences, including *IEEE TKDE*, *IJCAI*, *ACL*, *IEEE TMC*, *ACM CSUR*, *ACM TKDD*, *IEEE TNSE*, *IEEE TII*, *IEEE TITS*, *IEEE NETW*, *IEEE COMMUN MAG*, *IEEE TVT*, *IEEE IOJ*, *IEEE TSMC*, *IEEE TETC*, *IEEE TASE*, *IEEE TCSS*, *ACM TSON*, *ACM TSAS*, *WWWJ*, etc. His research interests include big data, network science, and computational social science. He is a Distinguished Member of CCF, a Senior Member of IEEE, a Full Member of Sigma Xi, and a Member of ACM.

Wei Wang

Wang Wei is currently a Professor with the Artificial Intelligence Research Institute, Shenzhen MSU-BIT University, Shenzhen, China, and also with the School of Medical Technology, Beijing Institute of Technology, Beijing, China. He has been a UM Macao Research Fellow at the University of Macau, Macau, SAR. He received his PhD degree in software engineering from Dalian University of Technology in 2018. His research interests include computational social science, data mining, the internet of things, and artificial intelligence.

Han Liu

Han Liu is currently a tenured associate professor at Dalian University of Technology. He received his B.Eng. degree in software engineering and his Ph.D. degree in computer application technology from Dalian University of Technology in 2012 and 2018, respectively. Before he joined Dalian University of Technology, he worked as a postdoctoral fellow in the Department of Computing at The Hong Kong Polytechnic University. He has published over 60 papers in top journals and conferences, including *TKDE*, *TKDD*, *NeurIPS*, *KDD*, *AAAI*, *IJCAI*, *SIGIR*, *WWW*, *CVPR*, *ACL*, *EMNLP*, etc. He is a PC member or regular reviewer of *NeurIPS*, *ICLR*, *ICML*, *KDD*, *AAAI*, *IJCAI*, *ACL*, *EMNLP*, *CVPR*, *ICCV*, *ECCV*, *MM*, *TPAMI*, *AI*, *TNNLS*, *TOIT*, *TITS*, *PR*, *INS*, *NN*, etc. He is a member of IEEE, ACM, and CCF.

applied sciences

MDPI

Article

Generative Aspect Sentiment Quad Prediction with Self-Inference Template

Yashi Qin and Shu Lv *

School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, China; qinyashi1215@163.com
* Correspondence: lvshu@uestc.edu.cn

Abstract: Aspect Sentiment Quad Prediction is a research topic of paramount significance and complexity within the Aspect-Based Sentiment Analysis task. Leveraging the generative paradigm of the T5 model, we achieve end-to-end extraction of aspect sentiment elements by paraphrasing the original text into sentences predefined by templates. Current research predominantly confines templates to single sentences or directly concatenates sentiment elements using a few symbols, limiting the model's reasoning opportunities. In this work, we introduce a Self-Inference Template (SIT) to guide the model in thoughtful reasoning, facilitating a step-by-step inference generation process. This approach enables the model to more accurately identify aspect sentiment elements and their interdependencies. Experimental results demonstrate a significant improvement in quadruplet prediction performance under constant time costs, effectively mitigating overfitting issues caused by limited data volume to some extent.

Keywords: aspect-based sentiment analysis; aspect sentiment quad prediction; aspect-category-opinion-sentiment; chain of thought; prompt

Citation: Qin, Y.; Lv, S. Generative Aspect Sentiment Quad Prediction with Self-Inference Template. *Appl. Sci.* **2024**, *14*, 6017. https://doi.org/10.3390/app14146017

Academic Editor: Rui Araújo

Received: 27 February 2024
Revised: 28 June 2024
Accepted: 9 July 2024
Published: 10 July 2024

1. Introduction

The research on Aspect-Based Sentiment Analysis (ABSA) mainly involves four sentiment elements: Aspect Term, Aspect Category, Opinion Term, and Sentiment Polarity. The study of ABSA tasks aims to identify sentiment elements related to specific text items, which can be individual elements such as aspect term extraction [1,2], aspect category detection [3,4], or multiple dependent sentiment elements like aspect-opinion pair extraction [5,6], aspect sentiment triplet extraction [7], aspect-category-sentiment detection [8], etc. Clearly, the more sentiment elements identified, the better the understanding of aspect-level opinions in the text. In 2021, Cai [9] first proposed the Aspect-Category-Opinion-Sentiment(ACOS) quadruple extraction task, which includes implicit aspect and opinion elements. In the same year, Zhang [10] introduced the Aspect Sentiment Quad Prediction (ASQP) task, excluding implicit opinions. Thus, the task of aspect sentiment quadruple extraction officially emerged.

Zhang [10] also proposed a novel modeling paradigm based on the T5 generative model. This paradigm involves paraphrasing the original sentence into the form "x_{ac} is x_{sp} because x_{at} is x_{ot}", making it easy to extract quadruplets from the paraphrased sentences. In this context, x_{ac} represents the aspect category, x_{sp} represents the sentiment polarity, x_{at} represents the aspect term, and x_{ot} represents the opinion term. Subsequently, numerous studies in aspect-level sentiment analysis based on the generative paradigm emerged, some of which explored template settings. Hu [11] investigated the impact of the order of sentiment tuples in templates on aspect sentiment quadruplet prediction. They simplified templates by directly connecting symbols and elements, such as "[AC] x_{ac} [AT] x_{at} [SP] x_{sp} [OT] x_{ot}". Joseph [12] also redefined model templates, setting them as "x_{ac} | the x_{at} is x_{ot} | x_{sp}". These templates, through different orders and forms of placing sentiment

elements, attempted to improve the prediction of aspect sentiment quadruplets. However, these templates were all concise sentences, expecting the model to directly provide a sentence containing the quadruplet. We consider the possibility of allowing the model to think slowly to provide answers. This is because there are complex dependencies among various sentiment elements in some sentences. For example, the aspect category is not only determined by the aspect term but also related to the opinion term. In the two examples in Figure 1, although the aspect terms are both "sandwiches", the aspect categories are different. Therefore, if we could design a template to assist the model in reasoning and analyzing the relationships among various sentiment elements, it might be beneficial for aspect sentiment quadruple prediction.

Example-1

Sentence: Also , the sandwiches did n't come with anything like chips or a side .

Gold Label: (sandwiches, food style_options, negative, did n't come with)

Example-2

Sentence: The sandwiches are dry , tasteless and way overpriced .

Gold Label: (sandwiches, food quality, negative, dry), (sandwiches, food quality, negative, tasteless), (sandwiches, food prices, negative, overpriced)

Figure 1. Example of Aspect Sentiment Quad Prediction.

Inspired by the chain-of-thought approach proposed by Jason [13], we incorporated intermediate reasoning steps into our templates. This inclusion guides the model to progressively reason through the generation process based on our template, step by step inferring each sentiment element. We term this approach the Self-Inference Template (SIT). Simultaneously, during this process, there might be a repetitive generation of aspect terms, aspect categories, opinion terms, and sentiment polarities. To address this, we conduct a voting mechanism on the repetitively generated sentiment elements to obtain the final quadruplet. This approach helps ensure the correctness of predictions to some extent.

We transform the gold labels into a Self-Inference Template form, denoted as y, and the original text denoted as x, is input into the model for supervised training, resulting in a $p_\theta(y|x)$ model. To train the θ parameters effectively, a large amount of supervised data is usually required. However, due to the complexity and high cost of ABSA data annotation, the commonly used ABSA datasets are relatively small. The rise of prompts can help models learn in few-shot or even zero-shot scenarios [14]. Therefore, we add prefix prompts to the text data to assist in model training.

Our goal is to extract the required sentiment elements from sentences, similar to entity recognition, requiring the model to have a deeper understanding of the text. Some current studies have found that using noisy text during model training can effectively improve model performance. For instance, focusing the noise on entities within the sentence can result in particularly high predictive performance for entities [14]. Currently, noise includes four types: Masking, Replacement, Deletion, and Permutation. BERT [15] employs Masking and Replacement to process training texts. To encourage the model to understand the text, we applied MASK processing to a small number of tokens in the text, forcing the model to comprehend the text better, thereby improving the identification of sentiment elements.

In summary, we made three improvements to the model: first, introducing a Self-Inference Template to guide the model to think and reason step by step; second, adding prompt prefixes to the text data to help the model quickly adapt to the data; third, implementing a MASK strategy within the text to force the model to deeply understand the text, aiding in the identification of sentiment elements.

The experiments demonstrate that the optimal model, combining the self-inferencing template with two additional methods, outperforms the Paraphrase model. Specifically, on the ASQP datasets Rest15 and Rest16, there is an improvement of 3.07% and 4.06%,

respectively. In the case of ACOS datasets for Restaurant and Laptop, the improvement is 3.32% and 1.45%, respectively.

In summary, our work contributes in the following three aspects:

- We designed a Self-Inference Template that guides the model in step-by-step reasoning and significantly improves the results of aspect sentiment quadruplet prediction. To our knowledge, this work is the first to approach aspect sentiment quadruplet prediction from the perspective of encouraging the model to contemplate and reason gradually.
- We created prompt texts based on the training tasks to help the model train on small datasets. Experiments on both Paraphrase and SIT models demonstrated the effectiveness of prompts.
- We boldly experimented with applying MASK operations to ABSA text data to help the model effectively identify sentiment elements, providing more possibilities for future research on ABSA tasks.

2. Related Work

The Chain of Thought (CoT) is a prompting method that significantly enhances the capabilities of large language models in complex reasoning tasks [13]. It achieves this by presenting a small number of examples to the model, explaining the reasoning process in these examples, and guiding the model to generate intermediate reasoning steps. The introduction of the Chain of Thought has led to substantial progress in large language models. Scholars have also applied the chain-of-thought approach to sentiment analysis. Fei [16] utilized the CoT framework to simulate human-like reasoning processes in implicit sentiment analysis, step-by-step extracting implicit aspects, opinions, and sentiment polarity, achieving outstanding results in implicit sentiment analysis.

Currently, the main modeling paradigms for ABSA tasks are Sequence-level Classification (SeqClass), Token-level Classification (TokenClass), Machine Reading Comprehension (MRC), and Sequence-to-Sequence Modeling (Seq2Seq) [17]. SeqClass and TokenClass paradigms are mostly used for single ABSA tasks and cannot meet the current demand for extracting multiple sentiment elements. The MRC paradigm extracts sentiment elements by constructing relevant questions, with the model predicting the start position of words in the original text. This method requires the extracted elements to appear in the original text, making it ineffective for texts containing implicit aspects or implicit opinions. In contrast, the generative paradigm of Seq2Seq can be widely applied to various ABSA tasks, offering high flexibility and providing a unified framework for ABSA task modeling. Zhang [10] transformed the ASQP task into a paraphrase generation process, demonstrating for the first time the excellent capabilities of generative paradigms in handling ABSA tasks. Joseph [12] using a generative model combined with contrastive learning, achieved optimal performance in quadruplet extraction on ACOS datasets containing implicit language. This approach also significantly improved the extraction of implicit aspects and opinions. This indicates that the generative paradigm has the potential for datasets containing implicit terms and requiring strong reasoning abilities.

Inspired by the generative paradigm and the Chain of Thought approach, we propose a Self-Inference Template based on generative aspect sentiment quadruplet prediction. By guiding the model to generate the reasoning process for aspect sentiment elements, our approach helps the model better comprehend the text and improves the results of aspect sentiment quadruplet prediction.

3. Methodology

3.1. Aspect Sentiment Quad Prediction Based on the Generative Paradigm

Aspect sentiment quadruplet prediction aims to predict all aspect terms (AT), aspect categories (AC), opinion terms (OT), and sentiment polarities (SP) within a given sentence x. Aspect terms and opinion terms are generally words present in the sentence, but sometimes aspect terms and opinion terms may be implicitly represented in the sentence, denoted as

"NULL" in such cases. Aspect categories belong to a predefined set V_c. A sentence may contain multiple quadruplets.

Currently, aspect sentiment quadruplet prediction based on the generative paradigm involves arranging the quadruplets in the dataset into a template format to create targets. The text and targets are then fed into a Sequence-to-Sequence model for fine-tuning training. Finally, the trained model generates targets, which are split into quadruplets based on the template format. The key component in achieving this task is the learning process of the Sequence-to-Sequence model. This involves learning parameters θ, maximizing the probability $p_\theta(y|x)$, where x is the original sentence, and y is the target sentence to be obtained. Since the target sentence is generated token by token, the $i-th$ token of y is determined by x and the preceding $i-1$ tokens of y.

$$p_\theta(y_i|x, y_1, \ldots, y_{i-1}) = softmax(W^T y_{i-1}) \tag{1}$$

In the process, W maps y_{i-1} to a vector of vocabulary size and subsequently utilizes the softmax function to determine which word from the vocabulary the model should choose as the next token.

During training, we chose the T5 model [18] to initialize the parameters. The T5 model, proposed by Google in 2020, is a pre-trained model designed to handle various text tasks through a unified framework. It converts all tasks into text-to-text problems and completes different tasks by adding different prefix prompts to the text, such as translation and summarization. T5 follows the standard Transformer encoder-decoder structure. We initialized the model parameters with T5-base and input the ABSA text data into the model. The data are converted into a sequence of word vectors through the word embedding layer and then passed into the Transformer encoder. The encoder transforms it into high-dimensional hidden representations. The decoder combines the encoder's output and the previously generated text to autoregressively generate new words step by step. As shown in Figure 2.

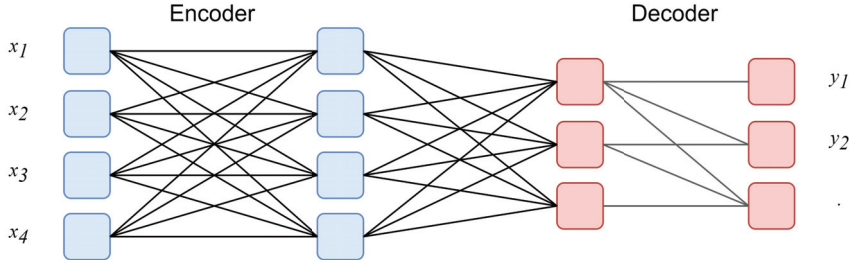


Figure 2. The architecture of the T5 model [18].

During training, T5 uses cross-entropy loss to measure the difference between the generated text and our Self-Inference Template target, updating the model parameters accordingly.

$$L(x, y) = -\sum_{i=1}^{n} log p_\theta(y_i|x, y_1, \ldots, y_{i-1}) \tag{2}$$

where n represents the length of the target sequence y.

3.2. Self-Inference Template

CoT provides specific thought processes in prompts, allowing large models to learn the way of thinking provided in the thought chain. The model then follows the thought chain, step by step, to enhance its reasoning abilities. However, in the generative model employed in this paper, we use the T5-base model, which has fewer parameters compared to large models. It is not suitable to train the model through examples.

Therefore, we directly formulate the intermediate reasoning process into the form of a template, as illustrated in Figure 3. This approach guides the model to reason step by

step according to the template's thought process. The first half of the template initially obtains the aspect term and opinion term, then infers the sentiment polarity based on the opinion term. In the second half of the template, the aspect category is deduced based on the obtained aspect term and opinion term. Finally, the aspect category and sentiment polarity are generated again to confirm the correctness of generating sentiment elements.

$\{at_1\}$ is $\{sp_1\}$ because $\{at_2\}$ is $\{ot_1\}$, according to $\{at_3\}$ and $\{ot_2\}$ this aspect belongs to $\{ac_1\}$, So $\{ac_2\}$ is $\{sp_2\}$

Figure 3. Self-Inference Template.

In the template, the aspect term is repeated three times, while the aspect category, opinion term, and sentiment polarity are each generated twice. Each output result may vary, so a numerical annotation is added in the lower right corner of each sentiment element to facilitate distinction. Leveraging CoT's self-consistency [19], a voting aggregation is applied to the repetitively generated sentiment elements, ultimately resulting in aspect sentiment quadruplets. The specific model structure is illustrated in Figure 4.

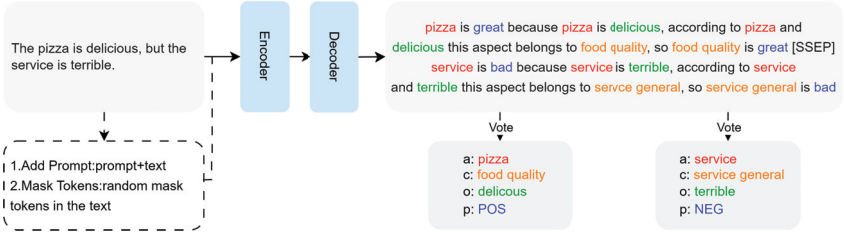


Figure 4. Model Architecture.

3.3. Addition Prompt

In recent years, prompts have been widely employed in language model processing. Research indicates that by selecting appropriate prompts, the model's behavior can be manipulated, enabling the language model to predict the desired outputs without additional training [20]. Our chosen T5 model [18] also supports prompt addition, aiding in model training. Therefore, we experimented with adding prompt prefixes to the text, specifying the task for the model. Experimental results demonstrate that prompts effectively assist the model in improving aspect sentiment quadruplet prediction capabilities.

3.4. Mask Tokens

Bert [15] utilizes a random token masking strategy to force the model to understand the text, enhancing the model's error correction ability and overall accuracy.

To deepen the model's understanding of the text and improve its ability to recognize sentiment elements, we applied a masking strategy to the data. We masked 10% of the text in the dataset. For the sentences to be masked, we randomly selected 10% of the tokens. Among these, 80% of the tokens were replaced with [mask], while the remaining 20% were randomly replaced with a word from the vocabulary. Experimental results indicate that the combined use of masking and prompt addition effectively aids the model in predicting quadruplets.

4. Experimental Setup

4.1. Dataset

To understand the performance of our model on different datasets, we conducted experiments on two main types of datasets, primarily focusing on explicit terms and datasets containing implicit opinions. The first type consists of the ASQP dataset curated by Zhang [10], including Rest15 and Rest16. This type of dataset does not include implicit opinion terms. The second type is the ACOS dataset proposed by cai [9], including

ACOS_Restaurant and ACOS_Laptop. In this type of dataset, over 33% of the sentiment quadruples contain implicit opinions or aspect terms, placing higher demands on the model's inference capability. The specific statistics for the four datasets are provided in Table 1. The proportions of explicit and implicit terms are illustrated in Figure 5.

Table 1. Data statistics. #C, #S, #+, #0, and #− denote the number of aspect categories, the number of sentences, the number of positive, neutral and negative quads, respectively.

		Train	Dev	Test
Rest15	#C	13	12	12
	#S	834	209	537
	#+	1005	252	453
	#0	34	14	37
	#−	315	81	305
Rest16	#C	12	13	12
	#S	1264	316	544
	#+	1369	341	583
	#0	62	23	40
	#−	558	143	176
ACOS_Restaurant	#C	12	13	12
	#S	1530	171	583
	#+	1656	180	667
	#0	95	12	44
	#−	733	69	205
ACOS_Laptop	#C	114	71	81
	#S	2934	326	816
	#+	2583	279	716
	#0	227	24	65
	#−	1362	137	380

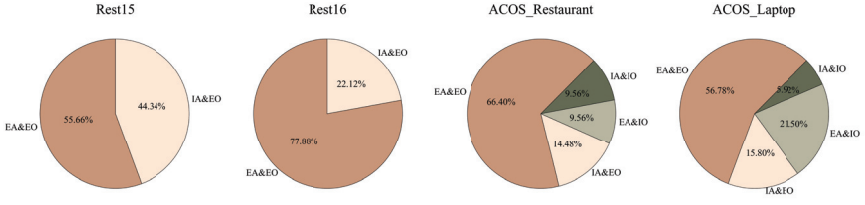


Figure 5. Distribution of Explicit and Implicit Terms in the Dataset. EA represents explicit aspect terms, EO represents explicit opinion terms, IA represents implicit aspect terms, and IO represents implicit opinion terms. The proportions are illustrated for each category.

For ACOS_Restaurant and ACOS_Laptop, the aspect categories are in a form similar to LAPTOP#GENERAL, which may be challenging for generative models to comprehend semantically. Inspired by Joseph [12], we replaced the aspect categories in the ACOS dataset with human-readable forms. For example, LAPTOP#GENERAL was replaced with "the laptop overall" to facilitate the model's understanding and sentence rewriting.

4.2. Experiment Details

We opted for the T5-base [18] as the pre-trained generative model, with a training batch size set to 16, a learning rate of 3×10^{-4}, and a fixed random seed of 42 to eliminate experimental bias caused by random factors. All experiments were conducted for 20 training epochs, and during the inference process, we employed greedy decoding to generate output sequences. Our experiments were performed on an Nvidia 4080 GPU.

4.3. Baselines

To assess the effectiveness of our approach compared to previous methods, we selected several strong baseline methods:

HGCN-BERT+BERT-Linear HGCN [21] jointly extracts aspect categories and sentiment polarities, utilizes BERT to extract corresponding aspect terms and opinion terms [22], and applies a linear layer for final aggregation.
HGCN-BERT+BERT-TFM Modification of the above model with the final linear layer replaced by Transformer blocks (BERT-TFM).
TASO-BERT-Linear TAS [8], originally designed for extracting unified triples of aspect categories, aspect terms, and sentiment polarities, is extended to TASO for handling ASQP tasks. Linear classification layers are used for prediction.
TASO-BERT-CRF A variant of the TASO model with a Conditional Random Field layer in the prediction stage.
TAS-BERT-ACOS On the basis of the TAS method, cai [9] designed a two-step pipeline approach that incorporates BERT to extract quadruples from ACOS data.
Extract-Classify-ACOS This method first extracts aspect terms and opinion terms from the original sentence and then classifies aspect categories and sentiment polarities based on these extracted terms [9].
GAS A generative baseline [23], modified by [10] to directly generate aspect sentiment quadruplets as the target sequence in the generative model.
Seq2Path Transforming the generation order of sentiments into the path of a tree, using a constrained beam search, automatically selecting valid paths with the help of additional tokens [24].
PARAPHRASE This method extracts (at, ac, sp, ot) by paraphrasing the original sentence as "ac is sp because at is ot" [10].
DLO Considering the impact of the order of generating each element in the quadruplet in generative models [11], 24 template orders were experimented with. The final template order was chosen based on the overall quadruplet extraction performance on the dataset.
ILO Similar to DLO, after experimenting with 24 template orders, the template order for each instance was chosen individually based on its own performance.

5. Results and Discussion

5.1. Main Results

The experimental results for various methods are reported in Table 2. For the ASQP dataset that does not contain implicit opinions, our model significantly improves various metrics compared to the Paraphrase method. The F1 scores for Rest15 and Rest16 are increased by 2.05% and 2.33%, respectively. In comparison to DLO and ILO methods, our Self-Inference Template slightly lags behind ILO on Rest15 but outperforms DLO and ILO on Rest16 without increasing the time cost. After adding prefix prompts, the model achieves the best results on Rest16. Combining prefix prompts and Mask operations on the smaller dataset Rest15 leads to a substantial improvement in model performance. With the assistance of these two methods, the model achieves optimal results on both datasets, with improvements of 1.02% and 1.73% compared to the Self-Inference Template.

For the ACOS dataset containing implicit opinion terms, the Self-Inference Template, compared to the Paraphrase method, showed a 2.94% improvement in F1 score on ACOS_Restaurant. This indicates that the Self-Inference Template, by guiding the model to think step by step, indeed enhances the model's reasoning ability. However, for the ACOS_Laptop dataset, the improvement in the Self-Inference Template was marginal. This could be attributed to the excessive number of aspect categories in ACOS_Laptop, coupled with imbalanced data distribution among different aspect categories. The training set of ACOS_Laptop comprises a total of 114 aspect categories, with only 10 categories appearing in the tuples more than 100 times, and over half of the aspect categories appearing in tuples fewer than 10 times. The model struggles to adequately learn from each aspect category's data. Despite the guidance provided by the Self-Inference Template for thoughtful reason-

ing, the model faces challenges in correctly classifying aspect categories with numerous classes and limited training examples. However, with the addition of prefix prompts and Mask operations, the F1 score for ACOS_Laptop increased by 1.44%. This indicates that our two methods effectively assist the model in learning.

Table 2. Evaluation results compared with baseline methods in terms of precision (Pre, %), recall (Rec, %) and F1 score (F1, %). PT stands for the Add Prompt method, MT stands for the Mask Tokens method, and PM represents the combination of both the Add Prompt and Mask Tokens methods. The best scores are marked in bold. The prefix prompts are the optimal prompts for each dataset in Section 5.2. For Rest15 and Rest16, the experimental results of the baseline methods, $*$ are from [10], and $\star$ are from [11]. For ACOS_Restaurant and ACOS_Laptop, the experimental results of the baseline methods, ♠ are from [9], ▲ are from [25], and ♣ are from [24]. ▼ indicates the reproduction of the official method on our dataset.

Methods	Rest15			Rest16			ACOS_Restaurant			ACOS_Laptop		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
HGCN-BERT+BERT-Linear *	24.43	20.25	22.15	25.36	24.03	24.68	-	-	-	-	-	-
HGCN-BERT+BERT-TFM *	25.55	22.01	23.65	27.40	26.41	26.90	-	-	-	-	-	-
TASO-BERT-Linear *	41.86	26.50	32.46	49.73	40.70	44.77	-	-	-	-	-	-
TASO-BERT-CRF *	44.24	28.66	34.78	48.65	39.68	43.71	-	-	-	-	-	-
TAS-BERT-ACOS ♠	-	-	-	-	-	-	26.29	46.29	33.53	47.15	19.22	27.31
Extract-Classify-ACOS ⋆♠	35.64	37.25	36.42	38.40	50.93	43.77	38.54	52.96	44.61	45.56	29.48	35.80
GAS *▲	45.31	46.70	45.98	54.54	57.62	56.04	53.57	54.34	53.95	40.70	40.17	40.43
Seq2Path ♣	-	-	-	-	-	-	62.38	55.02	58.47	41.46	41.00	41.23
Paraphrase *▼	46.16	47.72	46.93	56.63	59.30	57.93	61.02	59.73	60.37	44.87	44.10	44.48
DLO ⋆	47.07	49.33	48.18	57.92	61.80	59.79	-	-	-	-	-	-
ILO ⋆	47.78	50.38	49.05	57.58	61.17	59.32	-	-	-	-	-	-
SIT	47.89	50.13	48.98	58.98	61.60	60.26	63.13	63.49	63.31	44.38	44.61	44.49
SIT+PT	48.41	49.75	49.07	**60.78**	**63.24**	**61.99**	**63.54**	**63.83**	**63.69**	43.12	42.78	42.95
SIT+MT	47.93	49.50	48.70	58.30	60.96	59.60	61.79	63.27	62.52	44.46	44.35	44.41
SIT+PM	**49.63**	**50.38**	**50.00**	59.22	61.66	60.44	62.88	63.38	63.13	**45.95**	**45.91**	**45.93**

Table 3 records the runtime of Paraphrase, SIT, and the combination of SIT with two methods. It can be observed that, compared to the Paraphrase model, the runtime of the Self-Inference Template has almost remained unchanged. The addition of the two small enhancements to the model has also had no impact on runtime.

Table 3. Model Runtime (Unit: Seconds).

Methods	Running Time			
	Rest15	Rest16	ACOS_ Restaurant	ACOS_ Laptop
Paraphrase	152.24	224.81	266.91	501.65
SIT	151.16	225.55	263.80	495.60
SIT+PT	153.52	224.05	259.83	495.52
SIT+MT	154.39	225.97	268.03	496.58
SIT+PM	153.86	225.32	270.12	498.00

5.2. Determination of Prefix Prompts

Currently, prefix prompts can be broadly categorized into hard prompts and soft prompts [14]. Hard prompts, also known as discrete prompts, are manually crafted prompts typically consisting of semantically meaningful phrases. On the other hand, soft prompts, also known as continuous prompts, are continuously updated and iterated during training, resembling a kind of updatable parameter without clear human-interpretable semantics.

Training with soft prompts requires a substantial amount of data for iterative updates, and the existing datasets for aspect sentiment quadruplet prediction are relatively small, making them unsuitable for training with soft prompts. Therefore, we opt for hard prompts, where we manually create prompt texts to assist the model's understanding during training. We generated six prompt texts, as illustrated in Figure 6. Three of them were created based on the original template, and the other three were created based on the Self-Inference Template, informing the model about the task it needs to perform in three different forms. The experimental results are presented in Table 4.

prompt1: Rewrite the given sentence in the [ac] is [sp] because [at] is [ot] form. Please provide a clear and concise response that accurately represents the original sentence's structure and meaning:

prompt2: Rewrite the given sentence in the [at] is [sp] because [at] is [ot], according to [at] and [ot] this aspect belongs to [ac] so [ac] is [sp] form. Please provide a clear and concise response that accurately represents the original sentence's structure and meaning:

prompt3: Your task is to analyze the given sentence and identify the aspect category [ac], aspect term [at], opinion term [ot], and sentiment polarity [sp]. Once identified, rewrite the sentence in the form of '[ac] is [sp] because [at] is [ot]'.

prompt4: Your task is to analyze the given sentence and identify the aspect category [ac], aspect term [at], opinion term [ot], and sentiment polarity [sp]. Once identified, rewrite the sentence in the form of '[at] is [sp] because [at] is [ot], according to [at] and [ot] this aspect belongs to [ac] so [ac] is [sp]'.

prompt5: Identify the aspect category [ac], aspect term [at], opinion term [ot], and sentiment polarity [sp]. Rewrite the sentence in the form of '[ac] is [sp] because [at] is [ot]':

prompt6: Identify the aspect category [ac], aspect term [at], opinion term [ot], and sentiment polarity [sp]. Rewrite the sentence in the form of '[at] is [sp] because [at] is [ot], according to [at] and [ot] this aspect belongs to [ac] so [ac] is [sp]':

Figure 6. Prefix Prompt Texts.

Table 4. Experimental Results Combining Different Prefix Prompts with the Self-Inference Template. The best scores are marked in bold.

Prompt Text	Rest15			Rest16			ACOS_Restaurant			ACOS_Laptop		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
SIT	47.89	50.13	48.98	58.98	61.60	60.26	63.13	63.49	63.31	44.38	44.61	44.49
+Prompt1	48.41	49.75	49.07	**60.78**	**63.24**	**61.99**	**63.54**	**63.83**	**63.69**	43.79	43.57	43.68
+Prompt1+MT	**49.63**	**50.38**	**50.00**	59.22	61.66	60.44	62.88	63.38	63.13	43.18	42.96	43.07
+Prompt2	48.20	48.99	48.59	58.35	61.09	59.69	62.13	62.13	62.13	44.16	43.74	43.95
+Prompt2+MT	48.94	49.37	49.15	58.65	60.58	59.60	62.60	62.81	62.71	45.23	44.52	44.87
+Prompt3	46.99	48.11	47.54	53.50	56.15	54.79	61.43	61.22	61.33	44.20	43.74	43.97
+Prompt3+MT	45.41	46.10	45.75	57.89	60.46	59.14	61.20	62.59	61.88	44.70	44.70	44.70
+Prompt4	43.85	43.07	43.46	54.25	58.30	56.20	58.68	58.62	58.65	41.99	41.48	41.73
+Prompt4+MT	46.45	46.98	46.71	54.43	56.02	55.22	58.33	58.73	58.53	42.45	41.83	42.14
+Prompt5	47.43	48.87	48.14	59.21	61.09	60.14	63.46	63.61	63.53	43.93	43.39	43.66
+Prompt5+MT	47.27	49.12	48.18	57.95	61.47	59.66	61.88	62.59	62.23	43.46	43.30	43.38
+Prompt6	47.77	48.49	48.13	58.29	60.58	59.42	61.01	60.32	60.66	43.12	42.78	42.95
+Prompt6+MT	46.48	47.36	46.91	57.58	59.70	58.62	61.51	60.88	61.20	**45.95**	**45.91**	**45.93**

Based on the experimental results, it can be observed that, for the Rest15, Rest16, and ACOS_Restaurant datasets, the first type of prefix prompt, which directly instructs the model to rewrite, can significantly help improve the model's reasoning ability. For ACOS_Laptop, the third type of prefix, instructing the model to first identify the four sentiment elements and then rewrite, combined with the Mask operation, leads to the optimal results. The reason might be that Rest15, Rest16, and ACOS_Restaurant datasets have fewer aspect categories, allowing the model to adequately learn the data for each aspect category during training and understand the task requirements without the need for prompting the model to recognize sentiment elements. However, ACOS_Laptop has more aspect categories, and many of them have fewer occurrences, making it challenging for the

model to fully learn each class of data, resulting in an insufficient understanding of the task requirements. Therefore, for ACOS_Laptop, the third type of prefix, prompting the model to recognize sentiment elements first and then rewrite, can provide the maximum assistance in helping the model quickly understand task requirements and enhance its capabilities.

The second type of prefix prompt created in a task assignment manner yielded the worst results, possibly due to the relatively limited parameter count of our T5-base model. Unlike larger language models like GPT, which can engage in task-oriented dialogues, our model may not benefit as much from prompts crafted in a task assignment format. Therefore, describing the task directly as a prefix prompt proves to be more effective. Interestingly, among the prefix prompts, instructing the model to rewrite the original template resulted in a higher improvement compared to using the Self-Inference Template. This may be attributed to the shorter nature of the prompt in the original template, mainly informing the model about the rewriting task it is about to perform. When rewriting sentences, the model learns the template based on the data, eliminating the need for extensive text prompts. Therefore, based on the experimental results, for Rest15, Rest16, and ACOS_Restaurant, selecting Prompt1 as the prefix prompt and for ACOS_Laptop, choosing Prompt6 as the prefix prompt proves to be most effective.

5.3. Ablation Study

On the Self-Inference Template, we proposed two methods to assist in the experiments. To understand the respective contributions of the Self-Inference Template and the two methods, we incorporated each method separately into the Paraphrase model and the Self-Inference Template. The experimental results are shown in Table 5. Overall, the Self-Inference Template proves beneficial for sentiment quadruple extraction across all four datasets. The addition of prefix prompts effectively enhances the ability of both the Paraphrase model and the self-inference model to extract sentiment quadruples in Rest15, Rest16, and ACOS_Restaurant datasets. The use of Mask Tokens on the Paraphrase model results in a decrease in performance, but when combined with prefix prompts on the Self-Inference Template, it helps the model achieve the best results on Rest15 and ACOS_Laptop. This suggests that the combination of the Self-Inference Template and the two methods yields impressive performance on some datasets, but the effectiveness of Mask Tokens is unstable and requires careful experimentation.

Table 5. Results of ablation experiments for four datasets. The best results are in bold.

Methods	Rest15			Rest16			ACOS_Restaurant			ACOS_Laptop		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
Paraphrase	46.16	47.72	46.93	56.63	59.30	57.93	61.02	59.73	60.37	44.87	44.10	44.48
Paraphrase+PT	48.46	49.56	49.00	58.99	61.58	60.26	60.07	59.40	59.73	44.51	43.32	43.91
Paraphrase+MT	45.51	46.54	46.02	58.19	61.33	59.72	57.71	57.84	57.78	43.53	42.89	43.21
Paraphrase+PM	47.58	48.30	47.94	57.11	58.82	57.95	60.09	60.29	60.19	44.54	43.24	43.88
SIT	47.89	50.13	48.98	58.98	61.60	60.26	63.13	63.49	63.31	44.38	44.61	44.49
SIT+PT	48.41	49.75	49.07	**60.78**	**63.24**	**61.99**	**63.54**	**63.83**	**63.69**	43.12	42.78	42.95
SIT+MT	47.93	49.50	48.70	58.30	60.96	59.60	61.79	63.27	62.51	44.46	44.35	44.41
SIT+PM	**49.63**	**50.38**	**50.00**	59.22	61.66	60.44	62.88	63.38	63.13	**45.95**	**45.91**	**45.93**

5.4. Model Overfitting Analysis

Due to the limited amount of data, the original model exhibits a significant overfitting issue, as shown in Figure 7. In all four datasets, the training set's loss steadily decreases, but the validation set's loss increases instead of decreasing. After applying our Self-Inference Template, a notable reduction in the validation set's loss is observed. Although there is still a subtle upward trend, it is considerably alleviated compared to the original model, indicating a significant reduction in overfitting.

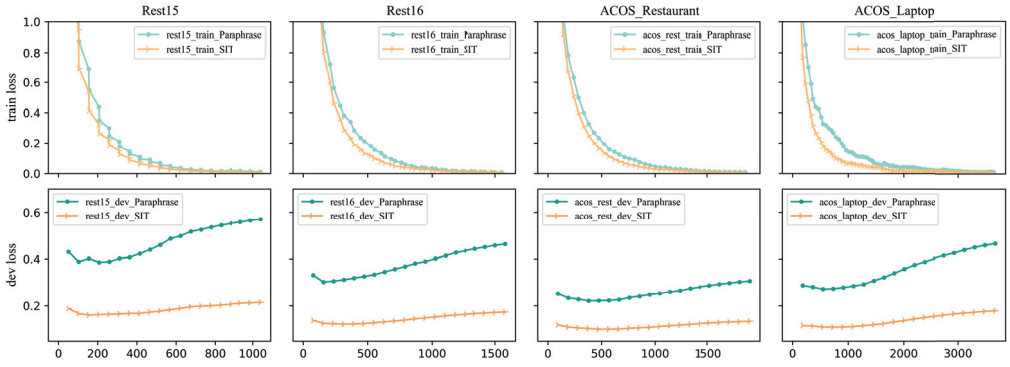


Figure 7. Training and validation set loss for Rest15, Rest16, ACOS_Restaurant, and ACOS_Laptop.

5.5. Error Analysis and Case Study

To understand the issues our model may encounter during inference, we conducted an error analysis and case study. We randomly sampled 100 data points from the test set of each dataset and performed sentiment quadruple extraction. Subsequently, we compared the quadruples inferred by the model with the ground truth labels, tallying the frequency of errors in predicting each sentiment element. Additionally, we recorded instances where the model overpredicted or underpredicted quadruples, as shown in Figure 8.

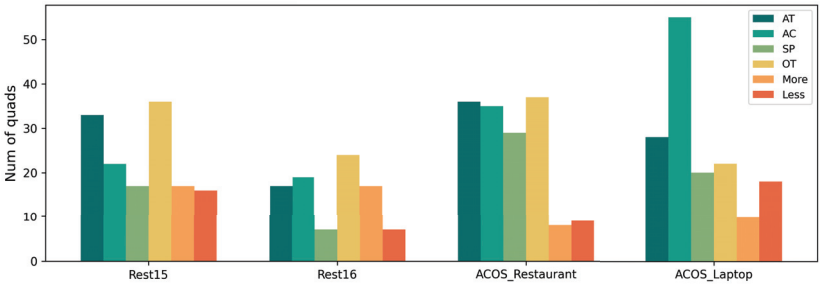


Figure 8. Quadruple Error Statistics.

On the Rest15, Rest16, and ACOS_Restaurant datasets, similar to the findings by [10], the opinion term is the most challenging sentiment element to predict. The model struggles to grasp the length of opinion term extraction. Following that, we have aspect terms and aspect categories, where the model finds it difficult to discern implicit aspect terms. If the aspect prediction is incorrect, it can easily lead to further errors in predicting aspect categories, as illustrated in Example 1 in Figure 9. Apart from predicting sentiment elements incorrectly, the model also tends to overgenerate or undergenerate quadruples, as shown in Example 2 in Figure 9. Therefore, determining how to make the model generate an appropriate number of quadruples is a question that deserves more consideration. For ACOS_Laptop, aspect category prediction errors are most frequent, as discussed in Section 5.1, mainly due to the abundance and imbalance of aspect categories in ACOS_Laptop, leading to insufficient learning, and the model tends to get confused, as shown in Example 3 in Figure 9.

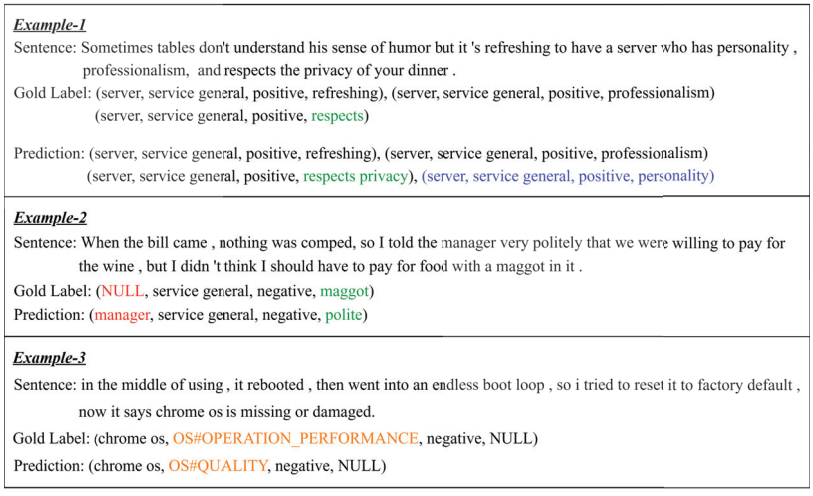


Figure 9. Cases of Quadruple Extraction Errors.

5.6. Practical Insights

In our work, in addition to the aforementioned methods, we also conducted some other experiments. When we initially observed the model's overfitting problem, we tried to mitigate overfitting through data augmentation using pseudo-labels. We crawled 10,000 restaurant reviews from the internet, then cleaned the data and filtered it down to 3000 entries. We first extracted 1000 entries and used the models trained on Rest15 and Rest16 to infer these 1000 reviews. We then performed an intersection of the inference results from the two models, ultimately obtaining 300 entries. We added these 300 entries to the dataset and experimented with adding them to the training set and test set in various proportions. We found that the model was very sensitive to the data, with different addition proportions causing significant fluctuations in the model's results. Therefore, we abandoned this method. These are some of our trial-and-error experiences, which we hope can provide some reference for future research.

6. Conclusions

In this work, we introduced a Self-Inference Template that leverages a chain of thought to assist the model in reasoning about aspect sentiment quadruples. Without increasing the time cost, this approach not only significantly improves the prediction results of quadruples but also effectively mitigates the overfitting issue caused by the limited amount of data. Additionally, we experimented with adding prefix prompts to the text and applying MASK operations to the text to assist in model training, which improved the model's results to some extent. This indicates the research significance of these two methods, suggesting potential avenues for further exploration in future studies. Finally, we conducted experiments on both the ASQP dataset, which does not contain implicit opinions, and the ACOS dataset, which contains implicit opinions. The results showed that the Self-Inference Template improved by 3.07%, 4.06%, 3.32%, and 1.45% on Rest15, Rest16, ACOS_Restaurant, and ACOS_Laptop, respectively, compared to Paraphrase, demonstrating significant effectiveness.

Author Contributions: Conceptualization, S.L. and Y.Q.; methodology, Y.Q.; software, Y.Q.; validation, S.L. and Y.Q.; formal analysis, S.L.; investigation, Y.Q.; resources, S.L.; data curation, Y.Q.; writing—original draft preparation, Y.Q.; writing—review and editing, S.L.; supervision, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: [9,10].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liu, P.; Joty, S.; Meng, H. Fine-grained opinion mining with recurrent neural networks and word embeddings. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1433–1443.
2. He, R.; Lee, W.S.; Ng, H.T.; Dahlmeier, D. An unsupervised neural attention model for aspect extraction. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 388–397.
3. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; AL-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; De Clercq, O.; et al. Semeval-2016 task 5: Aspect based sentiment analysis. In Proceedings of the ProWorkshop on Semantic Evaluation (SemEval-2016). Association for Computational Linguistics, San Diego, CA, USA, 16–17 June 2016; pp. 19–30.
4. Zhou, X.; Wan, X.; Xiao, J. Representation learning for aspect category detection in online reviews. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29.
5. Zhao, H.; Huang, L.; Zhang, R.; Lu, Q.; Xue, H. Spanmlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 3239–3248.
6. Chen, S.; Liu, J.; Wang, Y.; Zhang, W.; Chi, Z. Synchronous double-channel recurrent network for aspect-opinion pair extraction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6515–6524.
7. Peng, H.; Xu, L.; Bing, L.; Huang, F.; Lu, W.; Si, L. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8600–8607.
8. Wan, H.; Yang, Y.; Du, J.; Liu, Y.; Qi, K.; Pan, J.Z. Target-aspect-sentiment joint detection for aspect-based sentiment analysis. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 9122–9129.
9. Cai, H.; Xia, R.; Yu, J. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Virtual Event, 1–6 August 2021; pp. 340–350.
10. Zhang, W.; Deng, Y.; Li, X.; Yuan, Y.; Bing, L.; Lam, W. Aspect sentiment quad prediction as paraphrase generation. *arXiv* **2021**, arXiv:2110.00796.
11. Hu, M.; Wu, Y.; Gao, H.; Bai, Y.; Zhao, S. Improving aspect sentiment quad prediction via template-order data augmentation. *arXiv* **2022**, arXiv:2210.10291.
12. Peper, J.J.; Wang, L. Generative aspect-based sentiment analysis with contrastive learning and expressive structure. *arXiv* **2022**, arXiv:2211.07743.
13. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24824–24837.
14. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **2023**, *55*, 1–35. [CrossRef]
15. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
16. Fei, H.; Li, B.; Liu, Q.; Bing, L.; Li, F.; Chua, T.S. Reasoning Implicit Sentiment with Chain-of-Thought Prompting. *arXiv* **2023**, arXiv:2305.11255.
17. Zhang, W.; Li, X.; Deng, Y.; Bing, L.; Lam, W. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 11019–11038. [CrossRef]
18. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 5485–5551.
19. Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv* **2022**, arXiv:2203.11171.
20. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
21. Cai, H.; Tu, Y.; Zhou, X.; Yu, J.; Xia, R. Aspect-category based sentiment analysis with hierarchical graph convolutional network. In Proceedings of the 28th International Conference on Computational Linguistics, Online, 8–13 December 2020; pp. 833–843.
22. Li, X.; Bing, L.; Zhang, W.; Lam, W. Exploiting BERT for end-to-end aspect-based sentiment analysis. *arXiv* **2019**, arXiv:1910.00883.

23. Zhang, W.; Li, X.; Deng, Y.; Bing, L.; Lam, W. Towards generative aspect-based sentiment analysis. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Virtual Event, 1–6 August 2021; pp. 504–510.
24. Mao, Y.; Shen, Y.; Yang, J.; Zhu, X.; Cai, L. Seq2path: Generating sentiment tuples as paths of a tree. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, 22–27 May 2022; pp. 2215–2225.
25. Li, S.; Zhang, Y.; Lan, Y.; Zhao, H.; Zhao, G. From Implicit to Explicit: A Simple Generative Method for Aspect-Category-Opinion-Sentiment Quadruple Extraction. In Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 18–23 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–8.

applied sciences

MDPI

Article

Interpretability in Sentiment Analysis: A Self-Supervised Approach to Sentiment Cue Extraction

Yawei Sun [1,2], Saike He [3,*], Xu Han [4] and Yan Luo [5]

[1] Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China; sunyawei@bupt.edu.cn
[2] School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China
[3] State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[4] Institute of Scientific and Technical Information of China, Beijing 100038, China
[5] Institute of Medical Information, Chinese Academy of Medical Sciences & Peking Union Medical College, Beijing 100020, China
* Correspondence: saike.he@ia.ac.cn

Abstract: In this paper, we present a novel self-supervised framework for Sentiment Cue Extraction (SCE) aimed at enhancing the interpretability of text sentiment analysis models. Our approach leverages self-supervised learning to identify and highlight key textual elements that significantly influence sentiment classification decisions. Central to our framework is the development of an innovative Mask Sequence Interpretation Score (MSIS), a bespoke metric designed to assess the relevance and coherence of identified sentiment cues within binary text classification tasks. By employing Monte Carlo Sampling techniques optimized for computational efficiency, our framework demonstrates exceptional effectiveness in processing large-scale text data across diverse datasets, including English and Chinese, thus proving its versatility and scalability. The effectiveness of our approach is validated through extensive experiments on several benchmark datasets, including SST-2, IMDb, Yelp, and ChnSentiCorp. The results indicate a substantial improvement in the interpretability of the sentiment analysis models without compromising their predictive accuracy. Furthermore, our method stands out for its global interpretability, offering an efficient solution for analyzing new data compared to traditional techniques focused on local explanations.

Keywords: sentiment cue extraction; self-supervised learning; interpretable machine learning

Citation: Sun, Y.; He, S.; Han X.; Luo, Y Interpretability in Sentiment Analysis: A Self-Supervised Approach to Sentiment Cue Extraction. *Appl. Sci.* **2024**, *14*, 2737. https://doi.org/10.3390/app14072737

Academic Editor: Douglas O'Shaughnessy

Received: 26 February 2024
Revised: 20 March 2024
Accepted: 20 March 2024
Published: 25 March 2024

1. Introduction

In the rapidly evolving landscape of the information age, the prolific growth of textual data on various online platforms has propelled Natural Language Processing (NLP) into a position of increased importance. Within this domain, sentiment analysis [1], also referred to as opinion mining, stands out as a critical area. This process involves the automatic detection and interpretation of sentiments, emotions, and subjective information within textual data [2]. The application of sentiment analysis spans a wide spectrum, from the analysis of customer feedback in product reviews to the evaluation of public sentiment on social media platforms [3].

In the ever-evolving digital landscape, the exponential growth of textual data across various online platforms has elevated NLP to a critical technological frontier. Among the myriad applications of NLP, sentiment analysis plays a pivotal role. This field, focusing on the automatic detection and interpretation of sentiments, emotions, and subjective information within textual content, finds widespread application from analyzing customer feedback in product reviews to monitoring public sentiment on social media platforms.

Despite the remarkable advances and successes in sentiment analysis, a significant hurdle persists: the challenge of interpretability, which encompasses the difficulty of understanding and explaining how sentiment analysis models make their decisions, particularly in terms of identifying specific factors or textual elements that influence these decisions [4]. Traditional sentiment analysis models are often criticized for their "black-box" nature, which obscures the transparency of their decision-making processes [5]. This opacity generates concerns about accountability and dependability, especially in scenarios where precision and reliability are paramount.

To mitigate these concerns, our research introduces a novel self-supervised framework focused on sentiment cue extraction. This approach involves the identification and extraction of crucial linguistic elements—such as specific words, phrases, or syntactic patterns, referred to as "sentiment cues" in this paper—that significantly influence sentiment determination. Our approach is instrumental in demystifying the decision-making process of sentiment analysis models, thus contributing to a deeper understanding and trust in these systems.

For example, in finance, discerning the exact cues that drive sentiment predictions can be a game changer for market analysis [6–8]. Similarly, in healthcare, the analysis of sentiment cues in patient feedback, particularly from online sources, is essential to improve the quality of healthcare services. By evaluating positive and negative sentiments expressed in patient reviews, healthcare providers can identify strengths and areas for improvement in their services, such as facility cleanliness, staff behavior, and general patient care [9].

Our study introduces a groundbreaking framework based on self-supervised learning that incorporates sequence labeling techniques to significantly improve the interpretability of sentiment analysis models. Traditional approaches in sentiment cue extraction often involve labor-intensive and time-consuming data annotation processes. Existing interpretability methods for text classification models, while offering partial solutions, primarily depend on local interpretative methods. These local methods typically require individual training for each data instance, presenting significant challenges in efficiently handling new data.

In contrast, our innovative approach uses the abundance of existing annotated sentiment classification data through self-supervised learning. This enables our framework to interpret sentiment classification models in scenarios where explicit annotation is lacking, effectively facilitating sentiment cue extraction. Importantly, this methodology transcends the boundaries of local interpretability techniques, offering a global interpretability approach. Such a global perspective allows for a more holistic and comprehensive understanding of the model's decision-making process across various instances, rather than being confined to localized, instance-specific explanations.

To the best of our knowledge, ours is the first work to combine Monte Carlo methods with self-supervised learning to address the global interpretability issue in binary text classification [10–12]. The key contributions of our research are as follows.

- We propose a Self-Supervised Sentiment Cue Extraction (SS-SCE) method. This approach, inspired by the concept of interpretability in text classification models, accomplishes the extraction of sentiment cues from texts under conditions of scarce labeled data through a global interpretability analysis of the text classification models.
- We have developed a pseudo-label generation scheme for sentiment cue extraction models. This scheme selects appropriate mask sequences as pseudo labels for the sentiment cue extraction model based on the prediction results of a trained text classification model. Furthermore, we enhance the efficiency of pseudo-label generation by employing a Monte Carlo Sampling strategy.
- We have introduced the Mask Sequence Interpretation Score (MSIS) metric, designed to evaluate generated mask sequences based on the prediction results of a text classification model, thereby providing a basis for the generation of pseudo labels. Empirical evidence demonstrates the effectiveness of our MSIS metric.

The remainder of this paper is organized as follows: Section 2 discusses related work, providing background on sentiment analysis, self-supervised methods for information extraction, interpretability in machine learning, and the use of Monte Carlo methods. Section 3 details our methodology, explaining the sentiment cue extraction process, the use of Monte Carlo sampling, label sequence selection, and the sentiment cue extraction algorithm. In Section 4, we present our experimental setup, the datasets used, and a thorough evaluation of the performance of the SS-SCE framework. This includes an in-depth analysis of our results and a comparative study with state-of-the-art interpretability methods. Finally, Section 5 concludes the paper, summarizing our key findings, discussing the implications and potential applications of our work, and suggesting avenues for future research.

2. Related Works

2.1. Sentiment Analysis

Sentiment analysis, also known as opinion mining, is a crucial subfield of NLP that focuses on discerning and categorizing opinions expressed in text [13,14]. Its primary goal is to determine the writer's position toward specific topics or the general polarity of the sentiment of the text. This analysis typically involves categorizing text polarity at various levels: document, sentence, or feature/aspect level, determining whether the expressed opinion is positive, negative, or neutral [3].

With the advent of deep learning, sentiment analysis has undergone significant advances. Models such as Bidirectional Encoder Representations from Transformers (BERT) and its variants have been extensively employed for nuanced sentiment analysis, enhancing context and semantic understanding [15,16]. Moreover, transformer-based models like GPT-3 have pushed the boundaries further in generating human-like text, which is advantageous for more intricate sentiment analysis scenarios [17].

Sentiment analysis finds extensive applications across various domains, from customer service and market research to social media monitoring and political campaigns. It is essential for businesses and organizations to gauge public opinion, conduct market research, monitor the reputation of the brand and the product, and understand customer experiences [1].

In today's era of advanced NLP technology, sentiment analysis has emerged as a highly focused research area within the field, benefiting from a plethora of readily available high-quality datasets, such as IMDb [18] and SST-2 [19]. This availability has injected significant vitality into research in this direction. However, the "black box" nature of many deep learning models used in sentiment analysis poses another major limitation. These models, while powerful, often lack transparency in their decision-making processes, making it difficult for users to understand and trust their predictions.

Furthermore, sentiment analysis faces challenges in detecting nuances such as sarcasm, irony, and context-dependent meanings. Future research may involve more sophisticated models that understand complex human emotions and incorporate multimodal data (text, images, and videos) to better understand sentiments [20].

The field of sentiment analysis in NLP continues to be dynamic, with ongoing efforts to enhance the accuracy and versatility of sentiment detection algorithms. As computational models evolve, their ability to discern sentiments from text is expected to become increasingly refined and sophisticated.

2.2. Self-Supervised Methods for Information Extraction

Self-supervised learning in NLP has emerged as a fundamental approach to information extraction, harnessing the potential of unlabeled data to train predictive models. This paradigm involves creating learning tasks in which models predict certain parts of the input using other parts [21–23].

By utilizing large volumes of unlabeled data, self-supervised learning allows models to learn rich representations. These representations are beneficial for diverse downstream

NLP tasks, especially valuable in contexts where labeled data are scarce or expensive to acquire [16,21].

Among the popular methodologies in self-supervised learning, Masked Language Modeling (MLM) stands out. MLM is a key technique in self-supervised learning, notably used by BERT. It involves hiding some words in a sentence and training the model to predict these hidden words using the surrounding context. This process aids in understanding the context and relationships between words [16].

Permutation-based language modeling, as introduced by XLNet, is another significant methodology. It extends the concept of MLM to predict a token based on all permutations of tokens in a sentence. This approach offers a more comprehensive context understanding [22].

Additionally, models like BART [23] and Text-to-Text Transfer Transformer (T5) [24] utilize a corrupted text generation task for pre-training. In this approach, models learn to reconstruct the original text from a corrupted version, thereby enhancing their understanding of language structure and coherence [23].

In the evolving landscape of self-supervised learning models, the Generative Pre-trained Transformer (GPT) series by OpenAI marks a pivotal juncture [17,25,26]. Unlike BERT, renowned for its bidirectional approach to language comprehension, GPT models excel at text generation by predicting the subsequent word in a sequence. Consequently, while BERT shines in nuanced language understanding tasks, GPT excels in producing coherent and contextually apt text.

Continuing this trajectory, ChatGPT (https://chat.openai.com (accessed on 19 March 2024)), a notable addition to the GPT lineage, heralds further breakthroughs. Specifically, ChatGPT exemplifies the prowess of large-scale language models across an array of uses, from crafting human-like narratives to conducting nuanced sentiment analyses. Its adaptability for fine-tuning targeted tasks significantly expands its utility and effectiveness in addressing diverse NLP challenges. Parallel to ChatGPT's emergence, a myriad of other large language models like Gemini (https://gemini.google.com/ (accessed on 19 March 2024)) and ERNIE Bot (https://yiyan.baidu.com/ (accessed on 19 March 2024)) have surfaced, enriching the field with their distinct contributions.

However, these advancements are not without challenges. ChatGPT's closed-source nature hinders research transparency and restricts community-driven enhancements. Moreover, the substantial computational resources required to operate or fine-tune such models often necessitate reliance on cloud-based APIs provided by the developers. This reliance raises concerns regarding cost-effectiveness, latency issues, and data privacy implications [27,28].

Self-supervised learning has achieved remarkable success in tasks such as named entity recognition, relation extraction, and event extraction. By pretraining on extensive text corpora, these models capture nuanced language patterns, significantly increasing their task performance [29,30].

2.3. Interpretability of Deep Learning Models

The interpretability of deep learning models in NLP is a vital research area, concentrating on deciphering and explaining how these models make decisions. This aspect is particularly critical in applications where trust and transparency are paramount [4,31].

Interpretability in deep learning models is essential to validate and improve model performance, ensure fairness, and provide information on model behavior, especially in areas such as healthcare, finance, and legal applications [31].

Several techniques have been developed to enhance the interpretability of deep learning models. These include attention mechanisms, which underscore parts of the input data most relevant to the model decision [32], and Local Interpretable Model-Agnostic Explanations (LIME), which approximate the model locally using interpretable models [5]. In addition, researchers also use topic models such as Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA) to achieve interpretability [33]. For example, Xiong and Li [34] combined LDA with deep learning models to not only grade student essays but also identify the characteristics of excellent essays in terms of language expression.

Despite these advances, understanding deep learning models, particularly transformers, remains challenging. Their black-box nature often hinders the understanding of their predictive reasoning [4].

Future research in model interpretability is likely to focus on developing more robust generalizable techniques that offer clear explanations of model decisions, including integrating interpretability directly into model architecture and training [35].

As deep learning models continue to advance and find application in critical domains, the significance of interpretability will only escalate. Ensuring that these models are transparent and that their decisions are understandable is key to their successful and ethical application.

2.4. Monte Carlo Methods

Monte Carlo methods represent a class of computational algorithms that employ repeated random sampling to yield numerical outcomes. In the realms of NLP and machine learning, these methods are applied across a spectrum of tasks, including optimization, numerical integration, and probabilistic inference [36,37].

The foundational principle of Monte Carlo methods is the utilization of randomness to address problems that, while theoretically deterministic, are complex in nature. These methods are particularly effective in computing quantities that are challenging for deterministic algorithms, largely because of their high-dimensional characteristics.

In the field of NLP, Monte Carlo methods have found extensive applications in language modeling, particularly in tasks that encompass uncertainty and probabilistic models. A notable example of their application is in Bayesian learning methodologies, where they are instrumental in estimating the posterior distributions of model parameters [38].

Recent progress in Monte Carlo methods has geared towards enhancing both efficiency and accuracy, especially within the context of deep learning. Techniques such as Markov Chain Monte Carlo (MCMC) have been adapted for compatibility with complex model structures, including deep neural networks [37].

A primary challenge in the implementation of Monte Carlo methods within NLP pertains to the computational demands, which are accentuated when large datasets and intricate model architectures. Consequently, future research is anticipated to focus on the development of more efficient sampling techniques and the integration of Monte Carlo methods with other machine learning approaches [39].

3. Task Definition

The primary aim of this study is to enhance the interpretability of sentiment classification models applied to texts. Specifically, our focus is on identifying the key factors—words or phrases within a text—that sentiment classification models rely on to determine the sentiment polarity of that text. These influential words or phrases are collectively referred to as "Sentiment Cues". Therefore, we term the task we explore in this paper as Sentiment Cue Extraction (SCE). This endeavor seeks to uncover and articulate the rationale behind sentiment polarity judgments made by these models, making the decision-making process more transparent and understandable to both users and researchers.

To clarify the task of SCE more distinctly, let us illustrate with the following two examples:

- `Instance 1`: Very **friendly** customer service.
- `Instance 2`: If I could give a **zero star**, I would!

Here, the word "friendly" in the first instance allows us to identify its sentiment as positive; similarly, the phrase "zero star" in the second instance indicates a negative sentiment. Hence, "friendly" and "zero star" serve as what we define as sentiment cues.

4. Methodology

4.1. Overview of Our Method

To address the SCE task, this paper conceptualizes SCE as a sequence labeling task. This perspective allows for a systematic approach to identifying sentiment cues across varying textual instances.

Given an instance $X = \{x_1, x_2, \cdots, x_n\}$, our objective is to assign a corresponding label set $Y = \{y_1, y_2, \cdots, y_n\}$, where $y_i = 1$ signifies that the element constitutes a significant sentiment cue. For example, regarding Instance 1, the corresponding X and Y are as illustrated in Equations (1) and (2), respectively.

$$X = \{\text{“}Very\text{”}, \text{“}friendly\text{”}, \text{“}customer\text{”}, \text{“}service\text{”}, \text{“}.\text{”}\}, \tag{1}$$

$$Y = \{0, 1, 0, 0, 0\}. \tag{2}$$

However, a principal challenge within this work is the absence of annotated data for the SCE task, meaning that Y is unknown within the dataset. To address this, we introduce a Self-Supervised Sentiment Cue Extraction (SS-SCE) method that employs self-supervised learning to tackle the SCE task. In the SS-SCE framework, we utilize a sentiment classification model, which has been widely labeled, to generate pseudo labels for the SCE task. These pseudo labels, derived from samples X in the sentiment classification dataset, serve as inputs and outputs for constructing the SCE training dataset, thereby enabling the training of an SCE sequence labeling model. The fundamental steps of this approach are depicted in Figure 1.

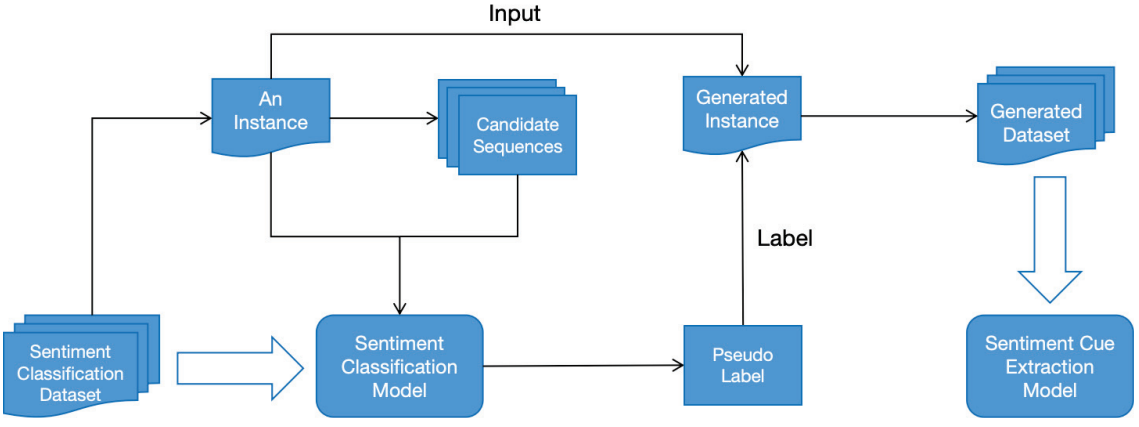


Figure 1. This figure illustrates the workflow of our self-supervised sentiment cue extraction method. “Input” and “Label” represent the roles of “An Instance” and “Pseudo Label” within the “Generated Instance”, respectively. The bold arrows indicate the process of training the corresponding models using the dataset.

Figure 1 illustrates the basic workflow of our method. Initially, we train a sentiment classification model based on a sentiment classification dataset. Building on this, we generate candidate sequences of pseudo labels (referred to as the Candidate Sequences in the figure) for an instance within the dataset and then use the sentiment classification model to select one sequence from these candidates as the pseudo label. Thus, by taking an instance as input and using the obtained pseudo label as the label, we can form a sequence labeling instance (referred to as the Generated Instance in the figure). By generating such generated instances for other instances in the sentiment classification, we can compile a

dataset (referred to as the Generated Dataset in the figure) that is suitable for training the SCE model. Based on this dataset, the SCE model can then be trained.

In the following sections, we will introduce our method in detail.

4.2. Generating Dataset for Sentiment Cue Extraction

4.2.1. Generation of Candidate Sequences

As described earlier, for an input X, it is necessary to generate several candidate sequences of pseudo labels, denoted by $Y^c = \{y_1^c, y_2^c, \cdots, y_n^c\}$. Theoretically, for an input X of length n, there are 2^n possible configurations for Y^c. This implies that for $n = 20$, Y^c could have over 1 million possible combinations—a daunting figure. This calculation pertains to just a single data instance, whereas training the SCE model requires thousands of such pseudo-labeled data instances. Enumerating all possible combinations is impractical, both in terms of time and computational resources. Therefore, we employ the Monte Carlo Sampling [40] method to randomly generate a specified number of candidate sequences, significantly reducing the time complexity associated with generating these candidate sequences. The Monte Carlo Sampling algorithm we use is outlined in Algorithm 1. This approach allows us to efficiently produce a manageable subset of potential label sequences for further analysis and selection, ensuring the feasibility of the SCE model training process.

Algorithm 1 Monte Carlo Sampling for generating one candidate sequence

Require: instance X, sampling ratio p
$Y^c \leftarrow \emptyset$
for $i \in [1, 2, \ldots, n]$ **do**
  generate g uniformly at random in the range [0, 1]
  if $g < p$ **then**
    $y_i^c \leftarrow 1$
  else
    $y_i^c \leftarrow 0$
  end if
  Add y_i^c to Y^c
end for
return Y^c

In Algorithm 1, we commence by specifying a sampling ratio p. For each element x_i in X, we randomly generate a decimal number g uniformly within the range of 0 to 1. If $g < p$, then y_i^c is set to 1; otherwise, it is set to 0. This mechanism ensures that each y_i^c has a probability p of being assigned the value 1. Consequently, it can be inferred that the proportion of elements labeled 1 in the generated sequence Y^c is expected, on average, to be p.

This method does more than simply allow for the manipulation of positive label density within candidate sequences; it also facilitates the emulation of varied labeling densities in scenarios devoid of pre-annotated data by modulating the p-value. Ideally, p should mirror the proportion of tokens in the text X that significantly influence the sentiment classification model's decision-making process, equivalent to the proportion of elements valued at 1 in Y. However, this proportion is unknown. Therefore, to generate candidate sequences as comprehensively as possible, we employ multiple values for p during the sampling process, conducting sampling under these varied p-values.

4.2.2. Sentiment Classification Model

As demonstrated in Figure 1, selecting an optimal pseudo label from the array of candidate sequences involves scoring each candidate. Within the SS-SCE framework, this scoring process is facilitated by a sentiment classification model. Herein, we provide an overview of the sentiment classification model used in the SS-SCE context.

In our research, the sentiment classification model is built on BERT as the encoding mechanism, mainly due to its ability to effectively capture contextual nuances within the

text. BERT, a transformer-based model, stands out for its dynamic encoding capabilities, compared to static word vector methods, such as GloVe [41], which may not fully grasp the context-dependent aspects of language.

Moreover, compared to GPT [25,26], another transformer-based architecture, BERT is more aligned with our needs. While GPT excels in text generation tasks due to its unidirectional nature, BERT's bidirectional training strategy makes it particularly suitable for understanding the nuanced expressions of sentiment in texts. This bidirectionality allows BERT to gather context from both sides of a token, offering a richer representation of the input text and enhancing the model's ability to discern the underlying sentiment.

Additionally, sentiment classification models within the academic community often leverage BERT-based architectures, facilitating straightforward comparisons with other models in the field.

When encoding the input X using BERT for our sentiment classification model, it is necessary to prepend a [CLS] token at the beginning and append a [SEP] token at the end of X. Thus, the actual sequence inputted into BERT becomes $X = \{[\text{CLS}], x_1, x_2, \cdots, x_n, [\text{SEP}]\}$. For text classification tasks, the encoding of the [CLS] token is typically utilized to represent the encoding of the entire sentence.

To facilitate comparisons with other models, we have constructed a remarkably straightforward binary text classification model f_{sc} based on the base version of BERT. In this model, f_{sc}, the enhanced input X is encoded using BERT, resulting in a 768-dimensional vector representation, h_X^{768}. This vector, specifically derived from the encoding of the [CLS] token, is then transformed into a two-dimensional vector, h_X^2, via a fully connected layer. Subsequently, a softmax function converts h_X^2 into a pair of probabilities that indicate the likelihood of X belonging to categories 0 (negative sentiment) and 1 (positive sentiment), respectively. The sentiment classification model f_{sc} can thus be expressed as:

$$f_{\text{sc}}(X) = \text{softmax}(\text{FC}^{768\times 2}(\text{BERT}(X)_{[\text{CLS}]})) \tag{3}$$

where $\text{FC}^{768\times 2}$ denotes the fully connected layer mapping the 768-dimensional BERT encoding to a 2-dimensional output, and $\text{BERT}(X)_{[\text{CLS}]}$ refers to the representation of the [CLS] token produced by BERT, which serves as the aggregate representation of the enhanced input text for classification purposes.

Accordingly, for an input, the model yields the following probability pair:

$$(p_{c0}, p_{c1}) = f_{\text{cls}}(X), \tag{4}$$

where p_{c0} and p_{c1} correspond to the probabilities of X being classified under negative and positive sentiments, respectively. Consequently, the sentiment prediction for X by f_{cls} is determined as:

$$C = \arg\max(p_{c0}, p_{c1}), \tag{5}$$

This procedure also facilitates the computation of the Probability Discrepancy between the categories:

$$\Delta P = |p_{c0} - p_{c1}|. \tag{6}$$

In this context, ΔP, referred to as "Probability Discrepancy", is utilized to assess the intensity of the sentiment inclination prediction made by f_{sc} for X. A larger ΔP value indicates a more pronounced sentiment inclination in X, reflecting the model's confidence in its sentiment classification.

4.2.3. Mask Sequence Interpretation Score

In the SCE task, for a given input X with labels Y, there exists an inverse sequence $\bar{Y} = \{1 - y_1, 1 - y_2, \cdots, 1 - y_n\}$. As defined by the task, if the token x_i in X is identified as an SC within X, then the corresponding label y_i is assigned a value of 1; if not, y_i is set to 0. This principle suggests that masking all tokens x_i in X for which $y_i = 1$, resulting in a masked input $X^{\bar{Y}}$, would hinder the sentiment classification model's ability to accurately

determine the sentiment inclination of $X^{\bar{Y}}$. Conversely, retaining only the tokens in X, where $y_i = 1$, and masking those with $y_i = 0$, to create a new input X^Y, should enable the sentiment classification model to predict its sentiment inclination effectively.

In typical scenarios, to obtain X^Y, it is necessary to replace tokens x_i in X corresponding to $y_i = 0$ in Y with a meaningless symbol like [MASK]. However, BERT provides a more straightforward solution for us. By using Y as the attention mask directly input into BERT, it automatically disregards tokens x_i in X corresponding to $y_i = 0$ in Y.

Therefore, for X, when using Y as the mask sequence, we obtain:

$$(p_{c0}^Y, p_{c1}^Y) = f_{\text{sc}}(X, Y), \tag{7}$$

yielding the sentiment category prediction:

$$C^Y = \arg\max(p_{c0}^Y, p_{c1}^Y), \tag{8}$$

and calculating the Probability Discrepancy as:

$$\Delta P^Y = |p_{c0}^Y - p_{c1}^Y|. \tag{9}$$

Similarly, when using the inverse sequence $\bar{Y}$ as the mask sequence, we can determine:

$$(p_{c0}^{\bar{Y}}, p_{c1}^{\bar{Y}}) = f_{\text{sc}}(X, \bar{Y}), \tag{10}$$

with the corresponding sentiment category determined by:

$$C^{\bar{Y}} = \arg\max(p_{c0}^{\bar{Y}}, p_{c1}^{\bar{Y}}), \tag{11}$$

and the Probability Discrepancy for $X^{\bar{Y}}$ calculated as:

$$\Delta P^{\bar{Y}} = |p_{c0}^{\bar{Y}} - p_{c1}^{\bar{Y}}|. \tag{12}$$

When selecting a candidate sequence Y as the pseudo label, the ideal scenario aims to maximize ΔP^Y while ensuring that $C^Y = C$, and simultaneously minimize $\Delta P^{\bar{Y}}$. However, this approach might lead to an extreme case where all elements of Y are set to 1 and all elements of $\bar{Y}$ are set to 0. In such a scenario, X^Y would be identical to X, and $X^{\bar{Y}}$ would contain no informative content, which, while adhering to the principle, is not desirable for effective sentiment cue extraction. To circumvent this issue, it is preferable to have as few elements labeled as 1 in Y as possible. To achieve this balance, we introduce the Ratio of Cue Tokens (RCT), calculated as follows:

$$RCT = \frac{\sum(Y)}{n} \tag{13}$$

where $\sum(Y)$ represents the number of elements valued at 1 in Y, and n denotes the total number of elements in Y.

Moreover, within X, there may be tokens that inversely affect the prediction of X's sentiment inclination. Such tokens might cause f_{sc} to predict the sentiment category of $X^{\bar{Y}}$ as being entirely opposite to that of X. In these situations, it is desirable for $\Delta P^{\bar{Y}}$ to be as large as possible to reflect a clear differentiation in sentiment inclination.

Taking into consideration the principles mentioned above, we propose an evaluation metric named the Mask Sequence Interpretation Score (MSIS) as follows:

$$MSIS = \begin{cases} \dfrac{\Delta P^{Y}}{\Delta P^{\bar{Y}} \cdot RCT^{2}}, & C = C^{Y} \cap \Delta P^{\bar{Y}} < \Delta P^{Y} \\ \dfrac{\Delta P^{Y} \cdot \Delta P^{\bar{Y}}}{RCT^{2}}, & C = C^{Y} \neq C^{\bar{Y}} \cap \Delta P^{\bar{Y}} \geq \Delta P^{Y} \\ 0, & Others \end{cases} \tag{14}$$

Algorithm 2 outlines the procedure for evaluating the MSIS for a given candidate sequence Y^c associated with an instance X.

Algorithm 2 Evaluating the candidate sequence Y^c

Require: instance X, the sentiment category of the instance C, length of the instance n, well trained sentiment classification model f_{sc}, candidate sequence $Y^c = \{y_1^c, y_2^c, \cdots, y_n^c\}$

$\bar{Y}^c \leftarrow \emptyset$
for $i \in [1, 2, \ldots, n]$ **do**
    Add $1 - y_i^c$ to $\bar{Y}^c$
end for
$(p_{c0}^{Y^c}, p_{c1}^{Y^c}) = f_{\text{sc}}(X, Y^c)$, $(p_{c0}^{\bar{Y}^c}, p_{c1}^{\bar{Y}^c}) = f_{\text{sc}}(X, \bar{Y}^c)$
$C^{Y^c} = \arg\max(p_{c0}^{Y^c}, p_{c1}^{Y^c})$, $C^{\bar{Y}^c} = \arg\max(p_{c0}^{\bar{Y}^c}, p_{c1}^{\bar{Y}^c})$
$\Delta P^{Y^c} = |p_{c0}^{Y^c} - p_{c1}^{Y^c}|$, $\Delta P^{\bar{Y}^c} = |p_{c0}^{\bar{Y}^c} - p_{c1}^{\bar{Y}^c}|$
$RCT_{Y^c} \leftarrow RCT(Y^c, n)$ ▷ Refer to Equation (13)
$MSIS_{Y^c} \leftarrow MSIS(C, C^{Y}, C^{\bar{Y}^c}, \Delta P^{Y^c}, \Delta P^{\bar{Y}^c})$ ▷ Refer to Equation (14)
return $MSIS_{Y^c}$

The process begins by creating an inverse sequence $\bar{Y}^c$, which serves as a complement to Y^c by inverting the binary values. This step ensures that we can compare the effects of including versus excluding specific tokens identified as sentiment cues on the predictions of the sentiment classification model.

Next, the algorithm employs f_{sc} to calculate the probabilities of X belonging to each sentiment category, both with and without the sentiment cues as indicated by Y^c and $\bar{Y}^c$, respectively. These probabilities allow the computation of the Probability Discrepancy (ΔP) for both sequences, offering insight into the decisiveness of the sentiment classification under different conditions.

The RCT for Y^c is then calculated, providing a measure of the proportion of tokens in X identified as sentiment cues by Y^c. This ratio is crucial for ensuring that the selection of sentiment cues is both significant and minimal, avoiding over-representation of cues.

Finally, the MSIS for Y^c is determined based on the sentiment category predictions and Probability Discrepancies for both Y^c and its inverse.

4.2.4. Process of Selecting Pseudo Label for X

Algorithm 3 details the comprehensive process for generating a pseudo label for an instance X. This process involves evaluating multiple candidate sequences generated under various sampling ratios, each with the aim of identifying the sequence that best represents the sentiment cues within X. The algorithm utilizes a well-trained sentiment classification model f_{sc} to calculate the MSIS for each candidate sequence, ultimately selecting the sequence with the highest MSIS as the pseudo label for X.

It should be noted that in order to ensure the RCT of the candidate sequences obtained through sampling is as uniform as possible, covering different instances, we will uniformly select several decimals between 0 and 1 to serve as sampling ratios.

By employing this algorithm for all instances in the dataset, a collection of pseudo labels is generated, forming a dataset that can be used to train the SCE model.

Algorithm 3 Process of selecting pseudo label for X

Require: instance X, the sentiment category of the instance C, length of the instance n, well trained sentiment classification model f_{sc}, set of sampling ratios $\{p_1, p_2, \ldots, p_k\}$, sampling number m

$Y \leftarrow \emptyset$ ▷ Initialize the pseudo label ad a empty set
$MSIS_Y \leftarrow 0$ ▷ Initialize $MSIS$ of Y with 0
for $p \in \{p_1, p_2, \ldots, p_k\}$ **do**
  for $i \in [1, 2, \ldots, m]$ **do**
    $Y^c \leftarrow$ Generate a candidate sequence with X and p ▷ Refer to Algorithm 1
    $MSIS_{Y^c} \leftarrow$ Evaluate Y^c ▷ Refer to Algorithm 2
    if $MSIS_{Y^c} > MSIS_Y$ **then**
      $Y \leftarrow Y^c$ ▷ Update Pseudo Label Y if a higher score is achieved
      $MSIS_Y \leftarrow MSIS_{Y^c}$ ▷ Update the score accordingly
    end if
  end for
end for
return Y

4.3. Sentiment Cue Extraction Model

Our SS-SCE approach conceptualizes the SCE task as a sequence labeling problem. This requires performing a binary classification for each token x_i within the input X. Consequently, the architecture of the SCE model is highly analogous to that of the sentiment classification model, with a key distinction: while the sentiment classification model focuses on classifying the [CLS] token to infer the overall sentiment of the input, the SCE model extends this classification to all tokens within X. Therefore, the SCE model can be formalized as follows:

$$f_{\text{sce}}(X) = \text{softmax}(\text{FC}^{768\times 2}(\text{BERT}(X))) \tag{15}$$

where $\text{BERT}(X)$ produces a sequence of 768-dimensional vector representations for each token in X. The fully connected layer, denoted as $\text{FC}^{768\times 2}$, maps each 768-dimensional vector to a 2-dimensional output, corresponding to the binary classification for sentiment cue detection. The softmax function is applied to these 2-dimensional vectors, yielding a probability distribution over two classes (cue vs. non-cue) for each token in X.

After generating pseudo labels for each instance X in the train set, these labels will be utilized as the ground truth for training the SCE model, f_{sce}. It is important to note that each X is augmented with [CLS] and [SEP] tokens at the beginning and end, respectively. While these tokens are essential for BERT's processing, they should not be overlooked by f_{sce}. Consequently, their corresponding labels in the pseudo label sequence are fixed to 1. However, we do not consider these specific tokens ([CLS] and [SEP]) as sentiment cues.

5. Experiments

5.1. Dataset

To rigorously evaluate the methodology proposed in this paper, we perform experiments using the IMDb [18] and SST-2 [19] datasets, both of which are sentiment classification datasets composed of English movie reviews. It is essential to note that BERT, the underlying model, is limited to processing sequences of a maximum of 512 tokens. Given that the IMDb dataset contains numerous instances exceeding this token limit, we selectively use instances with a length not surpassing 512 tokens for our experimental data.

Furthermore, we meticulously curate a subset of review data from the Yelp (https://www.yelp.com/dataset (accessed on 19 March 2024)) website. From the original Yelp dataset, we extract the top 14,000 reviews with the highest ratings and the bottom 14,000 reviews with the lowest ratings. After random swab, this dataset is divided into 20,000 reviews for training, 4000 for validation, and 4000 for testing.

To further extend the scope of our evaluation and validate the versatility of our methodology across different languages, we have incorporated the ChnSentiCorp dataset (https://aistudio.baidu.com/datasetdetail/10320 (accessed on 19 March 2024)). This dataset consists of Chinese-language hotel reviews, providing an opportunity to assess our model's performance in a non-English context.

The statistical characteristics of these four datasets are succinctly summarized in Table 1.

Table 1. This table shows the sizes of the training, validation, and test sets for four different datasets.

Dataset	Training Set Size	Validation Set Size	Testing Set Size
SST-2	60,000	7349	872
Yelp	20,000	4000	4000
IMDb	17,008	4310	21,500
ChnSentiCorp	9146	1200	1200

5.2. Experimental Setup

We initially train sentiment classification models for each of the three datasets. Then, for each instance X in the training and evaluation sets of each dataset, we generate candidate sequences using Algorithm 1. Since it is not possible to predict the proportion of tokens in X that are sentiment cues, denoted as p, we test different values of p from the set {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7}.

For both the sentiment classification model and the sentiment cue extraction model, we employed the bert-base-uncased (https://huggingface.co/bert-base-uncased (accessed on 19 March 2024)) architecture as our encoder and employed softmax [42] as the decoder. The learning rate is set to 0.00001, and we use the Adam optimizer with the applied cross-entropy loss function.

In the training phase of the SCE model, we primarily use cross-entropy loss as the main evaluation metric. We systematically selected the model parameters that achieved the minimum loss in the validation set as the final parameters of the model.

All computations are performed on a Tesla V100-SXM2-16GB GPU manufactured by NVIDIA Corporation, headquartered in Santa Clara, CA, USA. Due to variations in the maximum length of samples in the three datasets and limitations in GPU memory, the number of candidate sequences generated per run differed. Specifically, we generated 100 mask sequences for SST-2 and Yelp in a single run, while for IMDb and ChnSentiCorp, we could only generate 10 mask sequences per run.

In training the classification and SCE models, we adjust the batch size based on the dataset to optimize resource utilization and training efficiency. For SST-2 and Yelp, the batch size is set to 32, accommodating a larger number of instances per training step due to their relatively shorter text lengths. In contrast, for IMDb and ChnSentiCorp, which consist of longer text instances, the batch size is set to 8.

5.3. Evaluation Metrics

To assess the effectiveness of our SS-SCE approach, evaluations are conducted from both quantitative and qualitative perspectives.

5.4. Results and Analysis

5.4.1. Computational Efficiency of Monte Carlo Sampling

To assess the computational demands of our method, we performed Monte Carlo Sampling in the training and validation sets of the SST-2, Yelp, IMDb, and ChnSentiCorp datasets. We generated a fixed number of 10,000 candidate sequences for each instance.

To elucidate the computational efficiency of our Monte Carlo Sampling process, detailed statistics are presented in Table 2. This table shows the Average Time Per Sampling (ATPS) in milliseconds (ms) and the Average Time for the Optimal Mask Sequence (ATOMS) in seconds (s) for each dataset.

Table 2. This table presents the average length of each instance in four datasets, the average time consumed per sampling, and the average time to obtain the optimal mask sequence.

Metric	SST-2	Yelp	IMDb	ChnSentiCorp
Average Length	32.02	79.57	265.02	90.49
ATPS (ms)	0.70	3.71	9.84	6.03
ATOMS (s)	1.49	19.84	49.68	30.27

As evident from Table 2, the time required for generating a single sample increases with the length of the text, as does the average time to complete the sampling process for obtaining the optimal mask sequence. This outcome indicates that our approach is relatively less efficient for longer texts. As the length of the text increases, more time is required to complete the sampling process.

5.4.2. Main Performance Evaluation

Given the absence of annotated data, it is challenging to directly apply traditional sequence labeling evaluation metrics to assess SS-SCE. According to the definition of the SCE task, the sentiment orientation of X^Y, obtained by masking X with the pseudo-label Y, should align with that of X. Therefore, we can indirectly evaluate SS-SCE by comparing the performance metrics of instances in the test set when using X as input versus using X^Y as input in the sentiment classification model. Specifically, we calculate the accuracy, precision, recall, and F1 scores for the test set when using X and X^Y as inputs, respectively, and measure the performance loss caused by using X^Y as input.

Additionally, to statistically assess the impact of our SS-SCE method on the performance of sentiment classification, we conduct a t-test comparing the predictions made by the sentiment classification model for both the original input X and the input with extracted sentiment cues X^Y. The null hypothesis (H0) posits that the SS-SCE method does not significantly reduce the performance metrics of sentiment classification compared to the original input X. The alternative hypothesis (H1), on the other hand, suggests a significant reduction in these performance metrics, which would indicate an effect of the SS-SCE method. We set the confidence level for this test at 0.01, meaning a p-value less than 0.01 is required to reject the null hypothesis. Rejecting H0 would imply that the SS-SCE method significantly impacts the performance of the model, whereas failing to reject H0 would suggest that the SS-SCE method can extract sentiment cues without substantially compromising classification accuracy.

However, relying solely on this is not sufficient, as there could be special cases where all values of Y are 1, leading to $X^Y = X$. To avoid this scenario, we also evaluate using RCT, which is the proportion of sentiment cues extracted by SS-SCE relative to the original input.

To demonstrate the effectiveness and detailed impact of SS-SCE on sentiment classification accuracy, including any performance loss, Table 3 offers a comprehensive comparison. This table contrasts the performance metrics—accuracy, precision, recall, and F1 scores—for the original input (X) and the input with extracted sentiment cues (X^Y), across various datasets. It quantifies the performance loss incurred using X^Y as input and includes RCT to indicate the proportion of sentiment cues identified. Additionally, the table details the results of the t-test, providing statistical insight into the significance of the differences observed between the performances of X and X^Y.

For the SST-2 dataset, compared to the original input X, the prediction results using X^Y as input show a decrease across all major metrics, but the decrease is within 0.1, and the p-value from the t-test is greater than 0.01. This indicates that our SS-SCE method effectively extracts the majority of sentiment cues from the SST-2 dataset, albeit with some minor losses. The Ratio of Cue Tokens (RCT) is 0.1682, which means that tokens identified as sentiment cues by SS-SCE constitute 16.82% of the total in the SST-2 test set. This performance suggests that SS-SCE can extract sentiment cues without significantly compromising the accuracy of sentiment classification.

Table 3. This table displays accuracy, precision, recall, F1 scores, and performance loss for original (X) versus cue-extracted (X^Y) inputs across SST-2, Yelp, IMDb, and ChnSentiCorp datasets. RCT values and *t*-test results are also included to assess the extraction's effectiveness.

	SST-2			**Yelp**			**IMDb**			**ChnSentiCorp**		
Metric	X	X^Y	**Loss**	X	X^Y	**Loss**	X	X^Y	**Loss**	X	X^Y	**Loss**
Accuracy	0.9300	0.8719	0.0585	0.9885	0.9748	0.0138	0.9328	0.8798	0.0531	0.9369	0.8367	0.1002
Precision	0.9379	0.9072	0.0307	0.9876	0.9723	0.0153	0.9305	0.8333	0.0971	0.9387	0.7940	0.1448
Recall	0.9182	0.8224	0.0958	0.9895	0.9776	0.0120	0.9359	0.9501	−0.014	0.9372	0.9174	0.0198
F1	0.9280	0.8627	0.0652	0.9886	0.9749	0.0136	0.9332	0.8879	0.0453	0.9380	0.8512	0.0867
RCT	-	0.1682	-	-	0.3795	-	-	0.2858	-	-	0.3148	-
$p_{t\text{-test}}$		>0.01			>0.01			<0.01			<0.01	

For the Yelp dataset, the decline in metrics for X^Y is notably subtle, with all reductions less than 0.02. Furthermore, the *t*-test results reveal no significant differences in metrics between X^Y and X within this dataset. However, the relatively higher RCT indicates that SS-SCE may employ a more lenient criterion when extracting sentiment cues on the Yelp dataset.

Regarding the IMDb dataset, the results with X^Y as input show the highest decrease in accuracy and precision among the three datasets, while the impact on recall is the opposite, even surpassing the performance using X as input. This phenomenon could be attributed to longer texts containing more distracting information, which our SS-SCE method is adept at effectively filtering out. The relatively lower RCT value among the three datasets corroborates this observation. Furthermore, the higher recall rate for X^Y suggests that SS-SCE effectively extracts sentiment cues from X, improving the model's ability to identify relevant sentiment information. The *p*-value of the *t*-test being less than 0.01 indicates a significant difference in the sentiment classification results between X and X^Y. Coupled with the increase in recall, we consider this impact positive.

On the ChnSentiCorp dataset, the RCT is 0.3148, indicating that 31.48% of tokens in X were extracted as sentiment cues. In this context, the loss in recall is minimal, only 0.0198, suggesting that SS-SCE likely captures the majority of sentiment cues. However, compared to the IMDb dataset, the performance metrics on ChnSentiCorp are noticeably poorer. This indicates that our SS-SCE method may have certain limitations when processing Chinese data. This could be due to BERT's character-level processing of Chinese, whereas Chinese semantics are typically conveyed at the word level. Therefore, during the sampling process, words might be segmented into characters that fail to express complete semantics, thereby affecting the model's performance.

In summary, the experimental results prove that our SS-SCE method achieves good results on English datasets, especially on datasets with longer text lengths, where the extraction of sentiment cues is more effective. However, there are clear deficiencies in the Chinese dataset. In future research, we will consider addressing the issues encountered in the Chinese dataset.

5.4.3. Model Generalization Tests

To ascertain the adaptability and generalizability of our proposed method, we conduct cross-testing on three English datasets. Specifically, this involves using the model trained on each dataset to test the other two datasets. Additionally, we combine the datasets generated by the SS-SCE method from all three datasets to train a single sentiment cue extraction model, which is then tested on all three datasets.

Additionally, we merge the datasets sampled from the three English datasets to train collectively and conduct tests on each dataset individually. For the amalgamated dataset, we use the term "combined" to denote it.

In the cross-testing, we continue to use the same evaluation metrics as those presented in Table 3. It is noted that we use subscripts to denote the training dataset of the sentiment

extraction model. For example, $X^{Y}_{SST\text{-}2}$ represents the X^{Y} generated by the sentiment cue extraction model trained on the SST-2 dataset.

As shown in Table 4, when models trained on Yelp and IMDb datasets are tested on SST-2, they show a notable performance decline, particularly in accuracy and recall. The most pronounced drop is observed in the model trained on IMDb, with a 16.97% decrease in accuracy. This can be attributed to the disparity in text length and complexity between IMDb and SST-2 datasets. Although the precision of the IMDb-trained model remained relatively stable, indicating a consistent ability to identify true positives, the substantial decrease in recall, especially for this model, suggests challenges in capturing the full range of sentiment cues in shorter SST-2 texts.

Moreover, the recall of $X^{Y}_{combined}$ shows an improvement, indicating that the incorporation of Yelp and IMDb enhances the ability to extract sentiment cues. However, this integration also introduces additional information, which adversely affects the accuracy and precision of the model.

Table 4. This table shows the test results on the SST-2 dataset for models trained on SST-2, Yelp, IMDb, and the combined dataset.

Metric	X	$X^{Y}_{SST\text{-}2}$	$loss_{SST\text{-}2}$	X^{Y}_{Yelp}	$loss_{Yelp}$	X^{Y}_{IMDb}	$loss_{IMDb}$	$X^{Y}_{combined}$	$loss_{combined}$
Accuracy	0.9300	0.8716	0.0585	0.8234	0.1067	0.7603	0.1697	0.8039	0.1261
Precision	0.9379	0.9072	0.0307	0.8549	0.0830	0.9195	0.0184	0.7495	0.1884
Recall	0.9182	0.8224	0.0958	0.7710	0.1472	0.5607	0.3575	0.9234	-0.005
F1	0.9280	0.8627	0.0652	0.8108	0.1172	0.6967	0.2313	0.8274	0.1006
RCT	-	0.1682	-	0.1256	-	0.1012	-	0.1489	-

In Table 5, the adaptability of the models to the Yelp dataset is more promising. The decrease in accuracy and the F1 score is less severe compared to their performance in the SST-2 dataset. This implies that the models are better equipped to handle the moderate text lengths and complexity of Yelp reviews. However, the performance of the model trained on the IMDb dataset is significantly poorer, especially in terms of recall. Similarly, the model trained on the combined dataset also experiences some degree of performance degradation, which may be attributed to the influence of the IMDb dataset.

Table 5. This table shows the test results on the Yelp dataset for models trained on SST-2, Yelp, IMDb, and the combined dataset.

Metric	X	$X^{Y}_{SST\text{-}2}$	$loss_{SST\text{-}2}$	X^{Y}_{Yelp}	$loss_{Yelp}$	X^{Y}_{IMDb}	$loss_{IMDb}$	$X^{Y}_{combined}$	$loss_{combined}$
Accuracy	0.9885	0.9650	0.0235	0.9748	0.0138	0.9260	0.0625	0.9655	0.0230
Precision	0.9876	0.9722	0.0154	0.9723	0.0153	0.9853	0.0023	0.9626	0.0250
Recall	0.9895	0.9577	0.0319	0.9776	0.0120	0.8655	0.1240	0.9484	0.0411
F1	0.9886	0.9649	0.0237	0.9749	0.0136	0.9215	0.0670	0.9554	0.0332
RCT	-	0.1397	-	0.3795	-	0.2698	-	0.2773	-

Table 6 indicates that models trained on shorter text datasets, such as SST-2 and Yelp, also perform effectively on the IMDb dataset, positively influencing accuracy. However, there is a negative impact on recall. This suggests that while the models retain their ability to correctly identify true positives in the context of longer texts, their capacity to capture the full range of sentiment cues across the broader dataset is somewhat diminished.

These results indicate that while models trained on shorter texts, such as SST-2, exhibit relatively better generalization capabilities across datasets, models trained on datasets with longer texts, such as IMDb, show limited adaptability to shorter texts. Additionally, when conducting cross-dataset experiments, training on a combination of multiple datasets, although generally not outperforming training on their own respective datasets, tends to yield better results than training on any single, different dataset. This implies that when

extending SS-SCE to new data, considering training across multiple similar datasets could enhance model performance. This strategy may leverage the diverse characteristics of each dataset to build a more robust and adaptable model.

Table 6. This table shows the test results on the IMDb dataset for models trained on SST-2, Yelp, IMDb, and the combined dataset.

Metric	X	$X^{Y}_{SST\text{-}2}$	$loss_{SST\text{-}2}$	X^{Y}_{Yelp}	$loss_{Yelp}$	X^{Y}_{IMDb}	$loss_{IMDb}$	$X^{Y}_{combined}$	$loss_{combined}$
Accuracy	0.9328	0.8925	0.0403	0.8709	0.0620	0.8798	0.0531	0.8934	0.0394
Precision	0.9305	0.9205	0.0099	0.8564	0.0761	0.8333	0.0971	0.9327	−0.0020
Recall	0.9359	0.8598	0.0762	0.8918	0.0441	0.9501	−0.014	0.8474	0.0921
F1	0.9332	0.8891	0.0441	0.8737	0.0594	0.8879	0.0453	0.8880	0.0452
RCT	-	0.1153	-	0.1186	-	0.2858	-	0.2478	-

5.5. Case Study: Comparing SS-SCE with Established Interpretability Methods

To evaluate the unique contributions and effectiveness of SS-SCE, we perform a comparative analysis with established interpretability methods in text classification, including LIME [5], LIG [43], OCC [44], SVS [45], and LDS [46].

For this comparison, we use the Thermostat tool (https://github.com/DFKI-NLP/thermostat (accessed on 19 March 2024)) [47], which integrates state-of-the-art interpretability methods, offering a unified platform for analysis. This tool allowed us to apply these methods in a standardized way, ensuring a fair and consistent comparison between different interpretability approaches.

Our analysis aimed not to compare SS-SCE directly with these methods, but to showcase how SS-SCE's focused approach on sentiment cues provides a different, potentially more nuanced perspective in understanding model decisions, especially in the context of sentiment analysis.

Using Thermostat, we applied interpretability models trained on various datasets, such as IMDb with pre-trained language models such as BERT and ALBERT [48]. For a fair comparison, we chose the interpretability model trained with BERT on the IMDb dataset. To facilitate a comparison with SOTA methods, we manually annotated two selected instances, a positive and a negative, from the IMDb test set. We then calculated the precision, recall, and F1 score for each method's sentiment cue extraction on these annotated instances. The results of this comparative analysis are presented in Tables 7 and 8.

Table 7 shows that the SS-SCE models, particularly SS-SCE$_{SST\text{-}2}$, demonstrate superior performance in extracting sentiment cues from the positive text instance when compared with SOTA interpretability methods, SS-SCE$_{SST\text{-}2}$ achieved the highest precision of 0.7778, recall of 0.8235, and F1 score of 0.8000, indicating a robust capability in accurately identifying and recalling relevant sentiment cues.

The SS-SCE models trained on Yelp and IMDb datasets showed varying degrees of effectiveness, with SS-SCE$_{Yelp}$ displaying moderate performance and SS-SCE$_{IMDb}$, showing decent accuracy but lower effectiveness compared to SS-SCE$_{SST\text{-}2}$. This variation suggests the influence of training data characteristics on the model's performance.

In contrast, the standard interpretability methods, while useful in their own right, exhibited lower performance metrics in comparison. LIME, LIG, OCC, SVS, and LDS demonstrated lower precision, recall, and F1 scores, indicating a potential limitation in their ability to capture the nuanced sentiment cues as effectively as the SS-SCE approach.

Table 8 presents the performance of different interpretability methods in extracting sentiment cues from the negative text instance. The results indicate that the SS-SCE models, particularly SS-SCE$_{IMDb}$ and SS-SCE$_{SST-2}$, perform effectively in this context, albeit with some variations in precision and recall.

Table 7. This table shows the performance comparison of our Self-Supervised Sentiment Cue Extraction (SS-SCE) model trained on three datasets with SOTA interpretability methods on a positive instance.

Method	Result	Precision	Recall	F1
human	This is a **great horror movie**. **Great plot**. And a person with a fear of midgets will definately **love** the evil midget! This is a **must see** for **any horror fan**. Finally a lower budget movie with **decent effects** and a **great cast**! **Highly recommended**.	-	-	-
SS-SCE$_{SST\text{-}2}$	This is a **great** horror **movie**. **Great plot**. And a person with a fear of midgets will definately **love** the evil midget! **This** is a **must see** for **any** horror **fan**. Finally a lower budget **movie with decent** effects **and** a **great cast**! **Highly recommended**.	**0.7778**	**0.8235**	**0.8000**
SS-SCE$_{Yelp}$	**This** is a **great** horror movie. **Great plot**. **And** a person with a fear of midgets will definately love the evil midget! **This** is **a must see for any** horror fan. **Finally a** lower budget movie **with decent** effects **and a great cast**! **Highly recommended**.	0.5556	0.6250	0.5882
SS-SCE$_{IMDb}$	**This is a great** horror movie**. Great** plot. And a person with a fear of midgets will definately **love** the evil midget! **This is a must** see for any horror fan. Finally a lower budget movie with **decent** effects and a **great** cast! **Highly recommended**.	0.6154	0.4706	0.5333
LIME	This **is** a great horror **movie**. **Great** plot. And a **person** with a fear of midgets will definately love the **evil midget**! This is a **must see** for any horror **fan**. **Finally** a **lower budget movie** with decent effects and a great cast! **Highly** recommended.	0.4286	0.3529	0.3871
LIG	This **is a great** horror **movie**. **Great** plot. **And** a person **with** a fear of **midgets** will definately **love** the evil midget! This **is** a **must** see for any horror **fan**. **Finally** a lower budget movie **with** decent effects and **a great** cast! **Highly recommended**.	0.5000	0.5294	0.5143
OCC	**This** is a great horror movie. Great **plot**. And a person with **a fear of midgets** will definately **love the evil midget**! This is a must see **for any horror** fan. Finally **a lower budget movie with decent effects** and **a great cast! Highly** recommended.	0.3600	0.5294	0.4286
SVS	**This is a great** horror **movie**. **Great plot**. **And** a **person** with a **fear** of midgets will definately **love the** evil midget**!** This is **a must** see for any horror **fan**. Finally a lower budget **movie** with decent effects **and** a **great** cast**! Highly recommended**.	0.4545	0.5882	0.5128
LDS	This is a great **horror movie**. Great plot. And a person with a fear of **midgets will definately love the evil** midget**!** This is a must see for any **horror** fan**.** Finally **a lower budget movie** with decent **effects** and a great **cast**! Highly **recommended**.	0.4118	0.4118	0.4118

The bold tokens represent the extracted sentiment cues.

SS-SCE$_{IMDb}$ achieved the highest precision (0.8571), reflecting its strong ability to accurately identify relevant negative sentiment cues. However, its recall (0.3750) is relatively lower, suggesting that, while it is precise, it may miss some relevant cues. Conversely, SS-SCE$_{SST-2}$, with a recall of 0.5000, demonstrates a balanced performance with a precision of 0.5714 and an F1 score of 0.5333. This balance indicates its ability to capture a broader range of relevant cues while maintaining accuracy.

SS-SCE$_{Yelp}$, despite having the highest precision (0.8333), shows a lower recall (0.3125), indicating a tendency to be very selective in cue extraction, which may lead to missing some pertinent sentiment indicators.

In comparison, traditional interpretability methods show lower performance in both precision and recall. LIME and LDS, in particular, demonstrate limited effectiveness in accurately identifying negative sentiment cues. The lower performance of these methods may be attributed to their design, which might not be as fine-tuned for sentiment cue extraction as the SS-SCE approach.

Overall, the comparative analysis of sentiment cue extraction presented in Tables 7 and 8 demonstrates the robustness and versatility of the SS-SCE models across both positive and negative text instances. The SS-SCE models, especially SS-SCE$_{SST-2}$, consistently exhibit a balanced performance in terms of precision and recall, highlighting their ability to accurately and comprehensively extract sentiment cues. This is particularly evident in SS-SCE$_{SST-2}$, which shows strong performance in both positive and negative contexts. While SS-SCE$_{IMDb}$ and SS-SCE$_{Yelp}$ demonstrate higher precision in specific in-

stances, they sometimes compromise on recall, indicating a more selective extraction of cues. In comparison to the SOTA interpretability methods, the SS-SCE approach stands out for its enhanced capability to identify both explicit and subtle sentiment indicators.

Table 8. This table shows the performance comparison of our Self-Supervised Sentiment Cue Extraction (SS-SCE) model trained on three datasets with SOTA interpretability methods on a negative instance.

Method	Result	Precision	Recall	F1
human	**Unfortunately**, this movie is **absolutely terrible**. It's **not even laughably bad**, **just plain bad**. The actors do their best with what is **the cheesiest script ever**. **How scary** can a movie be when the climax actually involves a roomful of millions of styrofoam peanuts?	-	-	-
SS-SCE$_{SST\text{-}2}$	**Unfortunately**, this movie is absolutely **terrible**. It's **not** even laughably **bad**, just plain **bad**. The actors **do** their **best** with what is the **cheesiest** script ever. **How scary** can a movie be when the climax actually **involves** a **roomful** of millions of **styrofoam peanuts**?	0.5714	**0.5000**	**0.5333**
SS-SCE$_{Yelp}$	**Unfortunately**, this movie is absolutely **terrible**. It's not even laughably **bad**, just plain **bad**. The actors do their best with what is the cheesiest script ever. How **scary** can a movie be when the climax actually involves a roomful of millions of styrofoam **peanuts**?	0.8333	0.3125	0.4545
SS-SCE$_{IMDb}$	**Unfortunately**, this movie is **absolutely terrible**. It's not even laughably **bad**, just plain **bad**. The actors do their **best** with what is the **cheesiest** script ever. How scary can a movie be when the climax actually involves a roomful of millions of styrofoam peanuts?	**0.8571**	0.3750	0.5217
LIME	**Unfortunately, this** movie is **absolutely terrible**. **It**'s not **even** laughably bad, just plain bad. The actors do their best with what is the **cheesiest** script ever. How scary can a **movie be** when **the climax** actually **involves a roomful** of millions of **styrofoam** peanuts?	0.3125	0.3125	0.3125
LIG	**Unfortunately, this** movie **is absolutely terrible**. It's **not** even laughably **bad,** just plain **bad.** The actors do their best with what is the cheesiest script ever. How scary can a **movie** be when the climax actually involves a roomful of millions of styrofoam peanuts?	0.5455	0.3750	0.4444
OCC	Unfortunately, **this movie is absolutely terrible.** It's not even laughably bad, just plain bad. The actors do their best with what is the cheesiest script ever. How scary can a movie be when the climax actually involves a roomful of millions of styrofoam **peanuts?**	0.2857	0.1250	0.1739
SVS	**Unfortunately**, this movie is absolutely **terrible**. It's not **even** laughably **bad**, **just** plain **bad**. The actors do their best with what is the cheesiest script ever. How scary can a movie be when the climax actually involves a roomful of millions of styrofoam peanuts**?**	**0.8571**	0.3750	0.5217
LDS	**Unfortunately,** this **movie** is absolutely **terrible**. It's not even laughably bad, just plain bad. The actors **do** their best with what is the **cheesiest** script ever**. How** scary can a **movie** be when the climax actually involves a roomful of millions of styrofoam peanuts?	0.5000	0.2500	0.3333

The bold tokens represent the extracted sentiment cues.

Simultaneously, it is important to note that our approach represents a global interpretability method, which significantly outperforms traditional techniques in terms of efficiency when applied to new data. This global perspective enables a comprehensive understanding of the model's decision-making process across various datasets and scenarios, rather than focusing on individual instances.

5.6. Ablation Study on MSIS

To validate the effectiveness and contribution of each component within the MSIS, we conduct an ablation study. This study systematically examines how the removal or alter-

ation of each MSIS component affects the overall performance of our SS-SCE framework. The components of MSIS are as follows.

Probability Discrepancy (PD): This component, denoted as ΔP^{Y}, assesses the clarity of sentiment cues within the candidate sequence. It ensures that elements marked with 1 in the candidate sequence effectively contribute to the sentiment classification model's decision-making process.

Inverse Probability Discrepancy (IPD): Represented as $\Delta P^{\bar{Y}}$, it evaluates the absence of sentiment cues within the inverse attention mask $\bar{Y}$.This ensures elements marked with 0 in Y do not contribute significantly to sentiment interpretation, emphasizing the specificity of extracted cues.

Ratio of Cue Tokens (RCT): This component aims to minimize the inclusion of irrelevant tokens in the candidate sequence, promoting a concise extraction of sentiment cues. It is calculated as the proportion of 1s in Y^{c}, with a higher RCT indicating a more focused extraction of sentiment cues.

The results of our ablation study are summarized in Tables 9–12 . Each row represents a variant of the MSIS, indicating the presence (+) or absence (-) of each component. Performance metrics include the accuracy, precision, recall, and F1 score of the sentiment cue extraction under each variant.

Table 9. This table shows the ablation study results for Mask Sequence Interpretation Score (MSIS) components of SST-2 dataset.

PD	IPD	RCT	Accuracy	Precision	Recall	F1	RCT
+	+	+	0.8716	0.9072	0.8224	0.8627	0.1682
-	+	+	0.7879	0.7595	0.8536	0.8038	0.1405
+	-	+	0.7397	0.6764	0.9369	0.7856	0.0540
+	+	-	0.8807	0.8586	0.9167	0.8867	0.5214

Table 10. This table shows the ablation study results for Mask Sequence Interpretation Score (MSIS) components of Yelp dataset.

PD	IPD	RCT	Accuracy	Precision	Recall	F1	RCT
+	+	+	0.9748	0.9723	0.9776	0.9749	0.3795
-	+	+	0.9635	0.9446	0.9844	0.9641	0.3928
+	-	+	0.8395	0.7614	0.9869	0.8596	0.0181
+	+	-	0.9838	0.9854	0.9819	0.9837	0.8910

Table 11. This table shows the ablation study results for Mask Sequence Interpretation Score (MSIS) components of IMDb dataset.

PD	IPD	RCT	Accuracy	Precision	Recall	F1	RCT
+	+	+	0.8798	0.8333	0.9501	0.8879	0.2858
-	+	+	0.8654	0.9529	0.7728	0.8535	0.1542
+	-	+	0.6207	0.9586	0.2637	0.4136	0.0038
+	+	-	0.9261	0.9176	0.9385	0.9279	0.8547

Table 12. This table shows the ablation study results for Mask Sequence Interpretation Score (MSIS) components of ChnSentiCorp dataset.

PD	IPD	RCT	Accuracy	Precision	Recall	F1	RCT
+	+	+	0.8367	0.7940	0.9174	0.8512	0.3148
-	+	+	0.8746	0.8444	0.9240	0.8824	0.5042
+	-	+	0.6641	0.6392	0.7818	0.7033	0.0198
+	+	-	0.9234	0.9227	0.9273	0.9250	0.8676

As shown in Tables 9–12 , the ablation study systematically evaluates the contribution of each component within the MSIS on the SST-2, Yelp, IMDb, and ChnSentiCorp datasets. This study offers a nuanced understanding of how each element influences the framework's ability to extract and utilize sentiment cues.

Removing the PD component results in performance degradation across most metrics, particularly evident in the reduction of precision and the F1 score. This suggests that PD is crucial for identifying clear sentiment cues within the text, ensuring that the elements marked as sentiment cues in the candidate sequence contribute effectively to the decision-making process of the sentiment classification model. However, on the Chinese dataset, the performance after removing the PD component is slightly better than the overall performance with the complete MSIS. This may be attributed to the fact that the Chinese language processes characters as the smallest units, rather than words. It is important to note that while the removal of PD results in a decrease in RCT by 0.1894, the F1 score only drops by 0.0312, illustrating the effectiveness of our method.

The absence of the IPD leads to a significant decrease in recall and a noticeable drop in the RCT, indicating a diminished ability to exclude non-sentiment-related tokens from being marked as sentiment cues. This highlights the IPD's role in refining the specificity of extracted cues by ensuring that elements marked with 0 in Y do not significantly contribute to sentiment interpretation.

Removing RCT results in an improvement in sentiment classification performance but at the cost of a substantial increase in RCT. This implies that while the RCT component restricts the inclusion of irrelevant tokens in the candidate sequence, its absence leads to a wider selection of tokens as sentiment cues, including potentially irrelevant ones.

In summary, each component of the MSIS plays a vital role in the sentiment cue extraction process. PD ensures the clarity and relevance of cues, IPD enhances the specificity of cue extraction, and RCT promotes conciseness and focus. The ablation study demonstrates the delicate balance between these components, underscoring their collective contribution to the effectiveness of the SS-SCE framework.

6. Conclusions

In conclusion, our research introduces a novel self-supervised framework for sentiment cue extraction that significantly improves the interpretability of sentiment analysis models. Through meticulous identification and extraction of key linguistic elements that influence sentiment determination, our approach demystifies the decision-making process of sentiment analysis models, thereby fostering greater trust and understanding in these systems.

Our innovative use of Monte Carlo Sampling for efficient cue identification and the development of the Mask Sequence Interpretation Score (MSIS) metric to evaluate the extraction of sentiment cues represent substantial advances in the field of sentiment analysis. Importantly, our methodology extends beyond traditional local interpretability techniques, providing a global interpretability approach that enhances understanding across various instances and datasets. The application of our method in diverse datasets, such as SST-2, Yelp, IMDb, and ChnSentiCorp, demonstrates its effectiveness in extracting pertinent sentiment cues.

However, our study is not without its limitations. The computational demands of our approach, especially in handling longer texts, highlight the need for further optimization to enhance efficiency without sacrificing accuracy. Additionally, while our method shows promising results in extracting sentiment cues, the performance variability across different text lengths and complexities suggests room for improvement in the generalizability and adaptability of the model. Furthermore, when processing Chinese data, our method faces additional challenges. This is partly due to BERT's character-level processing of Chinese, whereas Chinese semantics are more accurately represented at the word level. Consequently, during sampling, words may be segmented into characters that fail to convey full semantics, affecting the model's performance. This aspect highlights the

importance of tailoring our approach to better accommodate the linguistic characteristics of Chinese, suggesting a direction for future research to improve the method's applicability and effectiveness in handling Chinese texts.

Looking forward, we see several avenues for future research. Enhancing the computational efficiency of our Monte Carlo Sampling process and exploring alternative sampling techniques could address current limitations in processing longer texts. Further refinement of the MSIS metric to better balance accuracy and interpretability could also produce improvements in sentiment cue extraction. Moreover, extending our framework to incorporate multimodal data (text, images, and videos) could offer a more holistic approach to sentiment analysis, reflecting the multifaceted nature of sentiment expression across various media. Then, addressing the specific challenges of processing Chinese data, such as adapting our approach to better capture the word-level semantics often lost in character-level processing, also constitutes a critical area for future exploration. This would not only improve the model's performance on Chinese texts but also enhance its applicability and effectiveness across linguistically diverse datasets.

Ultimately, our work contributes to the ongoing efforts to bridge the gap between advanced sentiment analysis techniques and their interpretability, aiming to create more transparent, reliable, and user-friendly NLP models. By emphasizing global interpretability, our approach offers a scalable and comprehensive solution for understanding complex sentiment analysis models. By continuing to refine and expand upon the foundations laid by this study, we anticipate contributing to the development of sentiment analysis models that are not only highly accurate but also thoroughly interpretable, ensuring their ethical and effective application in sensitive domains.

Author Contributions: Conceptualization, Y.S.; methodology, Y.S., S.H. and X.H.; software, Y.S.; validation, Y.S.; formal analysis, Y.S., X.H. and Y.L.; investigation, Y.S.; resources, Y.S.; data curation, Y.S. and X.H.; writing—original draft preparation, Y.S.; writing—review and editing, Y.S., S.H., X.H. and Y.L.; visualization, Y.S.; supervision, S.H.; project administration, Y.S.; funding acquisition, S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the National Natural Science Foundation of China under Grant No. 71974187.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The research data can be found at https://drive.google.com/drive/folders/1tHogTUtHC5sqLCS2bCNYpYTJayQ-1WnS (accessed on 19 March 2024).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, B. *Sentiment Analysis and Opinion Mining*; Springer Nature: Berlin/Heidelberg, Germany, 2022.
2. Pang, B.; Lee, L. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.* **2008**, *2*, 1–135. [CrossRef]
3. Wankhade, M.; Rao, A.C.S.; Kulkarni, C. A survey on sentiment analysis methods, applications, and challenges. *Artif. Intell. Rev.* **2022**, *55*, 5731–5780. [CrossRef]
4. Gilpin, L.H.; Bau, D.; Yuan, B.Z.; Bajwa, A.; Specter, M.; Kagal, L. Explaining explanations: An overview of interpretability of machine learning. In Proceedings of the 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA), Turin, Italy, 1–3 October 2018; IEEE: New York, NY, USA, 2018; pp. 80–89.
5. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.
6. Chiong, R.; Fan, Z.; Hu, Z.; Dhakal, S. A novel ensemble learning approach for stock market prediction based on sentiment analysis and the sliding window method. *IEEE Trans. Comput. Soc. Syst.* **2022**, *10*, 2613–2623. [CrossRef]
7. McCarthy, S.; Alaghband, G. Enhancing Financial Market Analysis and Prediction with Emotion Corpora and News Co-Occurrence Network. *J. Risk Financ. Manag.* **2023**, *16*, 226. [CrossRef]

8. Bharti, S.K.; Tratiya, P.; Gupta, R.K. Stock Market Price Prediction through News Sentiment Analysis & Ensemble Learning. In Proceedings of the 2022 IEEE 2nd International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC), Odisha, India, 15–17 December 2022; IEEE: New York, NY, USA, 2022; pp. 1–5.
9. Greaves, F.; Ramirez-Cano, D.; Millett, C.; Darzi, A.; Donaldson, L. Use of sentiment analysis for capturing patient experience from free-text comments posted online. *J. Med. Int. Res.* **2013**, *15*, e2721. [CrossRef]
10. Nauta, M.; Trienes, J.; Pathak, S.; Nguyen, E.; Peters, M.; Schmitt, Y.; Schlötterer, J.; van Keulen, M.; Seifert, C. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable AI. *ACM Comput. Surv.* **2023**, *55*, 1–42. [CrossRef]
11. Madsen, A.; Reddy, S.; Chandar, S. Post-hoc interpretability for neural nlp: A survey. *ACM Comput. Surv.* **2022**, *55*, 1–42. [CrossRef]
12. Saeed, W.; Omlin, C. Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. *Knowl.-Based Syst.* **2023**, *263*, 110273. [CrossRef]
13. Yue, L.; Chen, W.; Li, X.; Zuo, W.; Yin, M. A survey of sentiment analysis in social media. *Knowl. Inf. Syst.* **2019**, *60*, 617–663. [CrossRef]
14. Zhang, L.; Wang, S.; Liu, B. Deep learning for sentiment analysis: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1253. [CrossRef]
15. Liu, Y. Fine-tune BERT for extractive summarization. *arXiv* **2019**, arXiv:1903.10318.
16. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
17. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
18. Maas, A.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 142–150.
19. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.Y.; Potts, C. Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1631–1642.
20. Kaur, R.; Kautish, S. Multimodal sentiment analysis: A survey and comparison. In *Research Anthology on Implementing Sentiment Analysis Across Multiple Disciplines*; IGI Global: Hershey, PA, USA, 2022; pp. 1846–1870.
21. Liu, X.; Zhang, F.; Hou, Z.; Mian, L.; Wang, Z.; Zhang, J.; Tang, J. Self-supervised learning: Generative or contrastive. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 857–876. [CrossRef]
22. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; Le, Q.V. XLNet: Generalized autoregressive pretraining for language understanding. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
23. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7871–7880.
24. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 5485–5551.
25. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018. Available online: https://www.mikecaptain.com/resources/pdf/GPT-1.pdf (accessed on 19 March 2024).
26. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *Openai Blog* **2019**, *1*, 9.
27. Yang, J.; Jin, H.; Tang, R.; Han, X.; Feng, Q.; Jiang, H.; Zhong, S.; Yin, B.; Hu, X. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Trans. Knowl. Discov. Data* **2023**, *epub ahead of print*. [CrossRef]
28. Tian, S.; Jin, Q.; Yeganova, L.; Lai, P.T.; Zhu, Q.; Chen, X.; Yang, Y.; Chen, Q.; Kim, W.; Comeau, D.C.; et al. Opportunities and challenges for ChatGPT and large language models in biomedicine and health. *Brief. Bioinform.* **2024**, *25*, bbad493. [CrossRef]
29. Chen, Y.P.; Lo, Y.H.; Lai, F.; Huang, C.H. Disease concept-embedding based on the self-supervised method for medical information extraction from electronic health records and disease retrieval: Algorithm development and validation study. *J. Med. Int. Res.* **2021**, *23*, e25113. [CrossRef]
30. Feldman, R.; Rosenfled, B.; Soderland, S.; Etzioni, O. Self-supervised relation extraction from the web. In Proceedings of the Foundations of Intelligent Systems: 16th International Symposium, ISMIS 2006, Bari, Italy, 27–29 September 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 755–764.
31. Doshi-Velez, F.; Kim, B. Towards A Rigorous Science of Interpretable Machine Learning. *Stat* **2017**, *1050*, 2.
32. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
33. Wheeler, J.M.; Cohen, A.S.; Wang, S. A Comparison of Latent Semantic Analysis and Latent Dirichlet Allocation in Educational Measurement. *J. Educ. Behav. Stat.* **2023**, 10769986231209446. [CrossRef]

34. Xiong, J.; Li, F. Bilevel Topic Model-Based Multitask Learning for Constructed-Responses Multidimensional Automated Scoring and Interpretation. *Educ. Meas. Issues Pract.* **2023**, *42*, 42–61. [CrossRef]
35. Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **2019**, *1*, 206–215. [CrossRef] [PubMed]
36. Hammersley, J. *Monte Carlo Methods*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
37. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT press: Cambridge, MA, USA, 2016.
38. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
39. Betancourt, M. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv* **2017**, arXiv:1701.02434.
40. Shapiro, A. Monte Carlo sampling methods. In *Handb. Oper. Res. Manag. Sci.* **2003**, *10*, 353–425.
41. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
42. Bridle, J.S. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing: Algorithms, Architectures and Applications*; Springer: Berlin/Heidelberg, Germany, 1990; pp. 227–236.
43. Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic attribution for deep networks. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 3319–3328.
44. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part I 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 818–833.
45. Castro, J.; Gómez, D.; Tejada, J. Polynomial calculation of the Shapley value based on sampling. *Comput. Oper. Res.* **2009**, *36*, 1726–1730. [CrossRef]
46. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
47. Feldhus, N.; Schwarzenberg, R.; Möller, S. Thermostat: A Large Collection of NLP Model Explanations and Analysis Tools. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Virtual Event, 7–11 November 2021; Adel, H.; Shi, S., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2021.
48. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv* **2019**, arXiv:1909.11942.

applied sciences

MDPI

Article

Aspect Sentiment Triplet Extraction Based on Deep Relationship Enhancement Networks

Jun Peng [1] and Baohua Su [1,2,*]

[1] School of Education, City University of Macau, Macau 999078, China; junpeng@cityu.edu.mo
[2] College of Chinese Language and Culture, Jinan University, Guangzhou 510632, China
* Correspondence: subaohua@hwy.jnu.edu.cn

Abstract: The task of aspect-based sentiment analysis (ASBA) is to identify all the sentiment analyses expressed by specific aspect words in the text. How to identify specific objects (i.e., aspect words), describe the modifiers of the specific objects (i.e., opinion words), and judge the sentiment analysis expressed by opinion words (sentimental classification) in one step has become a focus of research in ASBA. ASTE (Aspect Sentiment Triplet Extraction) based on DREN (Deep Relationship Enhancement Networks) has been proposed in this paper. It aims to extract the aspect words and opinion words in the review text in one-step. They can judge the sentiment analysis expressed by the opinion words. Therefore, the study defines ten kinds of word relations; then, the study uses the parts of the speech feature, syntactic feature, relative position feature and tree distance relative feature to enhance the word representation relationship, which enriches the table of information in the relational matrix. Secondly, based on the word representation of BERT and GCN, the structural information of the texts are extracted; then, further extraction of higher-level word semantic information and word relationship information through SWDA (Sliding Window Dilated Attention) occurs, as SWDA can capture the multi-granularity relationship in words. Finally, the experimental results show that the proposed method is effective.

Keywords: triplet extraction; Graph Neural Networks; attention mechanism

Citation: Peng, J.; Su, B. Aspect Sentiment Triplet Extraction Based on Deep Relationship Enhancement Networks. *Appl. Sci.* **2024**, *14*, 2221. https://doi.org/10.3390/app14052221

Academic Editors: Xiangjie Kong, Wei Wang and Han Liu

Received: 28 January 2024
Revised: 2 March 2024
Accepted: 3 March 2024
Published: 6 March 2024

1. Introduction

In recent years, the advancements in deep learning have led to the widespread application of sentiment analysis technology across various fields and platforms. Public opinion platforms enable governments to influence the course of events by monitoring netizens' opinions on specific matters. Similarly, on consumer platforms, businesses can enhance their products by conducting targeted improvement based on the analysis of consumer reviews, considering sentiment analysis [1]. Sentiment analysis can be conducted at different levels, including sentence, paragraph, and fine-grained sentiment analyses [1,2]. Sentence-based sentiment analysis aims to identify the overall sentiment conveyed within a single sentence, while paragraph-based sentiment analysis determines the overall sentiment expressed in a paragraph by comprehensively analyzing each sentence [3]. In contrast to sentence-based and paragraph-based sentiment analyses, aspect-based sentiment analysis falls under the category of fine-grained sentiment analysis. This approach seeks to identify the sentiment associated with different aspect words in text, offering greater research opportunities and application value. For example, in e-commerce scenarios, users often provide mixed reviews regarding multiple aspects of a product. Similarly, in movie reviews, users may express diverse opinions on aspects such as dubbing, plot, and casting. In such common yet slightly complex scenarios, sentence- and paragraph-based sentiment analyses prove insufficient. Hence, aspect-based sentiment analysis becomes particularly important.

Attribute-based sentiment analysis, also known as ABSA (aspect-based sentiment analysis), focuses on analyzing the sentiment orientation expressed towards multiple

aspects or aspects of a thing in a comment. For example, in the statement "The mobile operating system is very smooth, but the battery performance is not good", the user assesses both the operating system and the phone's battery, each with different sentiment orientations. Such evaluations of multiple aspects with diverse sentiment orientations are common in e-commerce scenarios and public opinion analyses, making aspect-based sentiment analysis a topic of considerable interest in academia and the business community.

In recent years, with the progress in deep learning, aspect-based sentiment analysis has witnessed significant advancement. Some studies have proposed the use of attention mechanisms and related variants to process ABSA tasks. Tang et al. [4] and Wang et al. [5] proposed using attention mechanism networks to determine the sentiment polarity associated with a given aspect term in the text. Ma et al. [6] suggested that there is a mutual relationship between aspect words and their context, introducing an interactive attention mechanism to give weights to the words in the context and aspect text. Li et al. [7] suggested using collaborative attention mechanism to associate aspect words with their context while concatenating the term, embedding the context of the term with the aspect words, enhancing the interaction between aspect words and their contexts.

The advent of models like BERT (Bidirectional Encoder Representations from Transformers) and GCN (Graph Neural Networks) has further fueled the emergence of a large number of models based on BERT and GCN that are widely used in ABSA. BERT was presented in [8], which used mask tasks and predict next sentence task to learn the semantic information connecting words and sentences. So, Pang et al. [9] used BERT to extract more auxiliary information about aspect words from the context of comments, while Zhou et al. [10] used the syntactic relationship in sentences to construct a graph, and then they used GCN to determine sentiment polarity expressed by aspect words in context. GCN [11] realized the convolution operation on the topological graph with the help of graph theory, and semantic information was extracted via graph instruction and convolution.

The above studies were all based on the given attributive words and the contextual information to judge the sentiment polarity expressed by the attributive words in the text. At present, great progress is being made in ASTE for given aspect words. However, a pressing and challenging issue that requires urgent attention is how to determine the polarity of aspect words and sentiment words in one step without specifying pre-existing aspect words. In this scenario, where aspect words are not specified in advance, it is necessary to first identify aspect words and opinion words and then combine aspect words and opinion words to obtain aspect words and their corresponding sentiment polarity (The aspect words, opinion words and sentiment polarity were referred to as triplets in this paper). For example, consider the sentence "The taste is good but the service could be better". In this case, the aspect words are "taste" and "the service", while the opinion words are "good" and "could be better". Consequently, the triplets derived from the above sentence are ("taste", "good", "positive") and ("the service", "could be better", "negative"). It is worth noting that obtaining all these triplets from a comment sentence in a single step poses a more intricate challenge compared with determining the sentiment polarity of aspect words when the aspects are specified.

Therefore, the research questions considered in this study are as follows:

Research Question 1: What method can be used to obtain the polarity (triplet) of aspect and sentiment in one step without giving the aspect?

Research Question 2: How can we solve the problem of possible discontinuity between aspect and opinion in ASBA?

Research Question 3: What is the effect of the method derived from research question 1?

To solve the problem of extracting triplets from comments, Chen et al. [12] introduces a methodology involving the construction of a term relationship matrix from all the words present in the comments. Triplets are then extracted based on the assessment of these term relationships. Based on Chen et al. [12], this paper introduces DREN designed to evaluate term relationships.

To address the above three issues, we proposed DREN. Our contribution is as follows: first, we will transform the problem of extracting triplets into a word relationship classification problem and define ten's word relationships, which obtain the polarity (triplet) of aspect and sentiment in one step. In words with more details, this paper employs BERT to extract term information from the text to identify term relationships comprehensive, solves the problem of nested entity term embedding, and obtains the relationship information between words through the LBAM (Linear Biaffine Attention Mechanism) and GCN. Secondly, since aspect words and opinion words may appear non-sequentially in the sentence, SWDA is employed to effectively capture the relationships between words situated at varying distances from each other. Thirdly, this paper also uses the part-of-speech features, syntactic features, relative position features, and tree distance relative features in the comment to fully utilize the information in the comment text and integrates these features with the text representation. The experimental results on a large number of publicly available ASTE-related datasets demonstrate the effectiveness of the proposed model.

2. Review of Literature

ABSA (aspect-based sentiment analysis) is a fine-grained sentiment analysis that involves three sequential tasks: aspect extraction, opinion extraction, and sentiment polarity determination for aspect–opinion pairs. While existing works often focus on individual tasks such as sentiment polarity with aspect words specified, there are also some works on AOTE (Aspect and Opinion Term Co-Extraction). However, the identification of aspect words, corresponding opinion words, and sentiment expressed by those opinion words proposed in this paper belongs to a sequential task of aspect words, opinion sub-term, and sentiment polarity judgement, and ASTE needs to be accomplished at once; therefore, this section will provide a detailed description of these three inter-related tasks.

Since the introduction of deep neural networks for ABSA [4], numerous studies have explored the application of deep neural networks to determine the sentiment orientation of aspect words in text. Wang et al. [5] proposed using LSTM (Long Short-Term Memory) combined with attention mechanism to address this task, and Ma et al. [6] proposed an interactive attention mechanism to assess the importance of words expressing aspect words in comments, while Li et al. [7] employed a collaborative attention mechanism to deepen the interaction between aspect words and their contexts. Fan et al. [13] argued that when both aspect words and contexts are long, the simplicity of the weighted sum attention mechanism would introduce some noise; thus, they proposed a multi-granularity attention mechanism to reduce irrelevant information. Since the application of BERT and GCN in the field of NLP, a large number of works based on BERT and GCN have also studied ABSA, such as Zhou et al. [10] and Zhang et al. [14].

In the AOTE task, Wang et al. [15] used the high-level semantic representation of each term in its context through dependency syntax and recursive neural network to extract aspect words and opinion words, the aspect words were associated with the opinion words, and, finally, the aspect words and opinion words in the sentence were identified through CRF (Conditional Random Fields). Dai and Song [16] proposed using BiLSTM-CRF (Bi-Directional Long Short-Term Memory Conditional Random Fields) to extract aspect words and opinion words. Wang et al. [17] argued that syntax-based methods exhibit low accuracy when applied to unstructured text, and methods relying on the HMM (Hidden Markov Model) and CRF require a large number of manual labels. Therefore, a coupled attention mechanism is proposed to extract opinion words and aspect words. This mechanism necessitates the use of two separate attention mechanisms for each sentence to capture the direct and indirect relationships between aspect and opinion words. In addressing the challenge of extracting aspect words and opinion words across domains, Wang and Pan [18] proposed an Interaction Memory Network featuring local and global memory units, which enabled aspect words and opinion words within the same domain to interact with each other, as well as facilitating interaction between aspect words and opinion words across different domains. Consequently, this approach achieved higher experimental results in

the extraction of aspect words and opinion words across domains, eliminating the need for additional resources. Chen et al. [19] stated that there are certain difficulties in extracting aspect words and opinion words, such as the fact that the relationships between aspect words and opinion words can be one-to-many, many-to-one, or even overlapping, while the extraction of aspect words and opinion words is interdependent rather than independent; therefore, when extracting these two types of entities, it is necessary to synchronously determine the relationship between entities. Therefore, a new SDRN (Synchronous Double-Channel Recurrent Network) model is proposed, which consists of an entity extraction unit, a relationship detection unit, and a synchronization unit, enabling the simultaneous extraction of entities and their relationships. Wu et al. [20] stated that it is unreasonable to divide the AOTE task into multiple sub-tasks and extract them through multiple channels and proposed a unified grid-labeling task to solve the AOTE task in an End-to-End manner. In ASPE, the problems of small data and insufficient training are compensated by jointly extracting aspect words and determining the sentiment classification task corresponding to aspect words, as well as by training on other data and transferring some parameters to the current network [21].

With the enormous prospects of social media cosmetic electronic word of mouth (eWOM), it is imperative to examine the influence of cosmetic eWOM on social media and for cosmetic marketers to understand the antecedents that result in cosmetic consumers making a purchase [22]. Previous research shows that electronic word-of-mouth spread through social media has a strong influence on customers' purchase decisions [23]. In recent years, the business community has delved into aspect words and their corresponding opinion words in comments and made judgments on sentiment polarity in one step (i.e., ABSA triplet judgment). Since Li et al. [24] proposed this task, ASTE has received a lot of attention from scholars. In words of triplet extraction, previous work divided it into several parts, namely AOTE and AFOE. ASTE integrates these two tasks, with Xu et al. [25] discovering a strong correlation between the three elements of a triplet and designing a sequence labeling joint model that can capture the relationship between the three elements to extract triplets. Mao et al. [26] transformed the triplet extraction task into two Machine Reading Comprehension (MRC) problems. Firstly, the task of triplet extraction was decomposed into tasks of AE, AOE, and SC; then, left MRC and right MRC were used to process AE, AOE, and SC separately. Faced with the various ABSA sub-tasks, Yan et al. [27] converted ABSA into a unified generative task. Based on the unified formula, the model solved all ABSA sub-tasks using the End-to-End framework of the pre-trained sequence-to-sequence model BERT and achieved good experimental results on ASTE. Chen et al. [12] dealt with the triplet extraction task as a term relationship classification task and defined ten categories of term relations. They used a multichannel network to extract features such as vocabulary, grammar, syntax, and position. Experimental results on a large dataset proved the effectiveness of this model.

In these works, the task of extracting aspect words, opinion words, and the sentiment expressed by opinion words in a cascading way has a wider application scenario, and cascading methods can also reduce losses and have higher accuracy.

Inspired by Chen et al. [12], we still define 10 word relationships to facilitate the extraction of triplets and process DREN. And the following research hypotheses were made:

Research Hypothesis 1: *DREN can obtain the polarity (triplet) of aspect and sentiment in one step without giving the aspect.*

Research Hypothesis 2: *SWDA can solve the possible discontinuity problem between aspect words and opinion words.*

Research Hypothesis 3: *DREN can achieve higher F1 values than GTS-BiLSTM and GTS-BERT.*

3. Model Building

3.1. Research Methodology

Inspired by Chen et al. and Jiao et al. [12,28], we proposed DREN to process the ASTE, as shown in Figure 1, which is divided into encoding layer, information extraction and fusion layer, and prediction layer. The three layers are introduced in detail in the following text. As we described before, BERT have good presentation in semantics, and GCN is good at extracting instruction information. So, in the encoding layer, we used BERT, the Biaffine Attention Mechanism, and GCN to extract information, which included semantic information, structural information, and syntactic information. We combined other information, including part-of-speech information, dependency syntax information, relative position information, and tree distance information, in the information extraction and fusion layer to obtain a good representation. After that, SWDA was used to mitigate redundancy in global attention and discontinuity between aspect words and opinion words. Finally, triplets were extracted by predicting the relationships between words in the prediction layer.

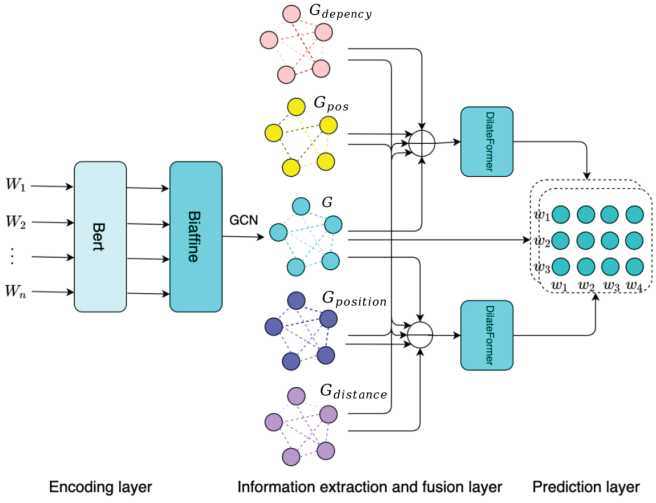


Figure 1. The Model of ASTE based on DREN.

When given a comment with n words, the task of the ATST was to extract the m aspect words a, opinion words o, and s pairs of sentiment polarities from the comment, i.e., $T = \{[a, o, s]_m\}$. According to the definition in Chen et al. [12], the triplet extraction task was transformed into a term relationship judgment task, and ten types of term relationships were defined (where o represents aspect words, while B_o,I_o,o represent the beginning, end, and opinion words, respectively; where a represents aspect words, while B_a,I_a,a represent the beginning, end, and opinion words, respectively; and POS, NEG, and NEU represent positive, negative, and neutral sentiment polarity, respectively; the sign * represents words that are not part of a triplet). Thus, the aspect term extraction, opinion term extraction, and the sentiment orientation expressed by aspect words in the comment were solved in one step. This paper uses the definitions of Chen et al.'s [12] pairs of triplet relationships and further improved the accuracy of relationship categories by using DREN. And this paper will continue to provide a detailed introduction to the model.

3.2. Encoding Layer

After obtaining the text $X = [W_1, W_2, \cdots W_n]$ containing m pairs of aspect words a, opinion words o, and sentiment polarity s, the semantic information of long dependency and structure was fully captured. Then, the model input X into BERT, which is the

encoder part of the Transformer [11]. A large number of studies have shown that BERT has better semantic feature extraction ability in natural language processing tasks compared with traditional LSTM and CNN. Afterward, through the last encoding layer of BERT, $H = [h_1, h_2, \cdots h_n] \in \mathbb{R}^{n \times d_h}$ is obtained, where d_h is the dimension of the vectors in BERT.

While this paper eventually transforms the triplet recognition task into a term relationship discrimination task, it fundamentally involves the initial steps of entity identification, followed by relationship determination between entities and subsequently sentiment polarity assignment. The recognition of entities heavily depends on other information, such as the syntactic structure, semantic structure, and part-of-speech structure within a sentence. Hence, how to identify entities becomes one of the challenges that this paper needs to address. To address this challenge, the obtained H, H is first input into LBAM. This mechanism serves to facilitate the model in solving the nesting problem between entities. As a classic neural network in entity recognition, the LBAM interacts through the boundaries of entity heads and tails, thereby better identifying the boundary problems between entities. The LBAM is shown in Formulas (1)–(5).

$$h_i^a = MLP_a(h_i) \tag{1}$$

$$h_j^o = MLP_o(h_j) \tag{2}$$

$$g_{i,j} = h_i^{aT} U_1 h_j^o + U_2\left(h_i^a \bigoplus h_j^o\right) + b \tag{3}$$

$$r_{i,j,k} = \frac{exp\left(g_{i,j,k}\right)}{\sum_{l=1}^{d_b} exp\left(g_{i,j,l}\right)} \tag{4}$$

$$B = Biaffine(MLP_a(H), MLP_o(H)) \tag{5}$$

In Formulas (1)–(4), U_1 and U_2 are parameter matrices, and b is the bias vector, where $\bigoplus$ represents a splice and MLP is a fully connected layer. Formulas (1)–(4) are used to measure the K relationships between the words W_i and W_j in the same sentence through the operations of full concatenation, planning etc. We used Formula (4) to normalize the kth relationships between word i and word j. So, the d_b relationships of the comment sentence with n words, namely $B = [b_1, b_2, \cdots b_n] \in \mathbb{R}^{n \times n \times d_b}$, are determined. Afterward, we obtained H through the Biaffine Attention Mechanism Network, where d_b is the dimension of the vectors in the LBAM, in order to enhance the identification of entity and structure relationships. In this paper, B is used to further obtain the graph information features through GCN to solve the problem of insufficient information in entity recognition. GCN [25] is also widely used in natural language processing, such as BERT. So, GCN is a variant of CNN (Convolutional Neural Networks). It is different from CNN and GCN regarding the use of text as the structure of graph. Then, we obtained $G = [g_1, g_2, \cdots g_n] \in \mathbb{R}^{n \times n \times d_g}$ after convoluting the information of graph, in which d_g is the dimension of the vector in GCN.

3.3. Information Extraction and Fusion Layer

Secondly, to fully exploit other information beyond semantics, this paper incorporated part-of-speech information, dependency syntactic information, relative position information, and tree distance information to broaden the expression of the semantic vector space and enrich vector representation. So, it could improve the accuracy of entity recognition, contributing to the improved judgment of relationships between words. The part-of-speech information, dependency syntactic information, relative position information, and tree distance information used in this paper are all provided in the dataset and are represented, respectively, as $G_{pos} = [g_1, g_2, \cdots g_n] \in \mathbb{R}^{n \times n \times d_g}$, $G_{depency} = [g_1, g_2, \cdots g_n] \in \mathbb{R}^{n \times n \times d_g}$, $G_{position} = [g_1, g_2, \cdots g_n] \in \mathbb{R}^{n \times n \times d_g}$, and $G_{distance} = [g_1, g_2, \cdots g_n] \in \mathbb{R}^{n \times n \times d_g}$ through

random initialization. After obtaining G, G_{pos},$G_{depency}$, $G_{position}$, and $G_{distance}$, this paper first adds and subtracts matrices and normalizes them through $SoftMax$ to integrate matrix representations and find differences between matrices, and it fully integrates this information and strengthens the relationships between words, as shown in Formulas (6) and (7).

$$G_{add} = softmax\Big(G + G_{pos} + G_{depency} + G_{position} + G_{distance} \Big) \tag{6}$$

$$G_{sub} = softmax\Big(G - G_{pos} - G_{depency} - G_{position} - G_{distance}\Big) \tag{7}$$

To address the issue of discontinuity between aspect words and opinion words in ABSA, this paper introduces a multiscale global network attention mechanism designed to link discontinuous words. The multiscale global network attention mechanism was initially introduced to mitigate redundancy in global attention modules in image processing [28]. Jiao et al. [28] asserts that attention mechanisms across different distances also prove effective at addressing different distance of pixels. We think that it could also process the discontinuity between the aspect words and opinion words. The structure of the SWDA is shown in Figure 2.

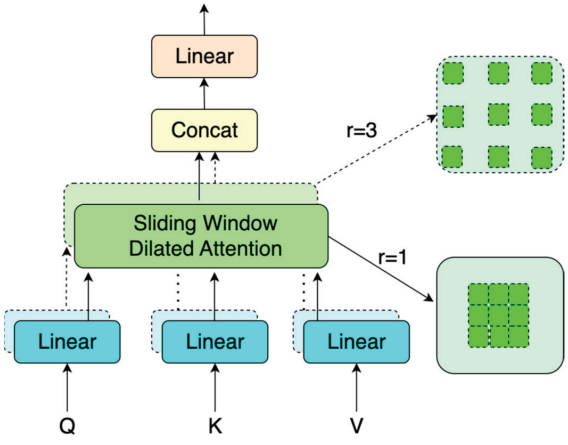


Figure 2. Structure of SWDA.

For G_{add}, when the scale $r = 1$, after the SWDA, Formula (8) is as follows.

$$X_1 = SWDA(Q, K, V, r = 1) \tag{8}$$

Among them, Q, K, and V are matrices of query, key, and value. They were the three feature matrices obtained by G_{add} through CNN, and each row of these three matrices represented a query/key/value feature vector. For the query at position (i, j), in the original feature mapping, the SWDA sparsely selected keys and values, with (i, j) as the center, to perform the attention mechanism in sliding windows of different sizes, as shown in Formula (9).

$$X_{i,j} = softmax(\frac{q_{i,j}K_r^T}{\sqrt{d_k}})V_r,\ 1 \le i \le W,\ 1 \le j \le H \tag{9}$$

W and H are the height and width of the feature map at position $X_{i,j}$. For position (i, j), the width and height have a certain limit range:

$$\left\{ \left(i' + j'\right) \middle| i' = i + p \times r, j' = j + p \times r \right\} \quad -1 <= p.q <= 1 \tag{10}$$

Similarly, when $r = 3$, X_3 is obtained, X_1 and X_3 are then spliced after passing through the linear layer. And X_{add} is finally obtained. X_{sub} is similarly obtained. In order to better

capture the relationships between words, the loss calculation used in this paper for G, X_{add} and X_{sub} consists of two parts, namely L, L_{add} and L_{sub}, as shown in Formula (5).

$$L = L^g + \alpha L^{add} + \beta L^{sub} \tag{11}$$

4. Experimental Process

4.1. Parameter Settings

In this paper, the text length n is defined as 102 because the text length of 100 can fit the lengths of most sentences in the dataset, while the start and end characters of BERT are +2. Therefore, any text that exceeds this limit will be truncated by experimentally looking for hyperparameters. The BERT used is BERT-base-uncased, featuring default dimensions of 768 and default layers of 12. The dimension of GCN is set to 300, with one layer. The part-of-speech information, dependency parsing information, relative position information, and tree distance information are initialized randomly with a dimension of 10. The batch size is set to 16, the learning rate is set to 10^{-3}, the dropout is set to 0.5, and α and β are set to 0.1. The values r are set to 1 and 3.

In addition, this paper uses F_1 as the evaluation standard for the experiment, as shown in Formulas (12)–(14).

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

$$F_1 = \frac{2 \times Precison \times Recall}{Precision + Recall} \tag{14}$$

where TP means that the true case is positive and the predicted value is positive, FP means that the true case is negative and the predicted value is positive, and FN means that the true case is positive and the predicted value is negative.

4.2. Dataset Description

This paper uses two ABSA datasets of D1 and D2 from SemEval. Each dataset contains information in the latter two domains, namely note reviews and restaurant data, in which the note data are derived from the SemEval of 2014, while the restaurant reviews are derived from the SemEval of 2014, 2015, and 2016. Each dataset is divided into training, validation, and testing sets. SemEval of 2014 is split into 14res and 14lap, and SemEval of 2015 and SemEval of 2016 are selected in the res domain. Table 1 provides specific details regarding the number of sentences in each dataset. Table 2 shows the numbers of aspect words in each dataset.

Table 1. Statistics regarding the number of sentences in each dataset.

Dataset	14lap	14res	15res	16res
D1_train	899	1259	603	863
D1_vaild	225	315	151	216
D1_test	332	493	325	328
D2_train	906	1266	605	857
D2_valid	219	310	148	210
D2_test	328	492	322	326

Table 2. Statistics regarding the number of aspect words in each dataset.

Dataset	14res	14lap	15res	16res
D1_train	1452	2356	1038	1421
D1_vaild	383	580	239	348
D1_test	547	1008	493	525
D2_train	1460	2338	1013	1394
D2_valid	346	577	249	339
D2_test	543	994	485	514

4.3. Comparison Experiment

To assess the effectiveness of the model of DREN in this paper, the Peng-two-stage-IOG [27], GTS-CNN/GTS-BiLSTM/GTS-BERT [28], S^3E^2 [20], CMLA+ [29], RINANTE [29], Li-unified-R [29], OTE-MTL [30], and JET-BERT [25] are chosen for comparison, in which CMLA+, RINANTE+, Li-unified-R, and Peng-two-stage are Pipeline methods, while GTS-CNN/GTS-BiLSTM/GTS-BERT, S^3E^2, OTE-MTL, and JET-BERT are End-to-End models. The following is a description of several baselines:

Peng-two-stage-IOG: It divides the triplet extraction task into two stages. In the first stage, BiLSTM and GCN are used to predict the possible descriptive aspect words and opinion words; then, in the second stage, the aspect words and opinion words predicted in the first stage are paired.

GTS-CNN/GTS-BiLSTM/GTS-BERT: It defines six kinds of word relationships, which are encoded via CNN/BilISTM/BERT and attention mechanism on the encoder. After pooling and some linear transformation operations, the SoftMax classifier is used to classify the word relationships.

S^3E^2: In order to solve the long tail problem in the extraction of aspect words and opinion words in review texts, the SoftProto framework makes samples related to each other, thus allowing rare words to be extracted.

CMLA: It considers that the extractions of aspect words and opinion words need to be correlated with each other and proposes using SWDA to correlate aspect words and opinion words. Each layer of attention mechanism is composed of a pair of attention mechanisms, one of which extracts aspect words and one of which extracts opinion words. Through interactive learning, the extraction performance is improved.

RINANTE: It proposes to use BiLSTM-CRF to train the data, which is composed of two parts: one is manually labeled data, and the other is automatically labeled based on dependency syntax and Part-of-Speech annotators.

Li-unified-R: It uses a two-layer recurrent neural network to process the extraction task. The first layer of the recurrent neural network is used to generate labeled results, and the second layer is used to label the target boundary.

OTE-MTL: It proposes a multi-task learning framework for joint extraction of attributive words and opinion words while using bilinear affine attention mechanism to record the sentiment dependency between words.

JET-BERT: It believes that attributive words, opinion words, and polarity judgments need to be extracted in End-to-End, rather than through multiple stages. The paper proposes using two kinds of location information to improve the extraction accuracy of recognition: one is to mark the distance between the sentiment words and attributive words, with attributive words as the center, and the other is to calculate the distance between attributive words and sentiment words, with sentiment words as the center.

There are comparative experimental baseline and DREN results for two datasets in Tables 3 and 4. Some baseline results can be found in [12,25,30,31].

Table 3. Experimental results of the value F1 for the Dataset D1.

Model	14res	14lap	15res	16res
Peng-two-stage-IOG	59.64	47.02	48.71	58.67
GTS-CNN	65.94	51.38	56.64	64.73
GTS-BiLSTM	64.49	51.30	56.29	65.56
S^3E^2	66.74	52.01	58.66	66.87
GTS-BERT	70.20	54.58	58.67	**67.58**
DREN	**72.27**	**56.75**	**61.41**	65.84

Table 4. Experimental results of the value F1 for the Dataset D2.

Model	14res	14lap	15res	16res
CMLA	42.79	33.16	37.01	41.72
RINANTE	34.95	20.07	29.97	23.87
Li-unified-R	51.00	42.34	47.82	44.31
Peng-two-stage-IOG	51.46	42.87	52.32	54.21
OTE-MTL	58.71	43.42	47.13	56.96
JET-BERT	62.40	51.04	57.53	63.83
GTS-BERT	68.81	55.42	58.60	**67.58**
DREN	**69.05**	**57.52**	**59.45**	67.20

4.4. Discussion

As shown in Tables 3 and 4, Peng-two-stage-IOG, GTS-CNN, GTS-BiLSTM, S^3E^2, GTS-BERT, and DREN are tested in D1, and CMLA, RINANTE, Li-unified-R, Peng-two-stage-IOG, OTE-MTL, JET-BERT, and GTS-BERT are tested in D2. It can be seen that the experiments on 14res, 14lap, and 15res in the DREN dataset all achieved the best results. Peng-two-stage demonstrated poor performance on D1 and D2. CMLA+, RINANTE+, and Li-unified-R had poor experimental results on D2, primarily attributed to the pipeline approach, which overlooked the correlation between aspect term extraction and opinion term extraction tasks, resulting in loss accumulation. Based on the performance of Peng-two-stage, CMLA+, RINANTE+, and Li-unified-R on the dataset, it can be inferred that the traditional two-step method of first identifying the attributed word and then identifying the sentiment of the attributed word in the context is defective. GTS-CNN/GTS-BiLSTM/GTS-BERT and S^3E^2 use End-to-End to identify relationships between words. GTS-BERT have a better result than GTS- BiLSTM in D1. The reason for this result is that BERT has been pre-trained on large-scale data and can provide a strong language understanding ability. However, this method will overlook the importance of other information in entity recognition, such as structural and syntactic information. OTE-MTL uses a bilinear affine attention mechanism to recognize the boundary relationship between words, and JET-BERT uses distance information. However, the information used by the above methods is limited, and the task features are not fully utilized for the attribute word extraction and sentiment orientation discrimination task. So, the DREN proposed in this paper not only considers various available information comprehensively, thus enriching the expression of semantics, and improves the accuracy of entity recognition, but it also further improves the accuracy of term relationship discrimination. In the 16res dataset, the DREN's performance is slightly less effective than that of GTS-BERT. While enhanced with deep relationships, DREN also introduces a small amount of noise, thus affecting the accuracy of the model.

This paper further illustrates the effectiveness of DREN through ablation experiments on D2, as shown in Table 5, where G_{other} represents the comprehensive information of G_{pos}, $G_{depency}$, $G_{position}$, and $G_{distance}$. It can be seen that in the absence of G_{pos}, $G_{depency}$,

$G_{position}$, and $G_{distance}$, the F1 values of the four datasets all decline to varying degrees. Compared with G_{pos}, $G_{depency}$, $G_{position}$, and $G_{distance}$, SWDA makes a larger contribution, thus further illustrating the effectiveness of SWDA on the model.

Table 5. Ablation experiment for Dataset D2.

Model	14res	14lap	15res	16res
DREN	69.05	57.52	59.45	67.20
(W/o) SWDA	67.45	55.05	58.89	65.85
(W/o) G_{other}	68.67	56.58	58.09	66.05

In order to validate the feasibility of the model, this section employs a case study for illustration. The triplets in this review, "The food is surprisingly good, and the decor is nice", from res14 in D1, are (food, good, positive) and (decor, nice, positive). DREN successfully identified the relationships between food and good and decor and nice. However, GTS-BiLSTM and GTS-BERT both mistakenly considered good and nice to be opinion words for the aspect term "food", leading to incorrect judgments, as shown in Figure 3.

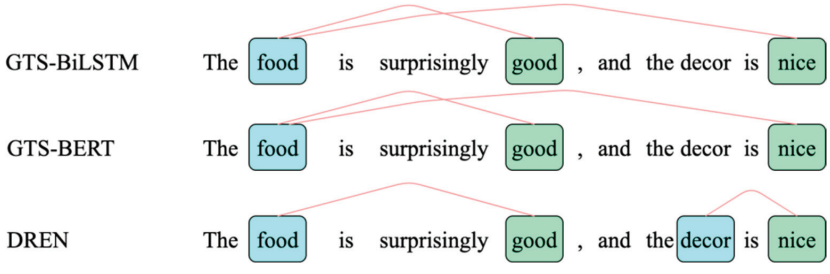


Figure 3. Case study.

The triplet extraction task is transformed into a term relationship judgment task, and ten types of term relationships are defined (where *o* represents aspect words, while *B_o*, *I_o*,*o* represent the beginning, end, and opinion words respectively; *a* represents aspect words, while *B_a*,*I_a*,*a* represent the beginning, end, and opinion words respectively; and POS, NEG, NEU represent positive, negative, and neutral sentiment polarity, respectively. Compared to the method of extracting aspect and opinion words first and then combining the two to judge sentiment. DREN can use the relationship between the words to extract the triplet in one step. As for Research Hypothesis 2, from Table 5 and the case study that we just discussed, we can see SWDA's importance to our results. It is also better for processing the difficulty of the main problem, which is how to associate aspect with opinion words. Compared with GTS BiLSTM and GTS BERT, the DREN proposed in this paper not only considers various available information comprehensively, thus enriching the expression of semantics, but also improves the accuracy of entity recognition, as shown in Tables 3 and 4.

5. Conclusions

This paper proposes a DREN for identifying extracting triplet problems in ABSA in one step. Firstly, the triplet extraction task is transformed into a relationship discrimination task between words. To improve the conversion rate of the discrimination task, the BERT and GCN methods are employed to enhance the semantic representation. Additionally, part-of-speech information, dependency syntax information, relative position information, and tree distance information are utilized to increase semantic richness and identify differences. Then, SWDA is used to mitigate redundancy in global attention and discontinuity between aspect words and opinion words. In capturing word relationships, the part-of-speech information, dependency syntactic information, relative position information, and tree

distance information enrich the representation. SWDA further improved the effect. Bert, GCN, and LBAM are also crucial. We have noticed that in ASTE, a large number of works are currently concentrated in the e-commerce field because there are already mature datasets in the e-commerce field, but there are serious deficiencies in the education field, especially in online courses and the reviews of online courses. Therefore, in the future, we will consider generating ASTE data in the online education field, before applying ABSA to the construction of the education field, and training the model by adding domain information through prompts and other methods.

Author Contributions: Conceptualization, J.P. and B.S.; methodology, J.P.; software, B.S.; validation, J.P. and B.S.; formal analysis, J.P.; investigation, J.P.; resources, J.P. and B.S.; data curation, J.P. and B.S.; writing—original draft preparation, B.S.; writing—review and editing, J.P. and B.S.; visualization, J.P.; supervision, B.S.; project administration, J.P. and B.S.; funding acquisition, J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This study was financially supported by the 2022–2023 Higher Education Fund of the Macao SAR Government (Project No: MF2315), the General Project of the National Social Science Foundation of Education (Project No: BBA230063), and the National Social Science Fund of China (BCA200090).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study can be obtained from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Feng, C.; Li, H.; Zhao, H.; Xue, Y.; Tang, J. Aspect-level Sentiment Analysis Based on Hierarchical Attention and Gate Networks. In Proceedings of the 19th Chinese National Conference on Computational Linguistics, Hainan, China, 30 October–1 November 2020; Chinese Information Processing Society of China: Haikou, China, 2020; pp. 688–697.
2. Shaikh, T.; Deshpande, D. A Review on Opinion Mining and Sentiment Analysis. *Int. J. Comput. Appl.* **2016**, *975*, 8887.
3. Su, J.; Ouyang, Z.; Yu, S. Aspect-Level Sentiment Classification for Sentences Based on Dependency Tree and Distance Attention. *Comput. Res. Dev.* **2019**, *56*, 1731–1745.
4. Tang, D.; Qin, B.; Liu, T. Aspect level sentiment classification with deep memory network. *arXiv* **2016**, arXiv:1605.08900.
5. Wang, Y.; Huang, M.; Zhu, X.; Zhao, L. Attention-based LSTM for aspect-level sentiment classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 606–615.
6. Ma, D.; Li, S.; Zhang, X.; Wang, H. Interactive attention networks for aspect-level sentiment classification. *arXiv* **2017**, arXiv:1709.00893.
7. Li, H.; Xue, Y.; Zhao, H.; Hu, X.; Peng, S. Co-attention networks for aspect-level sentiment analysis. In *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, 9–14 October 2019*; Springer International Publishing: Cham, Switzerland, 2019; pp. 200–209.
8. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
9. Pang, G.; Lu, K.; Zhu, X.; He, J.; Mo, Z.; Peng, Z.; Pu, B. Aspect-level sentiment analysis approach via BERT and aspect feature location model. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 5534615. [CrossRef]
10. Zhou, J.; Huang, J.X.; Hu, Q.V.; He, L. Sk-gcn: Modeling syntax and knowledge via graph convolutional network for aspect-level sentiment classification. *Knowl. Based Syst.* **2020**, *205*, 106292. [CrossRef]
11. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
12. Chen, H.; Zhai, Z.; Feng, F.; Li, R.; Wang, X. Enhanced multi-channel graph convolutional network for aspect sentiment triplet extraction. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; Association for Computational Linguistics: Dublin, Ireland, 2022; pp. 2974–2985.
13. Fan, F.; Feng, Y.; Zhao, D. Multi-grained attention network for aspect-level sentiment classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 3433–3442.
14. Zhang, S.; Zhao, T.; Wu, H.; Zhu, G.; Li, K. TS-GCN: Aspect-level sentiment classification model for consumer reviews. *Comput. Sci. Inf. Syst.* **2023**, *20*, 117–136. [CrossRef]
15. Wang, W.; Pan, S.J.; Dahlmeier, D.; Xiao, X. Recursive neural conditional random fields for aspect-based sentiment analysis. *arXiv* **2016**, arXiv:1603.06679.
16. Dai, H.; Song, Y. Neural aspect and opinion term extraction with mined rules as weak supervision. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5268–5277.

17. Wang, W.; Pan, S.J.; Dahlmeier, D.; Xiao, X. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. *Proc. AAAI Conf. Artif. Intell.* **2017**, *31*, 3316–3322. [CrossRef]
18. Wang, W.; Pan, S.J. Transferable interactive memory network for domain adaptation in fine-grained opinion extraction. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 7192–7199. [CrossRef]
19. Chen, S.; Liu, J.; Wang, Y.; Zhang, W.; Chi, Z. Synchronous double-channel recurrent network for aspect-opinion pair extraction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6515–6524.
20. Wu, Z.; Ying, C.; Zhao, F.; Fan, Z.; Dai, X.; Xia, R. Grid tagging scheme for aspect-oriented fine-grained opinion extraction. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Online, 16–20 November 2020; pp. 2576–2585.
21. He, R.; Lee, W.S.; Ng, H.T.; Dahlmeier, D. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 504–515.
22. Kohler, E.; Mogaji, E.; Erkan, İ. Save the Trip to the Store: Sustainable Shopping, Electronic Word of Mouth on Instagram and the Impact on Cosmetic Purchase Intentions. *Sustainability* **2023**, *15*, 8036. [CrossRef]
23. Anastasiei, B.; Dospinescu, N.; Dospinescu, O. The impact of social media peer communication on customer behaviour evidence from Romania. *Argum. Oeconomica* **2022**, *48*, 247–264. [CrossRef]
24. Li, X.; Bing, L.; Li, P.; Lam, W.; Yang, Z. Aspect term extraction with history attention and selective transformation. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, 13–19 July 2018; pp. 4194–4200.
25. Xu, L.; Li, H.; Lu, W.; Bing, L. Position-aware tagging for aspect sentiment triplet extraction. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Online, 16–20 November 2020; pp. 2339–2349.
26. Mao, Y.; Shen, Y.; Yu, C.; Cai, L. A joint training dual-mrc framework for aspect based sentiment analysis. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 13543–13551. [CrossRef]
27. Yan, H.; Dai, J.; Ji, T.; Qiu, X.; Zhang, Z. A unified generative framework for aspect-based sentiment analysis. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Virtual Event, 1–6 August 2021; pp. 2416–2429.
28. Jiao, J.; Tang, Y.M.; Lin, K.Y.; Gao, Y.; Ma, J.; Wang, Y.; Zheng, W.S. Dilateformer: Multi-scale dilated transformer for visual recognition. *IEEE Trans. Multimed.* **2023**, *25*, 8906–8919. [CrossRef]
29. Peng, H.; Xu, L.; Bing, L.; Huang, F.; Lu, W.; Si, L. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 8600–8607. [CrossRef]
30. Zhang, C.; Li, Q.; Song, D.; Wang, B. A multi-task learning framework for opinion triplet extraction. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Online, 16–20 November 2020; pp. 819–828.
31. Chen, Z.; Qian, T. Enhancing aspect term extraction with soft prototypes. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Online, 16–20 November 2020; pp. 2107–2117.

applied sciences

MDPI

Article

CCDA: A Novel Method to Explore the Cross-Correlation in Dual-Attention for Multimodal Sentiment Analysis

Peicheng Wang, Shuxian Liu * and Jinyan Chen

School of Information Science and Engineering, Xinjiang University, Urumqi 830017, China; 107552101310@stu.xju.edu.cn (P.W.); 107552103773@stu.xju.edu.cn (J.C.)
* Correspondence: liushuxian@xju.edu.cn

Abstract: With the development of the Internet, the content that people share contains types of text, images, and videos, and utilizing these multimodal data for sentiment analysis has become an important area of research. Multimodal sentiment analysis aims to understand and perceive emotions or sentiments in different types of data. Currently, the realm of multimodal sentiment analysis faces various challenges, with a major emphasis on addressing two key issues: (1) inefficiency when modeling the intramodality and intermodality dynamics and (2) inability to effectively fuse multimodal features. In this paper, we propose the CCDA (cross-correlation in dual-attention) model, a novel method to explore dynamics between different modalities and fuse multimodal features efficiently. We capture dynamics at intra- and intermodal levels by using two types of attention mechanisms simultaneously. Meanwhile, the cross-correlation loss is introduced to capture the correlation between attention mechanisms. Moreover, the relevant coefficient is proposed to integrate multimodal features effectively. Extensive experiments were conducted on three publicly available datasets, CMU-MOSI, CMU-MOSEI, and CH-SIMS. The experimental results fully confirm the effectiveness of our proposed method, and, compared with the current optimal method (SOTA), our model shows obvious advantages in most of the key metrics, proving its better performance in multimodal sentiment analysis.

Keywords: multimodality; sentiment analysis; attention mechanism

Citation: Wang, P.; Liu, S.; Chen, J. CCDA: A Novel Method to Explore the Cross-Correlation in Dual-Attention for Multimodal Sentiment Analysis. *Appl. Sci.* **2024**, *14*, 1934. https://doi.org/10.3390/app14051934

Academic Editor: Giacomo Fiumara

Received: 29 November 2023
Revised: 16 February 2024
Accepted: 20 February 2024
Published: 27 February 2024

1. Introduction

Multimodal sentiment analysis (MSA) is an important branch in the field of artificial intelligence. It aims to capture and understand human sentiment or emotion contained in text, speech, images, or other types of data, usually including positive, negative, neutral, or more specific emotional states such as joy, sadness, and anger [1]. In recent years, with the popularity of online social platforms, a large amount of multimodal data has emerged on the Internet. By analyzing data containing multiple modalities, computers can perceive human sentiment in the data [2]. Multimodal sentiment analysis has attracted widespread attention and it is widely applied in social media analysis [3,4], market research [5,6], and human–computer interaction [7,8].

In early studies on multimodal sentiment analysis, researchers have mainly used the following approaches to process multimodal data: The first one is early fusion, by concatenating different unimodal features and subsequently processing the features using different classifiers or models. For example, Morency et al. [9] used an HMM to process three unimodal features simultaneously. Poria et al. used CNN- [10] and LSTM-based [11] models to explore the contextual relationships between modalities. Zadeh et al. [12] used Multi-Attention Block(MAB) and Long-Short Term Hybrid Memory(LSTHM) to capture and store dynamics in multimodal features separately. Haohan et al [13]. used a Select Additive Learning based on CNN to improve the generalization performance of the model. The second method is late fusion, by training modality-specific classifiers for

each modality and then predicting sentiment according to the weight of the classifier's results. For example, Glodek et al. [14] used Kalman filters as combiners for decision-making. Cai et al. [15] first used several different CNNs and subsequently vectorized and fused the output of the features from each CNN. Alam et al. [16] used Sequential Minimal Optimization (SMO, a variant of SVM) with different kernel functions and fused their results in decision-making.

Although these two methods were relatively simple, when dealing with modal features, the model is unable to capture intra- and intermodality dynamics efficiently, which may lead to poor model performance. The researchers then combined the advantages of early and late fusion and proposed hybrid fusion. Poria et al. [17] used deep CNNs to extract features and fused multimodal features using MKL and determine the weights of textual modalities using a decision fusion approach in the final stage. Kumar et al. [18] used gating mechanisms to selectively learn cross-modal interaction information and used the results for sentiment prediction. Zhang et al. [19] used a multihead attention mechanism to extract semantic and sentiment analysis, then train multiple base classifiers and ultimately fuse the decisions of the base classifiers.

Word-level fusion fuses different modalities in a temporal step to obtain cross-modal correlations. For example, Zadeh et al. [20] proposed a memory fusion network (MFN), by simulating interactions within modalities and generalizing the temporal relationships between different modalities, the sequence is ultimately unified based on the relationships between unimodal word-level features. Subsequently, in [21], they proposed a Graph-Memory Fusion Network and performed word-level fusion by using a dynamic fusion graph. Paul et al. [22] proposed an LSTHM-based model to obtain cross-modal interactions by performing a multi-stage fusion of modalities features between each time step. Wang et al. [23] proposed a Recurrent Attended Variation Embedding Network (RAVEN), by modeling the fine-grained structure in word segments and transforming word representations based on nonverbal dynamic information.

Tensor fusion uses different tensor-based computation methods to allow different modalities to interact. Zadeh et al. [24] proposed Tensor Fusion Network (TFN), modal correlations are obtained by computing the outer product between the feature tensors. Zhun et al. [25] proposed Low-rank Multimodal Fusion (LMF) to solve the problem of excessive complexity in tensor computation. Barezi et al. [26] introduced a modality-specific deconstruction method in the model to reduce information redundancy. Liang et al. [27] proposed a regularization method to learn cross-time and cross-mode correlations in low-rank tensors. Tao et al. [28] correlated features at the same time step and further proposed a dual low-order multimodal fusion method. Jin et al. [29] used LSTM-based and tensor-based CNN networks to capture intra- and intermodal dynamic information encapsulated in asynchronous sequences.

In recent years, a number of attention-based approaches have emerged. Through the attention mechanism, the model can be made to acquire inter- and intramodal correlations more efficiently. Poria et al. [11] used attention units to capture dynamics across modalities. In [30–34], multihead and self-attention were used to perform cross-modal interactions, respectively, and perceive emotional information that is not within the modality. In addition, the researchers used other attention-based methods such as Gate Recursive Units (GRUs) [35,36] and Graph Convolutional Networks (GCNs) [37].

Nevertheless, there are still two main challenges in current multimodal sentiment analysis research. The first one is inefficiency in modeling the intramodality and intermodality dynamics. Multimodal sentiment analysis requires processing data from different modalities and correlating them to capture sentiment. It also needs to deal with sentiment dependencies within a single modality to help the model understand sentiment more accurately. The second one is the way in which different modal features are fused. Effective integration of features from different modalities can improve the accuracy and robustness of the model, which is crucial for the reliability of sentiment analysis in practical applications.

In this paper, we use a transformer-based approach to capture sentiment information and extract dynamics within and between modalities, and we introduce the relevant coefficient for the fusion of multimodal features. In addition, we propose a new cross-correlation loss function for investigating the correlations between different levels of attention mechanisms. Specifically, we obtain the intermodality dynamics between the global representation and unimodal representation by using the cross-attention mechanism, which is the component of the Transformer, so that they can strengthen themselves by learning about each other in this process. At the same time, we obtain the intramodality information by using the self-attention mechanism for three unimodal features, respectively. In addition, in our research, we hypothesized that there is some correlation between different levels of attention mechanisms, so we propose the cross-correlation loss to assess the interrelationship between cross-attention and self-attention. The contributions of this paper can be summarized as follows:

- We propose CCDA, a hierarchical model that studies intra- and intermodality correlations by using self-attention and cross-attention, respectively. Moreover, we introduce a new method to fuse multimodal features efficiently.
- We innovatively introduce a new cross-correlation loss function to study the correlation between different levels of attentional mechanisms in more depth. The objective function is minimized to cut down redundant information, which can help our model to better perceive sentiment information.
- Extensive experiments demonstrate the effectiveness of our proposed methodology. Our model achieves comparable results to the state-of-the-art (SOTA) approach in all evaluation metrics on the CMU-MOSI, CMU-MOSEI, and CH-SIMS datasets.

2. Related Works

Multimodal sentiment analysis aims to obtain sentiment information from different types of data. It provides additional sources of information for affective computing and enables computers to understand and perceive human sentiment more accurately [1–4]. A key challenge in this area is determining how to efficiently fuse data from different modalities so that the model can recognize sentiment precisely. This section presents related works on multimodal sentiment analysis, including early fusion, late fusion, hybrid fusion, word-level fusion, tensor-based fusion, attention-based methods (Table 1 provides a brief description of several of these methods), and other recent research approaches.

Early fusion combines all of the features from different modalities (text, audio, and visual) into a single feature vector, which is then used for sentiment prediction using a classification algorithm or model. Morency et al. input three unimodal features into the HMM model simultaneously [9]. Poria et al. proposed a method using CNN networks [10], by feeding unimodal features into a multikernel learning classifier. Following this, [11] proposed an LSTM-based model to deal with different unimodal features and explored the contextual relationships between modalities. Zadeh et al. [12] concatenated the multimodal features at each time step, used Multi-Attention Block to capture the dynamics between different modalities, and used a Long-short Term Hybrid Memory to store the dynamic information associated with each modality. Haohan et al. [13] proposed a Select Additive Learning based on CNN model (SAL-CNN) to improve the generalization performance of the model. The advantages of these approaches are that they can take into account the correlation between different modality features at the early stage. However, premature fusion of unimodal features can prevent the model from capturing information about the dynamics within the modalities, which can affect the model's ability to perform fine-grained classification.

In contrast to early fusion, late fusion employs independent classifiers separately for unimodal data and then fuses the outputs of each model to generate the final multimodal representation, or votes on the results of each model. Glodek et al. [14] used the Kalman filter as the combiner for temporally ordered classifier decisions. It is a linear dynamical system based on a Markov model which is well suited for real-time classifier fusion. Cai et al. [15] used text CNN, image CNN, and multi CNN to process unimodal features

and multimodal features, respectively; they used logistic regression as a classifier with the vectorized features in the penultimate layer of different CNNs. In [16], Alam et al. generated their classification models using Sequential Minimal Optimization(SMO, which is a variant of SVM) for each feature set, and different kernel functions were used for different feature sets. Finally, the results of classifiers for different feature sets were fused using decision fusion. While late fusion helped the model to better integrate semantic information. However, the model is not able to obtain the interactions between modalities during the training process, which would prevent the model from capturing cross-modal dynamic information. In addition, it is usually accompanied by a more complex model structure and a larger number of parameters.

Table 1. Related works in multimodal sentiment analysis.

Method Type	Description	Advantages	Flaws
Early fusion	Combines all of the features from different modalities into a vector.	Realizes modal interactions at the early stage.	Time asynchrony and information redundancy.
Late fusion	Employs independent classifiers separately for each modality.	Helps model to better integrate semantic information.	Usually involves more complex model structures.
Hybrid fusion	Combines the advantages of early fusion and late fusion	Balance the model's complexity.	Inefficiencies arising from the limitations of the backbone network.
Word-level fusion	Fuses word representation in the temporal dimension.	Helps model to understand the intrinsic relation of multimodal data.	Insufficient generalization.
Tensor-based	Utilizes various tensor-based methods to integrate information from different modalities.	Integrate multimodal data effectively and address the complexity and noise issues.	Excessive computation and lack of interpretability.
Attention-based	Learns the semantic and relevant information using different attention mechanisms or Transformer.	More flexible and accurate in processing temporal information and capturing interactions between different modalities.	Correlations between different attention mechanisms cannot be captured.

Hybrid fusion combines the advantages of early fusion and late fusion, capitalizing on their strengths and compensating for their weaknesses, respectively. Poria et al. [17] proposed a method for extracting text features using deep CNNs and fusing multimodal heterogeneous features using MKL, in addition to a decision-level fusion method that determines the weights of the text modalities by the coupling of the sentiment modalities. Kumar et al. [18] used gating mechanisms to selectively learn cross-modal interaction information and utilized post-interaction results for sentiment prediction. Zhang et al. [19] used multihead attention to extract accurate semantic and affective information in the representation fusion stage, followed by training multiple base classifiers to make independent judgments on different unimodal representations in the decision fusion stage, and finally fusing base classifiers' decisions. The core idea of this approach is to allow features to be fused at different stages of the model while avoiding some of the potential problems of early fusion and late fusion. However, the limitations of the baseline model itself at that time made this type of fusion method not perform well enough.

Word-level fusion is a method that fuses word representations in the temporal dimension to capture the interrelationships between different modalities. This approach emphasizes word-level information interactions and helps to understand the intrinsic structure and semantic relatedness of multimodal data in more detail. In [20], Zadeh et al. proposed a Memory Fusion Network (MFN); they first modeled interactions within modalities and generalized temporal relationships across modalities, ultimately unifying sequences based on relationships between unimodal word-level features. Subsequently, in [21], they used a Graph-Memory Fusion Network to perform unimodal, bimodal, and trimodal word-level fusion for unimodal features, and captured intermodal interactions by using a dynamic

fusion graph. Paul et al. [22] proposed an LSTHM-based model, obtaining cross-modal interactions by performing multiple stage fusion of modalities features between each time step. Wang et al. [23] proposed Recurrent Attended Variation Embedding Network (RAVEN) by modeling the fine-grained structure in word segments and transforming word representations based on nonverbal dynamic information. Word-level fusion enables the integration of affective information from different modalities in word representations. However, this approach may result in the loss of specific affective information in the original modality, and the complexity of word-level fusion increases further when multiple different modalities are involved.

Tensor fusion utilizes various tensor-based methods to integrate information from different modalities. These methods can effectively integrate multimodal data and address the complexity and noise issues in the data. The tensor fusion network (TFN) [24] obtains the dynamic correlation between modes by calculating the outer product of bimodal and trimodal features. Zhun et al. [25] proposed a Low-rank Multimodal Fusion (LMF) method to solve the problem of excessive computational complexity in TFN, and utilized modality-specific low-rank factors for multimodal fusion to improve the efficiency. The Modality-based Redundancy Reduction Fusion (MRRF) [26] introduces a modal-specific decomposition method into the model, which removes redundant information from the dependency structure and leads to fewer parameters with minimal loss of information. Liang et al. [27] proposed a regularization method to minimize the rank of the tensor and learn correlations across time and modes in low-rank tensors. Tao et al. [28] correlated the features of a single time step between multiple modalities and further proposed a dual low-order multimodal fusion method to reduce computational complexity. Jin et al. [29] used LSTM-based and tensor-based CNN networks to discover intra- and intermodal dynamics, and encapsulated them in an asynchronous sequence. However, tensor fusion is often accompanied by high-dimensional data representations, which, again, increases computational complexity while causing data sparsity. On the other hand, tensor fusion reduces the interpretability of the model, which may limit the credibility and acceptance of the model in practical applications.

Attention mechanism (Especially Transformer [38], proposed by Google in 2017) plays a significant role in multimodal sentiment analysis; it helps models better understand and leverage the interconnections and semantic information between different modalities, and be more flexible and accurate in processing multimodal data. Chen et al. [39] and Poria et al. [11] used an LSTM-based model as well as attentional units to capture the dynamics across modalities. In [30–34], multihead and self-attention were used to capture relevant information within or across modalities. In addition, the researchers additionally used other methods, e.g., Gate Recurrent Unit (GRU) [35,36] and Graph Convolutional Network (GCN) [37]. The Transformer exhibits strong generalization capabilities, making it suitable for different types of multimodal sentiment analysis tasks.

In addition, there are other methods in multimodal sentiment analysis, such as multitask contrastive learning [40], dynamic filtering mechanism [41], bidirectional multimodal dynamic routing mechanism [42], cross-modal hierarchical graph contrastive learning strategy [43], supervised contrastive learning [20,44], dynamic refined sentiment words [45], etc.

Previous studies have viewed modality self-attention and cross-modal attention as two separate units that cannot interact with each other. Therefore, in this study, we proposed Cross-Correlation in Dual-Attention model (CCDA) to capture the correlations that exist between the different attention layers, so that, after acquiring intra- and intermodal information, respectively, the model can also enable them to exchange information that is helpful for their respective learning. In addition, in the feature fusion stage, we propose a strategy to help the model converge quickly, by calculating the relevant coefficients between the unimodal self-attention features and the source feature representations to guide the multimodal feature fusion.

3. Methodology

3.1. Problem Definition

Multimodal sentiment analysis is a task that utilizes multiple modalities for the study of human sentiment. Typically, it includes three modalities: text, speech, and images. We define three modality feature sequences, $X_m = \{x_{m,1}, x_{m,2}, \ldots x_{m,n}\}$, and sample labels $Y = \{y_1, y_2, \ldots y_n\}$, where the modality is represented as $m \in \{t, a, v\}$ (t stands for text, a stands for audio, and v stands for visual) and n represents the number of samples in the dataset. Our goal is to input modality features $X_m \in \mathbb{R}^{T_m \times d_m \times n}$ into a model to obtain an accurate sentiment prediction label $y \in \mathbb{R}^1$, where T_m and d_m represent the sequence length and the dimension of modality features separately.

3.2. Model Structure

In this section, we provide a detailed overview of the architecture of the CCDA (Cross-Correlation in Dual-Attention) model, as shown in Figure 1. We first use three unimodal encoders to obtain the utterance representation $U_m^{d_m \times n}$ and embedding $F_m^{T_m \times d_m \times n}$ by using feature sequences X_m for each modality separately, which $m \in \{t, a, v\}$, $U_m^{d_m \times n}$ originate from the feature representation in each unimodal encoder. This helps the model understand the semantic and sentiment information in each modality.

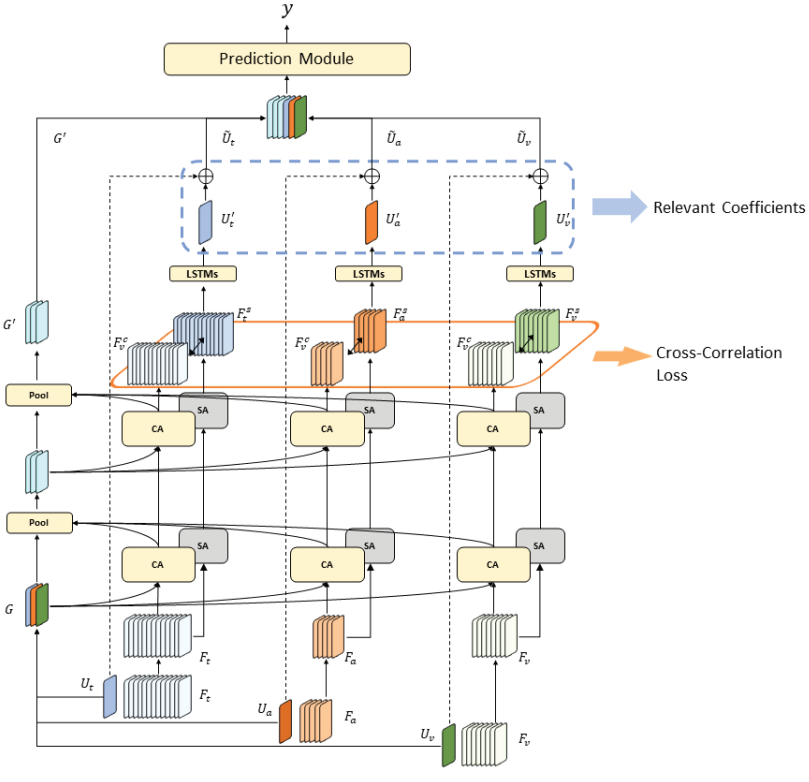


Figure 1. The structure of CCDA. The global representation G consists of three unimodal representations $\{U_t, U_a, U_v\}$. The model processes the global representation G and the unimodal features F_m using the dual-attention to obtain new global and unimodal representations $\{G', \widetilde{U}_t, \widetilde{U}_a, \widetilde{U}_v\}$ and fuses these representations for sentiment prediction. The unimodal features $\{F_t^S, F_a^S, F_v^S, F_t^C, F_a^C, F_v^C\}$ generated during this process are used to learn the correlation between the two attention mechanisms. The final objective function consists of the MSE loss $\mathcal{L}_{MSE}$ and the cross-correlation loss $\mathcal{L}_c$.

Next, we delve into the dual-attention mechanism (which contains self-attention and cross-attention), a core component of CCDA. By utilizing self-attention and cross-attention, CCDA can capture sentiment information and dynamics within a single modality (intramodality) and across different modalities (intermodality), respectively. This dual-attention mechanism enables the model to comprehensively analyze multimodal data and sentiment information, thereby improving the accuracy of sentiment analysis.

Following that, CCDA calculates cross-correlation losses between the embeddings generated by the two attention mechanisms while obtaining information about the intramodality and intermodality dynamics. This contributes to the indirect interaction between the two attention mechanisms and, thus, improves the model's performance. CCDA then uses relevant coefficients strategy to fuse the unimodal and multimodal representations obtained from these two attention mechanisms to generate the final sentiment representation.

In the following parts, we elaborate on the three main components of CCDA: unimodal encoders (Section 3.2.1), dual-level attention (Section 3.2.2), and fusion and prediction units (Section 3.2.3).

3.2.1. Unimodal Encoders

Similar to EMT [33], we employ the pretrained BERT model to encode textual tokens into context-aware word embeddings. Specifically, we notice that the [CLS] token of the BERT model contains a sequential representation of the text modality. Therefore, we use this token as the utterance representation for the text sequence, denoted as $u_t \in \mathbb{R}^{d_t}$. For the audio and visual modalities, we use LSTM recurrent neural networks to extract temporal information from the feature sequences. Ultimately, we select the hidden state of the last time step of the LSTM network for both the audio and visual modalities as their respective utterance representations: $u_a \in \mathbb{R}^{d_a}$ and $u_v \in \mathbb{R}^{d_v}$. Simultaneously, we need to process other tokens output by the BERT model and hidden states from LSTMs at different time steps for later use in self-attention and cross-attention mechanisms. These representations are denoted as $F_m \in \mathbb{R}^{T_m \times d_m}, m \in \{t, a, v\}$, representing the text, audio, and visual modalities, respectively.

$$\begin{aligned} F_t &= BERT(X_t) \\ F_a &= LSTM(X_a) \\ F_v &= LSTM(X_v) \end{aligned} \tag{1}$$

3.2.2. Dual-Level Attention

Attention mechanisms help the model better understand multimodal sentiment data and perceive emotional information. They enable the model to capture dynamics within a single modality or between different modalities during the multimodal sentiment processing. The Transformer [38] is a language model in the field of natural language processing; it is based on dot-product self-attention mechanisms. It employs self-attention to infuse global semantic information and consider long-range dependencies for every word in the sequence. Furthermore, the multihead mechanism allows the model to learn different subspaces of semantics.

In simple terms, the Transformer processes the input sequence $H \in \mathbb{R}^{T \times d}$ with positional encoding; it defines Query as $Q = HW_Q$, Key as $K = HW_K$, and Value as $V = HW_V$, where W represents the weight matrices during the feature sequence mapping process. Therefore, self-attention can be represented by Equation (2):

$$Self\text{-}Attention(H) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2}$$

In MulT [30], the Query and K–V pair in the self-attention computation process come from different modalities. Thus, MulT captures the interaction between the two modalities. MulT combines three modality pairs and calculates bidirectional modality interactions for

each pair. As shown in Equation (3), for two modality feature sequences H_1 and H_2, MulT defines Query as $Q_1 = H_1 W_Q$, Key as $K_2 = H_2 W_K$, and Value as $V_2 = H_2 W_V$. It calculates cross-modal attention in two directions between a pair of modalities:

$$\begin{aligned} Cross\text{-}Attention(H_1 \rightarrow H_2) &= softmax(\frac{Q_1 K_2^T}{\sqrt{d_k}})V_2 \\ Cross\text{-}Attention(H_2 \rightarrow H_1) &= softmax(\frac{Q_2 K_1^T}{\sqrt{d_k}})V_1 \end{aligned} \tag{3}$$

EMT [33] concatenates three unimodal utterance representations into a multimodal global representation. Inspired by EMT [33], we concatenate the utterance representations from each modality u_m as the global representation $G = Concat(u_t, u_a, u_v)$ during the cross-attention stage, where $m \in (t, a, v)$. Subsequently, we utilize a Transformer to calculate intermodality information between the modality feature sequences $F_m \in \mathbb{R}^{len \times d}$ and the global representation $G \in \mathbb{R}^{3 \times d}$, as shown in Figure 2 and Equation (4).

$$\begin{aligned} Attention(G \rightarrow F_m) &= Cross\text{-}Attention(G \rightarrow F_m) \\ Attention(F_m \rightarrow G) &= Cross\text{-}Attention(F_m \rightarrow G) \end{aligned} \tag{4}$$

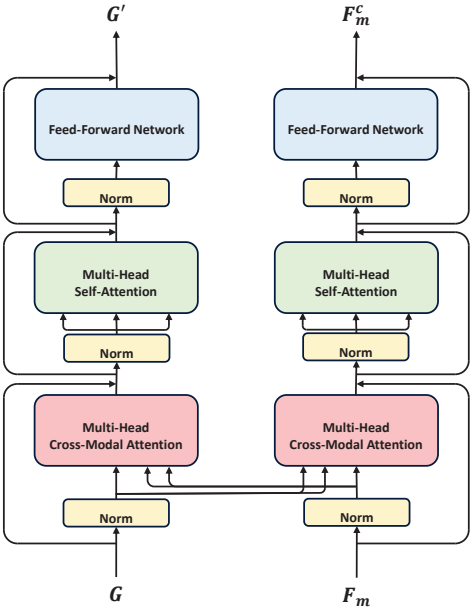


Figure 2. The structure of cross-attention. Cross-attention is used to capture dynamics between the global representation G and unimodal representations F_m.

On the other hand, we utilize modality-specific Transformer encoder layers, denoted as L_s, to capture intramodality information for each modality individually (using Equation (2)). After encoding each modality, we use the self-attention mechanism in Transformer to process the unimodal feature sequences separately, in which the embedding at each position is able to learn the semantic and emotional information contained in the sequences.

MulT [30] used directional encoders for bimodal interactions separately, and subsequently augmented these dynamics with self-attention mechanisms. EMT [33] achieved cross-modal interactions by making global representations and unimodal sequences learn

from each other, while ignoring modality-specific information present in the self-attention unit. CCDA used both cross-modal attention and self-attention; first the two attention mechanisms were isolated, and then it used the cross-correlation loss to make them to interact after sufficiently learning the relevant intra- and intermodal information, respectively. This preserves the specificity information of the different attention mechanisms and optimizes the global representation by backpropagating the cross-modal feature sequences during the training progresses. After the feature sequences in the self-attention module learn the intermodal information of the cross-modal feature sequences, they are able to increase the perceptual field of the final multimodal features and increase the generalization performance of the model.

The use of dual-attention allows the model to process and analyze multimodal data at two different levels, intermodality and intramodality, for a more comprehensive understanding and interpretation of multimodal sentiment data.

3.2.3. Modality Fusion

After passing through the cross-attention stage, the model obtains intermodality information, which is reflected through the global representation G', while in the self-attention stage, to maintain consistency with the global representation, we employ Bi-LSTMs to process the three single-modal feature sequences individually, obtaining each unimodal representation. Meanwhile, we propose the relevant coefficients, which are computed based on the relationship between the modal representation and the initial representation. Relevant coefficients strategy can fuse the representations obtained from dual-attention mechanisms and generate the final multimodal sentiment representation.

To be more specific, after learning intramodality information in the self-attention stage, the model utilizes Bi-LSTMs to transform unimodal feature sequences into feature representations $U'_m \in \mathbb{R}^{b\times d}$, which are specific to each modality. Subsequently, we calculate relevant coefficients based on the correlation between this representation and the initial modal representations $U_m \in \mathbb{R}^{b\times d}$:

$$r_m = \sum (Diag(tanh(U'_m) \bigotimes tanh(U_m)) - 1)^2 \tag{5}$$

where $\otimes$ denotes matrix multiplication, and $Diag(\cdot)$ represents all the diagonal elements of a square matrix. After obtaining the relevance coefficient r_m for each modality, we multiply it with U'_m to obtain the single-modal representation:

$$\widetilde{U}_m = r_m \times U'_m \tag{6}$$

Here, r_m is the relevance coefficient specific to each modality, and U'_m represents the feature representation of the corresponding modality obtained through Bi-LSTMs.

After obtaining the representations for both intermodality and intramodality $\{G', \widetilde{U}_l, \widetilde{U}_a, \widetilde{U}_v\}$, we concatenate the unimodal representations $\{\widetilde{U}_l, \widetilde{U}_a, \widetilde{U}_v\}$ with the global representation G' to create the representation for the sample. Finally, we employ several linear layers in combination with activation functions to make predictions for the ultimate result.

$$y = Pred(Concat(G', \widetilde{U}_l, \widetilde{U}_a, \widetilde{U}_v)) \tag{7}$$

3.3. Cross-Correlation Loss

Most of the current research uses attention mechanisms to capture relevant information from both intramodality and intermodality, but few scholars consider the relationship between these two different attention levels. In order to extract this relationship in dual-attention, we propose a cross-correlation loss to obtain relevant information. By adding it to the objective function, the model is able to accomplish an undirected interaction between two different kinds of attention.

As shown in Figure 3, we use linear projectors to expand the feature sequence dimensions of the two different attention mechanisms and perform modality-specific matrix multiplication to obtain a set of matrices with a shape of (batch, length, length).

$$C_m = F_m^S \bigotimes F_m^C \tag{8}$$

where C_m represents the cross-correlation matrix of the m modality's feature sequences in two different attention mechanisms, $m \in \{t, a, v\}$. The diagonal elements in this matrix represent the correlation between the corresponding positions of the two feature sequences, while the off-diagonal elements represent the redundant information.

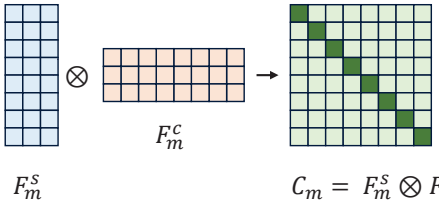


Figure 3. The cross-correlation matrix in dual-attention. We perform modality-specific matrix multiplication on the two types of unimodal feature sequences to obtain a cross-correlation matrix, and we use the diagonal elements of the matrix to represent the indirect interaction between these two feature sequences. The deeper the diagonal elements in the matrix C_m, the stronger the correlation between the two unimodal feature sequences at the corresponding positions is represented.

Taking the textual modalities of the samples in the CMU-MOSI dataset as an example, as shown in Figure 4, the model maximizes the diagonal elements in the intercorrelation matrix in order to capture the correlation between the different attentional mechanisms during the training process. At the same time, nondiagonal elements are minimized in order to reduce redundant information in this process.

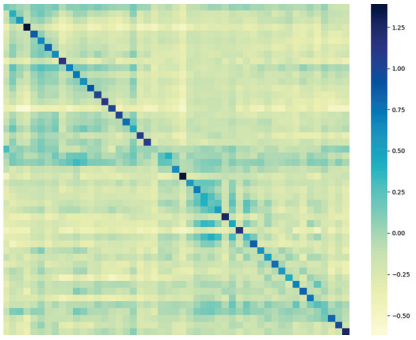


Figure 4. The cross-correlation matrix.

$$\mathcal{L}_{Corr} = \frac{1}{M} \cdot \sum_{m}^{M} (\sum_{i=j}^{n} (c_{ij} - 1)^2 + \sum_{i \neq j}^{n} c_{ij}^2) \tag{9}$$

As shown in Equation (9). The term $\sum_{i=j}^{n} (c_{ij} - 1)^2$ in $\mathcal{L}_{Corr}$ is the correlation term, which denotes the correlation between the sequence of modality features of m in different attention mechanisms, and the other term $\sum_{i \neq j}^{n} c_{ij}^2$ is the redundancy term. Intuitively, the model increases the correlation between different attentional mechanisms by making the

diagonal elements of the cross-correlation matrix close to 1. At the same time, it reduces the redundancy term by making the off-diagonal elements of the cross-correlation matrix close to 0.

3.4. Loss Function

We use MAE and Cross-Correlation loss as the final objective function. As shown in Equations (10) and (11):

$$\mathcal{L}_{MSE} = \frac{1}{N} \cdot \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{10}$$

$$\mathcal{L} = L_{MSE} + \lambda \cdot L_{Corr} \tag{11}$$

Where y denotes the true label of the sample and $\hat{y}$ denotes the predicted label of the model. Since the cross-correlation loss is calculated for all elements in the cross-correlation matrix, setting the weight of the cross-correlation loss too high in the objective function can cause the two attention mechanisms to lose their specificity and, thus, reduce the model performance. Therefore, we set a scaling factor λ in the cross-correlation loss according to the expansion of the feature sequence dimension. We conducted ablation experiments on different scaling weights on two datasets, as shown in Section 4.3.

4. Experiment

4.1. Preparations

4.1.1. Datasets

A multimodal dataset collects information from different modalities, such as text, speech, and vision, providing researchers with opportunities to gain a deeper understanding and analysis of sentiment expression. Three publicly available datasets are used in this article, including CMU-MOSI, CMU-MOSEI, and CH-SIMS. Figure 5 illustrates some samples from the CMU-MOSI and CMU-MOSEI datasets, and Figure 6 illustrates the CH-SIMS dataset.

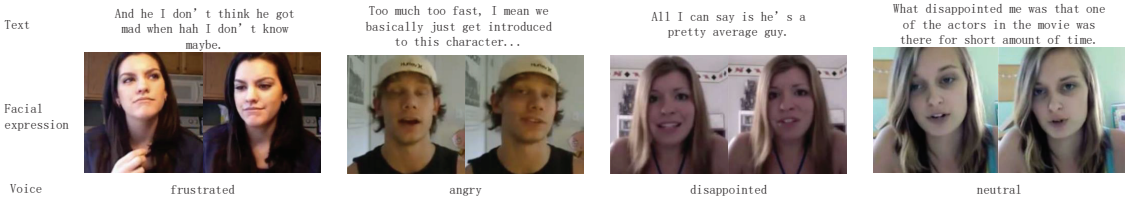


Figure 5. Examples in the CMU-MOSI and CMU-MOSEI datasets.

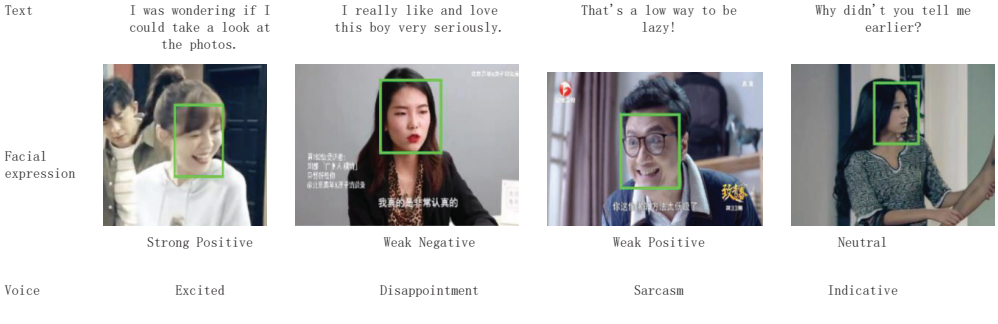


Figure 6. Examples in the CH-SIMS dataset. The green box in the image captures the speaker's facial expression.

CMU-MOSI [46] (Multimodal Opinion Level Sentiment Intensity) is a multimodal dataset with character subjective sentiment and sentiment intensity annotations. It contains 2199 multimodal samples from 93 YouTube videos, with each video ranging from 2–5 min and featuring 89 different speakers. Each video is annotated with sentiment intensity, ranging from strong positive to strong negative on a scale from -3 to 3.

Another dataset is CMU-MOSEI [21] (CMU Multimodal Opinion Sentiment and Emotion Intensity), an upgraded version of the CMU-MOSI dataset and one of the largest sentiment analysis datasets covering multiple fields, including sentiment recognition. CMU-MOSEI contains 23,453 manually annotated video clips from 5000 videos on YouTube, including 1000 different speakers and 250 different topics, covering almost all topics in daily life. CMU-MOSEI uses the same annotation method as CMU-MOSI.

In addition, considering the research on multimodal sentiment analysis in the Chinese community, we also used CH-SIMS [47], a refined Chinese multimodal dataset. It contains 2281 samples from 60 videos collected from movies, TV shows, and variety shows. Compared to the first two datasets, it not only includes multimodal sentiment labels but also provides independent fine-grained single-modality sentiment labels for each sample. Each label in this dataset is manually annotated from -1 (strongly positive) to 1 (strongly negative). The statistical information of these three datasets is shown in Table 2.

Table 2. Statistics of CMU-MOSI, CMU-MOSEI, and CH-SIMS datasets.

Dataset	Train	Validation	Test	All
CMU-MOSI	1284	229	686	2199
CMU-MOSEI	16,326	1871	4659	22,856
CH-SIMS	1368	456	457	2281

4.1.2. Data Processing

We targeted the different modalities for processing. For the text modality, we used the BERT-based-uncased model to encode the CMU-MOSI and CMU-MOSEI datasets. In addition, for the Chinese multimodal sentiment dataset CH-SIMS, we used the BERT-based-Chinese model for text encoding. This step helps to transform text data into vector representations with rich semantic information.

When processing the speech modality, we used the COVAREP tool to extract audio features, including pitch, glottal source parameters, and 12 Mel-frequency cepstral coefficients (MFCCs). These features capture sound frequencies, voice source properties, and acoustic features in speech, providing important information for sentiment analysis. For the CH-SIMS dataset, we used the Librosa toolkit in Python to extract speech features such as log fundamental frequency, constant-Q chromatograms, and 20 MFCCs.

For visual modality, we used the Facet tool to extract 35 facial features for the CMU-MOSI and CMU-MOSEI datasets, which record facial muscle movements related to sentiment. For the Chinese sentiment dataset CH-SIMS, we used the OpenFace 2.0 toolkit to extract 17 facial action units, 68 facial landmarks, and some features related to head posture and eye movements. These facial features capture information related to facial expressions in sentiment expression, providing important visual data for multimodal sentiment analysis.

4.1.3. Baseline

In the field of multimodal sentiment analysis, there exists a series of different baseline models, each with its own characteristics. In order to comprehensively verify the performance of the method proposed in this paper, we compared it with many current methods, which mainly include the following:

TFN [24]. The tensor fusion network is a tensor-fusion-based method that computes the triple Cartesian product between three modalities to explicitly capture intramodal-

ity and intermodality dynamic information. It utilizes tensor operations to capture the interaction and fusion of multimodal information.

LMF [25]. Similar to TFN, low-rank multimodal fusion also relies on tensor operations, but it cleverly uses modality-specific low-rank factors to more efficiently compute multimodal representations, improving fusion efficiency while ensuring information quality.

MulT [30]. Multimodal Transformer adopts a bidirectional cross-modal attention mechanism to calculate the relation between two different modalities separately. The method is based on Transformer architecture, which can better capture dynamic information between different modalities.

MISA [48]. Modality-invariant and-specific representations for multimodal sentiment analysis. MISA uses a subspace learning approach to map each modality to two different subspaces for learning, providing a comprehensive view of multimodal representation learning and achieving better fusion results.

Self-MM [49]. The self-supervised multitask multimodal sentiment analysis network designs an unimodal label generation module based on self-supervised learning to obtain independent unimodal representations. It utilizes self-supervised learning to improve model performance. Also, it jointly trains multimodal and unimodal tasks to learn modal consistency and variability.

AMML [50]. Adaptive multimodal meta-learning uses a meta-learning approach to train unimodal networks and applies them to multimodal inference. This method focuses on network adaptability and optimizes unimodal representations through adaptive learning rate adjustment for better multimodal fusion.

MMIM [51]. MultiModal InfoMax proposes a hierarchical maximization of mutual information framework, which improves the consistency and information density of multimodal representations by maximizing mutual information and preserves task-relevant information through multimodal fusion.

EMT [33]. Efficient Multimodal Transformer proposes an efficient network based on the Transformer architecture for integrating multimodal information. This network utilizes unimodal encoders to obtain multimodal representations and enables mutual learning between multimodal global representations and unimodal feature sequences.

4.1.4. Hyper-Parameter Setting

We use the Pytorch in deep learning to build our model and optimize it with the Adam optimizer, and we adopt an early-stop strategy. Table 3 shows the parameter settings for CCDA trained on CMU-MOSI, CMU-MOSEI, and CH-SIMS datasets. In the cross-attention section, we adopt the same hyper-parameter settings as EMT, and in the self-attention section, we use the Transformer parameter settings in MulT. To reflect the accuracy of the results, we conducted five experiments and averaged each metric in the experimental results.

Table 3. Hyper-parameter settings of CCDA on three datasets.

Hyper-Parameter	CMU-MOSI	CMU-MOSEI	CH-SIMS
Batch size	32	16	32
Early stop (epochs)	16	8	16
Learning rate	1×10^{-3}	1×10^{-4}	1×10^{-3}
Optimizer	Adam	Adam	Adam
Dimension of feature and representation	128	128	128
Transformer layers in cross-attention	3	2	4
Cross-attention heads	4	4	4
Transformer layers in self-attention	2	2	2
Attention dropout	0.1	0.1	0.1
Stacked LSTM layers for self-attention	2	2	2
Stacked LSTM dropout	0.1	0.1	0.1
λ in cross-correlation loss	5×10^{-5}	5×10^{-5}	1×10^{-3}
Projector dims in cross-correlation loss	1024	1024	256

4.2. Result Analysis

4.2.1. Evaluation Metrics

In regression tasks, we mainly use two metrics to measure model performance: mean absolute error (MAE) and Pearson correlation coefficient (Corr). MAE is used to measure the average absolute error between the model's predicted values and the true labels, with lower values indicating better model performance. Corr is used to measure the correlation between the model's predicted results and the true labels, with values closer to 1 indicating better model performance. Additionally, we also convert the model's output results into classification task metrics, including Acc-k and F1-score. Acc-2, Acc-5, and Acc-7 on the CMU-MOSI and CMU-MOSEI datasets and Acc-2, Acc-3, and Acc-5 on CH-SIMS are used to evaluate the model's accuracy in multiclassification tasks, with larger values indicating better model performance. F1-score represents the harmonic mean of precision and recall and is used to evaluate the balance between positive and negative categories. A higher F1-score indicates better model performance in classification tasks.

4.2.2. Quantitative Analysis

The experimental data for TFN, LMF, MulT, MISA, Self-MM, and MMIM come from [51]. For the other models, we conducted five experiments on each of the three datasets using publicly available source code and averaged the experimental results for each model. In all evaluation metrics, except for MAE, larger values indicate better model performance. The experimental results are compared in Tables 4–6.

Table 4 shows the model's results on the CMU-MOSI dataset. Compared to the EMT model, CCDA improved by 0.009 on the regression metrics MAE and Corr. In terms of classification task metrics, CCDA improved by 0.6% on Acc-2 and Acc-5 and 0.7% on Acc-7 and achieved a 0.6% improvement in F1-score over the best model. Similarly, as shown in Table 5, CCDA's performance on CMU-MOSEI improved by 0.003 on MAE, 0.006 on Corr, 0.5% on Acc-7, 0.4% on Acc-5, 0.6% on Acc-2, and 0.7% on F1-score compared to EMT. Table 6 shows the experimental results of the model on CH-SMIS, where CCDA achieved better results on some metrics, such as 0.006 on MAE, 0.005 on Corr, 1.4% on Acc-3, 1.2% on Acc-2, and 0.9% on F1-score. However, its performance on the 5-classification task was slightly worse than that of the EMT model. We believe that while CCDA improves coarse-grained sentiment classification, it does not improve much for fine-grained classification.

The experimental results show that the CCDA model using cross-modal attention and self-attention is able to learn intra- and intermodal dynamics. Dual-attention makes the model analyze the sample more comprehensively, and the cross-correlation loss enables some degree of interaction between different levels of dual-attention mechanism. In addition, the relevant coefficients guide the multimodal feature fusion stage, which allows the model to improve performance while increasing the model's generalization ability.

Table 4. Experiments on CMU-MOSI. Where ↑ indicates that the higher the metric the stronger the performance of the model, and ↓ is the opposite. Bold numbers indicate the model with the best results at that metric.

Models	MAE (↓)	Corr (↑)	Acc-7 (↑)	Acc-5 (↑)	Acc-2 (↑)	F1 (↑)
TFN [24]	0.901	0.698	34.9	-	80.8	80.7
LMF [25]	0.917	0.695	33.2	-	82.5	82.4
MulT [30]	0.846	0.725	40.4	46.7	83.4	83.5
MISA [48]	0.804	0.764	-	-	82.1	82.0
Self-MM [49]	0.717	0.793	46.4	52.8	84.6	84.6
MMIM [51]	0.712	0.790	46.9	53.0	85.3	85.4
AMML [50]	0.723	0.792	46.3	-	84.9	84.8
EMT [33]	0.705	0.798	47.4	54.1	85.0	85.0
Ours	**0.696**	**0.807**	**48.0**	**54.8**	**85.7**	**85.6**

Table 5. Experiments on CMU-MOSEI. Where ↑ indicates that the higher the metric the stronger the performance of the model, and ↓ is the opposite. Bold numbers indicate the model with the best results at that metric.

Models	MAE (↓)	Corr (↑)	Acc-7 (↑)	Acc-5 (↑)	Acc-2 (↑)	F1 (↑)
TFN [24]	0.593	0.700	50.2	-	82.5	82.1
LMF [25]	0.623	0.677	48.0	-	82.0	82.1
MulT [30]	0.564	0.731	52.6	54.1	83.5	83.6
MISA [48]	0.568	0.724	-	-	84.2	84.0
Self-MM [49]	0.533	0.766	53.6	55.4	85.0	85.0
MMIM [51]	0.536	0.764	53.2	55.0	85.0	85.1
AMML [50]	0.614	0.776	52.4	-	85.3	85.2
EMT [33]	0.527	0.774	54.5	56.3	86.0	86.0
Ours	**0.524**	**0.780**	**55.0**	**56.7**	**86.6**	**86.7**

Table 6. Experiments on CH-SIMS. Where ↑ indicates that the higher the metric the stronger the performance of the model, and ↓ is the opposite. Bold numbers indicate the model with the best results at that metric.

Models	MAE (↓)	Corr (↑)	Acc-5 (↑)	Acc-3 (↑)	Acc-2 (↑)	F1 (↑)
TFN [24]	0.437	0.582	-	-	77.1	76.9
LMF [25]	0.438	0.578	-	-	77.4	77.4
MulT [30]	0.442	0.581	40.0	65.7	78.2	78.5
MISA [48]	0.447	0.563	-	-	76.5	76.6
Self-MM [49]	0.411	0.601	43.1	66.1	78.6	78.6
MMIM [51]	0.422	0.597	42.0	65.5	78.3	78.2
AMML [50]	0.437	0.583	41.2	64.2	78.0	78.1
EMT [33]	0.396	0.623	**43.5**	67.4	80.1	80.1
Ours	**0.393**	**0.628**	43.3	**68.3**	**81.1**	**81.0**

4.3. Ablation Study

To validate the role of the dual-attention mechanism in the CCDA model and the effects of the multimodal fusion strategy and cross-correlation loss on the performance of the model, we conducted ablation experiments on two datasets, CMU-MOSI and CH-SIMS.

4.3.1. Dual-Attention Mechanisms

The MulT model first uses multiple cross-modal attention mechanisms between the bimodal features and later uses a Transformer encoder. Throughout the training process, the model does not capture modality-specific intramodal information, but, rather, directly interacts cross-modally. While this enables unimodal features to perceive affective information from neighboring modalities upfront, this will lose the modality specific information. EMT splices the unimodal representation as a global representation and selects the Transformer encoder to interact with the global and unimodal representations, but in the process does not model each modality individually, which can result in the model failing to capture affective information that exists within a single modality.

We designed a set of experiments to verify the effect of different mechanisms in dual-attention on model performance, as shown in Tables 7 and 8. The first rows of Tables 7 and 8 validate the model performance in the case of using only the unimodal self-attention mechanism, where we used the relevant coefficient to guide the unimodal representations. The final multimodal feature has only three unimodal representations and does not contain the global representation in standard CCDA. The second row verifies the model performance in the case where only the cross-modal attention mechanism is used, in which case the multimodal features are global representations, not containing unimodal representations, and the relevant coefficients cannot be used. Neither of these cases uses the cross-correlation loss. The third row indicates that we use the standard CCDA model for training.

Table 7. Impact of dual-attention in CCDA on CMU-MOSI. Where ↑ indicates that the higher the metric the stronger the performance of the model, and ↓ is the opposite. Bold numbers indicate the model with the best results at that metric.

	MAE (↓)	Corr (↑)	Acc-7 (↑)	Acc-5 (↑)	Acc-2 (↑)	F1 (↑)
Only self-attention	0.734	0.764	45.1	51.9	83.0	83.0
Only cross-attention	0.722	0.787	46.4	53.2	84.4	84.5
Standard CCDA	**0.696**	**0.807**	**48.0**	**54.8**	**85.7**	**85.6**

Table 8. Impact of dual-attention in CCDA on CH-SIMS. Where ↑ indicates that the higher the metric the stronger the performance of the model, and ↓ is the opposite. Bold numbers indicate the model with the best results at that metric.

	MAE (↓)	Corr (↑)	Acc-5 (↑)	Acc-3 (↑)	Acc-2 (↑)	F1 (↑)
Only self-attention	0.443	0.602	40.7	65.7	78.4	78.3
Only cross-attention	0.415	0.613	41.9	66.8	79.9	79.9
Standard CCDA	**0.393**	**0.628**	**43.3**	**68.3**	**81.1**	**81.0**

The data in the table show that when using self-attention, the model is unable to focus on cross-modal interaction information and only fuses the representations of each modality at a later stage. While the model performance improves when using only cross-attention, this is due to the fact that it discriminates the sentiment attributes of the sample as a whole from a global perspective, and compared to self-attention, cross-attention tends to select the information that is the most beneficial to the overall judgment when performing interactions. In the standard CCDA model, the model's performance is optimal when dual-attention is used at the same time, which suggests that CCDA retains as much of the affective information in dual-attention as possible.

4.3.2. Fusion Strategy with Relevant Coefficients

Before performing multimodal fusion in the model, we adjusted the unimodal representations based on the relevant coefficients computed between unimodal representations and their respective initial modality representations. Subsequently, these representations were concatenated with the global multimodal representation. To validate the effectiveness of our proposed fusion strategy, we conducted experiments on both Chinese and English datasets. We compared the performance of models with and without considering unimodal relevant coefficients, where the unimodal representations, computed after self-attention and subsequent Bi-LSTMs, were directly concatenated with the global multimodal representation, and then fed into the fusion and prediction module. We also compared these results with the standard version of CCDA. The comparative experimental results are shown in Tables 9 and 10.

According to Tables 9 and 10, it is evident that in multimodal fusion, the model's performance significantly improves when unimodal features are augmented with relevant coefficients compared to direct concatenation. Specifically, there is a 1.5% improvement in Acc-7. Therefore, the use of relevant coefficients in the multimodal feature fusion stage enables the model to analyze the relations between the self-attention modality representations and the source feature representations, and, thus, to achieve higher accuracy on multiclassification.

Table 9. Impact of correlation coefficients in fusion strategy on CMU-MOSI. Where ↑ indicates that the higher the metric the stronger the performance of the model, and ↓ is the opposite. Bold numbers indicate the model with the best results at that metric.

	MAE (↓)	Corr (↑)	Acc-7 (↑)	Acc-5 (↑)	Acc-2 (↑)	F1 (↑)
Direct Concat	0.713	0.790	46.5	53.8	85.2	85.2
Standard CCDA	**0.696**	**0.807**	**48.0**	**54.8**	**85.7**	**85.6**

Table 10. Impact of correlation coefficients in fusion strategy on CH-SIMS. Where ↑ indicates that the higher the metric the stronger the performance of the model, and ↓ is the opposite. Bold numbers indicate the model with the best results at that metric.

	MAE (↓)	Corr (↑)	Acc-5 (↑)	Acc-3 (↑)	Acc-2 (↑)	F1 (↑)
Direct Concat	0.408	0.614	41.2	66.4	80.4	80.4
Standard CCDA	**0.393**	**0.628**	**43.3**	**68.3**	**81.1**	**81.0**

4.3.3. Cross-Correlation Loss

Additionally, this study assumes a certain degree of cross-correlation between self-attention and cross-attention. Thus, we introduced a cross-correlation loss function to facilitate indirect interaction between these two attention mechanisms. To assess the impact of cross-correlation loss on model performance, we conducted ablation experiments on the CMU-MOSI and CH-SIMS datasets, as shown in Tables 11 and 12.

Table 11. Impact of cross-correlation loss in the objective function on CMU-MOSI. Where ↑ indicates that the higher the metric the stronger the performance of the model, and ↓ is the opposite. Bold numbers indicate the model with the best results at that metric.

	MAE (↓)	Corr (↑)	Acc-7 (↑)	Acc-5 (↑)	Acc-2 (↑)	F1 (↑)
w/o corr loss	0.708	0.795	47.4	54.2	84.9	84.9
Standard CCDA	**0.696**	**0.807**	**48.0**	**54.8**	**85.7**	**85.6**

Table 12. Impact of cross-correlation loss in the objective function on CH-SIMS. Where ↑ indicates that the higher the metric the stronger the performance of the model, and ↓ is the opposite. Bold numbers indicate the model with the best results at that metric.

	MAE (↓)	Corr (↑)	Acc-5 (↑)	Acc-3 (↑)	Acc-2 (↑)	F1 (↑)
w/o corr loss	0.400	0.610	42.0	66.7	80.1	80.1
Standard CCDA	**0.393**	**0.628**	**43.3**	**68.3**	**81.1**	**81.0**

It can be observed that adding cross-correlation loss to the objective function significantly enhances the model's performance. This improvement is particularly pronounced in multiclass tasks, indicating that cross-correlation loss has a substantial impact on model performance in multimodal sentiment analysis. Further analysis reveals that cross-correlation loss establishes a closer connection between self-attention and cross-attention in the model, enabling better integration of information from multimodal data. This indirect interaction helps the model better understand the relationships between different modalities, thereby improving overall sentiment analysis performance. In multimodal sentiment analysis tasks, such enhanced connectivity is highly beneficial. Moreover, the results on different datasets demonstrate the universality of the improvement brought by cross-correlation loss, indicating that it is not limited to specific datasets. This strengthens the scalability and generality of our approach.

4.3.4. Scaling Factor in Cross-Correlation Loss

When calculating the cross-correlation loss, the model expands the dimensions of the feature sequences. As a result, the values of elements in the correlation matrix become relatively large. To balance the cross-correlation loss in the objective function, we introduced scaling factors. Figure 7 illustrates the impact of scaling factors on the final results. Since we set different feature dimensions for unimodal features from different datasets (128 for CMU-MOSI and CMU-MOSEI, 32 for CH-SIMS), and applied different linear mapping layers for dimension expansion when calculating the cross-correlation loss for different datasets, the optimal scaling factors also vary. Specifically, we used 5×10^{-5} for CMU-MOSI and 1×10^{-3} for CH-SIMS.

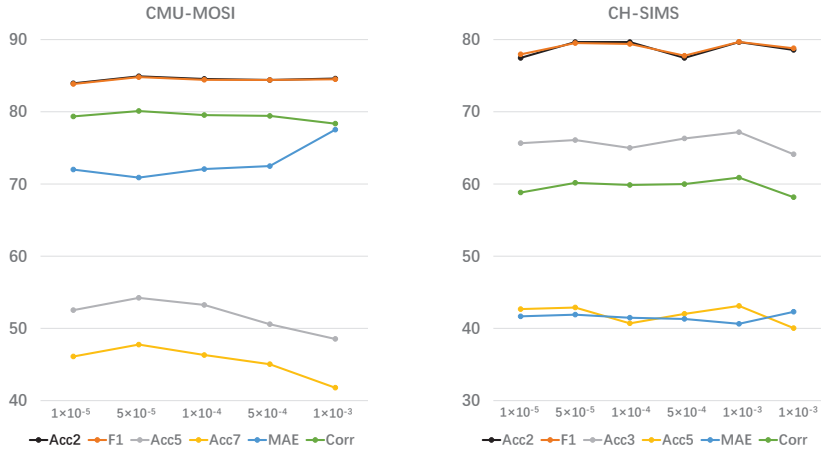


Figure 7. Impact of scaling weights in cross-correlation loss. Where a lower MAE (blue line) indicates better model performance, showing an opposite trend to the other metrics.

5. Conclusions

In this paper, we introduced the cross-correlation in dual-attention (CCDA) model aimed at fusing multimodal features and perceiving human sentiment analysis. We used dual-attention to obtain information about the intra- and intermodal dynamics contained in the samples from different perspectives, and in order to capture the relation that exists between different attention mechanisms, we propose the cross-correlation loss, which allows the cross-modal attention and the self-attention mechanism to complete a nondirective interaction. In addition, we introduce a new fusion strategy in the multimodal feature fusion stage by using correlation coefficients, which allows the initial unimodal representation to guide the multimodal fusion.

We conducted comprehensive experiments on three commonly used public datasets in the multimodal sentiment analysis domain, including CMU-MOSI, CMU-MOSEI, and CH-SIMS. We compared the CCDA model with baseline models and found that our model demonstrated a significant advantage on all three datasets. Through experimentation, we demonstrated the strong performance of the CCDA model in multimodal sentiment analysis tasks, offering new insights for further research and applications in this field.

Since the Transformer is used in this study, an issue that cannot be ignored is the number of parameters of the model, which increases rapidly as the number of attention block increases. In addition, the cross-correlation loss as well as the relevant coefficients in this study were calculated using matrix multiplication, which increases the computational complexity of the model, and there is still some redundant information in the calculation process.

Future research work:In view of the problems encountered in this study, future research efforts should focus on (1) reducing the number of parameters of the model while ensuring the model performance, (2) reducing the computational complexity of the model, and (3) further reducing the redundant information generated during the training process of the model.

Given the challenges faced in real-world multimodal sentiment analysis, especially in scenarios involving missing modal information, future research could focus on enhancing the model's robustness and accuracy in handling missing modal information. This would ensure the effectiveness and reliability of the model in a wider range of practical applications.

Author Contributions: Conceptualization, P.W.; methodology, P.W.; software, P.W.; validation, P.W., S.L., and J.C.; formal analysis, P.W.; investigation, P.W.; resources, P.W.; data curation, P.W.; writing—original draft preparation, P.W.; writing—review and editing, P.W.; visualization, P.W.; supervision, P.W.; project administration, P.W.; funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (61762085) and the Natural Science Foundation of Xinjiang Uygur Autonomous Region Project (2019D01C081).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in https://github.com/CMU-MultiComp-Lab/CMU-MultimodalSDK, reference number [21,46] and https://github.com/thuiar/MMSA, reference number [47]. These data were derived from the following resources available in the public domain: https://drive.google.com/drive/folders/1A2S4pqCHryGmiqnNSPLv7rEg63WvjCSk.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Atrey, P.K.; Hossain, M.A.; El Saddik, A.; Kankanhalli, M.S. Multimodal fusion for multimedia analysis: A survey. *Multimed. Syst.* **2010**, *16*, 345–379. [CrossRef]
2. Gandhi, A.; Adhvaryu, K.; Poria, S.; Cambria, E.; Hussain, A. Multimodal sentiment analysis: A systematic review of history, datasets, multimodal fusion methods, applications, challenges and future directions. *Inf. Fusion* **2023**, *91*, 424–444. [CrossRef]
3. Somandepalli, K.; Guha, T.; Martinez, V.R.; Kumar, N.; Adam, H.; Narayanan, S. Computational media intelligence: Human-centered machine analysis of media. *Proc. IEEE* **2021**, *109*, 891–910. [CrossRef]
4. Stappen, L.; Baird, A.; Schumann, L.; Bjorn, S. The multimodal sentiment analysis in car reviews (muse-car) dataset: Collection, insights and improvements. *IEEE Trans. Affect. Comput.* **2021**, *14*, 1334–1350. [CrossRef]
5. Soleymani, M.; Garcia, D.; Jou, B.; Schuller, B.; Chang, S.F.; Pantic, M. A survey of multimodal sentiment analysis. *Image Vis. Comput.* **2017**, *65*, 3–14. [CrossRef]
6. Poria, S.; Hazarika, D.; Majumder, N.; Mihalcea, R. Beneath the tip of the iceberg: Current challenges and new directions in sentiment analysis research. *IEEE Trans. Affect. Comput.* **2020**, *14*, 108–132. [CrossRef]
7. Cambria, E.; Das, D.; Bandyopadhyay, S.; Feraco, A. Affective computing and sentiment analysis. In *A Practical Guide to Sentiment Analysis*; Springer International Publishing: Cham, Switzerland, 2017; pp. 1–10.
8. Poria, S.; Cambria, E.; Bajpai, R.; Hussain, A. A review of affective computing: From unimodal analysis to multimodal fusion. *Inf. Fusion* **2017**, *37*, 98–125. [CrossRef]
9. Morency, L.P.; Mihalcea, R.; Doshi, P. Towards multimodal sentiment analysis: Harvesting opinions from the web. In Proceedings of the 13th International Conference on Multimodal Interfaces, Alicante, Spain, 14–18 November 2011; pp. 169–176.
10. Poria, S.; Chaturvedi, I.; Cambria, E.; Hussain, A. Convolutional MKL based multimodal emotion recognition and sentiment analysis. In Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM), Barcelona, Spain, 12–15 December 2016; pp. 439–448.
11. Poria, S.; Cambria, E.; Hazarika, D.; Majumder, N.; Zadeh, A.; Morency, L.P. Context-dependent sentiment analysis in user-generated videos. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 873–883.
12. Zadeh, A.; Liang, P.P.; Poria, S.; Vij, P.; Cambria, E.; Morency, L.P. Multi-attention recurrent network for human communication comprehension. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
13. Wang, H.; Meghawat, A.; Morency, L.P.; Xing, E.P. Select-additive learning: Improving generalization in multimodal sentiment analysis. In Proceedings of the 2017 IEEE International Conference on Multimedia and Expo (ICME), Hong Kong, China, 10–14 July 2017; pp. 949–954.
14. Glodek, M.; Reuter, S.; Schels, M.; Dietmayer, K.; Schwenker, F. Kalman filter based classifier fusion for affective state recognition. In Proceedings of the Multiple Classifier Systems: 11th International Workshop, MCS 2013, Nanjing, China, 15–17 May 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 85–94.
15. Cai, G.; Xia, B. Convolutional neural networks for multimedia sentiment analysis. In Proceedings of the Natural Language Processing and Chinese Computing: 4th CCF Conference, NLPCC 2015, Nanchang, China, 9–13 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 159–167.
16. Alam, F.; Riccardi, G. Predicting personality traits using multimodal information. In Proceedings of the 2014 ACM Multi Media on Workshop on Computational Personality Recognition, Orlando, FL, USA, 7 November 2014; pp. 15–18.

17. Poria, S.; Cambria, E.; Gelbukh, A. Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In Proceedings of the 2015 Conference on Empirical Methods in nAtural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 2539–2544.
18. Kumar, A.; Vepa, J. Gated mechanism for attention based multi modal sentiment analysis. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 4477–4481.
19. Zhang, S.; Li, B.; Yin, C. Cross-Modal Sentiment Sensing with Visual-Augmented Representation and Diverse Decision Fusion. *Sensors* **2022**, *22*, 74. [CrossRef]
20. Zadeh, A.; Liang, P.P.; Mazumder, N.; Poria, S.; Cambria, E.; Morency, L.P. Memory fusion network for multi-view sequential learning. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
21. Zadeh, A.B.; Liang, P.P.; Poria, S.; Cambria, E.; Morency, L.P. Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; pp. 2236–2246.
22. Liang, P.P.; Liu, Z.; Zadeh, A.; Morency, L.P. Multimodal language analysis with recurrent multistage fusion. *arXiv* **2018**, arXiv:1808.03920.
23. Wang, Y.; Shen, Y.; Liu, Z.; Liang, P.P.; Zadeh, A.; Morency, L.P. Words can shift: Dynamically adjusting word representations using nonverbal behaviors. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 29–31 January 2019; Volume 33, pp. 7216–7223.
24. Zadeh, A.; Chen, M.; Poria, S.; Cambria, E.; Morency, L.P. Tensor fusion network for multimodal sentiment analysis. *arXiv* **2017**, arXiv:1707.07250.
25. Liu, Z.; Shen, Y.; Lakshminarasimhan, V.B.; Liang, P.P.; Zadeh, A.; Morency, L.P. Efficient low-rank multimodal fusion with modality-specific factors. *arXiv* **2018**, arXiv:1806.00064.
26. Barezi, E.J.; Fung, P. Modality-based factorization for multimodal fusion. *arXiv* **2018**, arXiv:1811.12624.
27. Liang, P.P.; Liu, Z.; Tsai, Y.H.H.; Zhao, Q.; Salakhutdinov, R.; Morency, L.P. Learning representations from imperfect time series data via tensor rank regularization. *arXiv* **2019**, arXiv:1907.01011.
28. Jin, T.; Huang, S.; Li, Y.; Zhang, Z. Dual low-rank multimodal fusion. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16–20 November 2020; pp. 377–387.
29. Verma, S.; Wang, J.; Ge, Z.; Shen, R.; Jin, F.; Wang, Y.; Chen, F.; Liu, W. Deep-HOSeq: Deep higher order sequence fusion for multimodal sentiment analysis. In Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM), Sorrento, Italy, 17–20 November 2020; pp. 561–570.
30. Tsai, Y.H.H.; Bai, S.; Liang, P.P.; Kolter, J.Z.; Morency, L.P.; Salakhutdinov, R. Multimodal transformer for unaligned multimodal language sequences. In Proceedings of the conference. Association for Computational Linguistics. Meeting, Florence, Italy, 28 July–2 August 2019; NIH Public Access: 2019; Volume 2019, p. 6558.
31. Arjmand, M.; Dousti, M.J.; Moradi, H. Teasel: A transformer-based speech-prefixed language model. *arXiv* **2021**, arXiv:2109.05522.
32. Cheng, H.; Yang, Z.; Zhang, X.; Yang, Y. Multimodal Sentiment Analysis Based on Attentional Temporal Convolutional Network and Multi-layer Feature Fusion. *IEEE Trans. Affect. Comput.* **2023**, *14*, 3149–3163. [CrossRef]
33. Sun, L.; Lian, Z.; Liu, B.; Tao, J. Efficient multimodal transformer with dual-level feature restoration for robust multimodal sentiment analysis. *IEEE Trans. Affect. Comput.* **2023**, 1–17. [CrossRef]
34. Fu, Z.; Liu, F.; Xu, Q.; Qi, J.; Fu, X.; Zhou, A.; Li, Z. NHFNET: A non-homogeneous fusion network for multimodal sentiment analysis. In Proceedings of the 2022 IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, 18–22 July 2022; pp. 1–6.
35. Lian, Z.; Tao, J.; Liu, B.; Huang, J. Conversational emotion analysis via attention mechanisms. *arXiv* **2019**, arXiv:1910.11263.
36. Chen, Q.; Huang, G.; Wang, Y. The weighted cross-modal attention mechanism with sentiment prediction auxiliary task for multimodal sentiment analysis. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2022**, *30*, 2689–2695. [CrossRef]
37. Xiao, L.; Wu, X.; Wu, W.; Yang, J.; He, L. Multi-channel attentive graph convolutional network with sentiment fusion for multimodal sentiment analysis. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 22–27 May 2022; pp. 4578–4582.
38. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.***2017** *, 30.* . [CrossRef]
39. Chen, M.; Wang, S.; Liang, P.P.; Baltrušaitis, T.; Zadeh, A.; Morency, L.P. Multimodal sentiment analysis with word-level fusion and reinforcement learning. In Proceedings of the 19th ACM International Conference on Multimodal Interaction, Glasgow, UK, 13–17 November 2017; pp. 163–171.
40. Hu, G.; Lin, T.E.; Zhao, Y.; Lu, G.; Wu, Y.; Li, Y. Unimse: Towards unified multimodal sentiment analysis and emotion recognition. *arXiv* **2022**, arXiv:2211.11256.
41. Wang, F.; Tian, S.; Yu, L.; Liu, J.; Wang, J.; Li, K.; Wang, Y. TEDT: Transformer-Based Encoding–Decoding Translation Network for Multimodal Sentiment Analysis. *Cogn. Comput.* **2023**, *15*, 289–303. [CrossRef]
42. Tang, J.; Liu, D.; Jin, X.; Peng, Y.; Zhao, Q.; Ding, Y.; Kong, W. Bafn: Bi-direction attention based fusion network for multimodal sentiment analysis. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *33*, 1966–1978. [CrossRef]

43. Lin, Z.; Liang, B.; Long, Y.; Dang, Y.; Yang, M.; Zhang, M.; Xu, R. Modeling intra-and inter-modal relations: Hierarchical graph contrastive learning for multimodal sentiment analysis. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; pp. 7124–7135.
44. Wang, H.; Li, X.; Ren, Z.; Wang, M.; Ma, C. Multimodal Sentiment Analysis Representations Learning via Contrastive Learning with Condense Attention Fusion. *Sensors* **2023**, *23*, 2679. [CrossRef] [PubMed]
45. Wu, Y.; Zhao, Y.; Yang, H.; Chen, S.; Qin, B.; Cao, X.; Zhao, W. Sentiment word aware multimodal refinement for multimodal sentiment analysis with ASR errors. *arXiv* **2022**, arXiv:2203.00257.
46. Zadeh, A.; Zellers, R.; Pincus, E.; Morency, L.P. Mosi: Multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. *arXiv* **2016**, arXiv:1606.06259.
47. Yu, W.; Xu, H.; Meng, F.; Zhu, Y.; Ma, Y.; Wu, J.; Zou, J.; Yang, K. Ch-sims: A chinese multimodal sentiment analysis dataset with fine-grained annotation of modality. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 3718–3727.
48. Hazarika, D.; Zimmermann, R.; Poria, S. Misa: Modality-invariant and-specific representations for multimodal sentiment analysis. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 1122–1131.
49. Yu, W.; Xu, H.; Yuan, Z.; Wu, J. Learning modality-specific representations with self-supervised multi-task learning for multimodal sentiment analysis. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 2–9 February 2021; Volume 35, pp. 10790–10797.
50. Sun, Y.; Mai, S.; Hu, H. Learning to learn better unimodal representations via adaptive multimodal meta-learning. *IEEE Trans. Affect. Comput.* **2022**, *14*, 2209–2223. [CrossRef]
51. Han, W.; Chen, H.; Poria, S. Improving multimodal fusion with hierarchical mutual information maximization for multimodal sentiment analysis. *arXiv* **2021**, arXiv:2109.00412.

applied sciences

MDPI

Article

Implementing and Evaluating a Font Recommendation System through Emotion-Based Content-Font Mapping

Soon-Bum Lim [1], Young-Seo Ji [1], Byunghak Ahn [2], Jae Hong Park [3] and Yoojeong Song [4,*]

[1] Department of IT Engineering, Research Institute of ICT Convergence, Sookmyung Women's University, Seoul 04310, Republic of Korea; sblim@sookmyung.ac.kr (S.-B.L.); jyseo0102@sookmyung.ac.kr (Y.-S.J.)
[2] Visual Communication Design, School of Design, Hongik University, Seoul 04066, Republic of Korea; ahn.hisd@hongik.ac.kr
[3] Department of Visual Arts, Mokpo National University, Muan-gun 58554, Republic of Korea; jaehongpark@mnu.ac.kr
[4] School of Computer Science, Semyung University, Jecheon 27136, Republic of Korea
* Correspondence: yjsong@semyung.ac.kr

Abstract: Rapid digital content growth demands pivotal font selection for design and communication. Our study focuses on a font recommendation system that aligns fonts with content emotions. To achieve this, we define font-emotions and quantify them. Additionally, we leverage deep learning techniques for content analysis. Understanding common emotional perceptions, we aimed to align fonts with content emotions. After evaluating diverse mapping methods, we determined a correlation analysis-based model to be most effective. Implementing this model, we verified its utility through usability evaluations. Our proposed system not only assists users with limited design knowledge in receiving contextually fitting font suggestions but also extends its application across various digital content realms.

Keywords: font recommendation system; content emotion analysis; emotion calculation models; usability evaluation; emotion-based font recommendation

Citation: Lim, S.-B.; Ji, Y.-S.; Ahn, B.; Park, J.H.; Song, Y. Implementing and Evaluating a Font Recommendation System through Emotion-Based Content-Font Mapping. *Appl. Sci.* **2024**, *14*, 1123. https://doi.org/10.3390/app14031123

Academic Editors: Xiangjie Kong, Wei Wang and Han Liu

Received: 7 December 2023
Revised: 22 January 2024
Accepted: 24 January 2024
Published: 29 January 2024

1. Introduction

In recent years, there has been a significant and rapid increase in the volume of digital content, incorporating a diverse array of fonts. The selection of a font that complements the content is crucial for design, readability, and effective communication [1–3]. However, many users face challenges in the font selection process, often resulting in the use of inappropriate fonts.

Recent research has explored the use of deep learning and machine learning algorithms to recommend and predict optimal fonts for designers [4,5]. Nevertheless, studies focusing on technology that suggest fonts based on the emotions and atmosphere conveyed by the fonts are relatively limited. Fonts have the ability to evoke various moods and emotions through their shapes and forms, allowing people to perceive emotions from them.

To tackle this issue, our study is dedicated to designing and implementing a font recommendation system that suggests fonts based on the emotional context of input content. In our endeavor to recommend fonts that align with the emotional tone of the content, we conducted an analysis of font emotions using emotion keywords.

Recommending fonts based on the abstract concept of emotions is not straightforward. However, implementing a system that reflects emotions or feelings commonly perceived by many users is achievable. To do this, defining and mapping emotions felt within content and fonts is crucial.

In our previous research [6], the initial step involved measuring the typical impressions and emotions conveyed by Korean fonts. To achieve this, users were prompted to

directly select the emotions perceived from the fonts. In this paper, we aim to recommend fonts based on the impression data collected from previous research on fonts.

Ultimately, we successfully implemented a font recommendation system that suggests fonts aligned with the emotional context of the content. Given the distinct classification criteria of content emotion and font keywords, we needed a mapping model that could enable comparisons between different emotional classification standards. To achieve this, we designed two mapping methods: one that converts content and font emotions into PAD (Pleasure, Arousal, Dominance) values for comparison, and the other that relies on correlation analysis between content emotion classification standards and font keywords. We compared the effectiveness of these two mapping methods, presenting font recommendation lists for content using each calculation model. The results allowed us to evaluate and choose the model deemed most suitable for the task, confirming that the correlation analysis-based mapping method was more effective. This paper makes the following contributions:

- Design and Implementation of a Font Recommendation System: The research focused on designing and implementing a font recommendation system based on emotional context. This system enables designers to quickly and accurately receive font suggestions that align with the emotional tone of the input content.
- Development of Mapping Models for Font and Content Emotions: The study introduced two mapping models to bridge the gap between different emotional classification standards for content and font keywords. These models allow for comparisons between various emotional classification standards, laying the foundation for tailored recommendations.
- Implementation of a Font Recommendation Interface: Using a correlation analysis-based mapping method, the research implemented a font recommendation interface. This interface actualizes font suggestions based on content emotion to enhance the user experience. Usability evaluations were conducted to validate the effectiveness of the system.

Our study proposes a font recommendation system based on content emotion, enabling users with limited design experience to quickly and accurately receive font suggestions tailored to the context. The approach we have presented can find applications in various digital content fields, including video production, electronic publishing, and social networks. Figure 1 represents the entire process of our font recommendation system.

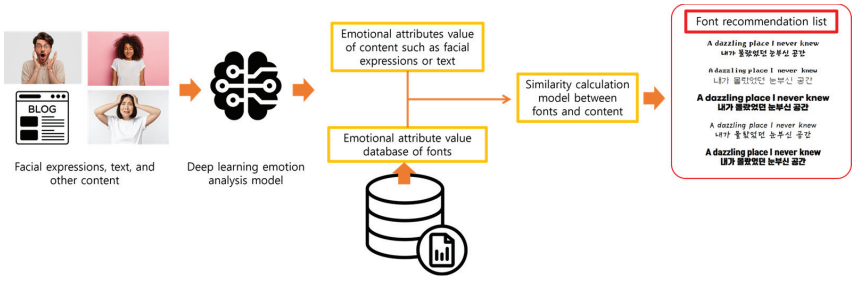


Figure 1. The entire process of the font recommendation system.

The structure of this paper is as follows: In Section 2, various studies related to font recommendation systems and emotion analysis are discussed. Section 3 presents the research findings regarding emotional keywords associated with fonts, including the selection of keywords to express font impressions and the calculation of keyword attribute values for each font. This research was conducted in-depth in previous studies. Section 4 explains the font recommendation system based on content emotion analysis, including the com-

parison and selection of calculation models. Section 5 covers the usability evaluation of the implemented system. Finally, in Section 6, the conclusion is provided.

2. Related Works

The exploration of mapping fonts to context and emotions has been a recurring area of investigation, underscoring the significance of font selection in various aspects of communication and user perception. In one study [7], researchers delved into the connections between the visual attributes of fonts and the linguistic context of the text. This research is expected to have a substantial impact on our understanding of how text and fonts interact. Additionally, current research [8] is concentrated on unraveling the emotional nuances conveyed by Chinese character fonts. This research promises to bring new insights into how fonts can evoke emotions, further enhancing our comprehension of the link between font choice and emotional expression. As highlighted by these studies, the selection of an appropriate font holds great significance for communication and the way users perceive information. Previous research, such as the exploration of font importance in email communication [9] and the emphasis on the scientific principles of font selection in academic writing [10], has demonstrated that specific fonts can influence the overall aesthetic appeal of text and the message's impact. Moreover, these studies have established methods for choosing fonts that improve text quality and readability.

Furthermore, research focusing on the psychological aspects of fonts and readability [11] has revealed that design elements like font style, thickness, spacing, and color can shape a reader's aesthetic sensibility and comprehension. Certain fonts can either emphasize or tone down the emotional undertones of the text, contributing to a more comprehensive understanding of the message. In this context, the importance of font selection in various digital media, such as blogs, cannot be overstated. It is a critical factor in effective communication and message reception in today's digital age.

In line with the importance of font selection, recent research has been actively utilizing state-of-the-art deep learning algorithms to recommend fonts [4,5]. These studies underscore the significance of context-specific font selection and employ deep learning and machine learning algorithms to assist font selectors in making more appropriate font choices.

To recommend fonts based on emotions, previous studies have measured and suggested the emotional attributes of fonts [12,13]. These studies conducted emotion analysis on both English and Korean fonts. Subsequently, after exploring fundamental attributes in the classification system of Korean fonts, emotional attribute values were acquired, leading to the implementation of an emotion-based recommendation system. Inspired by this research, our goal was to recommend fonts based on emotions perceived within the content.

Well-known commercial font recommendation systems include Adobe Fonts (Adobe Creative Cloud) [14], Google Fonts [15], and Canva (Font Space) [16]. These platforms facilitate font searching and assist in making appropriate font selections through features like simple filters. Adobe Fonts, integrated with some of Adobe's design tools, offers a vast array of fonts. Designers utilize Adobe Fonts to find fonts suitable for their projects, and the service provides recommendation features to assist users in selecting fitting fonts. Google Fonts provides a variety of free web fonts, allowing users to review detailed information about each font and search for fonts that suit their projects. However, users need to manually search using filters, and the service focuses more on selecting fonts based on their appearance than the mood of the context in which they will be used. Canva, a tool for video creation, social media posters, web design, and more, includes built-in font searching and recommendation features. Canva enables users to search for fonts based on style, using keywords like cool, cute, and fancy. However, some features are paid, and users need to manually search for fonts that match the desired mood. Our research aims to simplify this process by automatically mapping the emotions of content and fonts, eliminating the need for filters.

Existing font recommendation studies exist in various forms; however, they have certain limitations. Firstly, research related to the emotions conveyed by fonts has predominantly focused on measuring emotional values unique to fonts. This study takes a step further by implementing a system capable of matching fonts to content. Secondly, font recommendation studies often neglect the emotional aspect and predominantly center around image-based recommendations based on simple forms and shapes. Lastly, there is a scarcity of research that implements actual font recommendation systems and applies them in practical settings. In addressing these gaps, our research distinguishes itself.

3. Emotion Keywords Encapsulated in Fonts

To define the emotions associated with the images conveyed by fonts, we conducted crowdsourcing to classify the relevance of fonts and keywords among various individuals [8]. This method was employed to confirm the general atmosphere or emotions expressed by fonts. The emotions defined for fonts were later used in emotion keyword-based font recommendations after quantification.

First, the selection of emotion keywords was based on O'Donovan's research, which used various vocabulary to express the impressions of Korean fonts [12]. O'Donovan's study defined the impressions of fonts using 37 keywords for English fonts, as shown in Table 1.

Table 1. The impressions of fonts are defined in Donovan's study.

37 Expressions of Font Impressions						
Angular	Artistic	Attention-Grabbing	Attractive	Bad	Boring	Clam
Capitals	Charming	Clumsy	Complex	Cursive	Delicate	Disorderly
Display	Dramatic	Formal	Fresh	Friendly	Gentle	Graceful
Happy	Italic	Legible	Modern	Monospace	Playful	Pretentious
Serif	Sharp	Soft	Strong	Technical	Thin	Warm
Wide						

However, these keywords were originally defined for English fonts and merely represent form-based impressions of fonts, rather than human emotions. Therefore, some of the keywords that were difficult to apply to Korean fonts or those that only expressed the appearance of fonts were removed. Additionally, we included three distinctive keywords for Korean fonts, namely 'determined', 'harmonious', and 'stiff', resulting in an extended list of 41 keywords [17].

This process was carried out through a questionnaire-based survey in two phases. In the first phase, respondents were asked to select all the keywords that suited fonts from the 41 available. As a result, 29 keywords were selected. In the second phase, respondents were asked to determine whether each of the 29 keywords was closer to shape representation or emotional expression. Eventually, 19 emotion keywords, including angular, attention-grabbing, boring, calm, delicate, disorderly, dramatic, formal, friendly, gentle, graceful, modern, playful, strong, technical, warm, determined, harmonious, and stiff, were identified as the emotions that can be perceived in fonts [6].

To define the emotions associated with fonts, we had 61 users assess the impressions of each font based on the 19 emotion keywords. We measured the degree of each font's association with the emotions in a pool of around 200 Korean fonts. For example, the survey results for the DOS Gothic font are shown in Table 2. The impression of 'angular' for the DOS Gothic font was perceived as strong by 20 respondents, moderate by 32 respondents, and weak by 9 respondents.

Using the data collected in this manner, we determined the emotional attributes of fonts based on respondent numbers and weighed them accordingly. This process is illustrated in Equation (1). In this equation, F represents the font, k denotes the keyword, and F_k represents the keyword value of the font. f_{kh}, f_{km} and f_{kl} represent the number of respondents who rated the keyword as high, medium, and low for each font. n represents the total number of respondents [6].

$$F_k = \frac{(f_{kh} \times 1) + (f_{km} \times 0.5) + (f_{kl} \times 1)}{n} \quad (1)$$

Table 2. Results of the emotional keyword survey for DOS Gothic font, by the number of respondents.

DOSGothic [(ex) 도스고딕**]**			
<Emotional keywords>	High	Medium	Low
angular	20	32	9
attention-grabbing	13	17	31
boring	20	13	28
calm	4	24	33
delicate	1	19	41
disorderly	22	23	16
dramatic	12	14	35
formal	2	6	53
friendly	11	17	33
gentle	3	20	38
graceful	0	5	56
modern	1	22	38
playful	10	20	31
strong	1	17	43
technical	3	10	48
warm	3	15	43
determined	3	12	46
harmonious	0	11	38
stiff	29	20	12

4. Font Recommendation System through Content Emotion Analysis

The emotional values of fonts measured in Section 3 are ultimately utilized to implement a system that recommends fonts suitable for the extracted emotions of content based on user expressions and text content input. However, the emotional criteria for fonts and content differ. Therefore, two models for calculating the similarity between content emotions with different classification criteria and font keywords are designed. A comparison of the models is conducted to select the computational model, which, once chosen, can be applied to the implementation of the font recommendation system.

The "PAD model" [18] is a framework often used to describe emotions and affective states. It breaks down emotions into three fundamental dimensions:

- Pleasure: This dimension represents how pleasant or unpleasant an emotion is. Positive values indicate pleasurable emotions, while negative values indicate unpleasant ones.
- Arousal: Arousal measures the intensity or excitement level of an emotion. High values indicate highly aroused emotions, while low values represent calm or low arousal emotions.
- Dominance: This dimension reflects the sense of control or dominance associated with an emotion. Positive values represent emotions where the person feels in control or dominant, while negative values signify emotions where the person feels submissive or not in control.

The PAD model is commonly used in psychology and emotion recognition research. It is also applied in various fields, such as product design, advertising, and user experience design, to better understand and design for emotions and affective responses. By using this model, emotions can be described and analyzed in a more precise and structured manner, aiding in the improvement of products, services, and user interactions.

In this study, two calculation models were designed utilizing the emotional values derived from the PAD model, as described above. In Section 4.1, the model is explained, which recommends fonts based on distance calculation after converting the emotions of content and fonts into PAD values. Additionally, Section 4.2 elucidates the calculation model that determines similarity by calculating the correlation between the emotional classification criteria of content and font keywords using the Pearson correlation coefficient. Finally, in Section 4.3, the font recommendation results obtained through each calculation model are evaluated, and the final calculation model to be applied to the content-font mapping model is selected.

Additionally, in the experiments in Sections 4.1 and 4.2, using the maximum number of fonts could be effective. However, since the task involves classifying each font into 19 keywords, the workload increases with the growing number of fonts. This could potentially induce fatigue in evaluators, leading to reduced reliability in the results of the later stages of the evaluation. Therefore, in this study, we decided to select a representative minimum of fonts based on font classification for application in experiments. We collected over 200 fonts that are highly preferred by users and are available for free. After classifying them into general design attributes of Korean fonts, we categorized them into basic styles and modified styles within the classification system [19]. The classification was based on general design attributes of Korean fonts, using the criteria of serif characters for Myeongjo-style, sans-serif characters for Gothic-style, and others. The selected fonts included four basic Myeongjo-style, four basic Gothic-style, six modified Myeongjo-style, five modified Gothic-style, and seven miscellaneous fonts. To ensure diversity in font thickness, we structured the selected fonts within the same category to include both bold and thin characters.

4.1. Font Recommendation Model Utilizing PAD Value Distance Calculation

When mapping the 19-font keywords derived from Section 3 into a single PAD space, it becomes possible to calculate the distances between the PAD values of the content. To achieve this, the PAD values are used to multiply each keyword attribute value for each font and then calculate the average. However, this process led to an issue where the font's PAD values were distributed near the mid-value of PAD, which is 0.5, due to repetitive decimal multiplications. To address this concern, a new formula was introduced using the Sigmoid function. Formula (2) calculates the F_p, applying the Sigmoid function to the total sum of the product of the font's keyword values K_p. Here, F_k represents the keyword values of fonts, and the value -4.75 serves as the multiplication coefficient for the Sigmoid function.

This coefficient plays a role in determining the overall smoothness, or saturation, of the curve. The inflection point for the newly mapped font values in the PAD space is determined to be 4.75, considering the mid-value of the existing attribute values (0.5), the mid-value of the experiment (0.5), and the number of keywords, which is 19. Additionally, the multiplication coefficient for the Sigmoid function is set at 0.5, which plays a crucial role in determining the overall smoothness or saturation of the curve. Increasing this coefficient will distinctly differentiate the data clustered around the set mid-value, while more extreme values will result in less noticeable differences.

$$F_p = \frac{1}{1 + e^{-0.5 \times \{(\sum_1^k F_k \times K_p) - 4.75\}}} \tag{2}$$

4.2. Font Recommendation Model Utilizing the Pearson Correlation Coefficient

The Pearson correlation coefficient is a statistical measure used to assess the strength and direction of a linear relationship between two variables. It ranges between −1 and 1, and its interpretation is as follows:

- Close to 1: Indicates a strong positive linear correlation between the two variables. When one variable increases, the other tends to increase as well.
- Close to 0: Suggests a weak or no linear correlation between the two variables. The relationship between them may be weak or non-existent.
- Close to −1: Indicates a strong negative linear correlation between the two variables. When one variable increases, the other tends to decrease.

By utilizing the Pearson correlation coefficient, it is possible to calculate the correlation coefficient between the emotions of the content and the emotions of the font. Table 3 below represents the correlation coefficients between the emotions of the content and the font keywords.

Table 3. The correlation coefficient between content and font keyword emotions (Partial Data Included).

	Emotion of Content				
Font keywords (19)	angry	contempt	…	joy	trust
Angular	−0.7627	−0.909		0.388	0.914
Formal	−0.872	−0.972		0.557	0.975
Disorderly	0.93	0.995		−0.665	−0.996
Technical	0.132	−0.14		−0.14	0.152
Gentle	−0.984	−0.996		0.797	0.994
Modern	0.93	0.995		−0.665	−0.996
Harmonious	−0.95	−0.999		0.706	0.999
Stiff	0.837	0.658		−0.994	−0.649
attention-grabbing	−0.838	−0.954		0.501	0.958
…					
Determined	−0.94	−0.997		0.684	0.998
Playful	−0.645	−0.415		0.921	0.404

The emotion extracted from the content is transformed based on the font keyword using Pearson correlation coefficients as weights. Equation (3) represents the transformation formula, where the content keyword attribute value C is the sum of the product of the content's emotion E and the Pearson correlation coefficient $Corr$, and n represents the number of categories for content emotion classification. The values derived from this calculation formula can be used to determine the degree of similarity between the font's emotional values using a cosine similarity calculation method.

$$C = \sum_{1}^{n} E \times Corr \tag{3}$$

4.3. Comparison and Selection of Computational Models

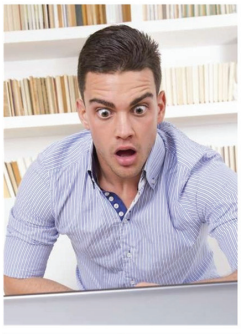
To evaluate the two computational models presented in this paper, participants were provided with ranked lists of recommended fonts based on specific expressions and text using each calculation method, as conducted in previous studies [6]. A total of 34 participants from diverse age groups, ranging from their twenties to their fifties, were involved in the evaluation. As a result, the model employing the Pearson correlation coefficient was assessed as the one that effectively recommends appropriate fonts based on the emotional content. Consequently, the font recommendation system proposed in this study is implemented using a computational model that utilizes the Pearson correlation coefficient.

4.4. Implementation of a Font Recommendation System

The font recommendation system, based on the emotions of the content, recommends fonts depending on the emotions extracted through deep learning emotion analysis APIs when content like images and text is input.

In the case of an image input, the DeepFace API [20] was utilized to extract emotions based on facial expressions. The API employs a model to determine emotion values. The DeepFace API is an application programming interface that provides access to the DeepFace facial recognition technology. It is commonly employed for face recognition, face detection, and facial attribute analysis in machine learning and computer vision applications. Developed by Facebook AI Research, DeepFace is renowned for its deep neural network models that can identify and analyze faces in images and videos.

Figure 2 below displays the list of recommended fonts and emotion numerical values based on the input image. The DeepFace API yields seven emotion values as results: 'angry', 'disgust', 'fear', 'happy', 'sad', 'surprise', and 'neutral.' In this instance, the 'surprise' emotion value is 1, while the values for the other emotions tend to converge toward 0. Fonts with irregular handwritten designs, rather than those from the Ming and Gothic font families, are recommended for expressions manifesting such emotions. Basic Ming fonts are ranked the lowest in the recommendations.



Top 5 recommended fonts from the list

What a surprise!
놀랄만한 소식이다!

What a surprise!
놀랄만한 소식이다!

What a surprise!
놀랄만한 소식이다!

What a surprise!
놀랄만한 소식이다!

What a surprise!
놀랄만한 소식이다!

Bottom 5 recommended fonts from the list

What a surprise!
놀랄만한 소식이다!

What a surprise!
놀랄만한 소식이다!

What a surprise!
놀랄만한 소식이다!

What a surprise!
놀랄만한 소식이다!

What a surprise!
놀랄만한 소식이다!

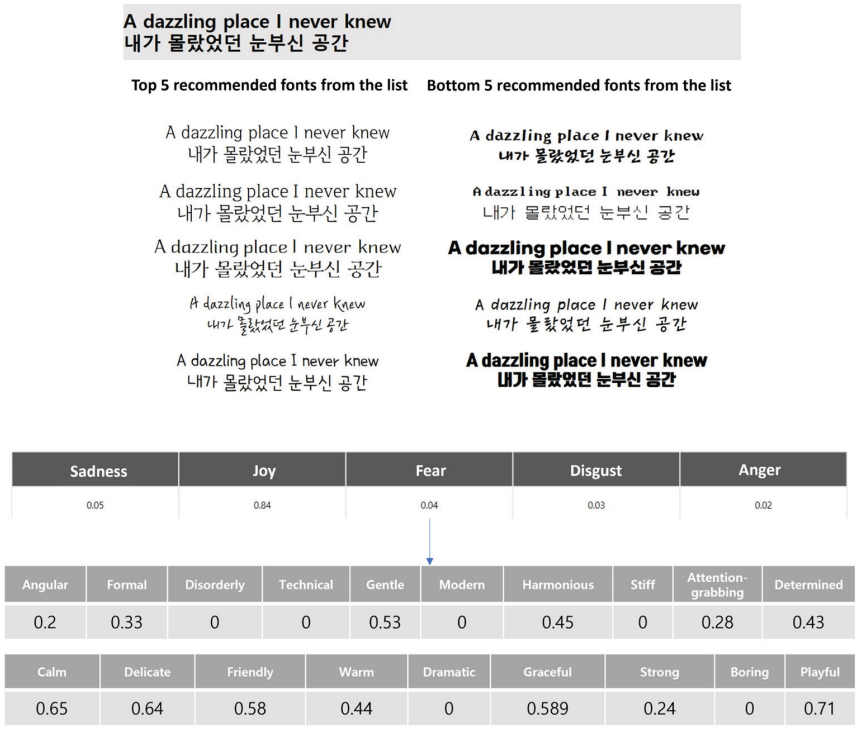
anger	disgus	fear	joy	sad	surprise	neutral
4.89X10^(-12)	7.68X10^(-19)	4.678X10^(-8)	8.342X10^(-15)	1.297X10^(-16)	1.0	0.149X10^(-23)

Angular	Formal	Disorderly	Technical	Gentle	Modern	Harmonious	Stiff	Attention-grabbing	Determined
0	0	0.17	0	0.26	0.19	0	0	0	0

Calm	Delicate	Friendly	Warm	Dramatic	Graceful	Strong	Boring	Playful
0.3	0.28	0.13	0	0.34	0.15	0	0	0.88

Figure 2. Displays the list of recommended fonts and corresponding emotion numerical values based on the input image.

The font recommendation system can also be employed for text input, and for sentence emotion analysis, the IBM Watson API [21] is utilized. This model analyzes the emotion of sentences into five categories: 'sadness', 'joy', 'fear', 'disgust', and 'anger.' An example of this is illustrated in Figure 3, where the input sentence results in the emotion 'JOY', and it displays the associated fonts' emotions.



Sadness	Joy	Fear	Disgust	Anger
0.05	0.84	0.04	0.03	0.02

Angular	Formal	Disorderly	Technical	Gentle	Modern	Harmonious	Stiff	Attention-grabbing	Determined
0.2	0.33	0	0	0.53	0	0.45	0	0.28	0.43

Calm	Delicate	Friendly	Warm	Dramatic	Graceful	Strong	Boring	Playful
0.65	0.64	0.58	0.44	0	0.589	0.24	0	0.71

Figure 3. Illustration of an example depicting the emotion 'JOY' generated from the input sentence, showcasing the corresponding emotions of associated fonts.

5. Usability Evaluation of the Font Auto-Recommendation System

5.1. Implementation Results

Based on the research findings, an interface has been developed to make it accessible to real users. This interface is built on the Django framework and is designed to have two main sections: one for users to input content and the other for displaying the recommended fonts. In the current system, fonts are recommended based on facial expressions and sentence emotion analysis. Users can choose to input content by selecting a facial expression or a sentence from the available options. Figure 4 represents the implementation of a font recommendation system on the web. At the top, there is an instructional message displayed, and the two buttons below allow the selection between entering an expression or a sentence. This example involves inputting an image with an expression and matching the emotions in the image with the top five recommended fonts. Clicking on the button to the right of the font enables redirection to a webpage featuring that font.

The input photos and sentences undergo emotion extraction through deep-learning emotion analysis models. Subsequently, a mapping model applying Pearson correlation coefficients transforms them into font keyword values. As a result, the transformed values are based on the same criteria as font keywords, enabling similarity evaluation. The transformed values are compared to the font keyword values using cosine similarity, and fonts with the highest similarity are displayed at the top.

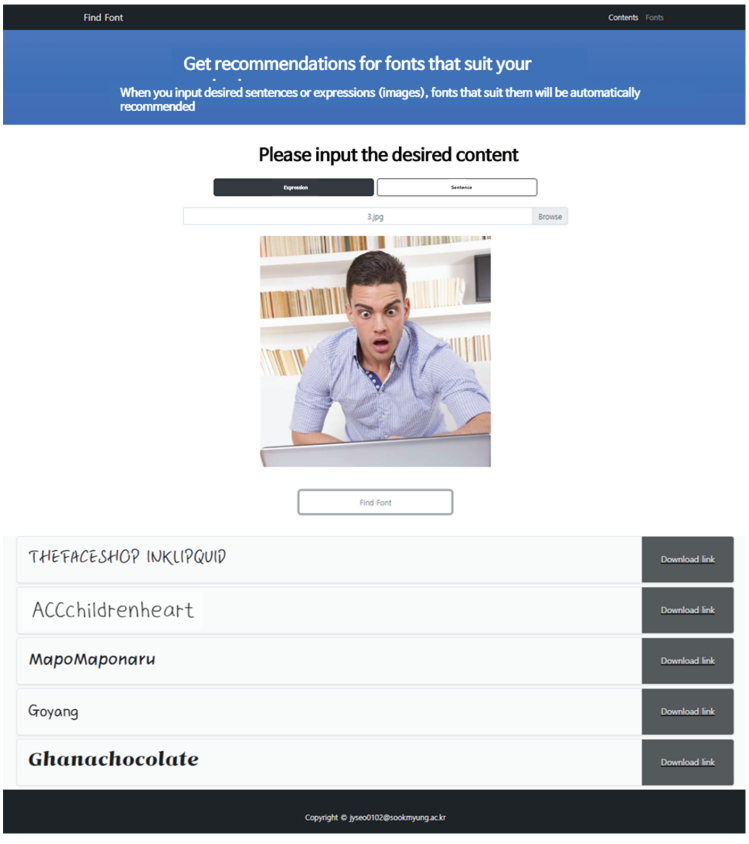


Figure 4. Web-based font recommendation system with expression and sentence input options, showcasing matching emotions in images to the top five recommended fonts, each linked to its webpage.

5.2. Application of the Implementation Results

In this paper, beyond a simple web-based recommendation system, we developed an additional service utilizing a font recommendation model to suggest fonts based on blog posts. The blog post font recommendation system analyzes the text and images within user-created blog posts to recommend suitable fonts for the content. Sentiment analysis of blog posts is performed in two different cases. First, an analysis of whether the images posted on the blog contain facial expressions is conducted. Images that are categorized by facial expressions are utilized to extract emotion keywords using the Deepface API.

In cases where facial expressions are not present, general images undergo image analysis using the Google Cloud Vision API [22]. This analysis recognizes objects, text, colors, object locations, and more within the images, with the results of the image analysis being derived as descriptive text.

Subsequently, the derived text undergoes sentiment analysis based on the IBM Watson Natural Language Understanding service. This service analyzes the text to identify the emotional tone of the given text, allowing the determination of the text's positive or negative emotions. As emotional analysis for Korean text is not supported, translation to English is performed before sentiment analysis.

The results of facial expression and image analysis, as well as text sentiment analysis, are integrated to determine the overall sentiment of the blog posts. Through such analysis, a more accurate understanding of the emotional aspects of the content within the posts

is achieved, providing richer emotional information through the combination of images and text.

An example of font recommendation results is as follows: Figure 5 displays a blog post containing both text and images. In this example blog post, there are no images depicting human emotions. In such cases, we generate descriptions for the pictures and analyze the emotions within these descriptions to recommend fonts. Additionally, as blog posts primarily consist of textual content, we analyze the content of the main body to suggest fonts.

Figure 5. Examples of recommended fonts derived from the analysis of blog content. (**a**) In the case of (**a**), it is a blog post containing content related to sadness, and below the red arrow are the font recommendations based on blog post content and image analysis, respectively. The font recommendation results can be applied to the blog's title, and user selection is possible. (**b**) Similarly, in the case of (**b**), it represents a blog post with challenging and angry content, along with the font recommendation results.

As a result, the system provides font recommendations aligned with the emotional analysis of both images and text, empowering users to make their preferred selections. In Figure 5a, fonts corresponding to the emotion "sadness" were recommended based on the analysis. Similarly, Figure 5b demonstrates the outcomes of blog post analysis conducted in a parallel fashion, resulting in font suggestions. Here, fonts associated with the emotion 'anger' were recommended, drawing from a comprehensive analysis of both image and text content.

5.3. Usability Evaluation

To evaluate the usability of the content emotion-based font recommendation system, three usability metrics were employed: the appropriateness of the recommended fonts concerning how well they matched the content, satisfaction with the recommendations provided by the system, and the efficiency of the content-based font recommendation system. The evaluation categorized the recommended fonts into four groups based on their ranking. Group 1 included fonts ranked from 1st to 6th; Group 2 included fonts ranked from 7th to 12th; Group 3 included fonts ranked from 13th to 19th; and Group 4 included fonts ranked from 20th to 26th.

Participants were presented with various facial expressions and text content to determine the appropriateness of the system's results. They were asked to select the most appropriate and least appropriate recommendation groups for each piece of content, providing an evaluation of the appropriateness of the system's results. Additionally, participants were asked to rate their satisfaction with the results for each recommendation group on a 7-point Likert scale. The Likert scale is commonly employed in usability evaluations for its simplicity, clarity, and effectiveness in quantifying and comparing user experiences across various aspects such as interface design, product features, or service quality. Its concise numeric measurements provide an integrated assessment, facilitating statistical analysis and trend observation. The scale's universality and ease of understanding make it applicable to diverse user groups, enhancing user feedback accessibility. Moreover, Likert scale scores exhibit consistency, allowing for reliable usability tracking and comparisons over time or among different user cohorts. Overall, the Likert scale serves as a versatile and objective tool in usability evaluations, promoting efficiency in data collection and analysis.

Furthermore, users were invited to directly experience the font recommendation system and compare it with the existing font recommendation system developed in previous research [13] to assess the efficiency of the system when receiving font recommendations. The evaluation was also conducted using a 7-point Likert scale. Table 4 provides details about the evaluation criteria and measurement methods.

Table 4. Measurement methods and evaluation scales for usability evaluation.

Evaluation Criteria	Measurement Methods
Appropriateness	• Verify the groups in which the content and recommended fonts have been classified according to their rankings. • Select the group of fonts that best matches the content. • Select the group of fonts that least suits the content.
Satisfaction	• Satisfaction evaluation of recommended fonts by font group (0–6 points).
Efficiency	• Selecting fonts suitable for desired content through the existing font recommendation system [13]. • Selecting fonts suitable for the desired content through our system. • Evaluating the efficiency of this system compared to the existing system (0–6 points).

The reason for utilizing the three metrics—Appropriateness; Satisfaction; and Efficiency—in the usability evaluation is as follows: First, each metric comprehensively assesses different aspects of the system. "Appropriateness" gauges the degree to which the system aligns with its purpose; "Satisfaction" measures the user's contentment; and "Efficiency" evaluates how efficiently tasks are performed. These metrics align with the core principles of user-centered design. "Appropriateness" ensures that the system meets user requirements and accomplishes user-centered goals. "Satisfaction" indicates how positively users experience the system, and "Efficiency" emphasizes reducing user effort and promoting efficient task performance, aligning with user-centered system design principles. These metrics are widely used in usability evaluations, addressing key aspects of the user experience. Appropriateness, Satisfaction, and Efficiency collectively reflect user needs, impacting the success of the system and user satisfaction. Moreover, these evaluation metrics are applicable to various types of systems and user groups and are widely recognized and utilized in usability evaluations [23–28].

A total of 15 participants took part in the evaluation. In the evaluation, the provided font groups were referred to as Group 1, which includes the top 6 recommended fonts;

Group 2, which includes fonts ranked 7th to 12th; Group 3, which includes fonts ranked 13th to 19th; and Group 4, which includes fonts ranked 20th to 26th. Based on the assessment of 12 different contents presented to users, Group 1, which contains the fonts recommended as the top-ranked ones, was consistently rated as the most suitable font group for each content. On average, 10.67 participants rated Group 1 as the most appropriate. In contrast, Group 4, which contains fonts with the lowest rankings, received an average rating of 1.68, with participants finding it suitable. Group 3 received the lowest average rating, with 0.94 participants finding it appropriate. However, statistically, there was no significant difference in ratings between Groups 2, 3, and 4. The graph depicting the appropriateness of the evaluation results is shown in Figure 6.

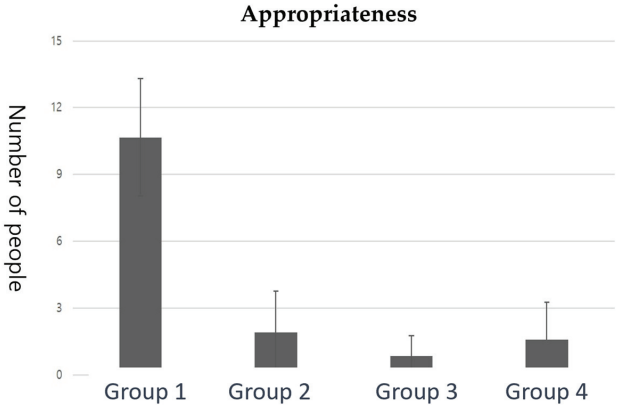


Figure 6. The average count of respondents who considered the results appropriate was classified by font recommendation rankings among grouped categories.

Conversely, in the evaluation where participants were asked to select the least appropriate font group, Group 4 received the highest number of selections, with approximately 7.9 participants choosing it, while Group 1 was chosen by the fewest participants, with only 0.75 individuals selecting it. However, for Groups 2, 3, and 4, there was no statistically significant difference in the choices made by participants. This can be seen in the following Figure 7.

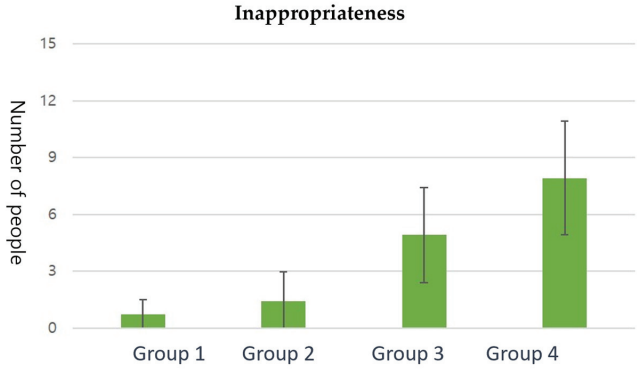


Figure 7. The average count of respondents who deemed the results inappropriate, categorized by font recommendation rankings among grouped categories.

When each font group was individually presented to users and font recommendation satisfaction was evaluated on a scale from 0 to 6 using the Likert score, Group 1 received the highest satisfaction with a score of 4.68. It was observed that as we move from Group 1

to Group 4, the average satisfaction gradually decreases. The group-specific results for satisfaction scores are illustrated in Figure 8.

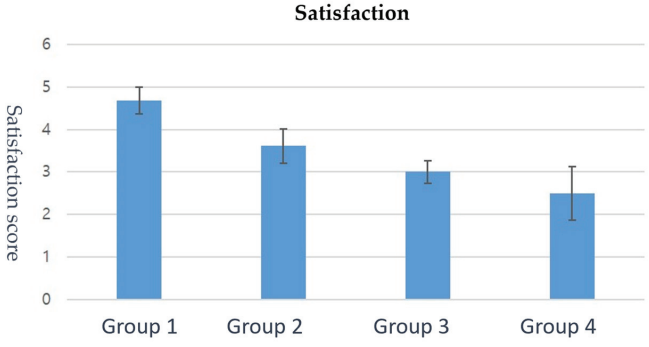


Figure 8. Group-specific results for satisfaction scores.

Finally, we aimed to assess the efficiency of our system through a comparison with the existing font search system, as illustrated in Figure 9 [13]. As a result, users highly evaluated the system that automatically recommends fonts based on content input, with an average efficiency score of 5.27. This score was measured after comparing it with the sentiment-based font recommendation system developed in previous research. The methodology involved initially having 15 participants use the existing system, followed by using our system. Participants were asked to rate the efficiency of the new sentiment-based font recommendation system compared to the existing one. They were prompted with the question, "Please select a score from 1 (very inefficient) to 7 (very efficient) indicating the efficiency of the new sentiment-based font recommendation system compared to the existing one." All 15 participants gave a score of 4 or higher. Therefore, the recommendation results of this system were considered both valid and efficient compared to the existing system.

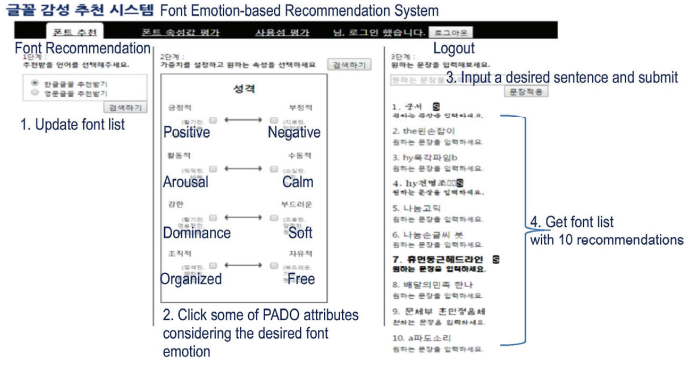


Figure 9. The existing font recommendation system was described in previous research. (Reprinted/adapted with permission from Ref. [13]. Copyright 2018, copyright Soon-bum Lim).

6. Conclusions

In conclusion, this research aimed to implement a system that recommends fonts based on the emotions of the content. To achieve this, the emotions associated with fonts were analyzed using keywords. We selected 19 keywords that are applicable to Korean fonts and easily understandable by users, based on keywords used in previous studies on English fonts. We conducted experiments to examine the relationship between keywords and fonts, selecting 26 fonts representative of various font classifications.

To recommend fonts based on content emotions, we designed new calculation models to allow for a consistent comparison between content emotions and font emotions. While we initially attempted to apply the PAD emotional model, convergence issues arose during the process of converting keyword attributes to PAD values. As a result, we designed two new calculation models: one with a new formula for PAD value conversion and another that utilized an analysis of the correlation between emotional classification criteria. The evaluation results favored the model that applied the correlation-based approach. Therefore, we decided to use this mapping method for font recommendations.

We implemented the recommendation interface using this model and conducted usability evaluations. The findings confirmed the appropriateness of the fonts recommended by the system and the efficiency of our implementation compared to the existing system.

In summary, this research successfully analyzed font emotions through keyword analysis and recommended fonts suitable for content with varying classification criteria. Furthermore, we expanded this recommendation system to platforms such as blogs, which encompass a diverse array of content. As a result, users can receive font recommendations based on the mood and emotions of their creative works during their creative activities, allowing for convenient and suitable font usage. Ultimately, this enables users to easily attain the positive effects achievable through font utilization.

The flexibility of our matching model allows for the recommendation of fonts for a wide range of diversified content. By enhancing font selection, even users with limited design expertise can improve the quality of their creations while reducing the cost associated with font selection. We believe that the concept introduced in this study can be extended to various types of content beyond facial expressions and sentences.

In the future, we plan to conduct an in-depth system evaluation targeting font designers and other experts, separate from the usability assessment. Through this, we aim to measure the accuracy of our font recommendation algorithm and system, assess user satisfaction, and identify areas for further enhancement. Subsequently, we will perform actual usability testing by providing font recommendations to users free of charge, aiming to refine the system for improved temporal efficiency in font selection.

Author Contributions: Conceptualization, S.-B.L. and Y.-S.J.; methodology, S.-B.L., B.A. and S.-B.L.; validation, S.-B.L. and Y.-S.J.; formal analysis, B.A. and J.H.P.; writing—original draft preparation, Y.S.; writing—review and editing, Y.S.; project administration, S.-B.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (2021R1I1A4A01059550). This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (grant number: RS-2022-00165818).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. However, the data cannot be publicly disclosed due to the inclusion of paid fonts and other proprietary elements, which restrict public access. For further inquiries, please contact the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Haenschen, K.; Tamul, D.J.; Collier, J.R. Font matters: Understanding typeface selection by political campaigns. *Int. J. Commun.* **2021**, *15*, 2894–2914.
2. Murchie, K.J.; Diomede, D. Fundamentals of graphic design—Essential tools for effective visual science communication. *Facets* **2020**, *5*, 409–422. [CrossRef]
3. Ueki, R.; Yokoyama, K.; Nakamura, S. Does the Type of Font Face Induce the Selection? In *International Conference on Human-Computer Interaction*; Springer Nature: Cham, Switzerland, 2023; pp. 497–510.

4. Zhao, N.; Cao, Y.; Lau, R.W. Modeling fonts in context: Font prediction on web designs. *Comput. Graph. Forum* **2018**, *37*, 385–395. [CrossRef]
5. Tsuji, K.; Uchida, S.; Iwana, B.K. Using Robust Regression to Find Font Usage Trends. In *Document Analysis and Recognition–ICDAR 2021 Workshops: Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part II 16*; Springer International Publishing: Berlin/Heidelberg, Germany, 2021; pp. 126–141.
6. Ji, Y.S.; Lim, S.B. Design and Application of Mapping Model for Font Recommendation System Based on Contents Emotion Analysis. In *International Conference on Intelligent Computing*; Springer Nature: Singapore, 2023; pp. 397–408.
7. Shirani, A.; Dernoncourt, F.; Echevarria, J.; Asente, P.; Lipka, N.; Solorio, T. Let me choose: From verbal context to font selection. *arXiv* **2020**, arXiv:2005.01151.
8. Zhang, S.; Wang, P.; Hou, W. Research on Font Emotion Based on Semantic Difference Method. In *Human Centered Computing, Proceedings of the 4th International Conference, HCC 2018, Mérida, Mexico, 5–7 December 2018*; Springer International Publishing: Berlin/Heidelberg, Germany, 2018.
9. Shaikh, A.D.; Fox, D.; Chaparro, B.S. The effect of typeface on the perception of email. *Usability News* **2007**, *9*, 1–7.
10. Lonsdale, M.D.S. Typographic features of text and their contribution to the legibility of academic reading materials an empirical study. *Visible Lang.* **2016**, *50*, 79–111.
11. Ho, A.G. Typography today: Emotion recognition in typography. In Proceedings of the IASDR 2013 Conference, Tokyo, Japan, 26–30 August 2013; Japanese Society for the Science of Design (JSSD): Tokyo, Japan, 2013; Volume 1, pp. 5573–5582.
12. O'Donovan, P.; Lībeks, J.; Agarwala, A.; Hertzmann, A. Exploratory font selection using crowdsourced attributes. *ACM Trans. Graph.* **2014**, *33*, 4. [CrossRef]
13. Kim, H.Y.; Lim, S.B. Emotion-based Hangul font recommendation system using crowdsourcing. *Cogn. Syst. Res.* **2018**, *47*, 214–225. [CrossRef]
14. Adobe Fonts. Adobe Fonts Website. 2024. Available online: https://fonts.adobe.com/ (accessed on 17 January 2024).
15. Google Fonts. Google Fonts Website. 2024. Available online: https://fonts.google.com/ (accessed on 17 January 2024).
16. FontSpace. Canva Fonts on FontSpace. 2024. Available online: https://www.fontspace.com/category/canva (accessed on 17 January 2024).
17. Eunmi, S. A Study on Image Classification for Hangul Font for Emotional Expression on Digital Media. Ph.D. Thesis, Yonsei University, Seoul, Republic of Korea, 2007.
18. Mohammad, S. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Volume 1, pp. 174–184.
19. Kim, H.Y.; Lim, S.B. Standardizing design-based font classification system for Hangul font services. *Comput. Stand. Interfaces* **2018**, *55*, 47–54. [CrossRef]
20. Wang, M.; Deng, W. Deep face recognition: A survey. *Neurocomputing* **2021**, *429*, 215–244. [CrossRef]
21. Carvalho, A.; Levitt, A.; Levitt, S.; Khaddam, E.; Benamati, J. Off-the-shelf artificial intelligence technologies for sentiment and emotion analysis: A tutorial on using IBM natural language processing. *Commun. Assoc. Inf. Syst.* **2019**, *44*, 43. [CrossRef]
22. Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E.L.; Norouzi, M. Photorealistic text-to-image diffusion models with deep language understanding. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 36479–36494.
23. Hashim, N.L.; Isse, A.J. Usability evaluation metrics of tourism mobile applications. *J. Softw. Eng. Appl.* **2019**, *12*, 267–277. [CrossRef]
24. Pu, P.; Chen, L.; Hu, R. A user-centric evaluation framework for recommender systems. In Proceedings of the Fifth ACM Conference on Recommender Systems, Chicago, IL, USA, 23–27 October 2011; pp. 157–164.
25. Seffah, A.; Donyaee, M.; Kline, R.B.; Padda, H.K. Usability measurement and metrics: A consolidated model. *Softw. Qual. Qual. J.* **2006**, *14*, 159–178. [CrossRef]
26. Bangor, A.; Kortum, P.T.; Miller, J.T. An empirical evaluation of the system usability scale. *Intl. J. Hum. Comput. Interact.* **2008**, *24*, 574–594. [CrossRef]
27. Lewis, J.R.; Sauro, J. The factor structure of the system usability scale. In *Human Centered Design: First International Conference, Proceedings of the HCD 2009, Held as Part of HCI International 2009, San Diego, CA, USA, 19–24 July 2009*; Springer: Berlin/Heidelberg, Germany, 2009; Proceedings 1; pp. 94–103.
28. Nielsen, J. *Usability Engineering*; Morgan Kaufmann: Burlington, MA, USA, 1994.

applied sciences

MDPI

Article

M2ER: Multimodal Emotion Recognition Based on Multi-Party Dialogue Scenarios

Bo Zhang [1,*], Xiya Yang [1], Ge Wang [1], Ying Wang [1] and Rui Sun [2]

[1] School of Information and Communication Engineering, Communication University of China, Dingfuzhuang, Chaoyang District, Beijing 10024, China; yangxiya@cuc.edu.cn (X.Y.); wg2530280660@163.com (G.W.); yingwang@cuc.edu.cn (Y.W.)
[2] School of Computing, Newcastle University, Newcastle upon Tyne NE1 7RU, UK; r.sun5@newcastle.ac.uk
* Correspondence: zhangbo2015@cuc.edu.cn

Abstract: Researchers have recently focused on multimodal emotion recognition, but issues persist in recognizing emotions in multi-party dialogue scenarios. Most studies have only used text and audio modality, ignoring the video modality. To address this, we propose M2ER, a **m**ultimodal **e**motion **r**ecognition scheme based on **m**ulti-party dialogue scenarios. Addressing the issue of multiple faces appearing in the same frame of the video modality, M2ER introduces a method using multi-face localization for speaker recognition to eliminate the interference of non-speakers. The attention mechanism is used to fuse and classify different modalities. We conducted extensive experiments in unimodal and multimodal fusion using the multi-party dialogue dataset MELD. The results show that M2ER achieves superior emotion recognition in both text and audio modalities compared to the baseline model. The proposed method using speaker recognition in the video modality improves emotion recognition performance by 6.58% compared to the method without speaker recognition. In addition, the multimodal fusion based on the attention mechanism also outperforms the baseline fusion model.

Keywords: multimodal; emotion recognition; feature extraction; feature-level fusion; attention mechanism; speaker recognition

Citation: Zhang, B.; Yang, X.; Wang, G.; Wang, Y.; Sun, R. M2ER: Multimodal Emotion Recognition Based on Multi-Party Dialogue Scenarios. *Appl. Sci.* **2023**, *13*, 11340. https://doi.org/10.3390/app132011340

Academic Editor: Douglas O'Shaughnessy

Received: 5 September 2023
Revised: 1 October 2023
Accepted: 11 October 2023
Published: 16 October 2023

1. Introduction

Emotions are unique and important forms of human expression [1]. When conducting early research on emotions, Ekman [2] classified people's basic emotions according to their needs. In 1977, Picard proposed the concept of emotional computing [3], aiming to equip computers with the ability to recognize, understand, express, and adapt to human emotions. An important direction in emotional computing research is emotion recognition, which can create more intelligent and harmonious user entities for applications such as lie detection, audiovisual monitoring, online conferences, and human–computer interaction (HCI) [4].

Researchers often rely on unimodal emotion recognition [5]. Recently, significant progress has been made in the research of unimodal approaches for text, audio, and video. Particularly, facial emotion recognition (FER) technology has a wide range of applications, including HCI, emotional chat, psychological diagnosis, and other tasks [6]. AffectNet [7] is a widely recognized corpus for video modality emotion recognition. Currently, the top three models in terms of accuracy for seven-class emotion recognition on AffectNet are POSTER++ (67.49%) [8], Emotion-GCN (66.46%) [9], and EmoAffectNet (66.37%) [10]. Other studies related to FER are as follows: Bakariya et al. [11] created a real-time system that can recognize human faces, assess human emotions, and recommend music to users. Meena et al. [12] proposed a facial image sentiment analysis model based on a CNN. It is discovered that more convolution layers, a strong dropout, a large batch size, and many epochs can obtain better effects. Savchenko [13] studied lightweight convolutional neural networks (CNNs) for FER task learning and verified the effectiveness of CNNs for FER.

Meena et al. [14] utilized Inception-v3, along with additional deep features, to enhance image categorization performance. A CNN-based Inception-v3 architecture was used for emotion detection and classification. In a study by Saravanan [15], they found that CNNs are highly effective for image recognition tasks due to their ability to capture spatial features using numerous filters. They proposed a model consisting of six convolutional layers, two max-pooling layers, and two fully connected layers, which performed better than decision trees and feed-forward neural networks on the FER-2013 dataset. Li [16] used a CNN, which extracts geometric and appearance features, and LSTM, which captures temporal and contextual information on facial expressions. This CNN–LSTM architecture allows for a more comprehensive representation of facial expressions by combining spatial and temporal information. Ming et al. [17] presented a facial expression recognition method that included an attention mechanism based on a CNN and LSTM. This model was able to effectively extract information on important regions, better than general CNN–LSTM-based models. Sang [18] focused on reducing intra-class variation in facial expression depth features and introduced a dense convolutional network [19] for the FER task.

There has been an increase in the combination of transformers in various FER methods. Xue [20] was the first to use the vision transformer for FER and achieved state-of-the-art results. VTFF [21] excels in dealing with facial expression recognition tasks in the wild due to its feature fusion. Chen et al. [22] introduced CrossViT, which uses dual branches to combine image patches of different sizes to produce more reliable features. Heo et al. [23] examined the benefits of pooling layers in ViT, similar to their advantages in CNNs.

However, in real-world scenarios, the video modality often presents complex data formats. For example, multiple faces often appear in the same frame in multi-party dialogue scenarios, and the presence of non-speaking individuals' faces can interfere with the final emotion recognition. This is the reason why most of the existing research on multimodal emotion recognition in multi-party dialogues has not utilized the video modality. Challenges such as speaker recognition, significant intra-class facial expression variations, and subtle inter-class differences further highlight the room for improvement in emotion recognition. Thus, there is still considerable scope for further research and exploration in the field of emotion recognition.

It is hard to obtain accurate emotional information only through a single modality [24,25]. Compared with unimodal emotion recognition, multimodal emotion recognition can make up for the noise interference caused by the single modality and make full use of the complementary features between different modalities. Zadeh [26] conducted multimodal sentiment analysis on three modalities of text, audio, and video for the first time and released the first dataset containing text, audio, and video modalities—the YouTube dataset. Rosas [27] proposed a multimodal research dataset—Moud—and conducted sentiment analysis in sentences. Zadeh [28] constructed a large-scale multimodal dataset CMU-MOSEI. In recent years, based on the above datasets, researchers have carried out many classic multimodal emotion analysis methods based on text, audio, and video modalities. Dai [29] combined multimodal feature extraction and fusion into a model and optimized it at the same time, which improved the accuracy of emotion recognition in real-time performance. Ren [30] used the self-supervised training model to fuse the features of text, audio, and video modalities into non-standard classes and achieved better results than the baseline model.

The focus of multimodal emotion recognition lies in how to extract features and perform subsequent fusion. However, most of the current research on multimodal emotion recognition only focuses on the stage of feature fusion, neglecting the initial stage of unimodal emotional feature extraction. For example, in the case of the audio modality, most studies directly extract audio features using open-source toolkits such as Librosa and OpenSmile [31,32] and fuse them with features from other modalities. In the context of multi-party dialogues, many researchers have focused on studying the text and audio modalities while neglecting the video modality. Extracting comprehensive features from individual modalities is a prerequisite for multimodal emotion recognition. The more

comprehensive the extraction of emotional features from each modality, the better it can reflect the characteristics of emotion.

There are three main methods of multimodal fusion: data-level fusion, feature-level fusion, and decision-level fusion. The specific process of feature-level fusion is illustrated in Figure 1. Feature-level fusion can fully leverage the advantages of each modality, effectively integrate information from different modalities, and consider the correlation between various data in different modalities. However, if the feature-level fusion is achieved by directly concatenating the feature vectors, it will result in high-dimensional vectors, leading to problems such as the curse of dimensionality.

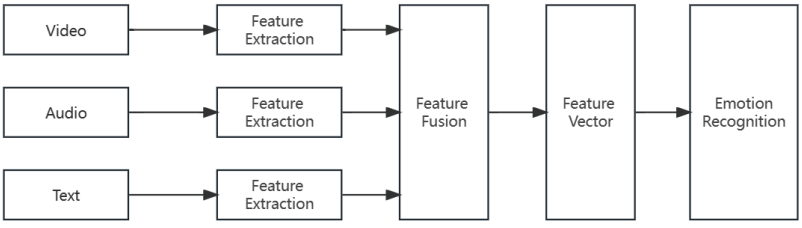


Figure 1. The specific process of feature-level fusion, which involves extracting emotional features from individual modalities, combining the obtained feature vectors in a specific way, and finally using an emotion classifier to recognize the fused features.

Recently, many research works have focused on attention-based fusion and its variants, such as self-attention, multi-head attention, and transformers [33]. The attention-based fusion integrates the advantages of early fusion and late fusion and compensates for their shortcomings [34]. The attention mechanism is a specialized structure that can be embedded in the framework of machine learning models. By employing the attention mechanism, the problem of information overload can be addressed. Furthermore, the attention mechanism can provide an effective resource allocation scheme in neural networks [35]. As the number of model parameters increases in deep neural networks, the model generally becomes more expressive and capable of storing a greater amount of information. However, the increasing number of parameters also demands significant computational resources during model training, making it challenging. By incorporating the attention mechanism into neural networks, it becomes possible to identify which data in the input sequence contributes more significantly to the task at hand [36]. Consequently, more limited attention can be allocated to the most valuable portions of information, while reducing attention or disregarding irrelevant information, thus efficiently utilizing computational resources [37]. Hu [38] proposed the Multimodal Dynamic Fusion Network (MM-DFN) to recognize emotions by fully understanding multimodal conversational context. Wang et al. [34] proposed a cross-attention asymmetric fusion module, which utilized information matrices of the acoustic and visual modality as weights to strengthen the text modality.

Based on the above situation, we propose M2ER that optimizes the key steps of multimodal emotion recognition in multi-party dialogue scenarios. We mainly focus on how to fully utilize video modalities. The contributions of M2ER are summarized as follows:

- We constructed suitable feature extraction models for text, audio, and video modalities. Addressing the challenge of multiple faces appearing in a single frame in the video modality, we propose a method using multi-face localization for speaker recognition, thus extracting features from facial expression sequences of the identified speaker.
- For the multimodal fusion model, we adopted the feature-level fusion approach utilizing a multimodal fusion model based on the attention mechanism. The extracted unimodal emotional features are combined using cross-modal attention to capture the intermodal interactions. Furthermore, the attention mechanism employed determines

the contribution of each modality to the final emotion classification, enabling the fusion with different weights.

- We conducted experiments on the Multimodal Emotion Lines Dataset (MELD) [39] using both unimodal and multimodal fusion methods and further evaluated the scalability of our models on the MEISD dataset [40]. The extensive experiments show that our unimodal feature-based emotion recognition model of M2ER outperforms the baseline models. The multimodal fusion model achieves higher recognition accuracy compared to the unimodal emotion recognition systems. Moreover, our fusion model of M2ER exhibits superior performance in multimodal emotion recognition tasks compared to directly concatenated models.

The remaining parts of the paper are structured as follows: Section 2 presents the detailed design of the proposed M2ER, including the extraction of unimodal features and the multimodal feature fusion model. In Section 3, we outline the experiments conducted on unimodal and multimodal emotion recognition separately and verify the scalability of the models. Furthermore, we discuss the advantages of our work as well as the limitations and future work in Section 4; Finally, Section 5 concludes the work of this paper.

2. Detailed Design

Figure 2 illustrates the overview framework of M2ER, which includes the extraction of emotional features from text, audio, and video modalities, as well as the multimodal fusion classification framework adopted in our work based on the attention mechanism. We will introduce the detailed scheme of the unimodal extraction model in Section 2.1 and fusion model information in Section 2.2.

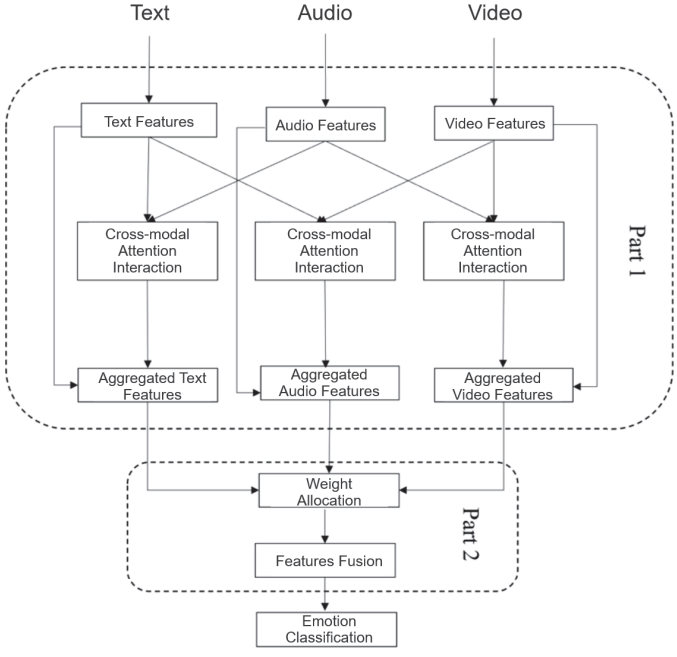


Figure 2. The framework of M2ER. Detailed information will be introduced in the following Sections 2.1 and 2.2.

2.1. The Unimodal Extraction Model of M2ER

We adopted a feature-level fusion method for multimodal emotion recognition, so we need to perform feature extraction for each modality in the first step. The detail of feature

extraction models of M2ER for the three modalities (including text, audio, and video) are described in Sections 2.1.1–2.1.3.

2.1.1. Text Modality Preprocessing and Feature Extraction

We used the Embeddings from Language Model (ELMo) [41] pre-trained model to obtain dynamic word vector features for the text modality. The core of ELMo lies in utilizing a bidirectional Long Short-Term Memory (LSTM) [42] recurrent neural network structure for feature extraction. During training, ELMo leverages the entire input text and considers both forward and backward input sequence information simultaneously to obtain more comprehensive text emotional features. We also adopted BERT [43] to extract semantic information at the sentence level. BERT can be used to extract text emotional features, where the proximity of words in the feature vector space reflects their semantic similarity [44]. The BERT model utilizes the transformer as a feature extractor. When processing a task, BERT first transforms the input text to obtain BERT input representation. Then, the transformer encoder performs computations on the input, then the computed results serve as the input for the next transformer encoder. This process is repeated, resulting in the representation of the entire text.

In conclusion, in the feature extraction part of text modality, we utilized the pre-trained model of ELMo and BERT to obtain text emotional features from the word-level and semantic-level perspectives, respectively. Finally, the extracted features from both parts were combined to obtain complete text emotional features. The process of text modality feature extraction is illustrated in Figure 3.

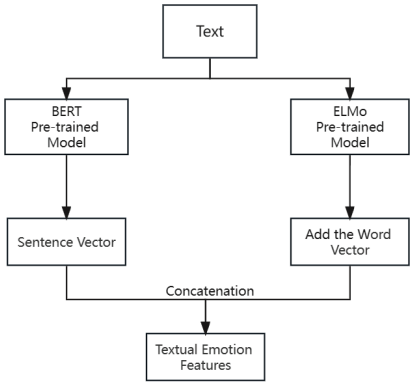


Figure 3. Text feature extraction model.

2.1.2. Audio Modality Preprocessing and Feature Extraction

The representation of audio signals is quite diverse, and the way audio signals are described greatly impacts the performance of subsequent feature extraction and emotion recognition. The purpose of the preprocessing is to transform audio signals with different quality into signals with smooth and uniform representative characteristics, which is convenient for the subsequent feature extraction. Preprocessing includes pre-emphasis, framing, and windowing. The next step is to process the data by transforming the raw audio into spectrograms, which contain both temporal and frequency domain information. These spectrograms are fed into a pre-trained model. Due to its excellent performance in audio emotion recognition, the pre-trained DenseNet [19] network model was selected for extracting emotional features from the spectrograms. The overall steps for audio emotion feature extraction are illustrated in Figure 4.

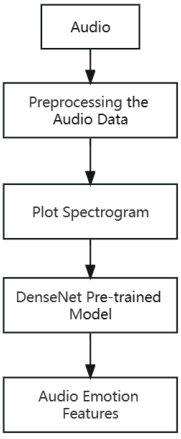


Figure 4. Audio feature extraction model.

Spectrograms visualize audio signals, and they can be regarded as color images in terms of their representation. By using the two-dimensional image to describe the three-dimensional information of time, frequency, and energy, the differences between different audio data can more effectively captured. Moreover, spectrograms are two-dimensional, colorful images, making them suitable for feature extraction using CNNs. The spectrograms corresponding to the seven emotions are shown in Figure 5.

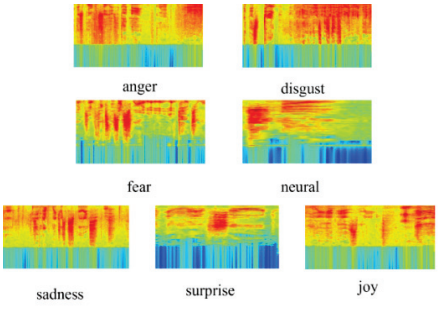


Figure 5. Spectrogram of seven emotions. The horizontal axis of the spectrogram represents the temporal information of the audio signal, while the vertical axis represents the frequency of the audio signal. The two-dimensional coordinates in the spectrogram represent the frequency of the audio at a specific moment, and the intensity of the coordinates also reflects the energy of the audio. The darker the color in the spectrogram, the higher the energy.

Finally, each spectrogram corresponding to each short-time frame is input into DenseNet to extract emotional features from the spectrogram. The detailed architecture of the DenseNet used for feature extraction is shown in Figure 6.

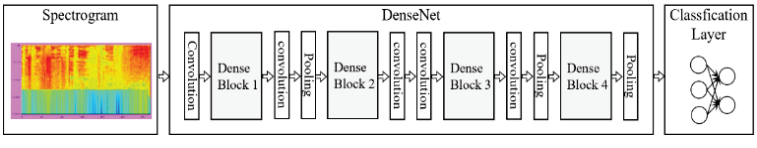


Figure 6. Structure of DenseNet in our method.

2.1.3. Video Modality Preprocessing and Feature Extraction

Capturing the emotional features of facial expressions from speakers accurately is a key challenge in implementing video-based emotion analysis. In real-life applications, the analysis is typically focused on the emotions of the subject (usually human) in a video, and human emotions tend to change slowly over time. Therefore, it is not necessary to analyze every frame in the video when extracting emotional features. Sampling frames from the video and analyzing those samples is sufficient. However, there are usually multiple faces present in the same frame in the case of multi-party dialogue scenarios. The facial expressions of unrelated persons can interfere with the analysis of the speaker's emotions. Therefore, the challenge in analyzing facial expressions in multi-party dialogue scenarios is how to isolate the facial expressions of the speaker.

We selected MELD, which is a widely used multi-party dialogue dataset. In our study, we first read all the sample data from the dataset. For all the video data, we sampled every fifth frame and applied multi-face localization to locate all the faces in the sampled frames, as shown in Figure 7. There are three persons: *Rachel*, *Monica*, and *Phoebe* with distinct facial expressions, i.e., neutral and anger. The facial expressions of the non-speakers (*Rachel* and *Monica*) in the frame can affect the emotion recognition of the real speaker (*Phoebe*). Therefore, it is necessary to exclude the faces of unrelated persons from the frames. Then, we extracted facial expression images of all the faces in the sampled frames.

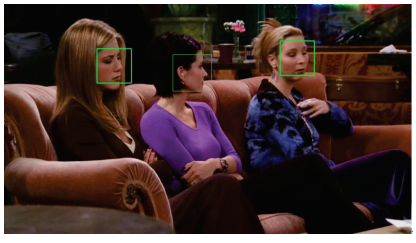


Figure 7. An example of multiple face detection technology locating all faces.

The facial expression image obtained in the previous step is for everyone in the picture, including both the speaker and the non-speaker. We use the speaker recognition method to extract the facial expression sequence of the speaker in the video. The length of the video modality in MELD is set to correspond to a single sentence in the text modality. The text modality also provides speaker annotations for each sentence in the dialogue. We can determine who is speaking in the video by loading the labels from the text modality. We applied speaker recognition to filter out the facial images of the speaker for each video segment. Figure 8 illustrates the changes in a speaker's facial expressions in a specific sequence of video segments.

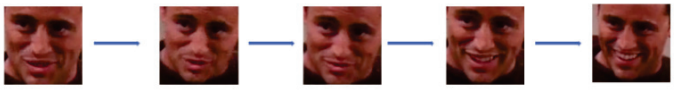


Figure 8. The changes in the speaker's facial expression. For each video segment, the number of facial expression images obtained through the previous steps is different. To ensure a consistent frame count for each video segment, a trimming and padding process is performed.

Firstly, the average number of facial expression frames obtained is calculated for each video segment in the dataset after the previous steps. During the experiment, we chose to retain a sequence of 30 frames for each video segment. For segments with fewer than 30 frames, zero-padding is used to fill the remaining frames, while for segments with more than 30 frames the sequence is trimmed to 30 frames. After that, the 30-frame facial expression sequence represents the entire video segment, and it can be directly input into the facial feature extraction model.

In the stage of feature extraction, we input the facial expression sequences of the speaker obtained from the previous steps into a pre-trained model VGG16 [45] to extract emotional features from each frame. Since facial expressions are slow-changing sequences, we also adopted LSTM to capture temporal context information through multiple rounds of training, thereby obtaining richer and more comprehensive emotional features.

The specific process for extracting emotional features from the video modality is shown in Figure 9. The preprocessing primarily involves using the OpenCV library to read video frame data. Then, our speaker recognition method of M2ER is applied to filter out the facial expressions of the speaker in the video. As a result, the complete video samples are processed into facial expression sequence images with dimensions of (30, 3, 224, 224). The feature extraction stage utilizes a combination of VGG16 and LSTM. The output from the fully connected layers is used as the emotional feature vector of the video modality.

Figure 9. Video feature extraction model.

2.2. Multimodal Emotion Recognition Based on Attention Mechanism

Building an effective multimodal fusion model is a crucial step in multimodal sentiment recognition. We adopted a feature-level fusion approach to combine the emotional features extracted from the text, audio, and video modalities obtained by the aforementioned models. Current research on multimodal sentiment recognition often relies on extracting a large number of features to identify emotion. However, directly concatenating these features can lead to the curse of dimensionality, and there is no distinction in their importance, which may result in the overshadowing of relatively significant features. Furthermore, there are often correlations among the features from the text, audio, and video modalities. Additionally, it is observable that people express emotion differently in real-life scenarios, but existing multimodal fusion models often overlook this phenomenon. The attention mechanism can be used in neural networks to achieve more effective resource allocation. Based on these issues, M2ER explores an attention-based multimodal fusion model.

Our fusion model consists of three main parts: (shown in Figure 2).

(1) Cross-Modal Attention Interaction—*Part 1*: This module utilizes cross-modal attention to capture the intermodal relationships and obtain the feature representation of the interaction between different modalities.
(2) Multimodal Attention Fusion—*Part 2*: This module employs the attention mechanism to determine the importance of each modality in the final fusion classification. It obtains the weight distribution of each modality's features in the fusion process and performs the fusion accordingly.
(3) Finally, the fused multimodal features are passed through the softmax classification layer for emotion recognition.

2.2.1. Cross-Modal Attention Interaction

Because multimodal emotion recognition often involves a large number of features, determining the importance of these features and capturing the relationships among multimodal emotional features are key issues. In our fusion model, we incorporate the emotional features extracted from the text, audio, and video modalities. This is achieved by utilizing cross-modal attention to facilitate the interaction among different modalities. The input to the Cross-Modal Attention Interaction—*Part 1* is the emotional features of the text, audio, and video modalities, represented as M^T, M^A, and M^V, respectively. The specific architecture of *Part 1* is shown in Figure 10.

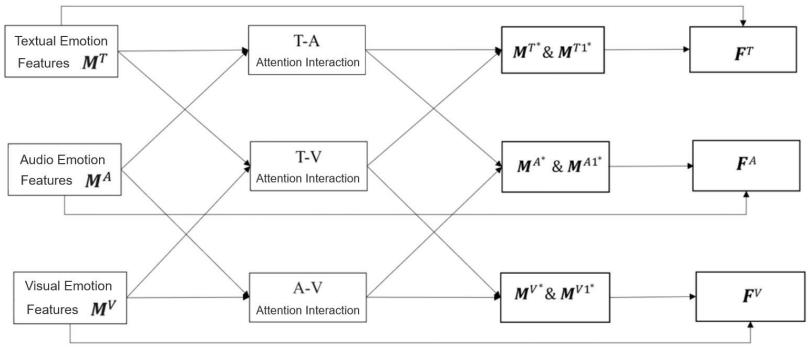


Figure 10. Cross-modal attention interaction architecture.

Taking the example of inputting the text and audio modality into the Cross-Modal Attention Interaction module, M^{T^*} represents the feature representation with interaction obtained from the text through this module. The calculation formula for M^{T^*} is shown in Equations (1)–(3). Similarly, M^{A^*} represents the feature representation with interaction obtained from the audio modality through the T-A attention module. The calculation formula for M^{A^*} is shown in Equations (4)–(6).

$$H^{TA} = M^T M^{A^T}, \tag{1}$$

$$\alpha^{TA} = softmax(H^{TA}, \tag{2}$$

$$M^{T^*} = (\alpha^{TA} M^T) * M^T, \tag{3}$$

$$H^{AT} = M^A M^{T^T}, \tag{4}$$

$$\alpha^{AT} = softmax(H^{AT}, \tag{5}$$

$$M^{A^*} = (\alpha^{AT} M^A) * M^A, \tag{6}$$

where H^{TA} and H^{AT} represent the cross-modal interaction information between the text and audio modalities. α^{TA} and α^{AT} represent the scores obtained for the text and audio modalities in the cross-modal attention interaction. By applying the soft attention mechanism to the emotional features of the input text and audio modalities in *Part 1* and multiplying M^T, M^A with the corresponding elements of their respective matrices, we obtain the feature representation with interaction for the text and audio modalities M^{T^*}, M^{A^*}.

Part 1 is divided into three main parts: text–audio attention interaction (*T-A*), text–video attention interaction (*T-V*), and audio–video attention interaction (A-V).

(1) *T-A*: The emotional features M^T, M^A of the input text and audio modalities in *Part 1* are used to obtain the interaction representation between text and audio M^{T^*}, M^{A^*} through cross-modal attention;
(2) *T-V*: The emotional features M^T, M^V of the input text and video modalities in *Part 1* are used to obtain the interaction representation between text and video, M^{T1^*}, M^{V^*} through cross-modal attention;
(3) *A-V*: The emotional features M^A, M^V of the input audio and video modalities in *Part 1* are used to obtain the interaction representation between audio and video M^{A1^*}, M^{V1^*} through cross-modal attention.

Finally, we obtained the interaction feature representations of the text modality: M^{T^*}, M^{T1^*}; the interaction feature representations of the audio modality: M^{A^*}, M^{A1^*}; and the interaction feature representations of the video modality: M^{V^*}, M^{V1^*}. These representations are concatenated with the respective emotional features of each modality using

fully connected layers to obtain the complete representation of the text, audio, and video modalities' emotional features. The calculation formulas for this process are shown in Equations (7)–(9).

$$F^T = \tanh(W^T[M^T \oplus M^{T^*} \oplus M^{T1^*}] + b^T), \tag{7}$$

$$F^A = \tanh(W^A[M^A \oplus M^{A^*} \oplus M^{A1^*}] + b^A), \tag{8}$$

$$F^V = \tanh(W^V[M^V \oplus M^{V^*} \oplus M^{V1^*}] + b^V), \tag{9}$$

W^T, W^A, W^V, b^T, b^A, b^V are the parameters to be learned, $\oplus$ denotes the concatenation operation. By performing the concatenation operation, we obtain the final text emotional features F_i^T, audio emotional features F_i^A, and video emotional features F_i^V for *Part 1*.

2.2.2. Multimodal Attention Fusion

People often express emotions in different ways in reality. Some people prefer to express their emotions through various facial expressions while others through different tones of voice. Based on this phenomenon, it can be inferred that different modalities of emotional features contribute differently to the final emotion classification. Therefore, in our fusion model, an attention mechanism was adopted to determine the importance of each modality in the final classification. Specifically, the attention mechanism is used to allocate attention weights to the emotional features F^T, F^A, and F^V obtained in *Part 1*. Finally, these weighted features are summed to obtain the fused emotional feature, denoted as F^*. The calculation process is illustrated in Equations (10)–(12):

$$H_X = \tanh(W_{att}^X F^X + b_{att}^X), \tag{10}$$

$$\beta_X = softmax(H_X), \tag{11}$$

$$F^* = \sum_X F^X \beta_X^T, \tag{12}$$

where X represents the modality, which can be text, video, or audio. H_X represents the hidden unit state, W_{att}^X represents the weights, and b_{att}^X represents the biases. Equation (11) is used to normalize the weight vector. The resulting F^* is then fed into the fully connected layer and the softmax classification layer for emotion classification.

3. Evaluation

3.1. Dataset Introduction

This study primarily adopted MELD for experiments. MELD is a multimodal dataset based on dialogues which is widely used for emotion recognition. The dataset consists of over 1400 dialogues, which contain more than 13,000 utterances. Due to the presence of multiple speakers in the same scenario, multi-party dialogues are more challenging than binary dialogues.

For each dialogue segment in MELD, researchers have annotated the corresponding emotion category for each utterance. Table 1 presents the emotion distribution in MELD. Table 2 provides several key statistical data of the dataset. By analyzing the emotion distribution in the training, validation, and test sets, it can be observed that the emotion distribution in the dataset is uneven. The majority of emotions are neutral, while the categories of fear and disgust have fewer instances. So, we conducted further experiments on the MEISD dataset with a more balanced emotional distribution to verify the scalability of our model.

Table 1. The emotion distribution in MELD.

Emotion	Training Set	Test Set	Validation Set
Anger	1109	153	345
Disgust	271	22	68
Fear	268	40	50
Joy	1743	163	402
Neutral	4710	470	1256
Sadness	683	111	208
Surprise	1205	150	281

Table 2. The detailed distribution of MELD. In the training, validation, and test sets, the average utterance length is almost the same.

MELD Statisic	Training Set	Test Set	Validation Set
No. of modalities	{a, v, t}	{a, v, t}	{a, v, t}
No. of unique words	10643	2384	4361
Avg./Max utterance length	8.0/69	7.9/37	8.2/45
No. of dialogues	1039	114	280
No. of dialogues dyadic MELD	2560	270	577
No. of utterances	9989	1109	2610
No. of speakers	260	47	100
Avg. No. of utterances per dialogue	9.6	9.7	9.3
Avg. No. of emotions per dialogue	3.3	3.3	3.2
Avg./Max No. of speakers per dialogue	2.7/9	3.0/8	2.6/8
No. of emotion shift	4003	427	1003
Avg. duration of an utterance	3.59 s	3.59 s	3.58 s

3.2. Experimental Setting

In our work, the experiments primarily utilized Python with PyTorch. Table 3 displays the hardware configuration used during the experiments. Python 3.7 with PyTorch 1.12.1 was installed on the PC via Anaconda.

Table 3. The server hardware configuration information.

Graphics Card	Server	RAM
NVIDIA GeForce RTX 3090 Ti	AMD Ryzen 9 5950X 16-Core Processor 3.4 GHz	32 G

Our experiment used the cross-entropy loss function and optimized the model parameters using the Adam optimizer [46] with the learning rate of 0.001. To prevent overfitting, We applied the dropout rate of 0.2. The model was trained for 100 epochs with the batch size of 64, which we found to be the most effective.

The experiment was mainly divided into three processes: training, validation, and testing. The model was trained on the training set of MELD, and the validation set was used to observe the training progress of the model and adjust relevant parameters based on the actual training process. Finally, the trained model was used to predict the results on the test set.

3.3. Performance Evaluation

Precision, Recall, and F1 Score are the main key performance indicators used to compare the performance of various models or algorithms [47]. Precision is the ability of the classifier not to label as positive a sample that is negative, and Recall is the ability of the classifier to find all the positive samples. The F1 Score can be interpreted as a weighted harmonic mean of the Precision and Recall. All of them were computed for the proposed model and other baseline models. In our experiment, the micro-F1 Score was used as the evaluation metric.

For binary classification evaluation metrics, the calculation formula for F1 Score is shown in Equations (13)–(15), as follows:

$$Recall = \frac{TP}{TP + FN}, \tag{13}$$

$$Precision = \frac{TP}{TP + FP}, \tag{14}$$

$$F1 = 2 * Recall * \frac{Precision}{Recall + Precision}, \tag{15}$$

where *TP* is true positives, *TN* is true negatives, *FP* is false positives, and *FN* is false negatives.

Multiclass evaluation metrics are derived from binary classification evaluation metrics. The micro-F1 Score takes into account the issue of class imbalance. This approach calculates the global Precision and Recall directly based on individual samples. The calculation formulas are shown in Equations (16)–(18), as follows:

$$Precision_{micro} = \frac{\sum_{i=1}^{L} TP}{\sum_{i=1}^{L} TP + \sum_{i=1}^{L} FP}, \tag{16}$$

$$Recall_{micro} = \frac{\sum_{i=1}^{L} TP}{\sum_{i=1}^{L} TP + \sum_{i=1}^{L} FN}, \tag{17}$$

$$micro - F1 = \frac{2 \cdot Precision_{micro} \cdot Recall_{micro}}{Precision_{micro} + Recall_{micro}}, \tag{18}$$

3.4. Results of Unimodal Experiments

This section primarily outlines the related experiments conducted on MELD, comparing our proposed method with commonly used baseline models for emotion recognition. Specifically, we compare the Text-CNN (text modality only) [48], bcLSTM [49], and DialogueRNN [50] models with our proposed model.

In these experiments, due to the imbalanced distribution of emotions within MELD, we utilized micro-F1 and weighted-average F1 (w-avg F1) as evaluation metrics. Table 4 presents the results for the seven emotion categories on the test set.

Table 4. Scores for unimodal emotion classification on the test set. To facilitate a clearer comparative analysis, this table was transformed into a bar chart as shown in Figure 11.

Model		Emotion							
		Anger	**Disgust**	**Fear**	**Joy**	**Neutral**	**Sadness**	**Surprise**	**w-avg F1**
text-CNN	text	34.49	8.22	3.74	49.39	74.88	21.05	45.45	55.02
bcLSTM	text	42.06	21.69	7.75	54.31	71.63	26.92	48.15	56.44
	audio	25.85	6.06	2.9	15.74	61.86	14.71	19.34	39.08
DialogueRNN	text	40.59	2.04	8.93	50.27	75.75	24.19	49.38	57.03
	audio	35.18	5.13	5.56	13.17	65.57	14.01	20.47	41.79
M2ER	text	40.45	15.24	5.55	53.31	77.57	37.85	52.42	60.05
	audio	31.51	9.02	5.25	29.08	64.84	13.06	20.13	43.39
	video	24.43	6.62	4.54	22.89	63.68	20.07	29.44	42.73

It can be observed that the emotion recognition performance in the text modality is generally better than that in the audio and video modalities. The text modality achieved a w-avg F1 of 60.05%, which is an improvement of 9.14%, 6.4%, and 5.3% compared to the Text-CNN, bcLSTM, and DialogueRNN models, respectively. The audio modality achieved a w-avg F1 of 43.39%, which is an improvement of 10.03% and 3.8% compared to the bcLSTM and DialogueRNN models, respectively. These results demonstrate the effectiveness of the text and audio modality feature extraction models, surpassing the performance of popular baseline models.

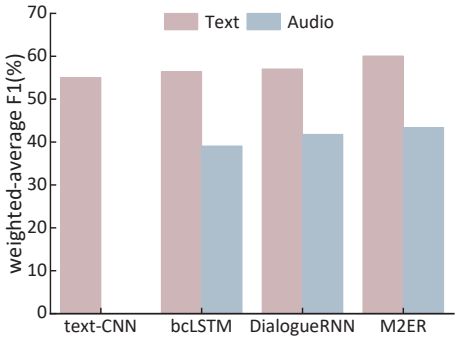


Figure 11. Performance results of different models of text and audio modalities on MELD. The figure shows the comparison of the w-avg F1 values for different models in text and audio modality emotion recognition on MELD.

3.5. Results of Multimodal Experiments

Relevant experiments were conducted on the training, validation, and test sets of MELD. The experimental results are shown in Table 5. This model utilizes cross-modal attention interaction to capture correlated information between modalities, obtaining feature representations with interactive effects. By using the attention mechanism, it determines the importance of each modality in the final fusion classification and combines the multimodal information.

Table 5. Scores for the seven emotion classifications on the test set. *"text + audio + video"* represents our attention-based multimodal fusion model. This table was transformed into a bar chart, as shown in Figure 12.

Model		**Emotion**							
		Anger	**Disgust**	**Fear**	**Joy**	**Neutral**	**Sadness**	**Surprise**	**w-avg F1**
bcLSTM	text + audio	43.39	23.66	9.38	54.48	76.67	24.34	51.04	59.25
DialogueRNN	text + audio	43.65	7.89	11.68	54.40	77.44	34.59	52.51	60.25
M2ER	text	40.45	15.24	5.55	53.31	77.57	37.85	52.42	60.05
	audio	31.51	9.02	5.25	29.08	64.84	13.06	20.13	43.39
	video	24.43	6.62	4.54	24.89	63.68	20.07	29.44	42.73
	text + audio + video	43.75	16.93	9.62	56.63	80.11	41.67	54.14	62.83

Comparing the results in Figure 12 shows that our three-modality fusion emotion recognition model achieved a w-avg F1 score of 62.83%, outperforming the individual modalities of text, audio, and video in emotion recognition. Compared to unimodal data, multimodal data can capture more diverse emotional features. Multimodal fusion can also compensate for the limitations of individual modalities. Furthermore, our multimodal fusion model shows improved fusion performance compared to several baseline models. It was found that the recognition results for disgust and fear were not satisfactory. To address this, we further conducted a five-class emotion recognition experiment on MELD, excluding the less frequent emotions of fear and disgust. The results of this experiment are shown in Table 6.

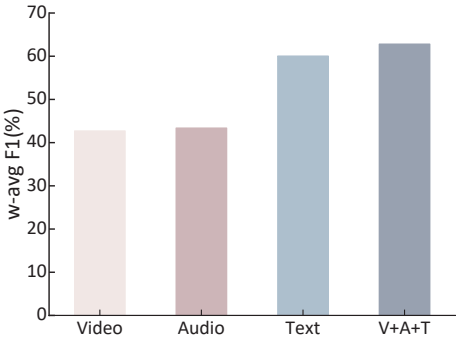


Figure 12. Results of unimodal and multimodal emotion classification of our model on the test set. They present a comparison of the w-avg F1 values for both unimodal and multimodal emotion recognition.

Table 6. Scores for the five emotion classifications on the test set.

Model		Emotion					
		Anger	**Joy**	**Neutral**	**Sadness**	**Surprise**	**w-avg F1**
bcLSTM	text + audio	45.9	52.2	77.9	11.2	49.9	60.6
DialogueRNN	text	41.7	53.7	77.8	21.2	47.7	60.8
	audio	34.1	18.8	66.2	16	16.6	44.3
	text + audio	48.2	53.2	77.7	20.3	48.5	61.6
M2ER	text	42.1	53.2	78.6	35.9	52.3	62.9
	audio	32.5	31.0	66.1	13.6	23.2	46.7
	video	29.2	26.0	65.7	19.5	29.6	46.3
	text + audio + video	45.6	55.1	79.4	36.3	53.9	64.3

By comparing the results in Tables 5 and 6, it can be observed that after excluding two less frequent emotions, the five-class emotion recognition performance significantly improved compared to the seven-class classification because there are very few samples for the emotions of fear and disgust in the training set. Additionally, distinguishing between anger, disgust, and fear is challenging as the differences between these emotions are subtle. This explains why the recognition results for disgust and fear were relatively poor in the seven-class emotion recognition experiment. Furthermore, the performance of the text modality in emotion recognition remained generally superior to that of the audio and video modalities in the five-class emotion recognition experiment, and the multimodal emotion recognition outperformed the single modality.

3.6. Ablation Experiments

We conducted ablation experiments to validate the effectiveness of different components designed in our multimodal feature fusion model by removing specific modules in the multimodal fusion part.

Our fusion model mainly consists of two parts: *Part 1* utilizes cross-modal attention to capture the interaction between modalities; *Part 2* utilizes the attention mechanism to determine the importance of each modality for the final classification and fuses the multimodal information. We conducted comparative experiments between direct concatenation and the fusion mechanism we adopted, as shown in Table 7.

Comparing the experimental results in Figure 13, *Fusion 2* improved the performance by 1.09% compared to *Fusion 1*. Compared with *Fusion 1* and *Fusion 2*, *Fusion 3* improved the performance by 3.4% and 2.3%, respectively, as *Fusion 2* and *Fusion 3* utilize the correlated information between modalities and effectively allocate importance weights to each modality. The ablation experiments confirmed the effectiveness of each module in the fusion model in our work—*Part 1* and *Part 2*.

Table 7. Results of ablation experiments for multimodal fusion emotion recognition. *Fusion 1* represents direct concatenation; *Fusion 2* represents the first part of the fusion model, *Part 1*, which considers only the interaction between modalities and within each modality; *Fusion 3* represents the complete fusion model, *Part 1* + *Part 2*. This table has been transformed into a bar chart, as shown in Figure 13.

Model			Emotion							
			Anger	**Disgust**	**Fear**	**Joy**	**Neutral**	**Sadness**	**Surprise**	**w-avg F1**
M2ER	Fusion 1	text + audio + video	41.81	11.63	7.10	54.37	77.84	40.09	53.41	60.74
	Fusion 2	text + audio + video	41.99	15.83	7.90	55.07	78.52	40.39	53.86	61.40
	Fusion 3	text + audio + video	43.75	16.93	9.62	56.63	80.11	41.67	54.14	62.83

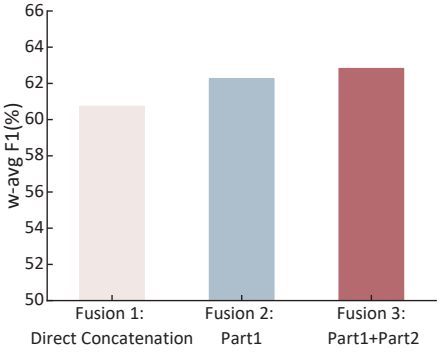


Figure 13. Experimental results of multimodal emotion recognition ablation. They present the comparison of the w-avg F1 scores for different variants of the multimodal fusion model on MELD.

Since the baseline models on MELD did not utilize the video modality, and it was found that most dialogue emotion recognition studies based on MELD also did not utilize the video modality through research, A comparative analysis was performed between the video modality emotion recognition models without speaker recognition and the models utilizing it in order to validate the effectiveness of our proposed speaker recognition method. The results are presented in Table 8.

Table 8. Results of the ablation experiments in the video modality. *video'* represents the method where our method was not used. *video* represents our proposed method of using speaker recognition.

Model	Emotion							
	Anger	**Disgust**	**Fear**	**Joy**	**Neutral**	**Sadness**	**Surprise**	**w-avg F1**
video'	22.64	5.41	3.32	23.07	60.34	14.15	24.16	40.05
video	24.43	6.62	4.54	22.89	63.68	20.07	29.44	42.73

By comparing the data in Table 8, it can be observed that our model achieved a w-avg F1 score of 42.73% for emotion classification in the video modality. Furthermore, by comparing the data of *video'* and *video*, it is evident that our proposed speaker recognition method improved the emotion recognition performance in the video modality by 6.58%. The comparison in Table 8 indicates that the method effectively enhances the efficiency of extracting emotional features in multi-party dialogue scenarios, highlighting the role of the video modality in emotion recognition during multi-party dialogues.

3.7. Model Scalability Verification

To validate the scalability of our multimodal emotion recognition model, we conducted emotion recognition experiments on the MEISD dataset, which is also a multi-party dialogue dataset. Additionally, we performed emotion recognition tests on some real-world

data using the fusion model and presented the test results to visualize the generalization performance of the model.

We further conducted multi-class emotion recognition experiments on the MEISD dataset, using the micro-F1 score as the evaluation metric. Since the distribution of each emotion label in the training, validation, and test sets of the MEISD dataset is relatively balanced, the prediction performance for each emotion label is almost the same. Therefore, we collected the overall w-avg F1 score for comparison and presentation in the experiments. The results are shown in Table 9.

Table 9. Scores for emotion classification on the MEISD dataset. This table has been transformed into a bar chart, as shown in Figure 14.

Model		w-avg F1
text-CNN	text	54.18
bcLSTM	text	57.05
	audio	41.17
	video	39.45
	text + audio + video	59.32
DialogueRNN	text	58.73
	audio	41.52
	video	40.87
	text + audio + video	60.57
M2ER	text	61.10
	audio	43.49
	video	42.91
	text + audio + video	62.97

Figure 14 shows that our multimodal emotion recognition approach also performs well on the MEISD dataset. The w-avg F1 score for text modality emotion recognition is 61.10%, for audio modality is 43.49%, and for the video modality is 42.91%. The performance of individual modalities in emotion recognition surpasses that of classical baseline models. The fusion model achieves a w-avg F1 score of 62.97%, outperforming the individual modality recognition results. These experimental results demonstrate the effectiveness and scalability of our multimodal emotion recognition model in multi-party dialogue scenarios.

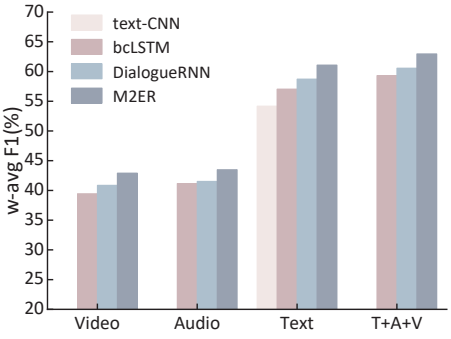


Figure 14. Emotional classification results of the MEISD. They present the comparison of w-avg F1 scores for unimodal and multimodal emotion recognition of various models on the MEISD dataset.

To better illustrate the scalability of the model, we tested it on some actual examples, as shown in Table 10. From the table, it can be observed that in the case of Example A, the facial image of the speaker obtained from the video modality shows an upward curvature of the mouth, indicating a smiling expression. Additionally, the speaker's voice has a high pitch and a cheerful speaking rate. The text also expresses a positive emotion, leading to

a predicted emotion of positive. In the case of Example B, the facial image of the speaker obtained from the video modality shows a furrowed brow, and the speaker's speech is slow and filled with a tone of sadness. The text modality also exhibits negative sentiment, resulting in a predicted emotion of negative, which aligns with the authentic label. In the case of Example C, the facial expression of the speaker obtained from the video modality is relatively neutral without a clear emotional color, but its text and audio modalities have evident negative sentiment, so the final prediction result is also negative, which is consistent with the real label. Through the analysis and presentation above examples, it is evident that our multimodal emotion recognition model based on multi-party dialogue scenarios can effectively identify the speaker and successfully fuse information from the text, audio, and video modalities.

Table 10. Examples of multimodal emotion recognition. *T* represents the dialogue text, *V* represents the speaker's visual information, and *A* represents the audio information in the video.

Example	Speaker	T	V	A	Authentic Emotion	Predicting Emotion
A	Phoebe	Ohh! I'm gonna be on the news.		high pitch, cheerful tone	Positive	Positive
B	Monica	So, I hear you, you hate me!		slow pace, downcast tone	Negative	Negative
C	Ross	Look! I did not feel like dancing. Okay?		downcast tone, high pitch	Positive	Positive

4. Discussion

4.1. Strengths

M2ER has solved the problem of speaker recognition when multiple faces appear in the same video frame, which enables utilization of the video modality for emotion recognition. When encountering multiple people in the same scene, M2ER eliminates the interference of other people by recognizing the speaker and using the facial expressions of the speaker in the video and the changes during video playback.

To incorporate the multimodal fusion model into our approach, we employed a feature-level fusion that relies on the attention mechanism. By utilizing cross-modal attention, we combined the extracted unimodal emotional features to effectively capture intermodal interactions.

4.2. Limitations and Future Work

In real-world scenarios, a variety of factors can cause modality absence, such as the faces in the video not appearing within the range of the camera at some moments. The datasets we used are also affected by modality absence, which definitely affects the accuracy of the modality. In future work, we will focus on addressing the issue of modality absence, which may enhance M2ER.

The proposed method involves multiple components, including face detection, face recognition, and attention-based fusion; while these components enhance the effectiveness of the model, they also introduce complexity that undoubtedly increases the difficulty of implementation for others.

Another limitation is that only two datasets were utilized for the experiment, without further testing the generalization of the model. Due to the potential impact of different data sources on performance, it is necessary to explore how well the proposed method generalizes to other datasets due to the potential impact of diverse data sources on performance in the future.

5. Conclusions

In this paper, our work primarily focuses on the research of multimodal emotion recognition in multi-party dialogue scenarios. We propose a novel approach using multi-face localization for speaker recognition in the video modality, thus enhancing the efficiency of utilizing the video modality in the field of multimodal emotion recognition. In the multimodal fusion part, we explore a multimodal feature fusion model based on attention mechanism to address dimension explosion and poor correlation in the directly concatenated fusion model. We conducted seven-class emotion experiments, five-class emotion experiments, and scalability experiments. The results validate the effectiveness of M2ER.

Author Contributions: Conceptualization, G.W.; methodology, B.Z. and X.Y.; validation, B.Z., X.Y. and G.W.; formal analysis, Y.W.; investigation, Y.W.; resources, X.Y.; data curation, G.W. and B.Z.; writing—original draft preparation, X.Y.; writing—review and editing, B.Z. and R.S.; visualization, Y.W; supervision, B.Z. and R.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ekman, P. An argument for basic emotions. *Cogn. Emot.* **1992**, *6*, 169–200. [CrossRef]
2. Ekman, P.; Friesen, W.V. Constants across cultures in the face and emotion. *J. Personal. Soc. Psychol.* **1971**, *17*, 124. [CrossRef] [PubMed]
3. Picard, R.W. *Affective Computing*; MIT Press: Cambridge, MA, USA, 2000.
4. Trigeorgis, G.; Ringeval, F.; Brueckner, R.; Marchi, E.; Nicolaou, M.A.; Schuller, B.; Zafeiriou, S. Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 5200–5204.
5. Zhang, S.; Zhang, S.; Huang, T.; Gao, W.; Tian, Q. Learning affective features with a hybrid deep model for audio–visual emotion recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 3030–3043. [CrossRef]
6. Perveen, N.; Roy, D.; Chalavadi, K.M. Facial expression recognition in videos using dynamic kernels. *IEEE Trans. Image Process.* **2020**, *29*, 8316–8325. [CrossRef] [PubMed]
7. Mollahosseini, A.; Hasani, B.; Mahoor, M.H. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Trans. Affect. Comput.* **2019**, *10*, 18–31. [CrossRef]
8. Mao, J.; Xu, R.; Yin, X.; Chang, Y.; Nie, B.; Huang, A. Poster v2: A simpler and stronger facial expression recognition network. *arXiv* **2023**, arXiv:2301.12149.
9. Panagiotis, A.; Filntisis, P.P.; Maragos, P. Exploiting emotional dependencies with graph convolutional networks for facial expression recognition. *arXiv* **2021**, arXiv:2106.03487.
10. Ryumina, E.; Dresvyanskiy, D.; Karpov, A. In search of a robust facial expressions recognition model: A large-scale visual cross-corpus study. *Neurocomputing* **2022**, *514*, 435–450. [CrossRef]
11. Bakariya, B.; Singh, A.; Singh, H.; Raju, P.; Rajpoot, R.; Mohbey, K.K. Facial emotion recognition and music recommendation system using cnn-based deep learning techniques. In *Evolving Systems*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 1–18.
12. Meena, G.; Mohbey, K.K.; Indian, A.; Khan, M.Z.; Kumar, S. Identifying emotions from facial expressions using a deep convolutional neural network-based approach. In *Multimedia Tools and Applications*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 1–22.
13. Savchenko, A.V. Facial expression and attributes recognition based on multi-task learning of lightweight neural networks. In Proceedings of the 2021 IEEE 19th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 16–18 September 2021; pp. 119–124.
14. Meena, G.; Mohbey, K.K.; Kumar, S. Sentiment analysis on images using convolutional neural networks based inception-v3 transfer learning approach. *Int. J. Inf. Manag. Data Insights* **2023**, *3*, 100174. [CrossRef]
15. Mehendale, N. Facial emotion recognition using convolutional neural networks (ferc). *SN Appl. Sci.* **2020**, *2*, 446. [CrossRef]
16. Li, T.H.S.; Kuo, P.H.; Tsai, T.N.; Luan, P.C. Cnn and lstm based facial expression analysis model for a humanoid robot. *IEEE Access* **2019**, *7*, 93998–94011. [CrossRef]

17. Ming, Y.; Qian, H.; Guangyuan, L. Cnn-lstm facial expression recognition method fused with two-layer attention mechanism. *Comput. Intell. Neurosci.* **2022**, *2022*, 7450637. [CrossRef] [PubMed]
18. Sang, D.V.; Ha, P.T. Discriminative deep feature learning for facial emotion recognition. In Proceedings of the 2018 1st International Conference on Multimedia Analysis and Pattern Recognition (MAPR), Ho Chi Minh City, Vietnam, 5–6 April 2018; pp. 1–6.
19. Huang, G.; Liu, Z.; Maaten, L.V.D.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
20. Xue, F.; Wang, Q.; Guo, G. Transfer: Learning relation-aware facial expression representations with transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 3601–3610.
21. Ma, F.; Sun, B.; Li, S. Facial expression recognition with visual transformers and attentional selective fusion. *IEEE Trans. Affect. Comput.* **2021**, *14*, 1236–1248. [CrossRef]
22. Chen, C.-F.R.; Fan, Q.; Panda, R. Crossvit: Cross-attention multi-scale vision transformer for image classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 357–366.
23. Heo, B.; Yun, S.; Han, D.; Chun, S.; Choe, J.; Oh, S.J. Rethinking spatial dimensions of vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 11936–11945.
24. Nguyen, D.; Nguyen, K.; Sridharan, S.; Ghasemi, A.; Dean, D.; Fookes, C. Deep spatio-temporal features for multimodal emotion recognition. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 1215–1223.
25. Guanghui, C.; Xiaoping, Z. Multi-modal emotion recognition by fusing correlation features of speech-visual. *IEEE Signal Process. Lett.* **2021**, *28*, 533–537. [CrossRef]
26. Zadeh, A.; Zellers, R.; Pincus, E.; Morency, L. MOSI: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. *arXiv* **2016**, http://arxiv.org/abs/1606.06259.
27. Zadeh, A.; Liang, P.P.; Poria, S.; Cambria, E.; Morency, L.-P. Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018. Available online: https://api.semanticscholar.org/CorpusID:51868869 (accessed on 15 July 2018).
28. Dai, W.; Cahyawijaya, S.; Liu, Z.; Fung, P. Multimodal end-to-end sparse model for emotion recognition. *arXiv* **2021**, arXiv:2103.09666.
29. Ren, M.; Huang, X.; Shi, X.; Nie, W. Interactive multimodal attention network for emotion recognition in conversation. *IEEE Signal Process. Lett.* **2021**, *28*, 1046–1050. [CrossRef]
30. Khare, A.; Parthasarathy, S.; Sundaram, S. Self-supervised learning with cross-modal transformers for emotion recognition. In Proceedings of the 2021 IEEE Spoken Language Technology Workshop (SLT), Shenzhen, China, 19–22 January 2021; pp. 381–388.
31. Lv, F.; Chen, X.; Huang, Y.; Duan, L.; Lin, G. Progressive modality reinforcement for human multimodal emotion recognition from unaligned multimodal sequences. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 2554–2562.
32. Xie, B.; Sidulova, M.; Park, C.H. Robust multimodal emotion recognition from conversation with transformer-based crossmodality fusion. *Sensors* **2021**, *21*, 4913. [CrossRef]
33. Zhang, L.; Liu, C.; Jia, N. Uni2mul: A conformer-based multimodal emotion classification model by considering unimodal expression differences with multi-task learning. *Appl. Sci.* **2023**, *13*, 9910. [CrossRef]
34. Wang, H.; Yang, M.; Li, Z.; Liu, Z.; Hu, J.; Fu, Z.; Liu, F. Scanet: Improving multimodal representation and fusion with sparse-and cross-attention for multimodal sentiment analysis. *Comput. Animat. Virtual Worlds* **2022**, *33*, e2090. [CrossRef]
35. Ma, H.; Wang, J.; Qian, L.; Lin, H. Han-regru: Hierarchical attention network with residual gated recurrent unit for emotion recognition in conversation. *Neural Comput. Appl.* **2021**, *33*, 2685–2703. [CrossRef]
36. Jiao, W.; Lyu, M.R.; King, I. Real-time emotion recognition via attention gated hierarchical memory network. *arXiv* **2019**, http://arxiv.org/abs/1911.09075.
37. Xing, S.; Mai, S.; Hu, H. Adapted dynamic memory network for emotion recognition in conversation. *IEEE Trans. Affect. Comput.* **2022**, *13*, 1426–1439. [CrossRef]
38. Hu, D.; Hou, X.; Wei, L.; Jiang, L.; Mo, Y. Mm-dfn: Multimodal dynamic fusion network for emotion recognition in conversations. In Proceedings of the ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 22–27 May 2022; pp. 7037–7041.
39. Poria, S.; Hazarika, D.; Majumder, N.; Naik, G.; Cambria, E.; Mihalcea, R. MELD: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv* **2018**, arXiv:1810.02508.
40. Firdaus, M.; Chauhan, H.; Ekbal, A.; Bhattacharyya, P. MEISD: A multimodal multi-label emotion, intensity and sentiment dialogue dataset for emotion recognition and sentiment analysis in conversations. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 4441–4453. Available online: https://aclanthology.org/2020.coling-main.393 (accessed on 8 December 2020).
41. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. *arXiv* **2018**, arXiv:1802.05365.
42. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]

43. Munikar, M.; Shakya, S.; Shrestha, A. Fine-grained sentiment classification using bert. In Proceedings of the 2019 Artificial Intelligence for Transforming Business and Society (AITB), Kathmandu, Nepal, 5 November 2019; Volume 1, pp. 1–5.
44. Jiang, D.; He, J. Tree framework with bert word embedding for the recognition of Chinese implicit discourse relations. *IEEE Access* **2020**, *8*, 162004–162011. [CrossRef]
45. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
46. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
47. Sharma, S.; Rana, V.; Kumar, V. Deep learning based semantic personalized recommendation system. *Int. J. Inf. Manag. Data Insights* **2021**, *1*, 100028. [CrossRef]
48. Kim, Y. Convolutional neural networks for sentence classification. *arXiv* **2014**, arXiv:1408.5882.
49. Poria, S.; Cambria, E.; Hazarika, D.; Majumder, N.; Zadeh, A.; Morency, L.-P. Context-dependent sentiment analysis in user-generated videos. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 873–883. Available online: https://aclanthology.org/P17-1081 (accessed on 30 July 2017).
50. Majumder, N.; Poria, S.; Hazarika, D.; Mihalcea, R.; Gelbukh, A.; Cambria, E. DialogueRNN: An Attentive RNN for Emotion Detection in Conversations. *arXiv* **2018**, arXiv:1811.00405.

applied sciences

MDPI

Article

Know an Emotion by the Company It Keeps: Word Embeddings from Reddit/Coronavirus

Alejandro García-Rudolph [1,2,3,*], David Sanchez-Pinsach [1,2,3], Dietmar Frey [4], Eloy Opisso [1,2,3], Katryna Cisek [5] and John D. Kelleher [5]

[1] Department of Research and Innovation, Institut Guttmann, Institut Universitari de Neurorehabilitació Adscrit a la UAB, 08027 Badalona, Spain
[2] Universitat Autònoma de Barcelona, Bellaterra (Cerdanyola del Vallès), 08193 Barcelona, Spain
[3] Fundació Institut d'Investigació en Ciències de la Salut Germans Trias i Pujol, 08916 Badalona, Spain
[4] CLAIM Charité Lab for AI in Medicine, Charité Universitätsmedizin Berlin, 10117 Berlin, Germany
[5] Information, Communication and Entertainment Research Institute, Technological University Dublin (TU Dublin), D7 EWV4 Dublin, Ireland
* Correspondence: alejandropablogarcia@gmail.com

Abstract: Social media is a crucial communication tool (e.g., with 430 million monthly active users in online forums such as Reddit), being an objective of Natural Language Processing (NLP) techniques. One of them (word embeddings) is based on the quotation, "You shall know a word by the company it keeps," highlighting the importance of context in NLP. Meanwhile, "Context is everything in Emotion Research." Therefore, we aimed to train a model (W2V) for generating word associations (also known as embeddings) using a popular Coronavirus Reddit forum, validate them using public evidence and apply them to the discovery of context for specific emotions previously reported as related to psychological resilience. We used Pushshiftr, quanteda, broom, wordVectors, and superheat R packages. We collected all 374,421 posts submitted by 104,351 users to Reddit/Coronavirus forum between January 2020 and July 2021. W2V identified 64 terms representing the context for seven positive emotions (gratitude, compassion, love, relief, hope, calm, and admiration) and 52 terms for seven negative emotions (anger, loneliness, boredom, fear, anxiety, confusion, sadness) all from valid experienced situations. We clustered them visually, highlighting contextual similarity. Although trained on a "small" dataset, W2V can be used for context discovery to expand on concepts such as psychological resilience.

Keywords: COVID-19; social media; Reddit; natural language processing; emotions; resilience

Citation: García-Rudolph, A.; Sanchez-Pinsach, D.; Frey, D.; Opisso, E.; Cisek, K.; Kelleher, J.D. Know an Emotion by the Company It Keeps: Word Embeddings from Reddit/Coronavirus. *Appl. Sci.* **2023**, *13*, 6713. https://doi.org/10.3390/app13116713

Academic Editor: Xiangjie Kong

Received: 20 March 2023
Revised: 20 May 2023
Accepted: 25 May 2023
Published: 31 May 2023

1. Introduction

Recent statistics show that there are 4.55 billion social media users around the world, equating to 57.6% of the total global population [1].

With the outbreak of the COVID-19 pandemic, social media on platforms such as Reddit [2] has become a critical communication tool for the generation, dissemination, and consumption of information [3].

Therefore, social media analysis is one of the most popular areas of research in recent days [4]. Many studies apply various Natural Language Processing (NLP) techniques to social media content [5]. Out of them, sentiment analysis and topic models are two of the most researched NLP topics, as concluded in a Lancet Digital Health scoping review [3]. Much less studied, word embeddings have been recently reported as a valuable text analysis technology in the pandemic context [6–8].

Understanding the meaning of a word is at the heart of NLP [9]; the approach followed by word embeddings is based on Firth's notion of "context of situation." In particular, his famous quotation: "You shall know a word by the company it keeps" [10]. Words that occur in similar contexts are prone to have similar meanings [11]. Firth's distributional

hypothesis is the foundation for the actual word embeddings implementations; one of the most popular is word2vec, developed at Google Labs [12].

Meanwhile, in the field of social sciences, as recently remarked, "Context is everything in Emotion Research" [13]. Few social scientists would refute that context fundamentally shapes psychological experience: our thoughts, feelings, and actions, as well as, to some extent, who we are, and we are all influenced by the context in which we find ourselves.

Context influences cognitions, emotions, and actions in a variety of ways, as well as how these outcomes are seen and understood by others [14,15].

Context is at the core of emotion. "Context is what gives rise to the diversity and depth of human emotional experience and the myriad thoughts and behaviors that stem from such experience" [13].

Existing research indicates that positive emotions support people to cope with stressful situations [16]. This concept is also applicable during times of extended stress, such as the COVID-19 worldwide crisis [17].

Despite the fact that they both share context as a central component, word embeddings have been rarely used in providing context to specific emotions, to the best of our knowledge.

Users of social media platforms, such as Reddit [2], often differ significantly from comparable groups that interact in person. For example, Reddit users are more inclined to discuss issues that they would feel uncomfortable addressing in person [18].

With more than 430 million monthly active users, the primary functionality of Reddit is the exchange of text-based postings through subforums, which are places set aside for users to assemble and communicate with one another on a common interest. The Reddit site name is a play on the words "I read it." At the end of 2021, there were more than 2.2M Reddit subforums [19] known as subreddits.

Therefore, users can publish posts (also known as submissions) and comments to a number of communities with shared interests called subreddits. Table 1 presents the subreddits related to COVID-19 with the highest number of subscribers. The Rank column shows the absolute position of each subreddit ranked by the number of subscribers as reported by Reddit stats [20]. The r/Music subreddit was included for comparison purposes, as shown in Table 1; r/music was ranked #12 with more than 20M subscribers, one of the most popular subreddits since Reddit was launched [21].

Table 1. The top COVID-19 subreddits and their position in the global rank.

Subreddit	Number of Subscribers	Rank	Posts per Day
r/Coronavirus	2,354,224	177	101
r/COVID19	336,253	1357	23
r/CoronavirusUS	140.913	3226	22
r/COVID-19Positive	113,778	3944	29
r/China_Flu	103,456	4261	19
r/CovidVaccinated	27,814	11,271	62
r/Music	20,350,355	12	410

Figure 1 was extracted from subreddit stats [20], and it plots the evolution of r/Coronavirus and r/Music (from January 2019 to July 2021), showing the tremendous increase in posts per day experienced by r/Coronavirus even when compared with one of the most popular subreddits as is the case of r/Music.

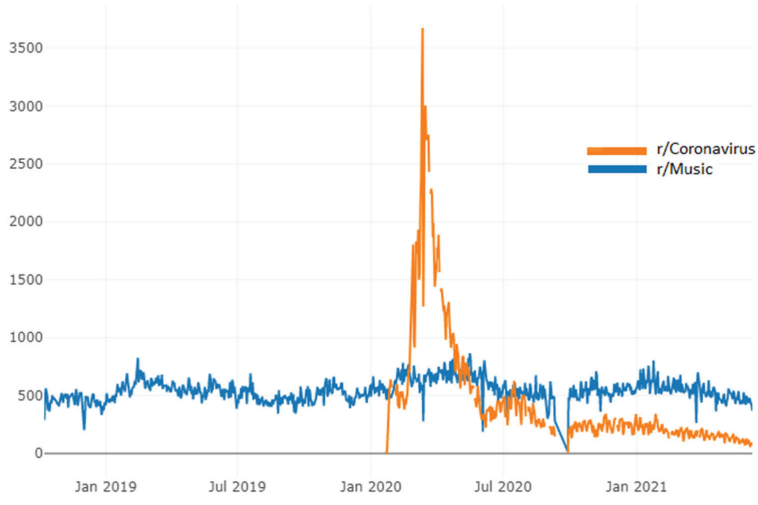


Figure 1. Number of posts per day for the Music and the Coronavirus subreddits.

The r/Coronavirus subreddit is a curated information platform. As presented in the r/Coronavirus official description [22], "This subreddit seeks to monitor the spread of the disease COVID-19, declared a pandemic by the WHO. This subreddit is for high-quality posts and discussion." As emphasized in the r/Coronavirus Rules: "There are many places online to discuss conspiracies and speculate, we ask you not to do so here." Otherwise users get the message: "Your post or comment was removed due to being low quality information" [22]. It is also worth noting that reposts are removed. A repost is a post that is created by taking a post from a while ago and posting it again in the same subreddit. The concept of reposting also covers new posts containing only information that has already been posted [22].

The number of subscribers and posts in the other COVID-19 subreddits are clearly lower and address more specific aspects; therefore, in this work, our data source was r/Coronavirus.

Users submit top-level postings, known as submissions, to each subreddit, while others respond with comments on the submissions. Submissions consist of a title (up to 300 characters) and either a web link or a user-supplied body text; in the latter case, the submission is also known as a self-post, while comments are always made up of a body text.

In this work, we focus on analyzing the titles of Reddit posts. There are two reasons why we believe titles will be a useful basis for NLP analysis.

First, Reddit strongly recommends double-checking the grammar, spelling, and punctuation of the titles: "Read over your submission for mistakes before submitting, especially the title of the submission. Comments and the content of self-posts can be edited after being submitted; however, the title of a post cannot be. Make sure the facts you provide are accurate to avoid any confusion" [23].

Second, Reddit also requests that posters make their titles factual, accurate, and relevant to the content of the post. As remarked in Rediquette: "Please don't editorialize or sensationalize your submission title, keep your submission titles factual and opinion free. If it is an outrageous topic, share your crazy outrage in the comment section. Do not be vague. Make sure redditors know what they are getting. People do not have time to click on every submission to find out what is inside. Contribute value to the community by writing titles that accurately describe what is being shared. Be relevant. Subreddit subscribers like to read about specific topics that are related to their subreddit. If your submission is out of place, it will not gain any attention" [23].

Another advantage of focusing on Reddit post titles is that Twitter has increased the available character space from 140 to 280 characters since November 2017, which is very similar to Reddit's 300 characters limitation of the titles. This provides an opportunity for linguistic comparisons between tweets and Reddit titles.

It is for these reasons that we focused our analysis on the titles of all posts extracted from r/Coronavirus.

A word embedding is a vector-based representation of a word. The vector representing a word can be understood as the coordinates of a word's position within a multi-dimensional feature space (where the dimensions of the feature space are equal to the size of the vector). Within the vector-based representation, the meaning of a word is encoded by its position within the feature space relative to other words in the space. From a linguistic semantics perspective, the concept of word embedding is related to the distributional hypothesis for Firth [10], which can be paraphrased as "you shall know the meaning of a word by the company it keeps." The relationship between the distributional hypothesis and word embeddings is that in well-trained word embedding models, words that occur in similar contexts (i.e., that keep the same company) are positioned close to each other in the feature space (i.e., they have similar vector representations).

Word2vec was created, patented, and published in two papers in 2013 by a team of researchers led by Tomas Mikolov at Google to learn word embeddings from a corpus of language [12]. It creates embeddings for the words in a corpus by training a neural network to predict words that co-occur with other words in the corpus.

Word2Vec includes two alternative strategies for training the neural network: Continuous Bag of Words (CBOW) and Skip-gram. In both of them, a preset length window is moved along the corpus. Using the CBOW strategy, at each step, the network is trained to predict the word in the center of the window based on the surrounding words. In the Skip-gram strategy, the network is trained to predict the other words in the window based on the central word. In both strategies, the learning signal for the network (and hence the information that is encoded in the embeddings the network generates) is the likelihood of one word co-occurring in the surrounding context of another word (i.e., within the same window). In the present paper, we use the Skip-gram model, which has shown better performance in semantic tasks [24].

Psychological resilience, as a general term, deals with how people manage stress and how they recover from traumatic events, encouraging constructive growth and promoting an optimistic outlook on the future [25]. Evidence suggests that when resilience-based abilities are applied to people's lives, they have many advantages (for example, a carry-over effect on other life domains) [26]. Resilience may be improved with deliberate practice; it is not necessary to be born with it [27]. However, within the research community, there is a lack of a unified definition for the concept [28]. This lack of consensus in definition can also be linked to the lack of consensus on how the concept should be operationalized in order to address community disasters [29]. As recently reported [30], positive and negative emotions have varied effects on developing a resilient attitude. People who go through higher levels of positive emotions (i.e., gratitude, compassion, love, relief, hope, calm, or admiration) exhibit a higher degree of resilience, whereas those who feel high levels of negative emotions (i.e., anger, loneliness, boredom, fear, anxiety, confusion, sadness) are associated with poorer resilience.

Typically, large general-purpose corpora (e.g., Wikipedia dumps with 3 billion words [31]) are used to learn word embeddings. Nevertheless, in this work, we hypothesized that word embeddings could be extracted from publicly available social media, using open source software, in sufficient numbers such that the embeddings (1) are relevant to provide meaningful context to specific emotions specifically linked to an ill-defined domain such as psychological resilience (2) verifiable by sound theoretical semantic tests such as the Battig and Montague norm [32] (3) consistent with current related scientific publications and (4) offering the possibility of providing actionable knowledge to on-field specialists.

Therefore the objectives of this work are to (1) train a model (W2V) for creating word associations (also known as embeddings) using a publicly available dataset (a subreddit on Coronavirus from January 2020 to July 2021, a period where emotions were exacerbated) and open access software (R libraries) able to retrieve meaningful closest terms. (2) Such a W2V model aims to be formally validated using the semantic categorization test by means of an updated and expanded version [33] of the Battig and Montague norm, with 65 categories; for each category, the silhouette coefficient of the model will be computed. As a complementary validation step, the extensive scientific literature is aimed to be included, supporting our findings. (3) We will then run W2V to discover the context for seven specific positive and seven negative emotions recently reported as related to resilience during the COVID-19 pandemic, and (4) such specific context will be supported using related scientific publications.

The article is organized as follows. A literature review is presented in Section 2. Materials and methods are introduced in Section 3. In Section 4, we initially report a descriptive analysis of the sample; we then present the results of our W2V model at three different levels (toy-example analogies, representative terms from a COVID-19 glossary, and resilience related terms) for both the COVID-19 glossary and resilience related terms. We support our findings with extensive scientific literature and then discuss performance using the Battig and Montague evaluation. The discussion and limitations are presented in Section 5. Lastly, in Section 6, we conclude the paper.

2. Related Work

For computer scientists and researchers, social media data are valuable assets for understanding people's sentiments regarding current events, especially those related to events with worldwide impacts, such as the COVID-19 pandemic. Therefore, the classification of these sentiments yields remarkable findings. For example, in one of the earliest related publications, Rajput and colleagues [34] classified (negative, positive, and neutral) tweets based on word-level, bi-gram, and tri-gram frequencies to represent word rates by power law distribution and applied the Python built-in package TextBlob to perform sentiment analysis. Samuel and colleagues [35] proposed machine learning models (naïve Bayes and logistic regression) to categorize sentiment tweets into two classes (positive and negative). Similarly, Aljameel et al. [36] analyzed a large Arabic COVID-19-related tweets dataset, applying uni-gram and bi-gram TF-IDF with SVM, naïve Bayes, and KNN classifiers to enhance accuracy. Muthausami et al. [37] classified the tweets into three classes (positive, neutral, and negative). They utilized different classifiers, such as random forest, SVM, decision tree, naïve Bayes, LogitBoost, and MaxEntropy. More recently, Jalil and colleagues [38] classified positive, negative, and neutral tweets using various feature sets and XGBoost (eXtreme Gradient Boosting) classifier. The authors of Rustam et al. [39] proposed a COVID-19 tweets classification approach based on a decision tree, XGBoost, extra tree classifier (ETC), random forest, and LSTM. Similarly, Dangi et al. [40] proposed a novel approach known as Sentimental Analysis of Twitter social media Data (SATD) based on five different machine learning models (logistic regression, random forest classifier, multinomial NB classifier, support vector machine, and decision tree classifier)

Rahman et al. [41] explored the performance of ensemble machine learning classifiers for sentiment analysis of COVID-19 tweets from the United Kingdom. Es-Sabery et al. [42] applied MapReduce opinion mining for COVID-19-related tweets classification using an enhanced ID3 decision tree classifier.

Basiri et al. [43] presented a model that combines five models such as naïve Bayes support vector machines (NBSVM), FastText, DistilBERT, CNN, and bidirectional gated recurrent unit (BiGRU) on COVID-19 tweets in eight highly affected countries. Ibrahim et al. [44] proposed a hierarchical Twitter sentiment model (HTSM) to show people's opinions in short texts. Bonifazi et al. [45] proposed a novel approach for investigating the COVID-19 discussions on Twitter through a multilayer network-based model. It yielded the identifica-

tion of influential users, which is much more important to analyze and can provide more valuable information.

Naseem et al. [46] correspondingly proposed the use of various pre-trained embedding representations—FastText, GloVe, Word2Vec, and BERT—to extract features from a Twitter dataset. Furthermore, for the classification, they applied deep learning methods Bi-LSTM and several classical machine learning classifiers, such as SVM and naïve Bayes.

Yan et al. [47] reported public sentiment toward COVID-19 vaccines across Canadian cities by analyzing comments on Reddit. In order to identify significant latent topics and classify sentiments in COVID-19-related English comments between January and March 2020, Jelodar et al., examined 563,079 comments from Reddit [48]. Lai et al. [49] analyzed 522 comments from a Reddit Ask Me Anything session about COVID-19. Reddit posts evaluated in this study were manually coded by two authors of this paper.

Pal et al. [50] showed that new knowledge could be captured and tracked using the temporal change in word embeddings from the abstracts of COVID-19 published articles. They found that thromboembolic complications were detected as an emerging theme as of August 2020. A shift toward the symptoms of long COVID complications was observed in March 2021, and neurological complications gained significance in June 2021.

Jha et al. [51] observed that the word2vec model performed better than the GloVe model on a COVID-19 Kaggle dataset. Another point highlighted by this work is that latent information about potential future discoveries was significantly contained in past papers and publications.

Batzdorfer et al. [52] used word embeddings to distinguish non-conspiracy theory content from conspiracy theory-related content and analyzed which element of conspiracy theory content emerged during the pandemic.

Didi et al. [6] proposed a tweets classification approach (negative, positive, and neutral) based on a hybrid word embedding method, combining several widely used techniques, such as TF-IDF, word2vec, Glove, and FastText, to represent posts.

Bhandari et al. [53] proposed a deep learning model with stacked word embeddings to the multi-class classification problem for three and five classes (extremely negative, negative, neutral, extremely positive, and positive). It outperformed the individual static pre-trained embedding representation, classical machine, and deep learning approaches.

To our knowledge, no previous analysis applied word embeddings to extract knowledge from Reddit to provide context about specific emotions involved in psychological resilience during the pandemic. Acute crisis and loss events, disruptions in many facets of life, continuous multi-stress problems, and always-changing conditions made the COVID-19 pandemic a perfect storm of stressors. The rapid spread of COVID-19 during the 2020–2021 period, when emotions were exacerbated [54], created a unique opportunity to extract knowledge about resilience in the face of global adversity, yet to be explored using NLP. We believe that a better understanding of resilience is important in developing strategies to cultivate and promote resilience.

3. Methods

Our research included different sequential phases starting with data collection from publicly available Reddit titles from the R/Coronavirus subreddit, data cleaning using open access R libraries, an initial descriptive analysis of the available data, word2vec model training, the formal model validation using semantic categorization test and visualization using hierarchical clustering and heatmaps. Each of them is described in this section.

3.1. Data Collection

Data from Reddit were obtained via pushshift.io through the pushshift.io API (Pushshift, 2023) [55]. In order to collect and distribute Reddit datasets for research purposes, academics can use Pushshift.io, a website that keeps all publicly accessible Reddit submissions and comments. Pushshift.io has been used in a large number of publications in related

research (e.g., Lama et al. [56]). In this work, the pushshiftr R package [57] was used as a wrapper for the pushshift.io API.

3.2. Data Cleaning

The quanteda R library [58] was used to create the final sample for analysis. The data cleaning process included lemmatization (where the phrases "dog," "dogs," and "dog's" are all changed to "dog"), nonprintable character removal (such as emojis), and basic normalizing (such as removing punctuation and lowercasing all text).

All analyses used are publicly available, anonymized data and comply with Reddit's terms of service, usage rules, and privacy guidelines. They were also carried out with institutional review board clearance from the authors' institutions.

3.3. Descriptive Initial Analysis

For descriptive analysis, we first processed the data into the tidy text format as one token (word) per row. The process of breaking text into tokens is known as tokenization. This one-token-per-row structure differs from how text is commonly kept in current studies (e.g., in a document-term matrix). For tidy text pre-processing, we used the tidytext [59], dplyr [60], ggplot2 [61], and broom [62] R packages.

In order to determine if the frequency of each word is rising or decreasing over time, we fitted a model (logistic regression) using the broom R package. Then, each term has a growth rate (represented by an exponential term) associated with it.

In the Supplementary Materials Figure S1, we present the number of titles per week. We confirm that the distribution is quite similar to the plot provided by the official Reddit statistics presented in Figure 1.

Figure S2 shows the most frequent words (after excluding COVID-19, Coronavirus, and pandemic, which due to their highest frequency, make all other terms not visible if put together in the same plot with all other terms). The top 10 are people, vaccine, China, positive, health, home, masks, world, death, and Trump.

Figure S3 shows the terms with the steepest increase in frequency. The highest one is for Donald Trump, right before the day of the Presidential Election in the United States (3 November 2020), with the highest decrease after it. When visualizing all four sub-plots in Figure S2, shown from left to right and from top to bottom, it can be seen that each of them refers to a specific aspect of this pandemic, each of them with special relevance at different time points: lockdown at the early stage, masks and Trump at intermediate stages, and vaccine increasing steadily until the final stages.

In Figure S4, we present a word cloud created using all the titles containing the term "stress."

3.4. Model Training: Word2vec

We applied the wordVectors [63] R package to train the word2vec model. It runs the original C code for word2vec [12].

A metric of the degree of similarity between two embedding vectors for the two words is provided to measure how similar the two words are. Given two vectors u and v, cosine similarity is defined as follows [12]:

$$CosineSimilarity(u, v) = \frac{u.v}{\|u\|_2 \|v\|_2} = \cos\theta \quad (1)$$

where $u.v$ is the dot product (or inner product) of two vectors, $\|u\|_2$ is the norm (or length) of the vector u, and θ is the angle between u and v.

The cosine distance is defined as the inverse of the cosine similarity; the shorter the cosine distance, the more similar the two vectors (words).

3.5. Model Validation: Semantic Categorization Test

We measured the capacity of the W2V model to represent the semantic categories based on the Battig and Montague category norms, which have been applied by researchers in several fields in over 1600 publications in more than 200 different journals [33]. In this work, we use Van Overschelde's [33] expanded and updated version of the Battig and Montague original norms.

In order to measure how well a word *i* is grouped in relation to the other words in its semantic category, we used the Silhouette Coefficients, *s(i)*, defined as:

$$s(i) = (b(i) - a(i))/max\{a(i),b(i)\}$$

where *a(i)* is the mean distance of word i with all other words within the same category, and *b(i)* is the minimum mean distance of word *i* to any words within another category (i.e., the mean distance to the neighboring category). Therefore, silhouette coefficients measure how close a word is to other words within the same category compared to words of the closest category [64].

3.6. Model Visualization: Hierarchical Clustering and Heatmaps

We used the superheat R package [65] to visualize the word vectors (obtained from Word2vec), highlighting contextual similarity. "The rows and columns are ordered based on a hierarchical clustering and are accompanied by dendrograms describing this hierarchical cluster structure" [65].

4. Results

4.1. Sample Description

We collected all 374,421 titles submitted by 104,351 different Redditors to the r/Coronavirus subreddit between 20 January 2020 and 14 July 2021.

In Figure 2, we show representative examples of the collected titles, the top 3 containing the term "resilience" and the bottom three randomly selected.

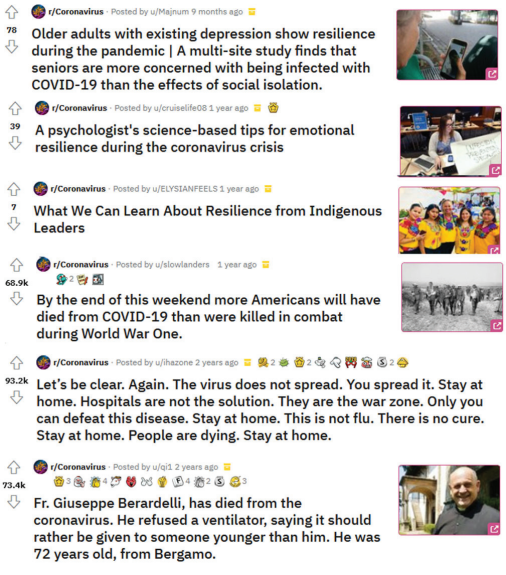


Figure 2. Examples of the collected titles containing the term "resilience" (top 3) and randomly selected (bottom 3).

4.2. A 3-Steps Validation of the Word2vec Embeddings

The train_word2vec function of the wordVectors R package was used to obtain the model (W2V) once the data had been generated. The following settings were used: "vectors = 200, threads = 4, window = 12, iter = 5, negative_samples = 0". These parameters have been applied by the *wordVectors* authors in related research [63].

We performed a three-step validation of W2V as in previous related research [66]. We utilized a subset of the original Mikolov article analogies [12] for the first one.

In NLP, the task of finding a word analogy is represented as "a is to b as c is to ___."

The classic Mikolov example is: king is to man as woman is to ___'—also represented as king – man + woman = ?

The human brain can recognize that the answer is the word 'queen'. However, for a machine to understand this pattern and fill in the blank with the most appropriate word requires a lot of training using a huge corpus (for example, the whole of Wikipedia; in our case, we are using only the obtained 374,421 titles from r/Coronavirus).

Using our obtained model (namely W2V), the example analogy is represented as: W2V("king") − W2V("man") + W2V("woman") = ?

We obtained promising results (as presented in Table 2) for several analogies from previous research [66], for example:

Analogy: brother − sister + husband = ?

Answer: wife (0.5985)

The number in brackets is the cosine distance between the vector embedding for the term 'wife' and the vector that is the result of the operations on the left-hand side of the equation.

Table 2. A subset of analogies from previous research [66] and the obtained results.

Category	Closest Terms (Cosine Distance)
paris − france + italy = ?	rome (0.584), milan (0.510)
brother − sister + husband = ?	wife (0.598)
dad − mom + father = ?	mother (0.546), family (0.569)
she − he + girl = ?	boy (0.375)
his − her + boy = ?	girl (0.570), schoolgirl (0.604)
she − he + mother = ?	father (0.373), husband (0.403)
boy − girl + man = ?	woman (0.553)
doctor − hospital + teacher = ?	school (0.577), teen (0.548)
cnn − news + netflix = ?	film (0.640), movies (0.692)
iphone − apple + android = ?	ios(0.406), tablet (0.4760), app (0.487)
moscow − putin + nyc	Blasio * (0.619), brooklyn (0.581)
young − teen + old	64 (0.633), aged (0.563)

* Bill de Blasio is an American politician serving as the 109th Mayor of New York City since 2014.

As the second step of W2V validation, from a representative list of specific terms related to COVID-19, we run our W2V model on each of them (for example, the term "anosmia") to identify its three closest terms using the following command:

nearest_to(W2V[["anosmia"]],3) = ?

As a result, we obtained the following set of the three closest terms to "anosmia":

{olfactory (0.463); parkinson (0.459); aspirin (0.496)}

In Table 3, we present the closest terms retrieved by our model and their cosine distances to several COVID-19 representative terms of a known COVID-19 glossary [67]. We proceeded through the closest terms and identified related publications and evidence supporting them, noting the high relevance of all the discovered terms in order to demonstrate the capacity of our W2V model to uncover relevant related terms (Table 3).

Table 3. Definitions extracted from COVID-19 Canadian Glossary and our obtained closest terms.

Glossary Term	Glossary Definition	Closest Terms
ards	Acute respiratory distress syndrome	remestemcel (0.364) [68], glucose (0.461) [69], epithelium (0.461) [70], anticoagulant (0.481) [71]
anosmia	The complete or partial loss of the sense of smell.	olfactory, (0.463) [72], parkinson (0.459) [73], aspirin (0.496) [74]
antibody	A protein that is produced in response to the introduction of an antigen in an organism	monoclonal (0.436) [75], regeneron (0.478) [76], serological (0.475) [77], bamlanivimab (0.539) [78]
antiviral	Medication used for treating viral infections	favipiravir (0.341) [79], remdesivir (0.344) [80], heparin (0.379) [81], interferón (0.385) [82], ritonavir (0.435) [83]

In Table 3, we show the closest terms retrieved by our model and their cosine distances to representative definitions from the initial terms of the glossary (terms starting with the 'A' letter). For each of the terms identified by our trained model, we included relevant published scientific literature. For example, the first term in the glossary was "ards" (acute respiratory distress syndrome); our model retrieved Remestemcel, and its cosine distance was 0.364. We referenced Mahendiratta et al. [68] because in their systematic review of Stem cell therapy in COVID-19, results on Remestemcel were recently reported. Similarly, for glucose, we referenced Lazzeri et al. [69] work, where they address the prognostic role of hyperglycemia and glucose variability in COVID-related acute respiratory distress, similarly, for all other terms in Table 3.

As the third step of W2V validation, we identified the closest terms to "resilience." Then we searched for all appearances of "resilience" in all 374,421 titles and identified the titles with the highest upvotes. We present them in Figure 2.

In Figure 2 (top 3 titles), we present the most upvoted titles, which explicitly include the term "resilience." Therefore, we used W2V to search for the closest terms to: resilience appearing in the same context with "older" with "indigenous" and with "tips." The obtained closest terms are presented in Table 4. We went through all the closest obtained terms and identified related publications and evidence remarking on the high relevance of all the identified terms.

Table 4. Resilience-related terms and our obtained closest terms.

Search Term	Closest Terms
resilience	wellbeing (0.569) [84], pessimism (0.611) [85], psychological (0.586) [85]
resilience + tips	mindfulness (0.580) [86], telehealth (0.588) [87], bedtime (0.577) [88], hobbies (0.546) [89]

Table 4. *Cont.*

Search Term	Closest Terms
resilience + older	addiction (0.570) [90], stress (0.588) [91], disability (0.588) [92], resentment (0.598) [93], depressive (0.617) [94]
resilience + indigenous	communities (0.520), tribe (0.565), minority (0.618), dignity (0.618), unequal (0.622), unicef (0.632), disparities (0.624) [95]

For example, as shown in Table 4, for "resilience" and "older," we identified several closest terms and included in Table 4 different publications addressing such aspects, e.g., addiction [90], stress [91], disability [92], resentment [93], and depression [94].

4.3. Semantic Categorization Test

For each of the first 65 semantic categories of the updated version of the Battig and Montague norm [33], we calculated the silhouette coefficients. The complete list of all the terms included in each category as well as distances and silhouette calculations, is presented in Supplementary Materials Table S1. A representative screenshot of the distances from the first eight semantic categories to representative terms is presented in Figure 3. For example, the first semantic category is "1. A precious stone", as detailed in Table S1. It is integrated into four terms (diamond, ruby, gold, and gem). We run our W2V model to calculate the distances from a representative term from each category to all the other terms. Therefore, as shown in Figure 3, the mean distance from the "diamond" term to all other terms in the "1. A precious stone" category is 0.66. Meanwhile, it is 1.01 to the "2. A unit time" category represented by the "hour" term, it is 1.00 to the "3. A relative" category represented by the "mother" term, and so forth. Therefore, Figure 3 represents such distances as a heatmap, with greener values to the closest distances. It can be seen that for each term, for every semantic category, the closest distances are to those terms related to the category where the term belongs, therefore showing encouraging results.

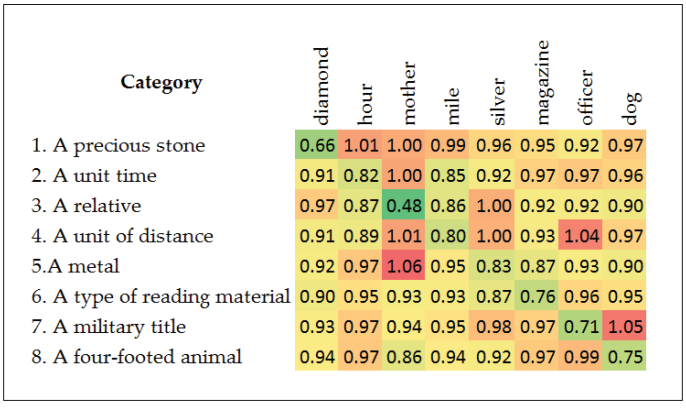


Figure 3. Heatmap representation of the mean distances between the first 8 semantic categories and their representative terms.

Table 5 presents the highest silhouette values calculated in Supplementary Materials Table S1. When analyzing the lower Silhouette scores, we identified remarkable reasons for the miscategorization of the terms. For example, as presented in Figure 3, the mean

distance from the "diamond" term to all other terms in the "1. A precious stone" category is 0.66, but as shown in Table S1, when considering the "51. A type of ship/boat" category, represented by the "cruise" term, such mean distance is 0.55, remarkably lower. A possible explanation for this is the existence of the Diamond Princess Cruise, which is mentioned in some of the Reddit titles used for training our W2V model.

Table 5. Top silhouette values obtained for 10 semantic categories of the updated version of the Battig and Montague norm.

Category	s
29. A sport	0.495
3. A relative	0.329
54. A city	0.233
55. A state	0.231
10. A color	0.169
58. A type of car	0.163
49. A disease	0.154
27. An occupation or profession	0.142
7. A military title	0,139
40. A science	0.137

4.4. Context for Positive and Negative Emotions

In Table 6, we present a list of specific positive emotions (gratitude, compassion, love, relief, hope, calm, and admiration) [30]. We ran our W2V model for each of them and identified several closest terms, providing the context where such emotions took place.

Table 6. Positive emotions and their obtained closest terms.

Search Term	Closest Terms
gratitude	paramedical (0.545), doctors (0.495), appreciation (0.368), selflessly (0.498), tirelessly (0.453), heroes (0.503), honor (0.516), tribute (0.540), hardworking (0.555), flashmovs (0.564)
compassion	dalai (0.684), lama (0.685), empathy (0.657), empathetic (0.662), mindfulness (0.633)
love	share (0.400), enjoy (0.440), friends (0.489), wish (0.519), god (0.526), smile (0.528), constructive (0.555), entertain(0.525)
relief	funds (0.325), aid (0.339), package (0.309), fund (0.341), trillion (0.394), billion (0.414), loan (0.409), liquidity (0.414), payments (0.429), payers (0.397), tax (0.436)
hope	Love (0.423), enjoy (0.477), brightens (0.471), help (0.513), inspire (0.517), smile (0.522), laugh (0.536), humor (0.569), fun (0.573), funny (0.573)
calm	Listen (0.532), sleep (0.435), meditation (0.544), Roads (0.521), streets (0.546), eerie (0.656), emptiness (0.668), scary (0.575), panic (0.567), nerves (0.546), keep (0.654)
admiration	clapping (0.431), clap (0.455), applause (0.455), balconies (0.484), applauding (0.522), windows (0.472), cheering (0.456), frontline (0.466), healthcare (0.532),

Similarly, Table 7 presents the list of negative emotions [30] (anger, loneliness, boredom, fear, anxiety, confusion, sadness) and their closest terms retrieved using W2V.

Table 7. Negative emotions and the obtained closest terms.

Search Term	Closest Terms
anger	frustration (0.471), confusion (0.474), tension (0.555), chaos (0.592), dishonesty (0.630), hostility (0.617), bureaucracy (0.625), fear (0.608), drought (0.579), outcry (0.598), outrage (0.618),
loneliness	profound (0.607), addiction (0.635), neuropsychiatric (0.630), opioid (0.658)
boredom	spotify (0.615), playlists (0.599), song (0.593), halo (0.596), fortnite (0.626), meditation (0.615), illustration (0.631), piano (0.632),
fear	conspiracies (0.587), xenophobia (0.612), racism (0.621), burnout (0.623), starving (0.636), sadness (0.532)
anxiety	stress (0.251), depression (0.431), meditation (0.578), obsessive (0.532), ideation (0.511), cope (0.529), coping (0.514), tips (0.570)
confusion	anger (0.474), frustration (0.546), chaos (0.543), distrust (0.561), tension (0.532), worries (0.522), doubts (0.533)
sadness	Disbelief (0.433), downfall (0.544), dislike (0.541), downvotes (0.541), fear (0.532), boredom (0.533), together (0.544), spinning (0.541)

Figure 4 graphically shows a dendrogram for the closest terms to two positive emotions (hope and gratitude) and two negatives (anger and anxiety) presented as clusters of the most similar closest terms. The darkest the color in the heatmap, the closest are the two terms; therefore, three clear clusters emerge in the heatmap diagonal.

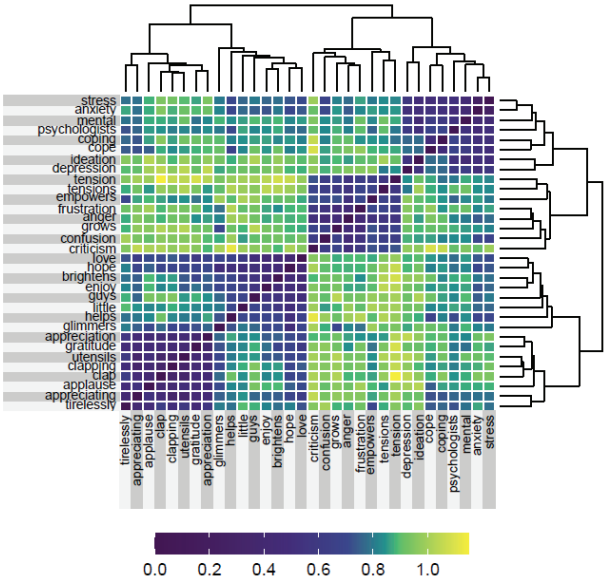


Figure 4. Dendrograms and heatmap for the closest terms to two positive (hope and gratitude) and two negative (anger and anxiety) emotions.

5. Discussion

In this study, we proposed social media (particularly a Reddit subforum) as a connection between word associations (also known as embeddings) and emotion research. Although they both share context as a critical component, to our best knowledge, word em-

beddings have rarely been used in the field of emotion research. Furthermore, COVID-19 created a unique opportunity for doing it.

Therefore, we trained a model for producing word embeddings using a publicly accessible dataset (a Coronavirus subreddit) and open-source tools (R libraries) capable of retrieving relevant content (closest words). This content was formally validated using a standard tool and supported by public evidence (scientific publications), and applied to the discovery of context for seven specific positive and seven negative emotions recently reported as related to resilience during the COVID-19 pandemic.

Our results confirmed our three initial hypotheses: word embeddings may be recovered in sufficient numbers from public domain-specific social media for the embedding to (1) be relevant to offer meaningful context to specific emotions, (2) be verifiable by sound theoretical semantic tests such as the Battig and Montague norm, and (3) be consistent with recent related publications, in spite of working with a relative "small" number of Reddit titles.

In relation to our fourth hypothesis (provide actionable knowledge to on-field specialists), current research reporting on the COVID-19 pandemic concluded that developing a resilient mentality differs depending on whether positive or negative emotions are present. Higher levels of positive emotions are correlated with higher levels of resilience, whereas high levels of negative emotions are associated with lower levels of resilience [30]. We associated seven positive and seven negative emotions to experienced situations. Specialists could therefore promote actions encouraging participation in activities related to positive emotions. For example, as shown in Table 6, "gratitude" and "admiration" were shown by means of activities taking place worldwide. People congregated on balconies while confined to their apartments to acclaim medical personnel working on the front lines, as well as to sing or take part in impromptu flash mobs [96]. Calm and compassion were associated with meditation and mindfulness. Hope was associated with humor, smiling, laughing, fun, and funny.

When analyzing negative emotions, we found racism and xenophobia mainly related to fear. Globally, migrants and minority groups were disproportionately affected by racism and xenophobia linked to COVID-19 [97]. They have an especially negative effect on people who already experience overlapping social, economic, and health-related vulnerabilities. They intensify current patterns of discrimination and unfairness. Minority groups in both the United States and Europe have endured discrimination and hate crimes. [98,99]. Anger was mainly related to frustration, bureaucracy, and confusion as in related research (e.g., Selman et al. [100]); loneliness was associated with addictions, while boredom was related to specific activities to overcome it, such as meditation, illustration, piano, Spotify, playlists or videogames (Halo, Fortnite).

Several recent studies addressed social media (particularly Reddit) during the pandemic. For example, Gozzi et al. [101] analyzed collective responses to media coverage. They performed mixed-methods analysis on web-based news articles, YouTube videos, English user posts and comments on Reddit, and views of Wikipedia pages related to COVID-19. They concluded that "collective attention was mainly driven by media coverage rather than epidemic progression [101]". Compared to other social media platforms, Reddit users were generally more concerned about health, data related to the new disease, and interventions needed to stop its spread [101]. In order to identify significant latent topics and classify sentiments in COVID-19-related English comments between January and March 2020, Jelodar et al., examined 563,079 comments from Reddit [48]. Lai et al. [49] analyzed 522 comments from a Reddit Ask Me Anything session about COVID-19 on 11 March 2020. Most posts addressed symptoms, followed by prevention recommendations. COVID-19 symptoms were also the most requested topic suggested by users for further discussion.

Word2vec has been scarcely used in small corpora. García-Rudolph et al. [66] analyzed 96,314 Reddit comments posted in r/disability from February 2009 to December 2019 by 10,411 Redditors. The highest reported silhouette value after the semantic categorization

test was s = 0.562 for the "3. A relative" category. Meanwhile, in our case, our highest silhouette value was s = 0.495 for the "29. A sport" category. In the "29. A sport" category, their reported silhouette was s = 0.475. Their top six higher silhouette values were reported for the following categories: 3. A relative, 29. A sport, 43. A vegetable, 10. A color, 55. A state and 49. A disease. In our case, the top six silhouette values were reported for 29. A sport, 3. A relative, 54. A city, 55. A state, 10. A color, and 58. A type of car. Therefore, very similar semantic categories yielded the highest silhouette scores for both studies. Nevertheless, in our case, we collected 374,421 titles (not comments) submitted by 104,351 users (ten times more users) to the Reddit/Coronavirus forum during a ten-times shorter period.

In another study applying word2vec in small corpora using the semantic categorization test, Stetten, the study included 37 k and 140 k documents to analyze and disambiguate the content of dreams [102]. This research area addresses questions such as "How do gender, cultural background, and waking life experiences shape the content of dreams?". To our knowledge, no previous work studied Reddit submission titles considering word embeddings in order to expand on the concept of resilience. We offer a tool for identifying terms of interest that can be addressed to practitioners in the field of psychology and social work.

A number of limitations to this study need to be highlighted. The analyzed sample was not meant to be exhaustive or representative of all titles posted by everyone living in any specific region during the period under study. It included all titles from only one of the COVID-19 subreddits; therefore, we did not include data from other subreddits addressing specific COVID-19 aspects (e.g., CovidVaccinated or COVID-19Positive). Nevertheless, r/Coronavirus was by far the subreddit with a higher number of subscribers and posts. It has been the most active subreddit during the period under study (between 20 January 2020 and 14 July 2021). We did not include comments in our analysis. We included only submissions' titles. The length limit in Reddit comments is 40,000 characters, more than 100 times larger than the titles' limit (300 characters). Therefore including comments would involve a different analysis, with different hypotheses, which is left as future work.

The potential impact of the data-cleaning process needs to be mentioned as another limitation, particularly in terms of the context of the text. For example, by removing emojis and other non-printable characters, we might have been removing some contextual information that could be relevant to understanding the sentiments or emotions. For example, Li et al. [103] presented an approach to classify microblog review sentiments that included emojis with an emoji-text-incorporating bi-LSTM (ET-BiLSTM) model. Their results showed that ET-BiLSTM enhances the performance of sentiment classification.

Another aspect of Reddit worth to be analyzed, not included in this study, involves NSFW (Not Safe For Work) posts. This term refers to user-submitted content not suitable to be viewed in public or in professional contexts. The phenomenon of NSFW posts on Reddit has been very little investigated, although it is very common in this social medium [104].

Other relevant factors to mention as limitations to our study include geographic location, spatial trajectory, or the time of day a submission was posted. Such factors, as noted by Padilla et al. [105] and Gore et al. [106], are relevant in social media. Geographic aspects were not analyzed in our study, but Reddit is most popular in the U.S., with American users far outnumbering those from any other country at 54% of Reddit users. After the U.S., the United Kingdom has the second-highest share of data traffic with 8%, while Canada ranks third with 6.4%. Reddit is most popular with young adults aged 25 to 34, who comprise more than half of the site's users. Nevertheless, there are also a large number of middle-aged users on Reddit. Previous studies have found that 33% of users are between the ages of 30 and 49, suggesting that Reddit is a viable platform for reaching both young and middle-aged adults. More than two-thirds of Reddit users are men who are particularly active on the site [107]. Compared to people living in rural areas, urban and suburban residents use Reddit much more frequently. Gozzi et al., also pointed out that

Reddit has developed into a self-referential community, reinforcing the site's propensity to concentrate on its own content rather than outside sources [101].

6. Conclusions

This study opens up interesting opportunities for exploration and discovery using, for the first time, a word2vec model trained with a small Coronavirus dataset of Reddit titles leading to immediate and accurate terms that can be used to expand our knowledge on specific concepts such as resilience, by identifying the context in which they take place. We presented a step forward in developing a tool that can be used by practitioners in the field of psychology or social work for identifying terms of interest describing the context in which specific positive and/or negative emotions related to psychological resilience took place. These may support clinicians in specific situations where individuals can be encouraged to get involved or promote positive emotions related to psychological resilience.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app13116713/s1, Figure S1: Number of titles per week of the r/Coronavirus subreddit; Figure S2: Top 50 most frequent words; Figure S3: Terms with the steepest increase in frequency; Figure S4: Wordcloud of all titles containing the "stress" term; Table S1: Semantic categorization test.

Author Contributions: Methodology: A.G.-R., D.S.-P., J.D.K. and K.C.; Software: A.G.-R., D.S.-P. and K.C.; Validation: E.O. and K.C.; Formal analysis: J.D.K. and A.G.-R.; Investigation: A.G.-R., D.S.-P. and E.O.; Resources: D.F. and E.O.; Data curation: A.G.-R. and D.S.-P.; Writing—original draft: A.G.-R. and D.S.-P.; Writing—review and editing: J.D.K. and E.O.; Visualization: A.G.-R. and D.S.-P.; Supervision: E.O., J.D.K. and D.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by PRECISE4Q Personalized Medicine by Predictive Modelling in Stroke for Better Quality of Life—European Union's Horizon 2020 research and innovation program under grant agreement No. 777107.

Institutional Review Board Statement: All analyses relied on public, anonymized data; adhered to the terms and conditions, terms of use, and privacy policies of Reddit; and were performed under Institutional Review Board approval from the authors' institution.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets generated and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Acknowledgments: Special thanks to Olga Araujo from Institut Guttmann–Documentation department for her continuous support of our bibliography requests.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

API	Application Program Interface
ARDS	Acute Respiratory Distress Syndrome
CVST	Cerebral Venous Sinus System
CBOW	Continuous Bag of Words
EMA	European Medicines Agency
IgM	Immunoglobin M
MHRA	Medicines Healthcare products Regulatory Agency
NLP	Natural Language Processing

References

1. Melton, C.A.; White, B.M.; Davis, R.L.; Bednarczyk, R.A.; Shaban-Nejad, A. Fine-tuned Sentiment Analysis of COVID-19 Vaccine-Related Social Media Data: Comparative Study. *J. Med. Internet Res.* **2022**, *24*, e40408. [CrossRef] [PubMed]
2. Reddit–Dive into Anything. Founded: June 23, 2005, Medford, Massachusetts, United States. Available online: https://www.reddit.com/ (accessed on 19 March 2023).

3. Tsao, S.F.; Chen, H.; Tisseverasinghe, T.; Yang, Y.; Li, L.; Butt, Z.A. What social media told us in the time of COVID-19: A scoping review. *Lancet Digit. Health* **2021**, *3*, e175–e194. [CrossRef] [PubMed]
4. White, B.M.; Melton, C.; Zareie, P.; Davis, R.L.; Bednarczyk, R.A.; Shaban-Nejad, A. Exploring celebrity influence on public attitude towards the COVID-19 pandemic: Social media shared sentiment analysis. *BMJ Health Care Inform.* **2023**, *30*, e100665. [CrossRef] [PubMed]
5. Al-Garadi, M.A.; Yang, Y.C.; Sarker, A. The Role of Natural Language Processing during the COVID-19 Pandemic: Health Applications, Opportunities, and Challenges. *Healthcare* **2022**, *10*, 2270. [CrossRef] [PubMed]
6. Didi, Y.; Walha, A.; Wali, A. COVID-19 Tweets Classification Based on a Hybrid Word Embedding Method. *Big Data Cogn. Comput.* **2022**, *6*, 58. [CrossRef]
7. Parikh, S.; Davoudi, A.; Yu, S.; Giraldo, C.; Schriver, E.; Mowery, D. Lexicon Development for COVID-19-related Concepts Using Open-source Word Embedding Sources: An Intrinsic and Extrinsic Evaluation. *JMIR Med. Inform.* **2021**, *9*, e21679. [CrossRef]
8. Sciandra, A. COVID-19 Outbreak through Tweeters' Words: Monitoring Italian Social Media Communication about COVID-19 with Text Mining and Word Embeddings. In Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020; pp. 1–6. [CrossRef]
9. Levy, O.; Goldberg, Y.; Dagan, I. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Trans. Assoc. Comput. Linguist.* **2015**, *3*, 211–225. [CrossRef]
10. Firth, J.R. A Synopsis of Linguistic Theory 1930–1955. In *Studies in Linguistic Analysis. Special Volume of the Philological Society*; Blackwell: Oxford, UK, 1957; pp. 1–32.
11. Harris, Z.S. *Distributional Structure*; Routledge: New York, NY, USA, 1954.
12. Mikolov, T.; Corrado, G.; Chen, K.; Dean, J. Efficient Estimation of Word Representations in Vector Space. In Proceedings of the International Conference on Learning Representations, Scottsdale, AZ, USA, 2–4 May 2013; pp. 1–12.
13. Greenaway, K.H.; Kalokerinos, E.K.; Williams, L.A. Context is Everything (in Emotion Research). *Soc. Personal. Psychol. Compass* **2018**, *12*, 12393. [CrossRef]
14. Barrett, L.F.; Mesquita, B.; Smith, E.R. The context principle. In *The Mind in Context*; Mesquita, B., Barrett, L.F., Smith, E.R., Eds.; Guilford Press: New York, NY, USA, 2010; pp. 1–22.
15. Ledgerwood, A. Evaluations in their social context: Distance regulates consistency and context dependence. *Soc. Personal. Psychol. Compass* **2014**, *8*, 436–447. [CrossRef]
16. Moskowitz, J.T.; Cheung, E.O.; Freedman, M.; Fernando, C.; Zhang, M.W.; Huffman, J.C.; Addington, E.L. Measuring positive emotion outcomes in positive psychology interventions: A literature review. *Emot. Rev.* **2020**, *13*, 60–73. [CrossRef]
17. Sun, R.; Balabanova, A.; Bajada, C.J.; Liu, Y.; Kriuchok, M.; Voolma, S.; Pavarini, G. Psychological wellbeing during the global COVID-19 outbreak. *PsyArXiv* **2020**. [CrossRef]
18. Welles, B.F.; González-Bailón, S. *The Oxford Handbook of Networked Communication*; Oxford University Press: Oxford, UK, 2020; ISBN 100190460512.
19. Basile, V.; Cauteruccio, F.; Terracina, G. How Dramatic Events Can Affect Emotionality in Social Posting: The Impact of COVID-19 on Reddit. *Future Internet* **2021**, *13*, 29. [CrossRef]
20. Subreddit Stats. 2023. Available online: https://subredditstats.com/ (accessed on 19 March 2023).
21. Subreddit Lists. 2023. Available online: https://redditlist.com/ (accessed on 19 March 2023).
22. Coronavirus Subreddit. Available online: https://www.reddit.com/r/Coronavirus/ (accessed on 19 March 2023).
23. Reddiquette: An Informal Expression of the Values of Many Redditors, as Written by Redditors Themselves. Available online: https://www.reddithelp.com/hc/en-us/articles/205926439 (accessed on 19 March 2023).
24. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of the NIPS'13: 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–8 December 2013; Volume 2, pp. 3111–3119.
25. Wu, G.; Feder, A.; Cohen, H.; Kim, J.J.; Calderon, S.; Charney, D.S.; Mathé, A.A. Understanding resilience. *Front. Behav. Neurosci.* **2013**, *7*, 10. [CrossRef] [PubMed]
26. Rutter, M. Resilience as a dynamic concept. *Dev. Psychopathol.* **2012**, *24*, 335–344. [CrossRef]
27. Newman, R. APA's resilience initiative. *Prof. Psychol. Res. Pract.* **2005**, *36*, 227–229. [CrossRef]
28. Vella, S.; Pai, N. A theoretical review of psychological resilience: Defining resilience and resilience research over the decades. *Arch. Med. Health Sci.* **2019**, *7*, 233–239. [CrossRef]
29. Tariq, H. Measuring Community Disaster Resilience at local levels: An adaptable Resilience Framework. *Int. J. Disaster Risk Reduct.* **2021**, *62*, 102358. [CrossRef]
30. Israelashvili, J. More Positive Emotions During the COVID-19 Pandemic Are Associated with Better Resilience, Especially for Those Experiencing More Negative Emotions. *Front. Psychol.* **2021**, *12*, 648112. [CrossRef]
31. Zhang, Y.; Wang, X.; Lai, S.; He, S.; Liu, K.; Zhao, J.; Lv, X. Ontology Matching with Word Embeddings. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*; NLP-NABD CCL 2014, Lecture Notes in Computer Science; Sun, M., Liu, Y., Zhao, J., Eds.; Springer: Cham, Switzerland, 2014; Volume 8801. [CrossRef]
32. Battig, W.F.; Montague, W.E. Category norms for verbal items in 56 categories: A replication and extension of the Connecticut norms. *J. Exp. Psychol.* **1969**, *80*, 1–46. [CrossRef]

33. Van Overschelde, J.; Rawson, K.; Dunlosky, J. Category norms: An updated and expanded version of the Battig and Montague (1969) norms. *J. Mem. Lang.* **2004**, *50*, 289–335. [CrossRef]
34. Rajput, N.K.; Grover, B.A.; Rathi, V.K. Word frequency and sentiment analysis of twitter messages during coronavirus pandemic. *arXiv* **2020**, arXiv:2004.03925.
35. Samuel, J.; Ali, G.; Rahman, M.; Esawi, E.; Samuel, Y. COVID-19 public sentiment insights and machine learning for tweets classi-fication. *Information* **2020**, *11*, 314. [CrossRef]
36. Aljameel, S.S.; Alabbad, D.A.; Alzahrani, N.A.; Alqarni, S.M.; Alamoudi, F.A.; Babili, L.M.; Aljaafary, S.K.; Alshamrani, F.M. A sentiment analysis approach to predict an individual's awareness of the precautionary procedures to prevent COVID-19 outbreaks in Saudi Arabia. *Int. J. Environ. Res. Public Health* **2021**, *18*, 218. [CrossRef]
37. Muthusami, R.; Bharathi, A.; Saritha, K. COVID-19 outbreak: Tweet based analysis and visualization towards the influence of coronavirus in the world. *Gedrag Organ. Rev.* **2020**, *33*, 8–9.
38. Jalil, Z.; Abbasi, A.; Javed, A.R.; Badruddin Khan, M.; Abul Hasanat, M.H.; Malik, K.M.; Saudagar, A.K.J. COVID-19 Related Sentiment Analysis Using State-of-the-Art Machine Learning and Deep Learning Techniques. *Front. Public Health* **2022**, *9*, 812735. [CrossRef] [PubMed]
39. Rustam, F.; Khalid, M.; Aslam, W.; Rupapara, V.; Mehmood, A.; Choi, G.S. A performance comparison of supervised machine learning models for COVID-19 tweets sentiment analysis. *PLoS ONE* **2021**, *16*, e0245909. [CrossRef]
40. Dangi, D.; Dixit, D.K.; Bhagat, A. Sentiment analysis of COVID-19 social media data through machine learning. *Multimed. Tools Appl.* **2022**, *81*, 42261–42283. [CrossRef]
41. Rahman, M.M.; Islam, M.N. Exploring the Performance of Ensemble Machine Learning Classifiers for Sentiment Analysis of COVID-19 Tweets. In *Sentimental Analysis and Deep Learning*; Shakya, S., Balas, V.E., Kamolphiwong, S., Du, K.L., Eds.; Advances in Intelligent Systems and Computing; Springer: Singapore, 2022; Volume 1408. [CrossRef]
42. Es-Sabery, F.; Es-Sabery, K.; Qadir, J.; Sainz-De-Abajo, B.; Hair, A.; Garcia-Zapirain, B.; De la Torre-Diez, I. A MapReduce opinion mining for COVID-19-related tweets classification using enhanced ID3 decision tree classifier. *IEEE Access* **2021**, *9*, 58706–58739. [CrossRef]
43. Basiri, M.E.; Nemati, S.; Abdar, M.; Asadi, S.; Acharrya, U.R. A novel fusion-based deep learning model for sentiment analysis of COVID-19 tweets. *Knowl.-Based Syst.* **2021**, *228*, 107242. [CrossRef]
44. Ibrahim, F.A.; Hassaballah, M.; Ali, A.A.; Nam, Y.; Ibrahim, A.I. COVID19 outbreak: A hierarchical framework for user sentiment analysis. *Comput. Mater. Contin.* **2022**, *70*, 2507–2524. [CrossRef]
45. Bonifazi, G.; Breve, B.; Cirillo, S.; Corradini, E.; Virgili, L. Investigating the COVID-19 vaccine discussions on Twitter through a multilayer network-based approach. *Inf. Process Manag.* **2022**, *59*, 103095. [CrossRef]
46. Naseem, U.; Razzak, I.; Khushi, M.; Eklund, P.W.; Kim, J. Covidsenti: A large-scale benchmark Twitter data set for COVID-19 sentiment analysis. *IEEE Trans. Comput. Soc. Syst.* **2021**, *8*, 1003–1015. [CrossRef] [PubMed]
47. Yan, C.; Law, M.; Nguyen, S.; Cheung, J.; Kong, J. Comparing public sentiment toward COVID-19 vaccines across Canadian cities: Analysis of comments on reddit. *J. Med. Internet Res.* **2021**, *23*, e32685. [CrossRef] [PubMed]
48. Jelodar, H.; Wang, Y.; Orji, R.; Huang, S. Deep Sentiment classification and topic discovery on novel coronavirus or COVID-19 online discussions: NLP using LSTM recurrent neural network approach. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 2733–2742. [CrossRef]
49. Lai, D.; Wang, D.; Calvano, J.; Raja, A.S.; He, S. Addressing immediate public coronavirus (COVID-19) concerns through social media: Utilizing Reddit's AMA as a framework for public engagement with science. *PLoS ONE* **2020**, *15*, e0240326. [CrossRef]
50. Pal, R.; Chopra, H.; Awasthi, R.; Bandhey, H.; Nagori, A.; Sethi, T. Predicting Emerging Themes in Rapidly Expanding COVID-19 Literature with Unsupervised Word Embeddings and Machine Learning: Evidence-Based Study. *J. Med. Internet Res.* **2022**, *24*, e34067. [CrossRef] [PubMed]
51. Jha, R.A.; Ananthanarayana, V.S. Gaining Actionable Insights in COVID-19 Dataset Using Word Embeddings. In *Pattern Recognition and Data Analysis with Applications. Lecture Notes in Electrical Engineering*; Gupta, D., Goswami, R.S., Banerjee, S., Tanveer, M., Pachori, R.B., Eds.; Springer: Singapore, 2022; Volume 888. [CrossRef]
52. Batzdorfer, V.; Steinmetz, H.; Biella, M.; Alizadeh, M. Conspiracy theories on Twitter: Emerging motifs and temporal dynamics during the COVID-19 pandemic. *Int. J. Data Sci. Anal.* **2022**, *13*, 315–333. [CrossRef]
53. Bhandari, A.; Kumar, V.; Thien Huong, P.; Thanh, D. Sentiment Analysis of COVID-19 Tweets: Leveraging Stacked Word Embedding Representation for Identifying Distinct Classes Within a Sentiment. In *Artificial Intelligence in Data and Big Data Processing*; ICABDE 2021, Lecture Notes on Data Engineering and Communications Technologies; Dang, N.H.T., Zhang, Y.D., Tavares, J.M.R.S., Chen, B.H., Eds.; Springer: Cham, Switzerland, 2022; Volume 124. [CrossRef]
54. Chan, A.Y.; Ting, C.; Chan, L.G.; Hildon, Z.J.L. "The emotions were like a roller-coaster": A qualitative analysis of e-diary data on healthcare worker resilience and adaptation during the COVID-19 outbreak in Singapore. *Hum. Resour. Health* **2022**, *20*, 60. [CrossRef]
55. Pushshift Reddit API Documentation. Available online: https://github.com/pushshift/api (accessed on 19 March 2023).
56. Lama, Y.; Hu, D.; Jamison, A.; Quinn, S.C.; Broniatowski, D.A. Characterizing Trends in Human Papillomavirus Vaccine Discourse on Reddit (2007–2015): An Observational Study. *JMIR Public Health Surveill.* **2019**, *5*, e12480. [CrossRef]
57. Pushshiftr: An R Package for Connection to the Pushshift.io API. Available online: https://github.com/dashstander/pushshiftr (accessed on 19 March 2023).

58. Benoit, K.; Watanabe, K.; Wang, H.; Nulty, P.; Obeng, A.; Müller, S.; Matsuo, A. quanteda: An R package for the quantitative analysis of textual data. *J. Open Source Softw.* **2018**, *3*, 774. [CrossRef]
59. Silge, J.; Robinson, D. tidytext: Text Mining and Analysis Using Tidy Data Principles in R. *J. Open Source Softw.* **2016**, *1*, 37. [CrossRef]
60. dplyr: A Grammar of Data Manipulation. Available online: https://cran.r-project.org/web/packages/dplyr/index.html (accessed on 19 March 2023).
61. ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics. Available online: https://cran.r-project.org/web/packages/ggplot2/ (accessed on 19 March 2023).
62. broom: Convert Statistical Objects into Tidy Tibbles. Available online: https://cran.r-project.org/web/packages/broom/index.html (accessed on 19 March 2023).
63. wordVectors: An R Package for Building and Exploring Word Embedding Models. Available online: https://github.com/bmschmidt/wordVectors (accessed on 19 March 2023).
64. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [CrossRef]
65. Barter, R.L.; Yu, B. Superheat: An R package for creating beautiful and extendable heatmaps for visualizing complex data. *J. Comput. Graph. Stat.* **2018**, *27*, 910–922. [CrossRef]
66. García-Rudolph, A.; Saurí, J.; Cegarra, B.; Bernabeu Guitart, M. Discovering the Context of People with Disabilities: Semantic Categorization Test and Environmental Factors Mapping of Word Embeddings from Reddit. *JMIR Med. Inform.* **2020**, *8*, e17903. [CrossRef]
67. The Official Website of the Government of Canada. Available online: https://www.btb.termiumplus.gc.ca/publications/covid19-eng.html (accessed on 19 March 2023).
68. Mahendiratta, S.; Bansal, S.; Sarma, P.; Kumar, H.; Choudhary, G.; Kumar, S.; Prakash, A.; Sehgal, R.; Medhi, B. Stem cell therapy in COVID-19: Pooled evidence from SARS-CoV-2, SARS-CoV, MERS-CoV and ARDS: A systematic review. *Biomed. Pharma-cother.* **2021**, *137*, 111300. [CrossRef] [PubMed]
69. Lazzeri, C.; Bonizzoli, M.; Batacchi, S.; Di Valvasone, S.; Chiostri, M.; Peris, A. The prognostic role of hyperglycemia and glucose variability in covid-related acute respiratory distress Syndrome. *Diabetes Res. Clin. Pract.* **2021**, *175*, 108789. [CrossRef] [PubMed]
70. Chilosi, M.; Poletti, V.; Ravaglia, C.; Rossi, G.; Dubini, A.; Piciucchi, S.; Pedica, F.; Bronte, V.; Pizzolo, G.; Martignoni, G.; et al. The pathogenic role of epithelial and endothelial cells in early-phase COVID-19 pneumonia: Victims and partners in crime. *Mod. Pathol.* **2021**, *34*, 1444–1455. [CrossRef]
71. Helms, J.; Severac, F.; Merdji, H.; Schenck, M.; Clere-Jehl, R.; Baldacini, M.; Ohana, M.; Grunebaum, L.; Castelain, V.; Anglés-Cano, E.; et al. Higher anticoagulation targets and risk of thrombotic events in severe COVID-19 patients: Bi-center cohort study. *Ann. Intensive Care* **2021**, *11*, 14. [CrossRef]
72. Chang, C.C.; Yang, M.H.; Chang, S.M.; Hsieh, Y.J.; Lee, C.H.; Chen, Y.A.; Yuan, C.H.; Chen, Y.L.; Ho, S.Y.; Tyan, Y.C. Clinical significance of olfactory dysfunction in patients of COVID-19. *J Chin. Med. Assoc.* **2021**, *84*, 682–689. [CrossRef]
73. Rethinavel, H.S.; Ravichandran, S.; Radhakrishnan, R.K.; Kandasamy, M. COVID-19 and Parkinson's disease: Defects in neurogenesis as the potential cause of olfactory system impairments and anosmia. *J. Chem. Neuroanat.* **2021**, *115*, 101965. [CrossRef]
74. Buchheit, K.; Bensko, J.C.; Lewis, E.; Gakpo, D.; Laidlaw, T.M. The importance of timely diagnosis of aspirin-exacerbated respiratory disease for patient health and safety. *World J. Otorhinolaryngol. Head Neck Surg.* **2020**, *6*, 203–206. [CrossRef] [PubMed]
75. Vandergaast, R.; Carey, T.; Reiter, S.; Lathrum, C.; Lech, P.; Gnanadurai, C.; Haselton, M.; Buehler, J.; Narjari, R.; Schnebeck, L.; et al. IMMUNO-COV v2.0: Development and Validation of a High-Throughput Clinical Assay for Measuring SARS-CoV-2-Neutralizing Antibody Titers. *mSphere* **2021**, *6*, e0017021. [CrossRef] [PubMed]
76. Baum, A.; Kyratsous, C.A. SARS-CoV-2 spike therapeutic antibodies in the age of variants. *J. Exp. Med.* **2021**, *218*, e20210198. [CrossRef]
77. Calitri, C.; Fantone, F.; Benetti, S.; Lupica, M.M.; Ignaccolo, M.G.; Banino, E.; Viano, A.; Pace, M.; Castella, A.; Gaido, F.; et al. Long-term clinical and serological follow-up of paediatric patients infected by SARS-CoV-2. *Infez Med.* **2021**, *29*, 216–223. [PubMed]
78. Kutzler, H.L.; Kuzaro, H.A.; Serrano, O.K.; Feingold, A.; Morgan, G.; Cheema, F. Initial Experience of Bamlanivimab Monotherapy Use in Solid Organ Transplant Recipients. *Transpl. Infect. Dis.* **2021**, *23*, e13662. [CrossRef] [PubMed]
79. Wadaa-Allah, A.; Emhamed, M.S.; Sadeq, M.A.; Ben Hadj Dahman, N.; Ullah, I.; Farrag, N.S.; Negida, A. Efficacy of the current investigational drugs for the treatment of COVID-19: A scoping review. *Ann. Med.* **2021**, *53*, 318–334. [CrossRef]
80. Hu, Y.; Meng, X.; Zhang, F.; Xiang, Y.; Wang, J. The in vitro antiviral activity of lactoferrin against common human coronaviruses and SARS-CoV-2 is mediated by targeting the heparan sulfate co-receptor. *Emerg. Microbes Infect.* **2021**, *10*, 317–330. [CrossRef]
81. Vergori, A.; Lorenzini, P.; Cozzi-Lepri, A.; Donno, D.R.; Gualano, G.; Nicastri, E.; Iacomi, F.; Marchioni, L.; Campioni, P.; Schininà, V.; et al. Prophylactic heparin and risk of orotracheal intubation or death in patients with mild or moderate COVID-19 pneumonia. *Sci. Rep.* **2021**, *11*, 11334. [CrossRef]
82. Li, C.; Luo, F.; Liu, C.; Xiong, N.; Xu, Z.; Zhang, W.; Yang, M.; Wang, Y.; Liu, D.; Yu, C.; et al. Effect of a genetically engineered interferon-alpha versus traditional interferon-alpha in the treatment of moderate-to-severe COVID-19: A randomised clinical trial. *Ann. Med.* **2021**, *53*, 391–401. [CrossRef]

83. Daoud, S.; Alabed, S.J.; Dahabiyeh, L.A. Identification of potential COVID-19 main protease inhibitors using structure-based pharmacophore approach, molecular docking and repurposing studies. *Acta Pharm.* **2021**, *71*, 163–174. [CrossRef]
84. Liu, Y.; Cooper, C.L.; Tarba, S.Y. Resilience, wellbeing and HRM: A multidisciplinary perspective. *Int. J. Hum. Resour. Manag.* **2019**, *30*, 1227–1238. [CrossRef]
85. Brog, N.A.; Hegy, J.K.; Berger, T.; Znoj, H. An internet-based self-help intervention for people with psychological distress due to COVID-19: Study protocol for a randomized controlled trial. *Trials* **2021**, *22*, 171. [CrossRef] [PubMed]
86. Park, C.L.; Finkelstein-Fox, L.; Russell, B.S.; Fendrich, M.; Hutchison, M.; Becker, J. Psychological resilience early in the COVID-19 pandemic: Stressors, resources, and coping strategies in a national sample of Americans. *Am. Psychol.* **2021**, *76*, 715–728. [CrossRef]
87. Ameis, S.H.; Lai, M.C.; Mulsant, B.H.; Szatmari, P. Coping, fostering resilience, and driving care innovation for autistic people and their families during the COVID-19 pandemic and beyond. *Mol. Autism* **2020**, *11*, 61. [CrossRef]
88. Tafoya, S.A.; Aldrete-Cortez, V.; Ortiz, S.; Fouilloux, C.; Flores, F.; Monterrosas, A.M. Resilience, sleep quality and morningness as mediators of vulnerability to depression in medical students with sleep pattern alterations. *Chronobiol. Int.* **2019**, *36*, 381–391. [CrossRef] [PubMed]
89. Ungar, M.; Ghazinour, M.; Richter, J. Annual Research Review: What is resilience within the social ecology of human development? *J. Child Psychol. Psychiatry* **2013**, *54*, 348–366. [CrossRef]
90. Yang, C.; Zhou, Y.; Xia, M. How Resilience Promotes Mental Health of Patients with DSM-5 Substance Use Disorder? The Mediation Roles of Positive Affect, Self-Esteem, and Perceived Social Support. *Front. Psychiatry* **2020**, *11*, 588968. [CrossRef]
91. Sterina, E.; Hermida, A.P.; Gerberi, D.J.; Lapid, M.I. Emotional Resilience of Older Adults during COVID-19: A Systematic Review of Studies of Stress and Well-Being. *Clin. Gerontol.* **2021**, *45*, 4–19. [CrossRef]
92. Buchman, A.S.; Yu, L.; Oveisgharan, S.; Petyuk, V.A.; Tasaki, S.; Gaiteri, C.; Wilson, R.S.; Grodstein, F.; Schneider, J.A.; Klein, H.U.; et al. Cortical proteins may provide motor resilience in older adults. *Sci. Rep.* **2021**, *11*, 11311. [CrossRef]
93. Koerner, S.S.; Shirai, Y. Latina/o Family Caregivers' Reactions to Limited Help from Relatives: From Frustration to Resilience. *J. Fam. Nurs.* **2019**, *25*, 590–609. [CrossRef]
94. Jané-Llopis, E.; Anderson, P.; Segura, L.; Zabaleta, E.; Muñoz, R.; Ruiz, G.; Rehm, J.; Cabezas, C.; Colom, J. Mental ill-health during COVID-19 confinement. *BMC Psychiatry* **2021**, *21*, 194. [CrossRef] [PubMed]
95. Brant-Birioukov, K. COVID-19 and In(di)genuity: Lessons from Indigenous resilience, adaptation, and innovation in times of crisis. *Prospects* **2021**, *51*, 247–259. [CrossRef] [PubMed]
96. Catungal, J.P. Essential workers and the cultural politics of appreciation: Sonic, visual and mediated geographies of public gratitude in the time of COVID-19. *Cult. Geogr.* **2021**, *28*, 403–408. [CrossRef]
97. Elias, A.; Ben, J.; Mansouri, F.; Paradies, Y. Racism and nationalism during and beyond the COVID-19 pandemic. *Ethn. Racial Stud.* **2021**, *44*, 783–793. [CrossRef]
98. Croucher, S.M.; Nguyen, T.; Rahmani, D. Prejudice toward Asian Americans in the Covid-19 Pandemic: The Effects of Social Media use in the United States. *Front. Commun.* **2020**, *5*, 39. [CrossRef]
99. Devakumar, D.; Shannon, G.; Bhopal, S.S.; Abubakar, I. Racism and Discrimination in COVID-19 Responses. *Lancet* **2020**, *395*, 1194. [CrossRef]
100. Selman, L.E.; Chamberlain, C.; Sowden, R.; Chao, D.; Selman, D.; Taubert, M.; Braude, P. Sadness, despair and anger when a patient dies alone from COVID-19: A thematic content analysis of Twitter data from bereaved family members and friends. *Palliat. Med.* **2021**, *35*, 1267–1276. [CrossRef]
101. Gozzi, N.; Tizzani, M.; Starnini, M.; Ciulla, F.; Paolotti, D.; Panisson, A.; Perra, N. Collective response to media coverage of the COVID-19 pandemic on Reddit and Wikipedia: Mixed-methods analysis. *J. Med. Internet Res.* **2020**, *22*, e21597. [CrossRef]
102. Stetten, N.E.; LeBeau, K.; Aguirre, M.A.; Vogt, A.B.; Quintana, J.R.; Jennings, A.R.; Hart, M. Analyzing the Communication Interchange of Individuals with Disabilities Utilizing Facebook, Discussion Forums, and Chat Rooms: Qualitative Content Analysis of Online Disabilities Support Groups. *JMIR Rehabil. Assist. Technol.* **2019**, *6*, e12667. [CrossRef]
103. Li, X.; Zhang, J.; Du, Y.; Zhu, J.; Fan, Y.; Chen, X. A Novel Deep Learning-based Sentiment Analysis Method Enhanced with Emojis in Microblog Social Networks. *Enterp. Inf. Syst.* **2022**, *17*, 2037160. [CrossRef]
104. Corradini, E.; Nocera, A.; Ursino, D.; Virgili, L. Investigating the phenomenon of NSFW posts in Reddit. *Inf. Sci.* **2021**, *566*, 140–164. [CrossRef]
105. Padilla, J.; Kavak, H.; Lynch, C.; Gore, R.; Diallo, S. Temporal and spatiotemporal investigation of tourist attraction visit sentiment on Twitter. *PLoS ONE* **2018**, *13*, e0198857. [CrossRef] [PubMed]
106. Gore, R.; Diallo, S.; Padilla, J. You Are What You Tweet: Connecting the Geographic Variation in America's Obesity Rate to Twitter Content. *PLoS ONE* **2015**, *10*, e0133505. [CrossRef] [PubMed]
107. Reddit's 2020 Year in Review. Available online: https://redditblog.com/2020/12/08/reddits-2020-year-in-review/ (accessed on 20 March 2023).

applied sciences

MDPI

Article

Predicting Consumer Personalities from What They Say

Hsiu-Yuan Tsao [1], Ching-Chang Lin [2,*], Hui-Yi Lo [1] and Ruei-Shan Lu [3]

[1] Department of Marketing, National Chung Hsing University, Taichung City 402, Taiwan; jodytsao@dragon.nchu.edu.tw (H.-Y.T.)
[2] Department of Business Administration, Taipei City University of Science and Technology, Taipei City 112, Taiwan
[3] Department of Management Information System, Takming University of Science and Technology, Taipei City 114, Taiwan
* Correspondence: cclin@ba.tpcu.edu.tw

Abstract: This study mapped personality based on the newly proposed extraction method from consumers' textual data and revealed the relevance (attention) and polarity (affection) of words associated with a specific personality trait. Furthermore, we illustrate how unique words are used to predict a consumer's behavior associated with certain personality traits. In this study, we employed the scales of the Kaggle MBTI Personality dataset to examine the methodology's effectiveness, extract the personality traits from the textual data into features, and map them into the traits/dimensions of the existing scale. Based on the results obtained in this study, we assert that using the TF-IDF algorithm is a good way to generate a custom dictionary. Furthermore, sentiment scoring with an AI-empowered machine learning algorithm provides useful data to filter and validate more coherent words to understand and, thus, communicate a particular aspect of personality. Finally, we proposed that four situations involving the interaction between attention (frequency) and affection (sentiment) allow us to better understand the consumer and how to use the feature words in terms of the interaction between attention (TF-IDF score) and affection (sentiment score).

Keywords: personality traits; sentiment analysis; text analytics; machine learning; MBTI

Citation: Tsao, H.-Y.; Lin, C.-C.; Lo, H.-Y.; Lu, R.-S. Predicting Consumer Personalities from What They Say. *Appl. Sci.* **2023**, *13*, 6148. https://doi.org/10.3390/app13106148

Academic Editor: Andrea Prati

Received: 28 February 2023
Revised: 12 May 2023
Accepted: 15 May 2023
Published: 17 May 2023

1. Introduction

An awareness of the personalities of those we interact with is beneficial because psychographic segmentation can increase the effectiveness of advertising, promotion, and other marketing activities and improve the measurement of job performance and related functions [1]. Nevertheless, most people assess customers' personality traits using psychological tests. The most widely used measures are the Big 5 model [2] and the MBTI model [3], the latter being a time- and energy-consuming method. Unfortunately, consumers can be reluctant to fill out tedious surveys and, instead, use social media, blogs, or comment threads to post text related to their interests, hobbies, lifestyles, and opinions.

Psychological research suggests that certain personality traits can correlate with linguistic behavior [4]. Furthermore, the automatic detection of personality traits from written messages has attracted significant attention from computational linguists and natural language processors [5]. Term frequency–inverse document frequency (TF-IDF) is a weighting scheme intended to measure how important a word is to a specific document (in our case, a user review) within a collection (or corpus) of documents. This scheme is widely used for information retrieval and summarization. TF-IDF can determine a word's importance by weighing its frequency within a particular document [6]. The highest-scoring words in a document are the most relevant to that document, more so than any other document [7]. Therefore, any personality trait can be regarded as a document. When all the personality traits undergo TF-IDF vectorization, those scores can then be used to classify a user's personality by having each document manually labeled with the aid of a psychology expert

or via a psychological test (such as the well-known MBTI). Therefore, TF-IDF, along with supervised data (via an expert or a psychological test), can provide a score for clusters of words (feature words) that are highly associated with a personality trait.

Past research has been devoted to automatic personality detection via TF-IDF and a machine learning algorithm. However, developing the predictive power of machine learning models that use the same features to predict consumer personality via textual data requires more exploration [1]. In other words, the stream of research devoted to the automatic detection of personality has focused on improving the efficiency and accuracy of personality prediction. However, the intrinsic concept of those features has not been sufficiently explored, and little is known about words that individuals with specific personalities use. Thus, one of the crucial research questions in this study is how to further extract and validate words that might reflect coherent aspects of personality.

Topic modeling, such as LDA, is a popular method for automatically categorizing words reflecting specific personality traits to explore which words a specific personality commonly uses. However, unsupervised topic models, e.g., LDA, often generate incoherent aspects [8]. Furthermore, these existing methods extract many aspects that are not relevant to the domain of interest. Scale-directed text analysis (SDTA) is a new method for generating custom dictionaries for any construct. It can even generate more valid words from constructs; however, the method relies heavily on knowledgeable oversight in the building process [9]. Therefore, it is worthwhile to explore more automatic semi-supervised approaches to develop sound techniques for automatic word extraction to identify consumers' personalities. In addition, the industry needs a rapid automatic dictionary generation method for each construct as well.

First, this study will attempt to extract words based on the TF-IDF scores to generate a dictionary of customer personality traits, as most past research has created. Second, we argue that the core technique of TF-IDF is to count the frequency of words, which is based on the extent of attention rather than the extent of affection (preference or valence) [10]. Hence, this study will consider words associated with a specific aspect and apply sentiment analysis to examine sentences that include words necessary to obtain the affection of an aspect rather than adopting TF-IDF scores to predict or compare the questionnaire ratings [8,11,12]. Previous research has utilized TF-IDF scores alone to compare or predict the questionnaire ratings or for manual labeling, which is not equivalent [9,13]. The main reason for this is that, besides focusing on attention, we also considered affection to be equivalent, comparing it to the results of a questionnaire or psychological test.

Second, the study will employ a sentiment score of featured words instead of only a TF-IDF score to predict the questionnaire ratings. Hence, the other crucial research question is the following. For those sentences of feature words relevant to personality traits, their sentiment score could be an effective source of information to filter and validate more coherent words to understand and, thus, communicate a particular aspect of personality. In other words, we need to identify feature words and the sentiment of the word associated with a personality trait.

Finally, we adopt the strategic analysis grid of FTTA (From Text to Action), which is an analysis framework based on an aspect to discover four interactions of attention (frequency) and affection (sentiment) [10] to further explore how consumers with specific personality traits use those featured words in term of the interaction of attention (TF-IDF score) and affection (sentiment score). The final research question in this study is whether people with specific personality traits intensively use specific feature words positively or negatively.

2. Research Methodology

As we mentioned in the Introduction, the personalities of those we interact with are beneficial because psychographic segmentation can increase the effectiveness of advertising, promotion, and other marketing activities [1]. Past research suggests that by taking advantage of insights into psychological factors, marketers can more effectively attract buyers through emotional involvement at the expense of functionality [14]. Additionally,

the consumer-perceived price also varies depending on the psychological traits of each individual [15]. As for automatic personality detection via algorithm and AI technology, some research provided evidence on improving advertising defectiveness [16,17]. Therefore, we attempted to adopt the scales of the Kaggle MBTI Personality dataset to examine the methodology's effectiveness, extract the personality traits from the textual data into features, and map them into the traits/dimensions of the existing scale to better understand what kinds of words are more intensively used for consumers with specific personality traits. The results should be useful for one-to-one advertising message communication.

2.1. How The Outcome Variable (MBTI) Is Transformed and Used

First, we employed the well-known scales of the Kaggle MBTI Personality dataset to examine the methodology's effectiveness at extracting the personality traits from the textual data into features and mapping them onto the traits/dimensions of the existing scale. This dataset contained over 8600 rows of data. Each row listed a person's type (the person's four-letter MBTI code/type) and the last 50 items they posted. The data were collected through the Personality Cafe forum (https://www.personalitycafe.com/, accessed on 15 December 2022). A sample of the dataset is shown in Table 1.

Table 1. Sample data of Personality Cafe forum dataset.

Type	Post
INFJ	http://www.youtube.com/watch?v=qsXHcwe3krw||| http://41.media.tumblr.com/tumblr_lfouy03PMA1qa1rooo1_500.jpg|||enfp and intj moments https://www.youtube.com/watch?v=iz7lE1g4XM4
ENTP	I'm finding the lack of me in these posts very alarming.|||Sex can be boring if it's in the same position often. For example, me and my girlfriend are currently in an environment where we have to creatively use cowgirl and missionary. There isn't enough...|||Giving new meaning to 'Game' theory.|||
INTP	Good one _____ https://www.youtube.com/watch?v=fHiGbolFFGw|||Of course, to which I say I know; that's my blessing and my curse.|||Does being absolutely positive that you and your best friend could be an amazing couple count? If so, than yes. Or it's more I could be madly in love in case I reconciled my feelings.

The Personality Cafe forum provides a large selection of people and their MBTI personality types, as well as what they have written. The dataset originated from the Personality Cafe forum in 2017, and its posts are predominantly in English, with an approximate corpus of 11.2 million words in more than 420,000 labelled points. Each row represents the last 50 posts of each user. Several studies exploring the MBTI personality adopted the Personality Cafe dataset to examine textual messages and personality traits. Most of the results indicated that using a dataset with an expert labelling the personality traits seems to be effective. Hence, we decided to utilize the dataset for this study.

The Myers–Briggs Type Indicator (MBTI) is a personality indicator that was developed based on Carl Jung's model. The MBTI assesses 16 different personality types (INTJ, INTP, ENTJ, ENTP, INFJ, INFP, ENFJ, ENFP, ISTJ, ISFJ, ESTJ, ESFJ, ISTP, ISFP, ESTP, and ESFP). They all differ in their characteristics and must be treated differently [3]. Each personality type (listed in Table 2 below) reflects a unique human psychological archetype.

Table 2. The definition of dimension and construct for MBTI personality traits.

Dimension	Construct	Definition
Mind	Introvert (I) or Extrovert (E)	shows how an individual interacts with others.
Information	Intuition (N) or Sensing (S)	shows how an individual sees the world and processes information.
Decision	Thinking (T) or Feeling (F)	shows how an individual makes decisions and copes with their emotions.
Structure	Judging (J) or Perceiving (P)	reflects an individual's approach to work, making decisions, and planning

Therefore, we collected raw data that could be arranged and scored to look like the figure below.

No. is the sequence number of the subjects; Type is the category of the MBTI personality; and mind, information, decision, and structure are the dimensions of the MBTI personality. The value of 0 for mind indicates the trait of an introvert, and 1 indicates the trait of an extrovert. Please refer to Tables 3 and 4 for the operational definitions of the other dimensions, constructs, and sample data.

Table 3. The sample data of the category of four dimensions of MBTI personality traits.

No	Type	Post	Mind	Information	Decision	Structure
1	ENTJ	I was referring to in every careers always a good memory is required, but	1	0	1	1
2	INTJ	I be well, but I feel different psychically and I like it. I'm sure some of	0	0	1	1
3	INTJ	Hell is other people INTJs are often portrayed as villains due to a lack of	0	0	1	1

Table 4. The operational definition of four dimensions of MBTI personality traits.

Dimension	Construct	Indicator
Mind	Introvert (I)	Value of mind = 0
	Extrovert (E)	Value of mind = 1
Information	Intuition (N)	Value of information = 0
	Sensing (S)	Value of information = 1
Decision	Thinking (T)	Value of decision = 0
	or Feeling (F)	Value of decision = 1
Structure	Judging (J)	Value of structure = 0
	Perceiving (P)	Value of structure = 1

2.2. Generation of a Custom Dictionary for the Construct

In this study, we attempted to extract feature words based on TF-IDF scores to generate a customer dictionary of construct/traits, as has been carried out in previous research. TF-IDF (term frequency–inverse document frequency) is a statistical measure that evaluates how relevant a word is to a document in a collection of documents [7]. This evaluation is performed by multiplying two metrics: (1) how many times a word appears in a document and (2) the inverse document frequency across a set of documents. The higher the score, the more relevant that word will be in that particular document but not in other documents.

Thus, the obtained MBTI scores indicated the positive or negative dimension of MTBI regarding the mind, information, decision, and structure. We classified those written texts as mind (1), mind (0), information (1), information (0), decision (1), decision (0), structure (1), and structure (0), respectively, and then calculated each word of TF-IDF. We filtered the higher and expected numbers of words to obtain more than 100 words for each construct (please refer to Table 5). Subsequently, we obtained the sample feature words for each dimension, as Table 6 shows (The programming language R provided the package superml (https://www.rdocumentation.org/packages/superml/versions/0.5.5, accessed on 20 December 2022) to easily obtain the score the TF-IDF).

Table 5. The sample value of TF-IDF feature words of MBTI personality traits.

Mind			Information			Decision			Structure		
Construct	**Word**	**TF-IDF**	**Construct**	**Word**	**TF-IDF**	**Construct**	**Word**	**TF-IDF**	**Construct**	**Word**	**TF-IDF**
mind(0)	contains	2.29×10^{-5}	information(0)	trump	4.40×10^{-5}	decision(0)	rhubarb	1.79×10^{-5}	structure(0)	believe	1.68×10^{-5}
mind(0)	dirt	1.98×10^{-5}	information(0)	hyper	3.02×10^{-5}	decision(0)	sm	1.49×10^{-5}	structure(0)	fap	1.28×10^{-5}
mind(0)	rigid	1.90×10^{-5}	information(0)	rings	2.83×10^{-5}	decision(0)	empathy	1.20×10^{-5}	structure(0)	stoned	1.28×10^{-5}
mind(0)	accomplishing	1.60×10^{-5}	information(0)	carried	2.58×10^{-5}	decision(0)	sighs	1.10×10^{-5}	structure(0)	lowered	1.08×10^{-5}
mind(0)	composition	1.60×10^{-5}	information(0)	collective	2.45×10^{-5}	decision(0)	snuggles	1.10×10^{-5}	structure(0)	luna	1.08×10^{-5}
mind(0)	socialism	1.52×10^{-5}	information(0)	hopeful	2.39×10^{-5}	decision(0)	alienated	9.96×10^{-6}	structure(0)	breasts	9.86×10^{-6}
mind(0)	buildings	1.45×10^{-5}	information(0)	atheism	2.33×10^{-5}	decision(0)	cheering	9.96×10^{-6}	structure(0)	devil	9.86×10^{-6}

Table 6. The sample feature words of each dimension of MBTI personality traits.

Mind(0)	**Mind(1)**	**Information(0)**	**Information(1)**	**Decision(0)**	**Decision(1)**	**Structure(0)**	**Structure(1)**
meditate	banned	destructive	jetplane	bob_toeback	radiation	algebra	energizes
dirt	cheaters	hyper	chow	sm	advantageous	fap	find
rigid	vous	rings	permissive	empath	devout	stoned	plethora
mew	type	heal	barbecued	probs	raping	memy	pufferfish
bees	asounds	produce	bitchiest	war	venue	shrooms	query

2.3. Categorization and Sentiment Analysis of Textual Data

This study proactively proposed that the core TF-IDF technique involves counting the occurrence/frequency of words. However, the frequency indicates the extent of attention instead of the extent of affection (preference or valence) [7]. Hence, this study applied the sentiment analysis of those sentences to obtain the affection of the aspect instead of adopting the TF-IDF score to predict or compare the questionnaire ratings [10]. Past research adopted the TF-IDF score or manual labeling to compare or predict the questionnaire ratings, which is different. Except for the attention aspects, we considered affection equivalent to comparing it with the results of a questionnaire or psychological tests [9,18].

The methods of personality measurement regarding sentiment are summarized as follows. For detailed theories and verification methods, please refer to the scale-directed text analysis (SDTA) developed by scholars [9,10]. R and PHP languages are used to develop programs to convert qualitative text content analysis into quantitative marketing scale scores based on the existing marketing scale (This study uses two word datasets, respectively, the AFINN sentiment lexicon (http://www2.imm.dtu.dk/pubdb/pubs/6010-full.html, accessed on 10 November 2022) and MBTI personality as developed by filtering the higher score of TF-IDF as described in a previous paragraph.).

2.4. Sentiment Analysis of MBTI Personality

The AFINN sentiment lexicon was used to distinguish the word polarity (positive, negative, or neutral) and the MBTI lexicon was used to distinguish the degree (via an interval scale from −5 to +5) (please refer to Table 7).

Table 7. Sample sentences illustrating how to calculate the sentiment score.

ID	**Dimension**	**Featured Word**	**Featured Sentence**	**Emotional Word**	**Sentiment Score**
1	Decision(1)	technologies	specific technologies meant countries denial	denial	−2
2	Information(0)	slang	worrying essay read biography Blank pieces paper Scattered stare blink Squirm time minutes slang depends word circumstance altruistic	worrying	−3

Sentiment analysis of textual data measures someone's words to determine their feelings. In some cases, it is considered more revealing than surveys because it is a more organic analytical method [8]. The performance of such sentiment classifiers depends on the domain or topic being analyzed [12]. We developed an automatic textual analysis system in the programming languages R and PHP to scan the collected textual data and compared it to the custom dictionaries of MBTI personalities that we developed using TF-IDF statistics. Based on keywords in the dictionaries, the program identifies relevant sentences and assigns each sentence to a construct of the MBTI personality dimension. The textual data for each construct's sentence was analyzed using sentiment analysis of the publicly available AFINN Sentiment Word List. This is a well-known list of English words manually developed by Finn Årup Nielsen, a researcher at the University of Denmark [19]. Specifically, the AFINN word list was used to rate the valence of each sentence using an integer ranging from –5 to +5 based on word strength. Our automated system also identifies and reverses the sentiment scores of sentences containing negative modifiers. Please refer to Table 7 for two examples of categorizing the sentences and scoring the sentiment polarity of textual data.

For example, the sentences shown in Table 7 were written by a participant. The keywords 'technologies' and 'slang' in those sentences can be found in the 'Decision(1)' and 'Information(0)' dimensions of the MBTI personality, respectively. Furthermore, emotional words, in this case, 'denial'-2 and 'worrying'-3, in those sentences were rated by the AFINN.

The categorization and sentiment analysis of the textual data revealed the sentiment score for each document, as shown in the columns mind(0), mind(1), information(0), information(1), decision(0), decision(1), structure(0), and structure(1) in Table 8. Those scores indicate the extent of the valance of the personality traits.

Feature selection is the process of reducing the number of input variables when developing a predictive model. Reducing the number of input variables is desirable to decrease the computational cost of modeling and, in some cases, improve the model's performance. From the perspective of text analytics, feature selection refers to feature word extraction when using the machine learning approach.

Statistical feature selection methods involve evaluating the relationship between each input variable and the target variable using statistics and selecting the input variables that have the strongest relationship with the target variable. These methods can be fast and also effective, although the choice of statistical measures depends on the data type of both the input and output variables. This current study, employing the machine learning approach, uses TF-IDF and sentiment analysis.

However, what is the central criterion to determine the baseline or cut-off threshold to filter more relative feature words for a specific trait? From the perspective of machine learning, feature importance refers to techniques that assign a score to input features based on how useful they are at predicting a target variable. Feature importance scores play an important role in a predictive modeling project, including providing insight into the data, the model, and the basis for dimensionality reduction and feature selection, which can improve the efficiency and effectiveness of a predictive model. Thus, in this study, we attempted to adopt a machine learning algorithm method, Random Forest, to calculate the relative importance of feature words and provide a mechanism to tune the amount and selection features of the words extracted from the TF-IDF and sentiment analysis.

Please refer to Table 9 for the list of feature words and scores of relative importance.

Table 8. The data preparation of sentiment score for prediction of MBTI personality traits.

No	Type	Clean_Post	Mind	Information	Decision	Structure	Mind(0)	Mind(1)	Information(0)	Information(1)	Decision(0)	Decision(1)	Structure(0)	Structure(1)
0	INFJ	http: //www.youtube. com/watch?v= qsXHcwe3krw	0	0	0	1	0	0	−15	0	0	−15	0	0
1	ENTP	I'm finding the lack of me in these posts very alarming.	1	0	1	0	−3	0	2	0	0	−3	0	0
2	INTP	Good one _____ https: //www.youtube. com/watch?v= fHiGbolFFGw Of course	0	0	1	0	0	0	4	0	0	0	0	0
3	INTJ	Dear INTP, I enjoyed our conversation the other day.	0	0	1	1	−4	0	−2	0	−2	0	0	0
4	ENTJ	You're fired. That's another silly misconception. That approaching is logically is going	1	0	1	1	0	−2	0	−4	1.5	0	0	0

Table 9. For the example of list of feature words and scores of relative importance.

Mind		Information		Decision		Structure	
Words	**Score**	**Words**	**Score**	**Words**	**Score**	**Words**	**Score**
hate	0.004983	word	0.00083	love	0.039867	infp	0.023256
able	0.003322	believe	0.00083	feel	0.02907	makes	0.01495
made	0.002492	talking	0.00083	info	0.020764	guys	0.011628
problem	0.002492	part	0.00083	life	0.01412	help	0.009967
stuff	0.002492	start	0.00083	feeling	0.010797	general	0.009967

2.5. Validation of a Custom Dictionary for the Construct

This study employed the score for the sentiment of extracted feature words instead of only the TF-IDF score to predict the questionnaire response. This study adopted the cross-validation function provided by R CARET. Furthermore, XGBoost is an increasingly popular machine learning algorithm due to its high performance and accuracy and its ability to solve overfitting (The programming language R provides easy use and is a powerful CARET package https://cran.r-project.org/web/packages/caret/caret.pdf (accessed on 25 November 2022) to implement the XGBooost algorithm). Before applying the ML algorithm to train and test the data, the input data were prepared as outlined below.

3. Results

3.1. Training Data for the MBTI Personality

The target variables in this instance were mind, information, decision, and structure, respectively, and the TF-IDF score for the words starting from the column think, know, etc., were the features used to predict the target variable. Please refer to Table 10.

Table 10. The TF-IDF score for predicting the MBTI personality traits.

No	Type	Clean_Post	Mind	Information	Decision	Structure	Think	People	Know	Time	Feel	Love
0	INFJ	http://www.youtube.com/watch?v=qsXHcwe3krw	0	0	0	1	0.000	0.052	0.000	0.219	0.000	0.000
1	ENTP	I'm finding the lack of me in these posts very alarming.	1	0	1	0	0.087	0.087	0.309	0.137	0.000	0.052
2	INTP	Good one _____ https://www.youtube.com/watch?v=fHiGbolFFGw Of course	0	0	1	0	0.152	0.305	0.103	0.000	0.000	0.061
3	INTJ	Dear INTP, I enjoyed our conversation the other day.	0	0	1	1	0.137	0.137	0.174	0.072	0.000	0.000
4	ENTJ	You're fired. That's another silly misconception. That approaching is logically is going	1	0	1	1	0.269	0.448	0.136	0.140	0.000	0.000

On the other hand, we prepared a dataset similar to that in Table 11. The target variables were mind, information, decision, and structure, respectively, and the sentiment score for each construct was the features. Please refer to Table 11.

Table 11. Data preparation of TF-IDF and sentiment scores predicting the MBTI personality traits.

Line	Mind	Information	Decision	Structure	Mind(0)	Mind(1)	Information(0)	Information(1)	Decision(0)	Decision(1)	Structure(0)	Structure(1)
0	0	0	0	1	0	0	−15	0	0	−15	0	0
1	1	0	1	0	−3	0	2	0	0	−3	0	0
2	0	0	1	0	0	0	4	0	0	0	0	0
3	0	0	1	1	−4	0	−2	0	−2	0	0	0
4	1	0	1	1	0	−2	0	−4	1.5	0	0	0

Given the result obtained via two sorts of features, the TF-IDF score and sentiment score of the construct, we can compare the accuracy of the kinds of features possible, as shown in Table 12.

Table 12. The comparison of performance prediction between TF-ID and sentiment.

Metrics	Mind	Information	Decision	Structure
TF-IDF	0.78	0.80	0.70	0.59
Sentiment	0.80	0.80	0.72	0.61
TF-IDF+Sentiment	0.80	0.91	0.74	0.66

3.2. Discovering How Consumers Use the Feature Words

We adopted the strategic analysis grid of FTTA (From Text to Action), which is an aspect-based analysis framework, to discover the four situations of the interaction of attention (frequency) and affection (sentiment) and further explore how a consumer uses those feature words in terms of the interaction of attention (TF-IDF score) and affection (sentiment score).

Tsao et al., 2022 [10], proposed that the data on the topics mentioned in a text (aspect), coupled with the data on the frequency with which they are mentioned (attention) and the sentiment they receive (opinion), can provide useful strategic insights, namely, the FTTA (From Text to Action) grid. This framework is based on a specific aspect or dimension, and the grid explores the interaction between attention and affection based on textual data.

First, the words appearing in the upper right quadrant are characterized by high attention and positive affection, which indicates that those words represent consumers with the corresponding personality traits and more positive affection.

Second, the words in the upper left quadrant, with high attention and negative affection, are most often used by consumers with the corresponding personality traits and negative affection.

Third, the words in the lower right quadrant, with high attention and high affection, indicate highly positive affection, but these are less used by those consumers with a corresponding personality trait.

Fourth, the words in the lower left quadrant, with low attention and log affection, indicate negative affection, and they are also used less frequently.

Please refer to Figure 1 for the sample words in the FTTA grid.

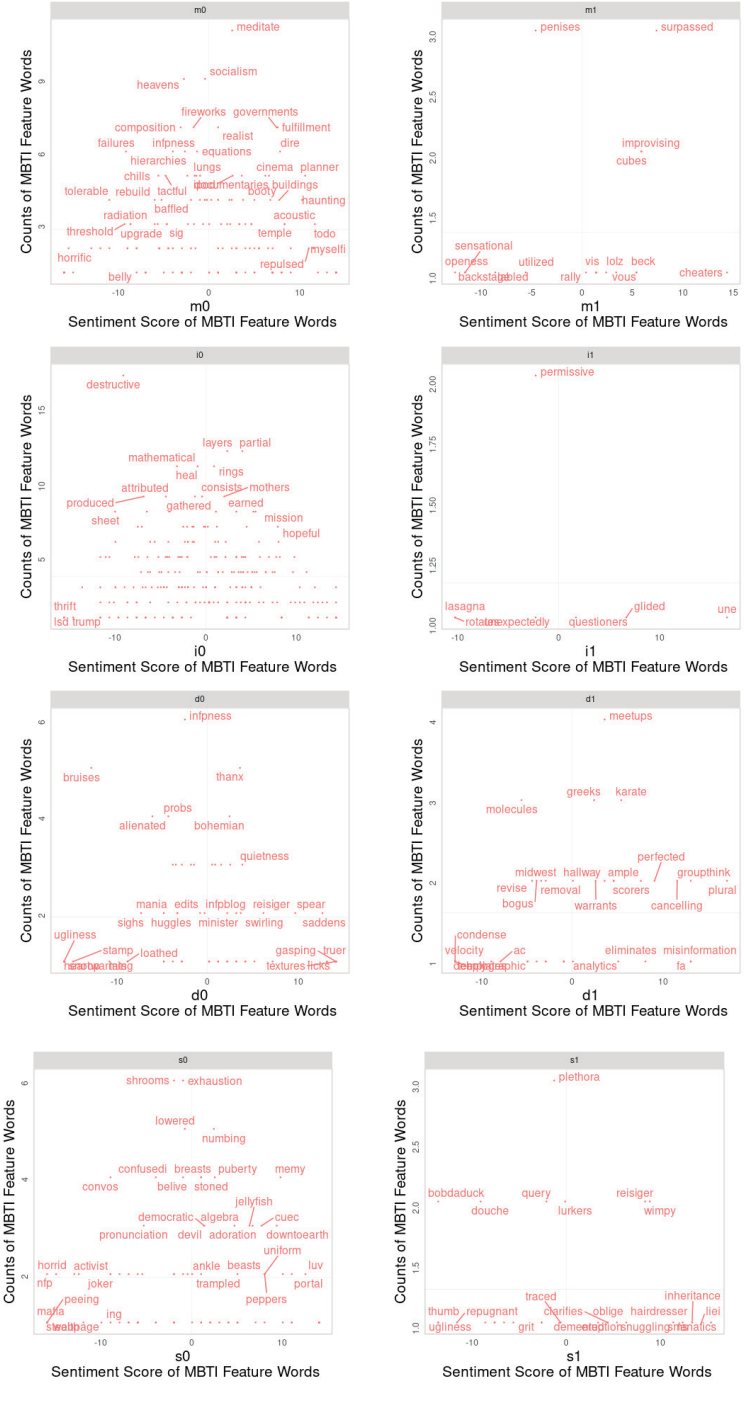


Figure 1. FTTA grid of MBTI personality.

We summarized some intensively used words for positive words and negative words for different traits of personality in Table 13.

Table 13. Sample words of MBTI Personality.

	Intensively Used Words	
MBTI Dimension	**Positive Words**	**Negative Words**
m0 (I)	realist, fulfilment, tactful, planner, myself	tolerable, haunting, horrific
m1 (E)	Improvising, sensational, openness	cheaters
i0 (N)	mathematical, produced	heal
i1 (S)	questioners, permissive, glided	-
d0 (T)	probs, edits	loathed, bruises, saddens
d1 (F)	standardized, meetups, devout	misinformation
s0 (J)	convos, democratic	query, improves, fanatics
s1 (P)	exhaustion, numbing, confused	wimpy, ugliness, lie

Note. I (Introversion): preferring self-reflection to social interactions and preferring to observe before participating in an activity. E (Extraversion): enjoying socializing and tending to be more enthusiastic, assertive, talkative, and animated. N (Intuition): referring to how people process data. They easily see the big picture rather than the details. S (Sensing): refers to processing data through the five senses. They focus on the present and prefer to "learn by doing" rather than thinking it through. T (Thinking): referring to how people make decisions. They are objective and base their decision on hard logic and facts. F (Feeling): they are more subjective. When making decisions, they consider other people's feelings and take them into account. J (Judging)**:** referring to how people outwardly display themselves when making decisions. They like order and prefer outlined schedules to working extemporaneously. P (Perceiving): they prefer flexibility, live their life with spontaneity, dislike structure, and prefer to adapt to new situations rather than plan for them [20,21].

3.3. Summary of Findings

First, in this study, we successfully obtained MBTI scores indicating the positive or negative dimension of MTBI regarding the mind, information, decision, and structure. We could then filter the higher value of TF-IDF for each construct to generate the feature words for each dimension.

Second, given the result obtained via two sorts of features, the TF-IDF score and sentiment score of the construct, we could compare the accuracy of the kinds of features possible via an AI-empowered machine learning algorithm, as shown in Table 12. The results support that the sentiment score is useful for filtering and validating more coherent words to communicate a particular aspect of personality.

Finally, we adopted the FTTA strategy analysis grid, allowing us to better understand the consumer by using the features of words in terms of the interaction between attention (TF-IDF score) and affection (sentiment score). In other words, individuals with specific personality traits tend to heavily use some words positively or negatively, as shown in the upper right and upper left quadrants, respectively, in Figure 1.

4. Conclusions

The results obtained in this study confirm that the TF-IDF algorithm can be used to generate a custom dictionary. Furthermore, sentiment scoring with an AI-empowered machine learning algorithm is effective for extracting more coherent words to communicate a particular aspect of personality.

In other words, we attempted to discover the association between words and their sentiments and specific personality traits. The TF-IDF and AI-empowered sentiment analysis can reveal intrinsic concepts of those features and words used by individuals with specific personalities. Furthermore, the strategic analysis grid of From Text to Action (FTTA), which is an analysis framework based on four situations of the interaction of attention (frequency score of TF-IDF) and affection (sentiment), allows us to better understand how consumers use feature words that are positively and negatively associated with personality traits, as Table 13 shows. However, we still require proposing a limitation of the usage of FTTA, that is, how to interpret the feature words is dependent on the realm and context of the research. While a deep dive into the original textual data is required to fully understand

the meaning behind the word mining, a domain expert is also needed to help with the interpretation. However, the FTTA grid still provides a data-driven pathway and cue to lead us to produce the insight. Furthermore, based on the results obtained in this study, a potential further research question could be explored, which is how to achieve automatic awareness of customers' personalities and a one-to-one advertising message-communication strategy [16,17,22].

Author Contributions: Conceptualization, H.-Y.T. and C.-C.L.; Methodology, H.-Y.T. and C.-C.L.; Investigation, H.-Y.L.; Resources, H.-Y.L. and R.-S.L.; Data curation, R.-S.L.; Writing—original draft, H.-Y.T.; Writing—review & editing, C.-C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: https://www.kaggle.com/datasets/datasnaek/mbti-type.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lukito, L.C.; Erwin, A.; Purnama, J.; Danoekoesoemo, W. Social media user personality classification using computational linguistic. In Proceedings of the 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia, 5–6 October 2016; IEEE: Piscataway, NJ, USA.
2. Costa, P.T.; McCrae, R.R. The Revised NEO Personality Inventory (NEO-PI-R). In *The SAGE Handbook of Personality Theory and Assessment*; Boyle, G.J., Matthews, G., Saklofske, D.H., Eds.; Sage Publications, Inc.: Thousand Oaks, CA, USA, 1992; Volume 2, pp. 179–198.
3. Myers, I.B.; Myers, P.B. *Gifts Differing: Understanding Personality Type*; Davies-Black Publishing: Mountain View, CA, USA, 1995.
4. Pennebaker, J.W.; Mehl, M.R.; Niederhoffer, K.G. Psychological aspects of natural language use: Our words, our selves. *Annu. Rev. Psychol.* **2003**, *54*, 547–577. [CrossRef] [PubMed]
5. Mairesse, F.; Walker, M.; Mehl, M.; Moore, R. Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text. *J. Artif. Intell. Res.* **2007**, *30*, 457–500. [CrossRef]
6. Ranjan, A.; Fernández-Baca, D.; Tripathi, S.; Deepak, A. An Ensemble Tf-Idf Based Approach to Protein Function Prediction via Sequence Segmentation. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2022**, *19*, 2685–2696. [CrossRef] [PubMed]
7. Mee, L.; Homapour, E.; Chiclana, F.; Engel, O. Sentiment analysis using TF–IDF weighting of UK MPs' tweets on Brexit. *Knowl. -Based Syst.* **2021**, *228*, 107238. [CrossRef]
8. Pang, B.; Lee, L. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In Proceedings of the Association for Computational Linguistics (ACL), Barcelona, Spain, 21–26 July 2004.
9. Tsao, H.-Y.; Campbell, C.; Sands, S.; Ferraro, C.; Mavrommatis, A.; Lu, S. A machine-learning based approach to measuring constructs through text analysis. *Eur. J. Mark.* **2019**, *54*, 511–524. [CrossRef]
10. Tsao, H.-Y.; Campbell, C.; Sands, S.; Ferraro, C.; Mavrommatis, A. From mining to meaning: How B2B marketers can leverage text to inform strategy. *Ind. Mark. Manag.* **2022**, *106*, 90–98. [CrossRef]
11. Culotta, A.; Cutler, J. Mining Brand Perceptions from Twitter Social Networks. *Mark. Sci.* **2016**, *35*, 343–362. [CrossRef]
12. Gunter, B.; Koteyko, N.; Atanasova, D. Sentiment Analysis: A Market-Relevant and Reliable Measure of Public Feeling? *Int. J. Mark. Res.* **2014**, *56*, 231–247. [CrossRef]
13. Tsao, H.-Y.; Chen, M.-Y.; Campbell, C.; Sands, S. Estimating numerical scale ratings from text-based service reviews. *J. Serv. Manag.* **2020**, *31*, 187–202. [CrossRef]
14. Deac, V.; Dobrin, C.; Girneata, A. Customer Perceived Value-An Essential Element in Sales Management. *Business Excell. Manag.* **2016**, *6*, 43–55.
15. Dobrin, C.O.; Girneata, A. Complaining Behaviour and Consumer Safety: Research on Romania Online Shopping. *Economic Stud.* **2015**, *24*, 161–175.
16. Winter, S.; Maslowska, E.; Vos, A.L. The effects of trait-based personalization in social media advertising. *Comput. Hum. Behav.* **2021**, *114*, 106525. [CrossRef]
17. Xia Liu, A.; Li, Y.; Xu, S.X. Assessing the Unacquainted: Inferred Reviewer Personality and Review Helpfulness. *MIS Q.* **2021**, *45*, 1113–1148.
18. Berger, J.; Humphreys, A.; Ludwig, S.; Moe, W.W.; Netzer, O.; Schweidel, D.A. Uniting the Tribes: Using Text for Marketing Insight. *J. Mark.* **2020**, *84*, 1–25. [CrossRef]
19. Nielsen, F.Å. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. *arXiv* **2011**, arXiv:1103.2903.
20. Myers, L.B. *Manual: The Myers-Briggs Type Indicator*; Educational Testing Services Publishing: Princeton, NJ, USA, 1962.

21. Myers, L.B.; McCaulley, M.H. *Manual: A Guide to the Development and Use of the Myers-Briggs Type Indicator*; Consulting Psychologists Press Publishing: Palo Alto, CA, USA, 1985.
22. Shumanov, M.; Cooper, H.; Ewing, M. Using AI predicted personality to enhance advertising effectiveness. *Eur. J. Mark.* **2022**, *56*, 1590–1609. [CrossRef]

applied sciences

MDPI

Article

A Pipeline for Story Visualization from Natural Language

Jezia Zakraoui, Moutaz Saleh *, Somaya Al-Maadeed and Jihad Mohamad Alja'am

Department of Computer Science, Qatar University, Doha 2713, Qatar; j.zakraoui@gmail.com (J.Z.); s_alali@qu.edu.qa (S.A.-M.); jaljaam@gmail.com (J.M.A.)
* Correspondence: moutaz.saleh@qu.edu.qa

Abstract: Generating automatic visualization from natural language texts is an important task for promoting language learning and literacy development for young children and language learners. However, translating a text into a coherent visualization matching its relevant keywords is a challenging problem. To tackle this issue, we proposed a robust story visualization pipeline ranging from NLP and relation extraction to image sequence generation and alignment. First, we applied a shallow semantic representation of the text where we extracted concepts including relevant characters, scene objects, and events in an appropriate format. We also distinguished between simple and complex actions. This distinction helped to realize an optimal visualization of the scene objects and their relationships according to the target audience. Second, we utilized an image generation framework along with different versions to support the visualization task efficiently. Third, we used CLIP similarity function as a semantic relevance metric to check local and global coherence to the whole story. Finally, we validated the scene sequence to compose a final visualization using the different versions for various target audiences. Our preliminary results showed considerable effectiveness in adopting such a pipeline for a coarse visualization task that can subsequently be enhanced.

Keywords: scene generation; story visualization; GAN; story understanding; language learning

Citation: Zakraoui, J.; Saleh, M.; Al-Maadeed, S.; Alja'am, J.M. A Pipeline for Story Visualization from Natural Language. *Appl. Sci.* **2023**, *13*, 5107. https://doi.org/10.3390/app13085107

Academic Editors: Xiangjie Kong, Wei Wang and Han Liu

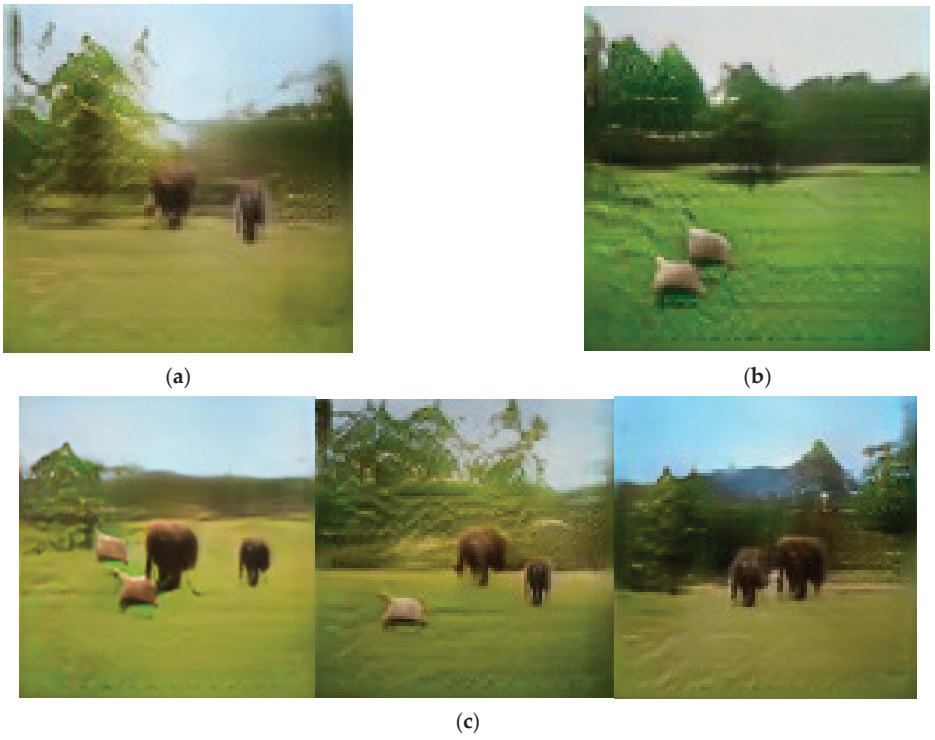
Received: 18 March 2023
Revised: 11 April 2023
Accepted: 17 April 2023
Published: 19 April 2023

1. Introduction

During the period of the COVID-19 pandemic, teachers had a full-time schedule to provide regular and online lessons to children, divided into several small groups. Both teachers and students encountered changes in teaching and learning habits, respectively. For instance, preparing a sequence of coherent images to visualize textual stories from an Arabic natural language text is a very challenging problem [1]. On the other hand, using only text-to-image retrieval methods is very inefficient for young children with special educational needs and learning difficulties (Senld). For instance, using retrieved images from diverse search engines to visualize non-common characters and actions from a story often requires enormous manual effort, yet it is more difficult to adapt this to meet each student's effective learning needs. The same applies for aligning images within a story. Sometimes, this task remains completely unresolved. We approached this unresolved issue using a semi-automatic scene sequence task, i.e., a visual story task to facilitate the learning process and inspire teachers, instructors, and students.

However, to create such story visualization efficiently, one needs to convert the story constituents into a sequence of image frames in a proper and coherent way. A sequence of images can illustrate the story events and characters that can contain multiple sentences. The sequence of images is defined as a continuous stream of consistent images that are part of the same story or event, as argued by the authors in [2]. Although visualized stories are difficult to generate in a robust way, they are more comprehensible, memorable, and attractive. Consequently, automatic story understanding and visualization has a broad application prospect in storytelling, while also representing an important step in many computer vision (CV) applications such as children learning natural language vocabularies. Essentially, our goal was to create a sequence of images to visualize an Arabic story where

the text was extended from sentence level to paragraph level for continuous visualization. In prior studies on text-to-image generation [3–5], the same sentence may have a significantly different generated image while depending largely on the contextual information; therefore, it is also necessary to pass the essential contextual information from the story text to the image generation framework. For instance, considering the sentences given in Figure 1, Figure 1b will vary widely without the context of the story, i.e., without the Figure 1a.



(**a**) (**b**)

(**c**)

Figure 1. Input sentences (translated from Arabic to English) and correspondingly generated single images and an image sequence: (**a**) The elephants are standing in a grassy field; (**b**) There are sheep around them; (**c**) The sheep approached the elephants.

To tackle the problem of objects and event extraction, we applied scene graphs [6] to represent the detailed semantics of each sentence from the story text. A scene graph (SG) is defined as a graph-based semantic representation having nodes and edges. The nodes represent objects, and the edges represent relationships between them. For simplicity, we rewrote each complex sentence in a simple form such as (object, attribute) or (object, relationship, object) tuples. This step abstracted away most of the lexical and syntactic challenges of natural language in the process.

To tackle the second challenge with image synthesis, we used an automatic image generation framework to allow different versions, namely single images and a sequence of images for common and non-common actions, respectively. However, the challenge here was how to display the logic narrative flow of the sequence of images to visualize the story characters and events in a coherent way. Specifically, the appearance of objects and the layout in the background must evolve as per the story narrative flow. Our method can be considered as a fast solution to visualize non-common characters and actions with multiple images whenever needed.

An emerging trend in CV combining deep learning models is regarded as a possible solution for approaching our task. Among these models, generative models construct scenes from sentences, either from short textual descriptions [5] or short dialog [7]. Additionally, previous studies have assumed that a space that synthesizes both vision and language modalities are indispensable to the performance of any text-to-image synthesis [5]. Notably, recent studies using generative adversarial networks (GANs) have presented good results. However, GAN cannot achieve expected results when the image to be generated contains multiple objects. Indeed, such a requirement is more challenging when multiple objects with complicated relationships and different locations are to be presented in the image [8]. Consequently, complex scene generation is still in the development stage and has not been elaborated upon.

We extended our previous study [9], which attempted to generate sequences of images. Despite producing visual sequences that capture the relevant content of the input text, i.e., characters and events, our previous method was limited to the extracted entities and relationships that exactly matched the model vocabulary, thereby ignoring other content from the input text. To tackle this issue, in this extended study, we added a vocabulary mapping module. Another limitation of our previous study was that the text–image alignment method was made by consecutively aligning the images, which showed sharp changes in visual content between the frames. In this study, we employed a multimodal similarity function to dynamically align the images in a sequence based on their similarity scores to the input text. Moreover, uncommon actions in the input text are hard to visualize and are left behind due to many reasons such as their rarity in the dataset; thus, we believed that a decomposition of such actions in a detailed image sequence could facilitate the visualization of such actions. Furthermore, we compared our method with two state-of-the-art models for generating images.

In this context, we proposed a framework based on a text-to-image approach and CLIP to generate and return the best image in the sequence corresponding to the input text. More specifically, the framework took the text as an input, generated a sequence of images, and highlighted the images whose CLIP embedding was most similar to the input text. Notably, we started with an NLP task, i.e., a story parsing task using handcrafted syntactic rules, followed by entities and relation extraction, and vocabulary mapping. Then, a semantic representation using SG was built upon all of the resulting triples denoted as (object, relation, object). Afterward, an image sequence generator for generating images from SG was applied. Subsequently, CLIP was used for image production and input text embedding, followed by computing similarity scores. Further, we evaluated the produced text–image alignments using different metrics.

In contrast to previous studies that have focused on single image generation, we applied detailed image sequence generation for non-common actions using a pre-trained model on a visual genome dataset [10] under the PyTorch framework [11]. Finally, we applied the CLIP [12] similarity function as a metric to check the sentence-level coherence to generated an image sequence, and it computed the cosine similarity between the feature vectors of the story sentences and each of the images. A higher similarity meant a closer match between the story sentence and the corresponding images. Based on these scores, the images were reranked to form an image sequence.

The rest of the paper is organized as follows: Section 2 describes the main approaches to scene generation, Section 3 specifically presents our method, Section 4 discusses the experimental setup, Section 5 shows our evaluation and obtained results, while Section 6 concludes the paper.

2. Related Studies

Early studies on text visualization and illustration [13–17] traditionally relied upon manually annotated image repositories collected from search engines using image retrieval techniques [18,19], and by using images produced by users [20]. Retrieval-based approaches compare texts and images across modalities [21] using different techniques such a canonical

correlation analysis [22]. Specifically, text-to-image systems use retrieval methods that focus on the matching of text and images. In addition, these studies have relied on massive amounts of labeled data, as stated by the authors in [23]. One of the early story visualization attempts was the story-picturing system [13]. The system retrieved landscape and art images from online repositories to illustrate ten short stories. It used keywords from the stories and image descriptions to match the linking between the images using the similarity function. A comparative study of early story illustrations, visualization systems, and tools can be found in [24].

A method worth mentioning was proposed by Huang et al. [14], using VizStory, as a visualization system of fairy tales, to transform the input texts to representative pictures. The system selected keywords from segments in the stories, while relevant pictures were searched for using online resources based on their tags. Finally, to represent the main ideas of the original segments, the final pictures were composed. Afterward, the authors built in a visual storytelling dataset (VIST) that was useful for image-in-sequence to story-in-sequence generation [25], thereby initiating the visual storytelling task.

Alternatively, the studies of the authors in [2,26] attempted to visualize a story with image sequences. The former proposed to enhance the single sentence representation with a global coherence vector and apply global and region matching to retrieve an image for each sentence. The latter proposed a framework with a story-to-image retriever. It selected relevant and inspirative cinematic images and used a storyboard creator that further refined and rendered the images to improve the relevancy and visual consistency. Both authors worked on VIST datasets to evaluate their work. Despite the method given by the authors in [26] scene images with a high resolution and multiple foreground objects were generated; however, it only used cartoon characters where the structures and shapes were poor, resulting in poor image quality.

Recently, Fang et al. [27] used the shooting time order and the storyline behind the images to construct a narrative collage image. First, they considered a set of semantic salient objects from each representative image for object extraction. Then, they used an image canvas according to layer graphs and scene graphs to visualize the extracted objects. Finally, they synthesized a new narrative collage image. More recently, Fang et al. [28] proposed a comprehensive text-to-image synthesis pipeline. They used segmented background scene image and foreground objects from the COCO dataset to generate complex and high-resolution scene images. Finally, they applied the constrained Markov chain Monte Carlo method to generate the optimal positions and scales for all foreground objects to look more realistic. However, these methods rely heavily on image retrieval and fail to generate images with a realistic look, since they just focus on text understanding, object selection, and text–object matching.

With the advances in CV using GANs [29], which are a more powerful class of implicit generative models, they have been successfully applied to various image synthesis methods such as text-to-image synthesis from short textual descriptions [3–5,30–32]. A key task in text-to-image generation is understanding longer and more complex input text, as in our case. Story visualization, however, is different from short textual descriptions, which places more emphasis on semantic coherency rather than simple descriptive text. A story text can contain different scene changes, many objects, different backgrounds, etc. An interesting study [33] has demonstrated dialogue-to-image generation, where the input was a complete dialogue session rather than a single sentence. However, this method was simply a text–image concatenation task and used a coarse sentence condition that, as a consequence, limited its overall performance.

Lee et al. [23] proposed the StoryGAN model to tackle the above the story visualization challenge. Their model employed a context encoder to track the story narrative flow. It used two discriminators; one at the story level and the other at image level to enhance image quality and the consistency of the generated images. However, well-known difficulties in training generative models such as instabilities in the training procedure [34] has limited these studies of specific domains, such as cartoon characters [23]. The study of Zeng

et al. [35] enhanced the latter study in several and significant ways, particularly in relation to image quality and consistency. First, they integrated a universal sentence encoder to incentive compliance of the generated images with textual descriptions. Second, they incorporated an attention-driven word feature into their model, making it more realistic in terms of image details. Finally, they introduced an image patches discriminator to determine whether parts of the image were real. However, this work was limited in its scope since only cartoons could be considered and the quality of the generated images needed further improvement.

More recently, Song et al. [36], Li et al. [37], and the authors in [38] improved upon StoryGAN [23] to emphasize the continuity between consecutive frames in generated video as well as to enhance the quality and relevance of the generated images. More recently, the authors proposed an approach [39] that decomposed the task of story visualization into three phases, namely semantic text understanding, object layout prediction, and image generation and refinement. In contrast to our study, only captions were considered, and only a single image was generated at each step. In addition, their model used two-stage image generation, aka StackGAN. A different model called *Text2Scene* has been proposed by Tan et al. [40]. It is a sequential framework [41] where, at every time point, it learns to generate objects and their associated attributes by attending to the words in the input text and the status of the current generated scene. This approach, however, is restricted to the composition of tasks of abstract scenes and object layouts. On the other hand, the quality of the generated image is usually not stable in most cases. Subsequently, it is difficult to directly apply generative models in complex and real-life scenarios such as scene generation for stories in the wild [42].

3. The Proposed Pipeline Framework

Image generation for the task of story visualization aims to generate representative and coherent images to convey the semantic in a given story text. This is a challenging task since it requires a deep understanding of the objects involved in the story as well as their mutual interactions, and semantical connections and co-relations. In this context, we proposed a framework consisting of (i) an NLP task followed by (ii) a semantic representation using SG, (iii) an image sequence generator for generating images from SG, and (iv) CLIP for producing images and input text embeddings followed by computing similarity scores. Further, we evaluated the produced text–image alignments using different metrics. Moreover, we compared our approach based on scoring images according to their semantic relevance to the input text. In the following section, we have presented the main components of the proposed story visualization pipeline, as depicted in Figure 2. The architecture consists of four consecutive parts as shown below:

1. **Natural language processing**: The first step was the language model where we applied a preprocessing pipeline, machine translation, tokenization, stop-word removal, co-reference resolution, and semantic parsing, i.e., the task of mapping natural language text into its semantic representation using a scene graph parser.
2. **Relation extraction and vocabulary mapping**: The second step involved constructing scene graphs of extracted triples so that the text was transformed into a directed graph $G = (O; R)$ of objects O (nodes) and their relations R (edges).
3. **Image sequence generation**: The third task was image sequence generation where we generated images from scene graphs for all mapped triplets using two different modes.
4. **Text–image alignment**: Finally, we applied CLIP similarity function to produce previsualizations with different sequences. The instructor could examine each image sequence and choose whether to use the single image version or the detailed image sequence.

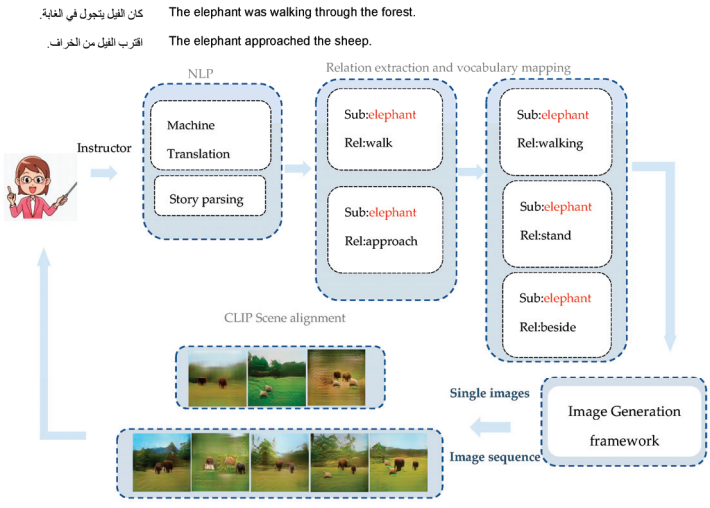


Figure 2. The overall pipeline of the proposed approach: a story text is piped into an NLP module to the first MT, which preprocesses and parses the sentences into scene graph triples. Then, the triples are mapped to model vocabulary to generate single images and image sequences. After applying CLIP model, the instructor is able to adjust the synthesized image sequences by choosing whether to use the single image version or the detailed image sequence.

3.1. Natural Language Processing (NLP)

We considered mainly children's stories featuring animals. After translating the stories from Arabic to English, we extracted the characters and scene objects that were necessary for visualization, including the relationships between them. Then, we proceeded with a neural coreference resolution of the pronouns to prepare the text as simply as possible for the next step. A pre-trained neural model NeuralCoref [43] was used to replace the ambiguous mention of pronouns with its corresponding nominal pronoun.

We obtained a set of relationships based on form (subject; relation; object) by using a scene graph parser. In many cases, the obtained list of relationships was noisy, for instance, objects may have multiple relationships, a passive form, a plural form, etc. To prepare the list for further processing, we pre-processed the list using different rules.

We defined a list of entities to record characters and scene objects that appeared in the text. We traversed every relationship and confirmed whether the involved relation and the entities existed in the vocabulary list. If they existed, they were appended in the *relationships* and *entities* list, respectively. Otherwise, we use word2vec-based (https://code.google.com/archive/p/word2vec/ (accessed on 10 February 2023)) similarity function to find the nearest token in the model vocabulary list. Finally, a dictionary output was created that included two lists, *entities* = [o_i, o_j, . . .] and *relationships* = [[x_i, r, x_j], . . .], where x_i is the index of o_i, x_j is the index of o_j in from the entities' list, and r $\in$ R is the set of model relationship categories. The described process is shown in Algorithm 1.

Algorithm 1: ParseStory parses input text and extracts triples as characters, entities, and relations

Input = AS: Arabic Story, args []: list of access parameters, vocabulary []
Output = DrawTriples {}, SceneGraphTriples {}, entities [], relationships []
Begin

```
rawtext = translateQCRI (AS, args)
rawtext = coreference_resolution (rawtext)
docx = nlp (rawtext)
SceneGraphTriples = sng_parser (docx)
for relation in SceneGraphTriples ['relations']
if (relation in vocabulary) then
x_i, x_j = relation ['subject'], relation ['object]//indices for both involved entities
o_i, o_j = SceneGraphTriples ['entities'].value (x_i, x_j)//get both involved entities
if (o_i, o_j in vocabulary)
entities.append (o_i, o_j)
relationships.append ([x_i, relation, x_j ])
end
else
o_i, relation, o_j = get_mapping (o_i, relation, o_j)//vocabulary mapping
entities.append (o_i, o_j)
relationships.append ([x_i, relation, x_j ])
end
end
DrawTriples ['entities] = entities
DrawTriples ['relationships'] = relationships
return DrawTriples
```

end

3.2. Relation Extraction and Vocabulary Mapping

We considered phrases that described the main animal characters' behavior. We also focused on some of their common and uncommon basic behaviors. Table 1 shows some common sample phrases used in this work as well as their related actions. It is worth noting that animal behavior that is not listed is considered to be non-common animal behavior. From the resultant phrases of the previous step, we obtained all of the triples in the form <*object*, *relationship*, *object*> using a scene graph parser. Due to practical reasons, it was not possible to create images for all of the extracted triples from the story text. Due to this restriction, as in the case of the visual genome dataset [10], the vocabulary mapping used a semantic similarity based on word2vec to find the nearest tokens from the model vocabulary, as shown in Algorithm 2.

Algorithm 2: Get_mapping extended extracted entities and relations with model vocabulary

Input = vocabulary [], triples []
Output = similar_triples []
Begin

```
model = gensim.models.Word2Vec (vocabulary, size = 100, min_count = 1, sg = 1)//initialize
from Gensim library (https://radimrehurek.com/gensim/models/word2vec.html
(accessed on 12 February 2023))
for entity in triples:
top_similar = model.wv.most_similar (positive = entity, topn = 1)//get most similar token to
entity
... similar_triples.append (top_similar)
end
return similar_triples
```

end

Table 1. An excerpt from stories' details related to the above sentences.

Sentences	Noun-Phrases	Dependency Parsing	Scene Graph Triples
The elephants are standing in a grassy field	The elephants	nsubj	<elephants, in, field>
	A grassy field	pobj	
The sheep are running behind the elephants	The sheep	nsubj	<sheep, behind, elephants>
	The elephants	pobj	
The sheep approached the elephants	The sheep	nsubj	<sheep, approached, elephants>
	The elephants	dobj	

Thus, the mapping also helped us to map non-common actions such as "*approach*" to the similar common action in the list such as "*stand*" and "*walk*". If we failed to find a match, we checked for a mapping while including the verb's preposition such as "*close to*", "*next to*", etc. For instance, for the tokens of the sentences mentioned earlier in Figure 1, we computed their similarities with the terms in the vocabularies and took the maximum value among them all. As an example, the token "*elephants"* was mapped to the term *"elephant"* with a similarity value of 1.0; however, the token *approached* was mapped to the term "*stand"* with a similarity value of 0.1, using the word2vec similarity function.

3.3. Image Generation

We split the image generation step into two main tasks. One task tackled the generation of a single image to visualize sentences in isolation. The second task was directed towards the detailed generation of image sequences, i.e., multiple images that were highly coherent with the whole story. After, obtaining the objects and relationships that composed the scene graph, we used a graph convolution network [11] composed of several graph convolution layers to process the scene graph.

Single image generation. We generated images from scene graph triples of actions and characters using a pre-trained model for PyTorch [11]. Basically, the architecture consisted of three main modules: a graph convolution network (GCN), a layout prediction network (LN) and a cascade refinement network (CRN). First, the GCN took a scene graph as an input and produced an embedding label vector output for each object. Then, these object embedding vectors were used by LN to compute a scene layout by predicting a segmentation mask and bounding box for each object. Given a scene layout, the CRN was then responsible for generating an image that respected the object relations in the scene layout. Finally, discriminators were used to generate realistic output images by adversarially training the image generation network against a pair of image discriminator networks and an object discriminator network. The generated realistic output images were adversarially trained by the image generation network against a pair of discriminator networks D_{image} and D_{object} to minimize the weighted sum of six losses [11]. The discriminator D_{image} attempted to classify its input x as real or fake by maximizing the following objective:

$$\mathcal{L}_{GAN} = \mathbb{E}_{x \sim p_{real}} \log D(x) + \mathbb{E}_{x \sim p_{fake}} \log(1 - D(x)) \quad (1)$$

where $x \sim p_{real}$ is the ground truth image and $x \sim p_{fake}$ is the fake image that is generated using the generator network. The discriminator D_{object} guarantees that the generated objects are identifiable by predicting the object's category. Both D_{object} and the generator network attempt to maximize the probability that D_{object} correctly classifies objects [11].

Image sequence generation. Non-common actions are typically hard to illustrate. To enable a fair visual understanding of such actions, it was necessary to decompose these actions into simple ones. This process enabled us to employ more detailed images rather than only one image. However, the decomposition for actions has only been explored for humans [44,45], even though representative actions with structured representations could lead to improved action recognition in general. Therefore, we applied an image generation

mode [11] to generate sequences of images rather than single isolated images. This is because the image sequence can give more details to support the visual understanding of complex actions. For instance, in Figure 1, it is hard to visualize the action "*approaching*" using a single image only; therefore, it is necessary to generate a sequence of images that decompose the flow of this action into several frames, similar to the way that humans actively perceive ongoing actions, i.e., a phenomenon referred to as event segmentation theory [46].

Specifically, for this category of actions, we generated sequences of images shot by shot using progressive additions of objects and relations. Where the input text described only one object, it was rendered in almost the middle of the scene. On the other hand, complex images were rendered by starting with simple characters and progressively adding others to build up to more complex images.

3.4. Text–Image Alignment

Once we generated all of the images, we subsequently computed the cosine similarity using CLIP feature vectors between the story text and each of the generated images. In CLIP, a visual encoder and a text encoder encode an input image and text independently, and the dot-product between the two encoder's output was used as the "alignment score" between the input image and text based on following Formula (2):

$$logits = X_{image}\, X_{text}^{T} \times e^{\tau} \tag{2}$$

where X_{image} image and X_{text}^{T} are normalized encoders outputs for the image and the text, respectively, and τ is a learned temperature parameter [12]. The CLIP model, which was already trained over an extremely large number of images, was capable of generating semantic encodings for arbitrary images without additional supervision.

Finally, an automatic alignment image sequence was suggested based on the CLIP scores. The instructor could choose whether to use the single image version or the detailed image sequence. He/she could then refine the image sequence by reordering and skipping frames, etc.

4. Experimental Setup

At this stage, we began by preprocessing the input stories as the input data set. We considered 80 short and simple phrases from Arabic stories in the animal domain [9]. We translated them from Arabic to English, and the selected 20 stories had 80 key phrases. The distribution of objects was consistent in number, where each object possessed five different actions. We selected phrases with a simple narrative structure to introduce concepts using animal characters and their common behaviors such as *running, eating*, *jumping*, etc., as well as non-common behavior *such as approaching, covering, looking at*, etc. The characters, objects, location, and background were explicitly mentioned in the text and were realistic. In the experimental set up, we further applied the following steps:

Story parsing was applied; it included coreference resolution, part-of-speech tagging, dependency parsing, relation extraction using linguistic patterns, and scene graph parser (https://github.com/vacancy/SceneGraphParser (accessed on 17 March 2023)). For example, we considered the sentences "*The elephants are standing in a grassy field. The sheep are running behind them. The sheep approached the elephants*". After applying the coreference resolution using NeuralCoref (https://github.com/huggingface/neuralcoref (accessed on 25 February 2023)) and manual adjustment, we obtained the following final representation for the sentences *"The elephants are standing in a grassy field. The sheep are running behind the elephants. The sheep approached the elephants"*. Table 1 shows the story parsing outputs. Of note, *nsubj*, *pobj*, *dobj*, and *iobj* denoted the nominal subject, object of a preposition, direct object, and indirect object, respectively.

1. To handle out-of-vocabulary words besides those in the training data set, we applied simple vocabulary mapping using word2vec, a pre-trained word embedding model,

to find the nearest vocabulary of the extracted triples in the trained model vocabulary. For example, the token *approached* was mapped to the terms *stand* and *beside*, with similarity values of 0.1 and 0.2, respectively.

2. We used all extracted relation triples and their mapped tokens to generate 128 × 128 images using the same configuration as in the *sg2im* model [11]. The sg2im model is pretrained on the visual genome dataset [10], a dataset with 108,077 128 × 128 images annotated with scene graphs. Each image has an average of 21 objects, 18 attributes, and 18 pairwise relationships between objects where animal categories are included, in addition to their visual relationships. All experiments were executed with Pytorch 0.4, CUDA v10, Cudnn v7, and Python 3. We generated single images and sequences of images for common and non-common actions, respectively depending on the type of actions. Table 2 provides excerpts of such phrases and the correspondingly generated images.
3. We applied object detection using the *PixelLib* (https://pixellib.readthedocs.io/en/latest/ (accessed on 2 March 2023)) model for all generated images. The detected object indicated whether the character mentioned in the story line also appeared in the image frame. We exploited the output of the model to estimate, to some extent, the coherence of the image sequence with the whole story text. We scored each correct image frame and summed the final score for each sequence.
4. Finally, we arrange the generated images for each story sequentially, as produced by the CLIP score, using two different versions: the single image version and image sequences, as can be seen in Table 3 below.

Table 2. An excerpt of phrase (containing non-common actions) and corresponding generated single images versus image sequences.

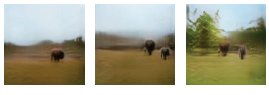
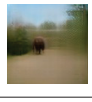
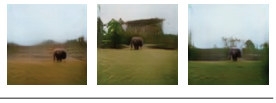
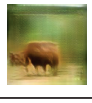
Id	Non-Common Phrases	Generated Single Image	Generated Image Sequence
1	Elephants approaching		
2	Elephant looking at sky		
3	Elephant attaching to tree		
4	Elephant covering in tree		
5	Elephant carrying wood		

Table 3. Single images, image sequences, and corresponding object detection details.

Generation Mode	Object Detection Mask	Character Relevance
Single image		1.2 correct images (counting only correct images, 1 + 0 + 1 = 2)
Image sequence		2.3 correct images (counting only correct images, 1 + 1 + 1 + 0 + 0 = 3)

5. Evaluation and Results

The evaluation of story visualization is complex due to the generative nature of the method. We conducted both quantitative and qualitative assessments as follows. First, we compared our method with BigGAN + CLIP [12] and Dall-E [47], two state-of-the-art models for generating images from user prompts. In these models, each image prediction was actually the result of an optimization process where the latent space of the generator directly maximized the CLIP score between the generated image and the description.

5.1. Quantitative Results

We demonstrated the image sequence quality of our method in a score-based manner with regard to two aspects: character relevance and the semantic relevance of the generated images. For our approach, we adopted a character relevance score and CLIP similarity score between the story text and each of the images.

Character relevance: Inspired by the studies of [23,35], we selected five of the most common characters and actions. Specifically, we selected the following animal characters, *elephant*, *sheep*, *cow*, *zebra*, *and giraffe*, each with five actions representing their behavior and relations. The results obtained from the experiment with each story character, e.g., *elephants*, and *sheep*, is summarized in Table 3. The five continuous images form a visualization version corresponding to a single story. For each image sequence version, we counted each image frame as correct if the characters mentioned in the input sentence appeared in the corresponding image frame, according to the object detection model. For instance, in Figure 1, in the second image (from left to right) the elephant character was not present in the image, so this image was counted as being incorrect. Since the ground truth of the object segmentation was unavailable in the visual genome dataset, we exploited a pre-trained salient object detection model to detect objects from all generated images. The object detection task gave an indication if the character mentioned in the story line also appeared in the image frame.

Semantic relevance: We measured the semantic relevance between the generated image and the story text features for each generated image sequence using CLIP. The images with the highest scores were marked with red borders and selected for final visualization, see also Table 4. We computed the sentence similarity score as local consistency (Table 4) and the story similarity score as global consistency (Table 5).

Table 4. Comparison with state-of-the-art model for real-world image synthesis for a one-sentence story sample: the images with the highest scores were marked with red borders and selected for final visualization.

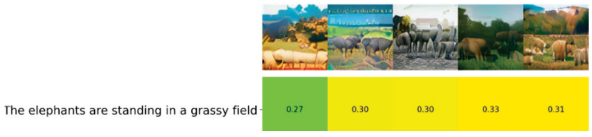
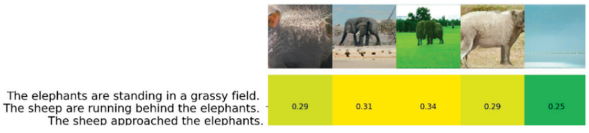
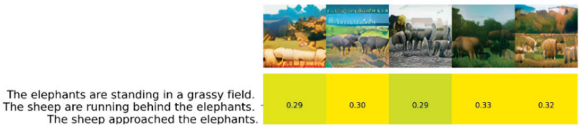
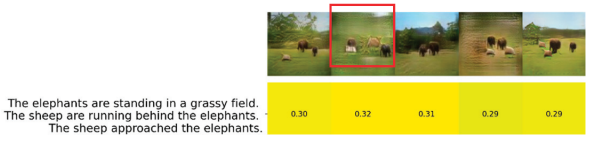
Method	Generated Images	CLIP Score
BigGAN + CLIP Radford, et al. (2021) [12]		0.30
Dall-E Ramesh, et al. (2021) [47]		0.30
Our		0.32

Table 5. Comparison with state-of-the-art model for real-world image synthesis for a story sample: the images with the highest scores were marked with red borders and selected for final visualization.

Method	Generated Images	CLIP Score
BigGAN + CLIP Radford, et al. (2021) [12]		0.29
Dall-E Ramesh, et al. (2021) [47]		0.30
Our		0.30

5.2. Qualitative Results

We evaluated the visual quality of image sequences, generated image sequences that contained multiple scene objects, and visually inspected them. Table 6 shows a scenario applied on the sentences from Figure 1. The results showed that image sequences that were coherent and consistent were preferred over any image sequence, according to our early evaluation. Consistent image sequence indicates visual similarity between images, while coherent image sequences show common characters in the story in terms of overall appearance.

Table 6. A scenario showing the results after vocabulary mapping step.

Characters	Actions and Relation	Resulted Triples	Generation Mode	Generated Images
Two Elephants	Standing in	<elephant, standing in, field>	Single	
Two Sheep, two elephants	Walking on	<sheep, walking on, field>	Multiple	
	Behind	<sheep, behind, elephant>		
Two Sheep, two elephants.	Stand	<sheep, stand, field>	Multiple	
	Beside	<sheep, beside, elephant>		

Our proposed story visualization pipeline saved us time and manual effort in delivering a robust visualization that is ready to use in schools under certain pandemic conditions. We further demonstrated the effectiveness of the proposed pipeline in more complex scenarios such as inter-related sentences and non-common actions. On one hand, using coreference resolution simplified such sentences so that relation extraction reflected the whole sentence meaning, including the story context embedded in previous neighbor sentences. In addition, identifying non-common actions supported the provision of detailed images, while the decomposition of such actions into simple spatio-temporal actions was helpful in explaining how objects and their relationships change as such action occurs.

Robustness: Since our testing set contained sentences of different types, it could exaggerate the contributions of the relation extraction task. Therefore, we resolved this issue by splitting them into two groups. One group included the stories as they are, while the other group included only sentences with co-referenced pronoun resolution. We report that relation extraction performance significantly improved in the second group. This is due to the simplification of inter-related and complicated sentence into multiple simpler sentences, each having a single action along with its participant characters, making it straightforward to extract necessary relations and actions.

Quality of different versions: Concerning the obtained results, we selected some examples from our test set which are shown in Tables 2 and 6, together with the generated images. In the single generated images, for actions such as *standing on*, *attaching*, *etc.*, we can clearly see that the single images visualized the characters and the actions in some cases. However, they were misleading for other non-common actions such as *covering*, *looking at*, etc., as they required more supporting detailed images. In contrast, for the sequence of images, it was observed that non-common actions such as *approaching* were decomposed by starting with simple graphs and progressively building up to more additional details. The addition of objects caused the shift of related objects so that the relationships were respected. However, many images capturing the same type of events can be vastly different in their visual structures, such as those seen in row#2 and row#3 in Table 6. Adding more images promoted the understanding of the input sentence; in contrast, using a single image with cluttered objects resulted in a crowded image plane. Moreover, we observed that using one version instead of the other version was correct based on challenge of using characters only.

Semantic consistency: CLIP guaranteed that each selected image was locally consistent since each selected image matched its corresponding sentence semantically by choosing a higher CLIP score. Our method is of global and local relevance, achieving the highest average rank in comprehensive relevance compared to two state-of-the-art image synthesis methods used for real-world scenarios. Visual examples are shown in Tables 4 and 5, where our method outperformed these two models in terms of global semantic consistency.

Character relevance: The evaluation of the story character classification results indicated that all characters mentioned in the story also appeared in any frame of the image, as was observed from the calculated character relevance score. Thus, this result also proves the effectiveness of our method in maintaining story character accuracy. Essentially, generating image sequences with a higher story coherence score can better comply with the story text, in addition to supporting visual learning

Image quality: Regarding the quality of the generated images, however, the promising results were still limited to generating a few categories of objects. For general stories where multiple objects co-exist with complex relationships, the realism and diversity of the generated images are not satisfactory and remain to be improved in relation to many aspects. Though experimentation with CLIP, the semantic relevance between the two modalities was enhanced.

To reduce the difficulty of synthesizing complex scenes in any real-world setting, we aimed to enrich our pipeline to cover a wide range of characters, objects, and diverse actions. However, we still faced some limitations and technical problems with the image generation task such as the low quality of the synthetized images. Likewise, the generated images still contained many obvious visual artifacts; therefore, models trained for this task are still far from being deployed in any real-world setting. Nevertheless, our work strongly argues that text visualization through a single image only will not produce a meaningful visualization to help with understanding stories. However, proposing a better solution that combines automatic image sequence generation and semi-manual adjustment can ensure flexibility and safety in the learning process.

6. Conclusions

We presented a pipeline overview to illustrate the use of Arabic story text with a sequence of generated images as a fast solution to support distance learning in schools. In summary, we applied an NLP module to process the story text and to obtain an appropriate semantic representation of the main characters, common events, and actions in each sentence. Extensive experimental results on an in-domain visual story test set demonstrated the effectiveness of the proposed pipeline, while the image generation framework was applied to complete the final visualization. Despite the challenges associated with evaluating such systems, our preliminary results showed considerable effectiveness in the adoption of such a pipeline for a coarse visualization task that can be subsequently enhanced. In addition, we expect our contributions to assist with the visualization of stories with a higher image quality when considering more detailed information regarding characters, objects, and relationships.

We are now positioned to conduct an Arabic story annotation effort, followed by implementation of the story visualization, following the outlined task modules detailed previously. Our pipeline and implementation details are algorithmically comprehensible. We anticipate state-of-the-art computer vision and language generation methodologies will provide a number of baselines for Arabic story visualization. For instance, to compare a computer vision algorithm that may over-identify objects against one focused on a specific story domain. Our pipeline allows us to easily prompt for different narrative versions and audiences. In the future, it will be necessary to compare different narrative sequences of images in terms of the cognitive and perception degree of students. Evaluation and release of the final image sequence must take into consideration the narrative goal and audience to ensure a flexible and safe learning environment. In addition, the evaluation must balance the correctness of the action flow, as well as the coherency of the generated story visualization. In particular, new quantitative and qualitative metrics for such tasks must be developed.

In the future, we would like to process more complex and meaningful text with multiple paragraphs. We would also extend the work to produce more professional and intelligent components to support the whole proposed pipeline. Indeed, such as pipeline

for a story visualization task can be extended to a video generation task, which is more challenging in terms of the temporal spatial consistency of the video content.

Author Contributions: Conceptualization, J.Z., M.S. and J.M.A.; Methodology, J.Z. and M.S.; Software, J.Z.; Validation, J.Z.; Writing—original draft, J.Z.; Writing—review & editing, M.S., S.A.-M. and J.M.A.; Supervision, M.S., S.A.-M. and J.M.A.; Project administration, S.A.-M.; Funding acquisition, S.A.-M. and J.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was made possible by an NPRP grant #10-0205-170346 from the Qatar National Research Fund (a member of the Qatar Foundation). The statements made herein are solely the responsibility of the authors.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used during this study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Zakraoui, J.; Elloumi, S.; Alja'Am, J.M.; Ben Yahia, S. Improving Arabic Text to Image Mapping Using a Robust Machine Learning Technique. *IEEE Access* **2019**, *7*, 18772–18782. [CrossRef]
2. Ravi, H.; Wang, L.; Muniz, C.; Sigal, L.; Metaxas, D.; Kapadia, M. Show Me a Story: Towards Coherent Neural Story Illustration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7613–7621. [CrossRef]
3. Xu, T.; Zhang, P.; Huang, Q.; Zhang, H.; Gan, Z.; Huang, X.; He, X. AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1316–1324.
4. Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; Metaxas, D.N. StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 1947–1962. [CrossRef] [PubMed]
5. Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; Lee, H. Generative Adversarial Text to Image Synthesis. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016.
6. Johnson, J.; Krishna, R.; Stark, M.; Li, L.J.; Shamma, D.; Bernstein, M.; Fei-Fei, L. Image retrieval using scene graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3668–3678.
7. El-Nouby, A.; Sharma, S.; Schulz, H.; Hjelm, D.; El Asri, L.; Ebrahimi Kahou, S.; Bengio, Y.; Taylor, G.W. Keep Drawing It: Iterative language-based image. In Proceedings of the Neural Information Processing Systems (NeurIPS) Visually-Grounded Interaction and Language (ViGIL) Workshop, Montreal, QC, Canada, 7 December 2018.
8. Tobias, H.; Stefan, H.; Stefan, W. Generating Multiple Objects At Spatially Distinct Locations. In Proceedings of the International Conference on Learning Representations (ICLR), Washington, DC, USA, 30 May–2 June 2019.
9. Zakraoui, J.; Saleh, M.; Al-Maadeed, S.; Alja'am, J.; Abou El-Seoud, M. Visualizing Children Stories with Generated Image Sequences. In *Visions and Concepts for Education 4.0. ICBL 2020. Advances in Intelligent Systems and Computing*; Auer, M.E., Centea, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; p. 1314. [CrossRef]
10. Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D.A.; et al. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *Int. J. Comput. Vis.* **2017**, *123*, 32–73. [CrossRef]
11. Johnson, J.; Gupta, A.; Fei-Fei, L. Image Generation from Scene Graphs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1219–1228.
12. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sutskever, I. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July2021.
13. Joshi, D.; Wang, J.Z.; Li, J. The Story Picturing Engine—A system for automatic text illustration. *ACM Trans. Multimed. Comput. Commun. Appl.* **2006**, *2*, 68–89. [CrossRef]
14. Huang, C.-J.; Li, C.-T.; Shan, M.-K. VizStory: Visualization of Digital Narrative for Fairy Tales. In Proceedings of the 2013 Conference on Technologies and Applications of Artificial Intelligence, Taipei, Taiwan, 6–8 December 2013.
15. Zakraoui, J.; Al Jaam, J.M. A Dynamic Illustration Approach For Arabic Text. In Proceedings of the IEEE 10th GCC Conference & Exhibition (GCC), Salmiya, Kuwait, 19–23 April 2019.
16. Andrej, K.; Li, F.-F. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3128–3137.
17. Krstajić, M.; Najm-Araghi, M.; Mansmann, F.; Keim, D.A. Story Tracker: Incremental visual text analytics of news story development. *Inf. Vis.* **2013**, *12*, 308–323. [CrossRef]

18. Dureja, A.; Pahwa, P. Image retrieval techniques: A survey. *Int. J. Eng. Technol.* **2018**, *7*, 215–219. [CrossRef]
19. Banharnsakun, A. Artificial bee colony algorithm for content-based image retrieval. *Comput. Intell.* **2020**, *36*, 351–367. [CrossRef]
20. Radiano, O.; Graber, Y.; Mahler, M.; Sigal, L.; Shamir, A. Story Albums: Creating Fictional Stories From Personal Photograph Sets. *Comput. Graph. Forum* **2017**, *37*, 19–31. [CrossRef]
21. Gu, Y.; Wang, C.; Ma, J.; Nemiroff, R.; Kao, D.L.; Parra, D. Visualization and recommendation of large image collections toward effective sensemaking. *Inf. Vis.* **2016**, *16*, 21–47. [CrossRef]
22. Hardoon, D.R.; Szedmak, S.; Shawe-Taylor, J. Canonical Correlation Analysis: An Overview with Application to Learning Methods. *Neural Comput.* **2004**, *16*, 2639–2664. [CrossRef] [PubMed]
23. Yitong, L.; Zhe, G.; Yelong, S.; Jingjing, L.; Yu, C.; Yuexin, W.; Lawrence, C.; David, C.; Jianfeng, G. StoryGAN: A Sequential Conditional GAN for Story Visualization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
24. Zakraoui, J.; Saleh, M.; Aljaam; Jihad, M. *Text-to-Picture Tools, Systems and Approaches: A Survey. Journal of Multimedia Tools and Applications*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 78, pp. 22833–22859.
25. Huang, T.-H.; Ferraro, F.; Mostafazadeh, N.; Misra, I.; Agrawal, A.; Devlin, J.; Girshick, R.; He, P.K.X.; Ba-tra, D.; Zitnick, L.; et al. Visual Storytelling. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 233–1239.
26. Chen, S.; Liu, B.; Fu, J.; Song, R.; Jin, Q.; Lin, P.; Qi, X.; Wang, C.; Zhou, J. Neural Storyboard Artist: Visualizing Stories with Coherent Image Sequences. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), 21–25 October 2019*; Association for Computing Machinery: New York, NY, USA, 2019.
27. Fang, F.; Yi, M.; Feng, H.; Hu, S.; Xiao, C. Narrative Collage of Image Collections by Scene Graph Recombination. *IEEE Trans. Vis. Comput. Graph.* **2017**, *24*, 2559–2572. [CrossRef] [PubMed]
28. Fang, F.; Luo, F.; Zhang, H.-P.; Zhou, H.-J.; Chow, A.L.H.; Xiao, C.-X. A Comprehensive Pipeline for Complex Text-to-Image Synthesis. *J. Comput. Sci. Technol.* **2020**, *35*, 522–537. [CrossRef]
29. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems. In Proceedings of the 27th Annual Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
30. Wenbo, L.; Pengchuan, Z.; Lei, Z.; Qiuyuan, H.; Xiaodong, H.; Siwei, L.; Jianfeng, G. Object-driven Text-to-Image Synthesis via Ad-versarial Training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 12166–12174.
31. Tingting, Q.; Jing, Z.; Duanqing, X.; Dacheng, T. MirrorGAN: Learning Text-to-image Generation by Redescription. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 1505–1514.
32. Han, Z.; Tao, X.; Hongsheng, L.; Shaoting, Z.; Xiaogang, W.; Xiaolei, H.; Dimitris, M. StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5908–5916.
33. Shikhar, S.; Dendi, S.; Vincent, M.; Samira, E.K.; Yoshua, B. ChatPainter: Improving Text to Image Generation using Dialogue. *arXiv* **2018**, arXiv:1802.08216.
34. Tim, S.; Ian, G.; Wojciech, Z.; Vicki, C.; Alec, R.; Xi, C. Improved techniques for training gans. In Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16), New York, NY, USA, 5–10 December 2016; pp. 2234–2242.
35. Gangyan, Z.; Zhaohui, L.; Yuan, Z. PororoGAN: An Improved Story Visualization Model on Pororo-SV Dataset. In Proceedings of the 3rd International Conference on Computer Science and Artificial Intelligence, Beijing, China, 6–8 December 2019.
36. Song, Y.-Z.; Tam, Z.-R.; Chen, H.-J.; Lu, H.-H.; Shuai, H.-H. Character-Preserving Coherent Story Visualization. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; Volume 12362.
37. Li, C.; Kong, L.; Zhou, Z. Improved-StoryGAN for sequential images visualization. *J. Vis. Commun. Image Repre-Sentation* **2020**, *73*, 102956. [CrossRef]
38. Maharana, A.; Hannan, D.; Bansal, M. Improving Generation and Evaluation of Visual Stories via Semantic Consistency. *arXiv* **2021**, arXiv:2105.10026.
39. Zakraoui, J.; Saleh, M.; Al-Maadeed, S.; Jaam, J.M. Improving text-to-image generation with object layout guidance. *Multimed. Tools Appl.* **2021**, *80*, 27423–27443. [CrossRef]
40. Tan, F.; Feng, S.; Ordonez, V. Text2Scene: Generating Compositional Scenes From Textual Descriptions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6703–6712.
41. Ilya, S.; Oriol, V.; Quoc, V.L. Sequence to sequence learningwith neural networks. In Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2 (NIPS'14), Montreal, QC, Canada, 8–13 December 2014; pp. 3104–3112.
42. Zakraoui, J.; Saleh, M.; Asghar, U.; AlJa'Am, J.M.; Al-Maadeed, S. Generating Images from Arabic Story-Text using Scene Graph. In Proceedings of the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2–5 February 2020; pp. 469–475.

43. Wolf, T.; Ravenscroft, J.; Chaumond, J.; Rebo, M. Coreference Resolution in Spacy with Neural Networks. HuggingFace. 2018. Available online: https://github.com/huggingface/neuralcoref (accessed on 15 January 2021).
44. Jingwei, J.; Ranjay, K.; Li, F.-F.; Juan Carlos, N. Action Genome: Actions As Compositions of Spatio-Temporal Scene Graphs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10236–10247.
45. Lillo, I.; Soto, A.; Niebles, J.C. Discriminative Hierarchical Modeling of Spatio-temporally Composable Human Activities. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 812–819. [CrossRef]
46. Kurby, C.A.; Zacks, J.M. Segmentation in the perception and memory of events. *Trends Cogn. Sci.* **2008**, *12*, 72–79. [CrossRef] [PubMed]
47. Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; Sutskever, I. Zero-Shot Text-to-Image Generation. In Proceedings of the International Conference on Machine Learning, Vienna, Austria, 18–24 July 2021.

applied sciences

MDPI

Article

SSEMGAT: Syntactic and Semantic Enhanced Multi-Layer Graph Attention Network for Aspect-Level Sentiment Analysis

Xiangzhe Xin [1], Aishan Wumaier [2], Zaokere Kadeer [2,*] and Jiangtao He [1]

[1] College of Information Science and Engineering, Xinjiang University, Urumqi 830046, China
[2] Xinjiang Laboratory of Multi-Language Information Technology, College of Information Science and Engineering, Xinjiang University, Urumqi 830046, China
* Correspondence: zuhra@xju.edu.cn

Abstract: Aspect-level sentiment analysis aims to identify the sentiment polarity of specific aspects appearing in a given sentence or review. The model based on graph structure uses a dependency tree to link the aspect word with its corresponding opinion word and achieves significant results. However, for some sentences with ambiguous syntactic structure, it is difficult for the dependency tree to accurately parse the dependencies, which introduces noise and degrades the performance of the model. Based on this, we propose a syntactic and semantic enhanced multi-layer graph attention network (SSEMGAT), which introduces constituent trees in syntactic features to compensate for dependent trees at the clause level, exploiting aspect-aware attention in semantic features to assign the attention weight of specific aspects between contexts. The enhanced syntactic and semantic features are then used to classify specific aspects of sentiment through a multi-layer graph attention network. Accuracy and Macro-F1 are used as evaluation indexes in the SemEval-2014 Task 4 Restaurant and Laptop dataset and the Twitter dataset to compare the proposed model with the baseline model and the latest model, achieving competitive results.

Keywords: aspect-level sentiment analysis; graph attention network; feature extract

Citation: Xin, X.; Wumaier, A.; Kadeer, Z.; He, J. SSEMGAT: Syntactic and Semantic Enhanced Multi-Layer Graph Attention Network for Aspect-Level Sentiment Analysis. *Appl. Sci.* **2023**, *13*, 5085. https://doi.org/10.3390/app13085085

Academic Editor: Christos Bouras

Received: 14 March 2023
Revised: 17 April 2023
Accepted: 17 April 2023
Published: 19 April 2023

1. Introduction

The rapid development of the Internet has changed people's way of life. For example, information is exchanged and shared through online service platforms, which generates a large number of comment information. These comments not only contain users' views and attitudes towards news events, which can help the government and other agencies monitor public opinion, but also contain preferences for products, which can help commercial companies quickly complete product analysis and make improvements. These comment data have great social and commercial value. It is of great significance to use sentiment analysis technology to study these comments. Aspect-level sentiment analysis is a subtask in sentiment analysis. It is a fine-grained sentiment analysis task, aiming at judging the sentiment tendency of different aspects of entities in comments. Recently, the syntax-based model has used the dependency tree to extract syntactic information and apply it to the aspect-level sentiment analysis task, which has achieved remarkable results. Dependency trees capture dependencies between aspect words and their corresponding opinion words, which can solve the problem of long-distance dependence [1]. Therefore, they are often used to extract syntactic information. Due to the arbitrary expression of online comments, there is no obvious syntactic structure, which leads to the introduction of noise (irrelevant dependency relation) in the parsing of dependency-tree-based methods, reducing the ability of a dependency tree to capture the sentiment-aware context [2].

Based on the above observations, we propose a syntactic and semantic enhanced multi-layer graph attention network (SSEMGAT). The dependency tree is used to represent the dependency between words at the word level; the constituent tree is introduced to

obtain syntactic information from a higher-level perspective. The attention mechanism is easily disturbed by other aspect words. It uses aspect-aware attention to redistribute the attention weight between specific aspect words and context. Then, the extracted syntactic and semantic features are fed into the multi-layer graph attention module for specific aspects of sentiment classification.

The main contributions of this paper are as follows:

(1) For the aspect-level sentiment analysis task, we propose a syntactic and semantic enhanced multi-layer graph attention network to extract features from syntactic and semantic perspectives and use pre-training knowledge to integrate syntactic and semantic features extracted to infer specific aspects of sentiment polarity.
(2) We introduce a constituent tree to make up for the defect in the dependency tree and combine different levels of syntactic information to align the position of the aspect word and its corresponding opinion word. At the same time, aspect-aware attention and multi-headed attention are used to construct local attention and global attention, respectively, to link sentiment information between specific aspects and contexts.
(3) Experimental results on three benchmark datasets show that the performance of the SSEMGAT model exceeds the baseline model and some recent models. Our model incorporates syntactic and semantic feature information well, which indicates that our work is effective.

The following sections of this paper are arranged as follows: In Section 2, we introduce the relevant work of aspect-level sentiment analysis, which is mainly divided into three categories: attention-based approach, syntax-based approach, and pre-training-based approach. In Section 3, we describe the proposed model in detail. In Section 4, we test our proposed model on the public benchmark datasets and analyze it separately. Finally, in Section 5, we summarize the whole paper and look forward to future work.

2. Related Work

Sentiment analysis (SA) is an important research direction in opinion mining. It is the process of using natural language processing technology (NLP) to analyze and summarize text content containing sentiment. Sentiment analysis is divided into sentence-level [3,4], chapter-level [5,6], and aspect-level analysis. The sentence level aims at comment text, which needs to judge its whole sentiment tendency and provide corresponding sentiment values, generally including positive, neutral, and negative. Chapter level refers to a document, which judges the overall sentiment tendency and provides the same sentiment value as the sentence level. Both methods judge the whole and generally only provide sentiment value, which belongs to coarse-grained sentiment. Aspect level aims at the multiple aspects of the entity contained in the review text; each aspect can be composed of different sentiment values, and different aspects can have different sentiment values, even conflict, while the sentence level and chapter level only have one direction of sentiment. Existing studies on aspect-level sentiment analysis can be broadly split into three categories:

(1) Attention-based methods: The attention mechanism models the dependency relationship between an aspect term and its corresponding opinion words. However, there may be several different aspect terms in a sentence. There have been studies to judge the sentiment of a particular aspect. Wang et al. [7] captured the importance of different contextual information to a given aspect word through the attention mechanism, and the attention mechanism and LSTM are combined to model the semantics of sentences and solve the problem of aspect-level sentiment analysis. Ma et al. [8] proposed an interactive attention network (IAN), which uses the attention mechanism to link the target and context for multi-level semantic classification. Chen et al. [9] used multiple attention mechanisms to capture connections between long-distance sentiment features, with strong robustness to irrelevant information. Huang et al. [10] introduced an attention-over-attention (AOA) module to capture the connection between aspects and context words. Fan et al. [11] proposed a multi-grained attention network (MGAN) to combine coarse-grained and fine-grained attention to capture

the interaction of aspect and context at the word level. The attention-based approach achieves attractive results. However, due to its defect, the attention mechanism is easily affected by the noise in the sentence, thus misjudging the sentiment polarity.

(2) Syntax-based methods: Some work explicitly uses dependency trees of a sentence to extract syntactic information. Zhang et al. [12] first proposed building a graph convolutional neural network on a dependency tree to learn the dependencies between nodes. Sun et al. [13] utilized the representation of sentence features learned from the bidirectional LSTM and enhanced embedding with the graph convolutional network. Zhang et al. [14] constructed a hierarchical syntactic graph and lexical graph via convolution on GNN embedding and BiLSTM embedding, respectively, and a bi-level interactive network was designed to learn information interaction. Chen et al. [15] combined information from the latent graph and the dependency graph via a gated attention mechanism. For the situation where the current node of the dependency tree pays average attention to adjacent nodes, Wang et al. [16] constructed an aspect-oriented dependency tree structure (R-GAT) by extending the graph attention network to encode graphs with labeled edges. Most syntax-based models only make use of dependency, without considering the type of dependency. Tian et al. [17] proposed T-GCN, which uses an attention mechanism to distinguish different edges in a graph and uses attention layer ensemble to comprehensively learn different layers of T-GCN. The use of syntactic knowledge only cannot obtain the best results, and some researchers have studied the use of other knowledge. Li et al. [18] proposed a dual graph convolutional neural network (DualGCN) to construct syntactic graphs and semantic graphs from the perspective of syntactic structure and semantic correlation, respectively. Zhang et al. [2] combined the attention matrix constructed by the attention mechanism and syntactic mask matrix to accomplish the interaction of syntactic structure and semantic information. Wu et al. [19] used a dependency tree and phrase tree to construct a phrase dependency graph and used the PD-RGAT model on it for the ABSA task. Compared with the attention-based model, the performance of the syntax-based method was greatly improved, but some shortcomings cannot be ignored. Since dependency trees have different syntactic sensitivities, the noise introduced to sentences without obvious syntactic structure will make it difficult for dependency trees to accurately capture sentiment-aspect context [17], and GCN cannot perfectly integrate topological structure and node features [20]. These problems limit the further development of graph neural networks.

(3) Pre-trained-based methods: Devlin et al. [21] used the left and right context to pre-train the depth bidirectional representation, requiring only one additional output layer to fine-tune the pre-trained BERT representation, achieving state-of-the-art results for a variety of tasks without basic task-specific architecture modifications. Xu et al. [22] proposed training on large-scale general domain data and fine-tuning on a small amount of downstream data, which provides a solution for the study of small sample data. Song et al. [23] designed an attentional encoder to generate hidden representations, and the BERT-SPC model is designed as a comparison model for sentence pair classification tasks. There are also some studies using a combination of pre-training and GCN. Jawahar et al. [24] found that BERT could capture a rich hierarchy of language information, with phrase features at the bottom, syntactic features in the middle, and semantic features at the top. Xiao et al. [25] integrated syntactic sequence information from BERT and knowledge from dependency trees to enhance graph convolutional neural networks for better coding dependency graphs. Tang et al. [26] regarded GCN as a special form of transformer and studied the representation between GCN and a transformer interactively.

3. Methodology

In this section, we introduce the syntactic and semantic enhanced multi-layer graph attention model, that is, SSEMGAT. The overall structure of the model is shown in Figure 1.

It is mainly divided into four parts: input layer, extraction layer, MGAT module, and fusion layer. Next, we will describe each module in the model in detail.

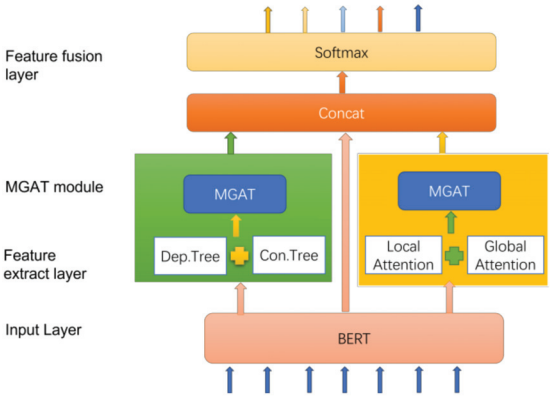


Figure 1. The framework of the proposed SSEMGAT model.

3.1. Input Layer

Given a sentence of n words s = $\{\omega_1, \omega_2, \ldots, a_1, a_2, \ldots, a_m, \ldots, \omega_n\}$, where $\{a_1, a_2, \ldots, a_m\}$ is aspect term, since BERT has a powerful representation learning capacity, we utilize BERT as a sentence encoder to generate contextual representations. To accommodate the input format of the BERT model, given target aspect, we follow BERT-SPC [23] to construct a BERT-based sequence: [CLS] + {sentence} + [SEP] + aspect + [SEP]. However, there may be multiple aspects in a sentence, so we use the form of [CLS] + {sentence} + [SEP] + aspect + [SEP] + aspect + [SEP] to construct the pattern sequence. Then, the output representation H is obtained by BERT,

$$\mathrm{H} = \left\{h_1^t, \ldots, h_n^t\right\} \tag{1}$$

3.2. Extraction Layer

The existing models based on graph structure often use the dependency tree to extract syntactic information, the attention mechanism to extract semantic information, and use GCN to construct syntactic graphs and semantic graphs; the above graphs are interactively learned, and good results are achieved.

3.2.1. Syntactic Feature Extraction

Generally, a dependency tree (Dep.Tree) can capture dependencies between aspect terms and their corresponding opinion words, maintaining valid in the long-distance dependency problem. Therefore, dependency trees are often used to extract syntactic information from sentences. However, not all information on the dependency tree is beneficial to our task, and introducing noise (unrelated relations of dependencies) makes it difficult for each aspect word to accurately capture the corresponding contextual sentiment information. For example, the dependency tree parsing of sentences is shown in Figure 2, and the "conj" relation between "delicious" and "terrible" is invalid for our task, but the aspect term "taste" may be associated with the opinion word "terrible", reducing the ability to accurately capture "delicious" in the opinion words.

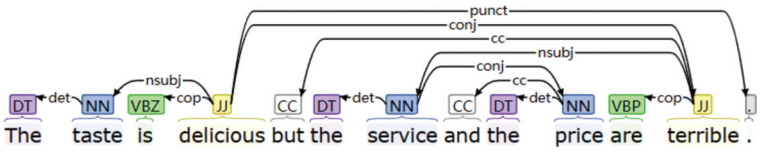


Figure 2. The result of dependency tree parses.

Moreover, the dependency tree reveals relations between words, the relationship between clauses and between aspects that is difficult to capture. Based on this, we use constituent trees, which mainly consist of phrase segmentation and hierarchical structures that help to correctly align aspect words with their corresponding opinion words of sentiment information. Phrase segmentation can easily divide a sentence into multiple clauses and refine the syntactic position of each word in the sentence. The structured hierarchy can distinguish different relationships between aspect words to infer different aspects of sentiment information from a clause-level perspective. For example, the result of parsing the constituent tree of sentences is shown in Figure 3. The whole sentence is divided into four parts: clause "The taste is delicious", phrase segmentation term "but", clause "the service and price are terrible", and ".". In hierarchical structure, according to the phrase segmentation term "and", we can find that the aspect words "service" and "price" have the same sentiment polarity, while according to the phrase segmentation term "but", it is concluded that it has the opposite sentiment polarity towards the aspect word "taste" and the aspect words of other clauses.

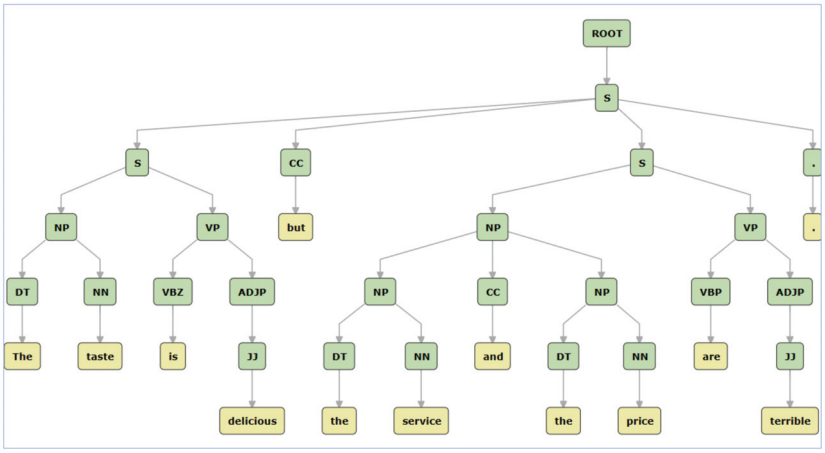


Figure 3. The result of constituent tree parses.

Integrating information from different structural levels can obtain more accurate syntactic information. Therefore, we construct the dependency adjacency matrix DA at the word level and the constituent adjacency matrix CA from the clause level, which is constructed as follows:

(1) Matrix *DA*: Using the dependency tree as an undirected graph, if there is a connection between the words w_i and w_j,

$$DA_{i,j} = \begin{cases} 1, & if\ w_i, w_j\ link\ directly\ in\ Dep.Tree \\ 0, & otherwise \end{cases} \quad (2)$$

(2) Matrix CA: The constituent tree has a hierarchical structure, and in each layer, if words w_i and w_j belong to the same clause phrase,

$$CA_{i,j}^{l} = \begin{cases} 1, & if\ w_i, w_j\ in\ same\ phrase\ of \left\{ ph_u^l \right\} \\ 0, & otherwise \end{cases} \tag{3}$$

Then, the CA and DA matrices are combined via position-wise addition as the extracted syntactic feature matrix A_{syn}:

$$A_{syn} = CA + DA \tag{4}$$

3.2.2. Semantic Feature Extraction

Attention mechanism is a common way to capture the interactions between the aspect and context words. However, the attention mechanism is easily disturbed by noise (other irrelevant aspects of words), and as clues, misjudge the sentiment polarity of the related aspects. Therefore, we use aspect-aware attention to learn local semantic information for a specific aspect, while using self-attention to learn global semantic information for sentences. After that, we fuse local attention with global attention to learn semantic correlation.

(1) Local attention: To enhance the attention of specific aspects to local contextual sentiment information, we use aspect-aware attention to prevent disturbance with other aspects of word information. The aspect-aware attention mechanism utilizes the aspect term as query conditions to calculate the attention feature information of related aspects,

$$A_{local}^{i} = \tanh\left(H_a W^a * \left(KW^K \right)^T + b \right) \tag{5}$$

where K is equal to the output H of the input layer, and W^a and W^K are learnable weights. We perform mean pool operation on output H and copy the processed output n times as H_a.

(2) Global attention: The attention mechanism captures the semantic correlation between any two words in a sentence. This is useful for grasping all of the semantic information in a sentence. Therefore, we use the multi-head attention mechanism [27] to construct the global semantic score matrix A_{global}^{i} of the sentence. The calculation process is as follows,

$$A_{global}^{i} = soft(\frac{QW^Q * \left(KW^K \right)^T}{\sqrt{d}}) \tag{6}$$

where W^Q and W^K are learnable weights

Then, we combine the local attention score with the global score to obtain semantic matrix A_{sem}:

$$A_{sem} = A_{global} + A_{local} \tag{7}$$

3.3. Multi-Layer Graph Attention Module (MGAT)

To utilize rich hierarchical syntactic information, we use the MGAT block stacked by several designed graph attention layers [28]. GAT is a new graph neural network architecture, including an attention mechanism, which enables one to assign different attention weights to the information provided by the feature aggregation of the central node according to different nodes and propagate the sentiment information of node to its neighboring nodes.

The set of input and output in the graph attention layer is $h = \left\{\overrightarrow{h_1}, \overrightarrow{h_2}, \ldots, \overrightarrow{h_N}\right\}$ and $h' = \left\{\overrightarrow{h'_1}, \overrightarrow{h'_2}, \ldots, \overrightarrow{h'_N}\right\}$, from which the attention coefficient between the central node and neighboring nodes is obtained:

$$e_{ij} = a([W\overrightarrow{h_i}||W\overrightarrow{h'_j}]) \tag{8}$$

where a is attention mechanism and W is the weight matrix.

GAT adopts a masked attention mechanism to prevent the dropping of all structural information and changes the previous situation where the self-attention mechanism will allocate attention to all nodes to allocate attention to neighboring nodes. In addition, the attention coefficient is normalized using the *softmax* function, so the attention coefficient after the update is:

$$\alpha_{ij} = softmax(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k=1}^{N} \exp(e_{ik})} \tag{9}$$

The multi-head attention mechanism is used to obtain the influence of adjacent nodes on the central node, and the node features extracted by K heads are represented to complete the splicing operation, and finally, the K average is used to replace the connection operation to obtain the final node representation:

$$\overrightarrow{h'_i} = \sigma\left(\frac{1}{K}\sum\sum_{j\in N} \alpha_{ij}^{k} W^{k} \overrightarrow{h_j}\right) \tag{10}$$

where α_{ij}^{k} is the normalized attention coefficients and W^k is the linear transformation correlation weight matrix.

By stacking the above update process multiple times, node updates in a multi-layer attention graph can be represented as follows:

$$H_A = MGAT(A) \tag{11}$$

The syntactic matrix A_{syn} and semantic matrix A_{sem} are fed to the *MGAT*, respectively, to obtain the syntactic feature H_{syn} and semantic feature H_{sem}:

$$H_{syn} = MGAT(A_{syn}) \tag{12}$$

$$H_{sem} = MGAT(A_{sem}) \tag{13}$$

3.4. Fusion Layers

Pre-trained language models such as BERT have rich hierarchical information, with phrase-level information at the bottom layer, syntactic feature information in the middle layer, and semantic feature information at the top layer [24]. In addition, according to [29], syntactic and semantic information is not completely isolated, and as the syntactic structure changes, the semantics also have some changes. Interactive learning between syntax and semantics can help us better understand sentences. Therefore, we combine the pre-trained knowledge to fuse and learn the semantic and syntactic information, then feed the output feature H_a into the softmax function for classification, and finally obtain the probability distribution $P(a)$ of the sentiment polarity:

$$H_a = [H_{sem}; H_{syn}; H] \tag{14}$$

$$P(a) = softmax(W_p H_a + b_p) \tag{15}$$

3.5. Loss Function

We use standard cross-entropy with L_2 as the loss function:

$$L = -\sum_i \sum_{j=1}^{C} P \log \hat{P} \tag{16}$$

4. Experiment

4.1. Datasets

We evaluate our model on three public datasets: Restaurants and Laptops dataset from Sem-Eval 2014 Task 4 [30] and Twitter dataset provided by Dong et al. [31]. Each sentence in the three datasets is labeled with aspects and opinion words, and sentiment includes three different polarities: positive, neutral, and negative. The statistics from the datasets are in Table 1.

Table 1. Statistics from datasets.

Dataset	Restaurant		Laptop		Twitter	
	Train	Test	Train	Test	Train	Test
Positive	2164	728	994	341	1507	173
Negative	807	196	851	128	1528	169
Neutral	637	196	455	167	3016	336

4.2. Experimental Environment and Parameter Setting

The computing hardware used in the experiment was GeForce GTX 2080Ti, and the deep learning framework was PyTorch. The specific configuration of the experimental environment is shown in Table 2. For model training, we use the bert-base-uncased version of BERT as the sentence encoder and Adam as the optimizer. The detailed parameters are shown in Table 3.

Table 2. Experimental environment.

Projects	Configuration
Operating Platforms	CUDA 11.3
Operating System	Linux
Memory	16 GB
Python Versions	Python 3.8
PyTorch Versions	PyTorch 1.12.0

Table 3. Model parameter settings.

Parameter Name	Parameter Value
batch size	12
learning_rate	0.0001
rnn_hidden	200
bert_dim	768
input_dropout	0.1
atten_head_	2
layer_dropout	0.2
num_epoch	20
attn_head	2

4.3. Evaluation Index

Following the previous work, we used Accuracy and Macro-F1 values as evaluation indexes of aspect-level sentiment analysis tasks.

4.4. Baseline Methods

We selected some mainstream baseline and lasted models to compare with the proposed models.

(1) IAN [8]: The aspect words and contextual representations generated by LSTM are used to learn interactively through attention.
(2) AOA [10]: The aspect words and context representations generated by LSTM are modeled by attention-over-attention neural networks to capture the interaction between aspect and context.
(3) RAM [9]: This proposes a recurrent attention network on memory to capture sentiment features between long distances.
(4) MGAN [11]: The alignment matrix is used to complete the coarse-grained interaction between the aspect word and the context, and the aspect alignment loss function is designed to complete the fine-grained interaction at the word level.
(5) TNet [32]: Use CNN to extract significant features from the transformed word representations from the bidirectional RNN layer.
(6) ASGCN [12]: The dependency tree is used to extract syntactic information and perform graph convolution operations on the dependency tree to learn the representation of nodes.
(7) CDT [13]: The feature representation of a sentence is learned by using bidirectional LSTM, and the embedded representation is enhanced by graph convolutional networks.
(8) BiGCN [14]: The hierarchical syntactic graph and lexical graph are constructed by convolution on GNN embedding and BiLSTM embedding, respectively, and a bi-level interactive network is designed to learn information interaction.
(9) kumaGCN [15]: It combines information from the latent graph and the dependency graph through a gated attention mechanism.
(10) R-GAT [16]: The dependency tree is rooted to the target aspect by reconstructing, and pruning is performed to preserve the edges that are directly dependent on the aspect term.
(11) DGEDT [15]: Considering the dependency tree as a special form of transformer, representations from the dependency tree and transformer are learned in an iterative interaction manner.
(12) DualGCN [26]: Syntactic graph and semantic graph are constructed at the same time, and a double affine mechanism is used to complete the information exchange between syntactic and semantic, and finally, all the information is fused for classification.
(13) SSEGCN [2]: The attention matrix constructed by the attention mechanism and syntactic mask matrix are combined to accomplish the interaction of syntactic structure and semantic information.
(14) RAG-TCGCN [33]: Multiple attention is used to combine syntactic and semantic features and their related features with word-level features parsed using residual attention gates.
(15) BERT [21]: MLM is used for pre-training bidirectional transformers to generate deep language representation, and good results can be achieved only with fine-tuning in downstream tasks.
(16) BERT-PT [22]: The pre-training language model is trained through a large number of general domain data and a small amount of downstream data. It provides a solution for small sample data research.
(17) AEN-BERT [23]: This uses an attention encoding network to interact aspect words with context and designs a processing form based on BERT word embedding.
(18) BERT4GCN [25]: Based on BERT's rich hierarchical structure information, the feature information in the middle layer is fused with the knowledge of the dependency tree, the enhanced dependency graph is constructed, and the convolution operation is performed in it.

4.5. Experimental Results and Analysis

Our proposed model is compared with three types of baseline model: the attention-based method, the syntax-based model, and the pre-training-based model. The attention-based model includes IAN, AOA, RAM, MGAN, and TNet. The syntax-based model includes ASGCN, CDT, BiGCN, kumaGCN, R-GAT, DGEDT, DualGCN, SSEGCN, and RAG-TCGCN. The pre-training-based model includes BERT, BERT-PT, AEN-BERT, and BERT4GCN. The main experimental results are reported in Table 4.

Table 4. Sentiment classification results. We directly introduce the result data from the original author's paper as the data for comparison, where "-" means that this part of the work is not revealed, and the best experimental results are shown in bold.

Models	Restaurant		Laptop		Twitter	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
IAN	78.60	-	72.10	-	-	-
AOA	80.53	69.84	72.88	67.48	72.25	69.96
RAM	80.23	70.80	74.49	71.35	69.36	67.30
MGAN	81.25	71.94	75.39	72.47	72.54	70.81
TNet	80.69	71.27	76.54	71.75	74.90	73.60
ASGCN-DG	80.77	72.02	75.55	71.05	72.15	70.40
ASGCN-DT	80.86	72.19	74.14	69.24	71.53	69.68
CDT	82.30	74.02	77.19	72.99	74.66	73.66
BiGCN	81.97	73.48	74.59	71.84	74.16	73.35
kumaGCN	81.43	73.64	76.12	72..42	72.45	70.77
R-GAT	83.30	76.08	77.42	73.76	75.57	73.82
DGEDT	83.90	75.10	76.80	72.30	74.80	73.40
DualGCN	84.27	78.08	78.48	74.74	75.92	74.29
SSEGCN	84.72	77.51	79.43	76.49	76.51	75.32
RAG-TCGCN	84.09	77.02	78.80	75.04	76.66	75.41
BERT-PT	84.95	76.96	78.07	75.08	-	-
BERT-SPC	84.46	76.98	78.99	75.03	73.55	72.14
AEN-BERT	83.12	73.76	79.93	76.31	74.71	73.13
BERT4GCN	84.75	77.11	77.49	73.01	74.73	73.76
Ours	**86.42**	**79.70**	**80.06**	**76.78**	**76.81**	**76.10**

Based on the experimental results in Table 4, we offer the following analysis:

(1) Our proposed model achieves better results compared with other last and baseline models. We believe that the primary reason is that the designed SSEMGAT model captures syntactic and semantic feature information more efficiently than other models, which also proves the effectiveness of our work.

(2) The model that considers syntactic structure and semantic information at the same time is better than the model that considers only semantic information or syntactic structure, which shows that syntax and semantics do not exist in isolation, and learning the interaction information between them is also very necessary.

(3) Compared with attention-based models, our proposed model has obvious advantages. From the analysis of this phenomenon, we believe that the attention mechanism is easily affected by the noise factor in the sentence when facing complex sentences and obscure structures and cannot accurately align the contextual and sentiment information. This reduces the performance of the model.

(4) Compared with syntax-based models, our model also has good results. This may be because we made up for the inherent defects in dependency trees in sentence parsing, thus enhancing their ability to capture aspect words and their corresponding opinion words and improving the model's ability to resist interference to noise elements introduced in the dependency tree.

(5) Compared with the model based on pre-training, our model also has better performance. BERT has strong representational learning ability and a rich hierarchical structure, while the dependency tree also has an obvious hierarchical structure, which

may be related in some way. When we use the enhanced feature extractor for extraction, we can better capture the correlation between syntax and semantics.

4.6. Ablation Study

We further conducted an ablation study to verify the validity of each module in our model. The result is in Table 5. In the ablation experiment, we removed the dependency tree (dep), constituent tree (con), aspect-aware attention (aaa), and multi-head attention (mha) for comparison and verification.

Table 5. The results of the ablation study.

	Restaurant		Laptop		Twitter	
	Accuracy	**Macro-F1**	**Accuracy**	**Macro-F1**	**Accuracy**	**Macro-F1**
SSEMGAT	**86.42**	**79.70**	**80.06**	**76.78**	**76.81**	**76.10**
w/o dep	85.69	78.69	79.15	75.97	74.26	73.56
w/o con	85.52	78.16	78.80	74.67	75.62	74.43
w/o aaa	86.05	79.66	79.75	76.00	76.22	75.70
w/o mha	85.25	78.02	78.31	75.73	75.31	75.22

First, removal of the dependency tree (dep) leads to a drop in accuracy of 0.73%, 0.91%, and 2.65% on the Restaurant, Laptop, and Twitter dataset, respectively, which demonstrates that the dependency tree is important for extracting syntactic information. Then, with the removal of the constituent tree (w/o con), the model performance decreases by 0.9%, 1.26%, and 1.19%, respectively. It is shown that the constituent tree can effectively supplement the syntactic information extracted from the dependency tree. After, the removal of aspect-aware attention (w/o aaa) causes a decay in the accuracy of 0.37%, 0.31%, and 0.59%. As for 'w/o mha', the accuracy decreases by 1.17%, 1.75%, and 1.5% on the Restaurants, Laptop, and Twitter datasets, respectively. As a result, the ablation experimental outcomes confirm the contribution of both components.

4.7. Case Study

To better understand the work of the SSEMGAT model, we selected two samples to review for visual case studies. In Table 6, we visualize the attention weights, predicted labels, aspect terms, and corresponding true labels for sentences.

Table 6. Visual analysis of attention in review sample.

Model	Aspect	Attention Visualization	Prediction	Label
AOA	environment	Its environment is elegant but price is expensive	Negative	Positive
	price	Its environment is elegant but price is expensive	Positive	Negative
	room	The look of the room is novel	Positive	Positive
ASGCN	environment	Its environment is elegant but price is expensive	Negative	Positive
	price	Its environment is elegant but price is expensive	Negative	Negative
	room	The look of the room is novel	Positive	Positive
BERT-BASE	environment	Its environment is elegant but price is expensive	Positive	Positive
	price	Its environment is elegant but price is expensive	Positive	Negative
	room	The look of the room is novel	Positive	Positive
SSEMGAT	environment	Its environment is elegant but price is expensive	Positive	Positive
	price	Its environment is elegant but price is expensive	Negative	Negative
	room	The look of the room is novel	Positive	Positive

The first sample contains two aspect terms where the corresponding sentiment polarity is opposite, and the second sample contains only one aspect term.

In the first example, the AOA model focuses on "elegant" and "but" at the same time, misjudges "environment" as negative sentiment polarity, while "price" focuses on "elegant" and "expensive" and allocates positive sentiment polarity. This shows that there is interference between different aspect terms. In the second example, with only one aspect term, the correct sentiment polarity was identified. The ASGCN model may misjudge sentiment by taking the relationship between "but" and "environment" as clues. The BERT model does not correctly align the sentiment information corresponding to "price". We speculate that the possible reason is that the corresponding sentiment words are randomly replaced with other irrelevant information when masking. The SSEMGAT model effectively combined syntactic structure and semantic correlation of the feature information and correctly predicted all aspects of terms related to sentiment tendency.

5. Conclusions and Future Work

In this paper, we proposed a syntactic and semantic enhanced multi-lay graph attention neural network (SSEMGAT) to solve the problem of introducing noise in dependent trees in sentences without obvious syntactic structure. Given the inherent defects in dependent trees, we introduced the composition tree structure, which can obtain more field-of-view information at the causal level, and we enhanced the syntactic features by merging syntactic information at different levels. The multi-head attention mechanism may misjudge the sentiment polarity due to the noise introduced by the interference of irrelevant words, so we construct local attention and global attention of specific aspects based on the attention mechanism to assign the attention weight between aspect and context. Facing feature information with a rich hierarchy, we used the multi-layer stacked graph attention module to aggregate different hierarchical information separately and used attention to give higher weight to the information most relevant to the feature. Finally, the extracted syntactic and semantic features are fused with the pre-training knowledge to obtain the most specific aspect of rich hierarchical feature information to achieve aspect sentiment classification.

In future research, we will continue to apply the model to different domains to verify the generalization performance and observe the model's performance in multilingual datasets. Current research still has challenges in mining deeper correlation information between syntax and semantics, and we will further develop methods that can dig deeper into the correlation between them.

Author Contributions: Conceptualization, X.X. and A.W.; methodology, X.X.; software, J.H.; validation, X.X.; formal analysis, X.X., A.W. and Z.K.; investigation, J.H.; data curation, A.W.; writing—original draft preparation, X.X. and A.W.; writing—review and editing, X.X.; supervision, J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Autonomous Region Natural Science Foundation Joint Fund Project, Research on Xinjiang Tourism Sentiment Analysis Technology Based on Deep Learning, under Grant 2021D01C081.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liang, S.; Wei, W.; Mao, X.-L.; Wang, F.; He, Z. BiSyn-GAT+: Bi-Syntax Aware Graph Attention Network for Aspect-based Sentiment Analysis. In *Findings of the Association for Computational Linguistics: ACL 2022*; Association for Computational Linguistics: Dublin, Ireland, 2022; pp. 1835–1848.
2. Zhang, Z.; Zhou, Z.; Wang, Y. SSEGCN: Syntactic and Semantic Enhanced Graph Convolutional Network for Aspect-Based Sentiment Analysis. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Seattle, WC, USA, 10–15 July 2022; Association for Computational Linguistics: Dublin, Ireland, 2022; pp. 4916–4925.
3. Yang, B.; Cardie, C. Context-Aware Learning for Sentence-Level Sentiment Analysis with Posterior Regularization. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Baltimore, MD, USA, 23–24 June 2014; Association for Computational Linguistics: Dublin, Ireland, 2014; pp. 325–335.
4. Severyn, A.; Moschitti, A. Twitter Sentiment Analysis with Deep Convolutional Neural Networks. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015; Association for Computing Machinery: New York, NY, USA, 2015; pp. 959–962.
5. Dou, Z.-Y. Capturing User and Product Information for Document Level Sentiment Analysis with Deep Memory Network. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; Association for Computational Linguistics: Dublin, Ireland, 2017; pp. 521–526.
6. Lyu, C.; Foster, J.; Graham, Y. Improving Document-Level Sentiment Analysis with User and Product Context. In Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 6724–6729.
7. Wang, Y.; Huang, M.; Zhu, X.; Zhao, L. Attention-Based LSTM for Aspect-Level Sentiment Classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; Association for Computational Linguistics: Dublin, Ireland, 2016; pp. 606–615.
8. Ma, D.; Li, S.; Zhang, X.; Wang, H. Interactive Attention Networks for Aspect-Level Sentiment Classification. *arXiv* **2017**, arXiv:1709.00893.
9. Chen, P.; Sun, Z.; Bing, L.; Yang, W. Recurrent Attention Network on Memory for Aspect Sentiment Analysis. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; Association for Computational Linguistics: Dublin, Ireland, 2017; pp. 452–461.
10. Huang, B.; Ou, Y.; Carley, K.M. Aspect Level Sentiment Classification with Attention-over-Attention Neural Networks. In Proceedings of the Social, Cultural, and Behavioral Modeling: 11th International Conference, SBP-BRiMS 2018, Washington, DC, USA, 10–13 July 2018; Thomson, R., Dancy, C., Hyder, A., Bisgin, H., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 197–206.
11. Fan, F.; Feng, Y.; Zhao, D. Multi-Grained Attention Network for Aspect-Level Sentiment Classification. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October-4 November 2018; Association for Computational Linguistics: Dublin, Ireland, 2018; pp. 3433–3442.
12. Zhang, C.; Li, Q.; Song, D. Aspect-Based Sentiment Classification with Aspect-Specific Graph Convolutional Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; Association for Computational Linguistics: Dublin, Ireland, 2019; pp. 4568–4578.
13. Sun, K.; Zhang, R.; Mensah, S.; Mao, Y.; Liu, X. Aspect-Level Sentiment Analysis via Convolution over Dependency Tree. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; Association for Computational Linguistics: Dublin, Ireland, 2019; pp. 5679–5688.
14. Zhang, M.; Qian, T. Convolution over Hierarchical Syntactic and Lexical Graphs for Aspect Level Sentiment Analysis. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online. 16–20 November 2020; Association for Computational Linguistics: Dublin, Ireland, 2020; pp. 3540–3549.
15. Chen, C.; Teng, Z.; Zhang, Y. Inducing Target-Specific Latent Structures for Aspect Sentiment Classification. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online. 19–20 November 2020; Association for Computational Linguistics: Dublin, Ireland, 2020; pp. 5596–5607.
16. Wang, K.; Shen, W.; Yang, Y.; Quan, X.; Wang, R. Relational Graph Attention Network for Aspect-Based Sentiment Analysis. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online. 6–8 July 2020; Association for Computational Linguistics: Dublin, Ireland, 2020; pp. 3229–3238.
17. Tian, Y.; Chen, G.; Song, Y. Aspect-Based Sentiment Analysis with Type-Aware Graph Convolutional Networks and Layer Ensemble. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online. 6–11 June 2021; Association for Computational Linguistics: Dublin, Ireland, 2021; pp. 2910–2922.

18. Li, R.; Chen, H.; Feng, F.; Ma, Z.; Wang, X.; Hovy, E. Dual Graph Convolutional Networks for Aspect-Based Sentiment Analysis. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online. 1–6 August 2021; Association for Computational Linguistics: Dublin, Ireland, 2021; pp. 6319–6329.
19. Wu, H.; Zhang, Z.; Shi, S.; Wu, Q.; Song, H. Phrase Dependency Relational Graph Attention Network for Aspect-Based Sentiment Analysis. *Knowl.-Based Syst.* **2022**, *236*, 107736. [CrossRef]
20. Wang, X.; Zhu, M.; Bo, D.; Cui, P.; Shi, C.; Pei, J. AM-GCN: Adaptive Multi-Channel Graph Convolutional Networks. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Long Beach, CA USA, 6–10 July2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1243–1253.
21. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics: Dublin, Ireland, 2019; pp. 4171–4186.
22. Xu, H.; Liu, B.; Shu, L.; Yu, P. BERT Post-Training for Review Reading Comprehension and Aspect-Based Sentiment Analysis. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics: Dublin, Ireland, 2019; pp. 2324–2335.
23. Song, Y.; Wang, J.; Jiang, T.; Liu, Z.; Rao, Y. Attentional Encoder Network for Targeted Sentiment Classification. *arXiv* **2019**, arXiv:1902.09314.
24. Jawahar, G.; Sagot, B.; Seddah, D. What Does BERT Learn about the Structure of Language? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 22019; Association for Computational Linguistics: Dublin, Ireland, 2019; pp. 3651–3657.
25. Xiao, Z.; Wu, J.; Chen, Q.; Deng, C. BERT4GCN: Using BERT Intermediate Layers to Augment GCN for Aspect-Based Sentiment Classification. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021; Association for Computational Linguistics: Dublin, Ireland, 2021; pp. 9193–9200.
26. Tang, H.; Ji, D.; Li, C.; Zhou, Q. Dependency Graph Enhanced Dual-Transformer Structure for Aspect-Based Sentiment Classification. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online. 6–8 July 2020; Association for Computational Linguistics: Dublin, Ireland, 2020; pp. 6578–6588.
27. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
28. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2018**, arXiv:1710.10903.
29. Pylkkänen, L. The Neural Basis of Combinatory Syntax and Semantics. *Science* **2019**, *366*, 62–66. [CrossRef] [PubMed]
30. Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, 23–24 August 2014; Association for Computational Linguistics: Dublin, Ireland, 2014; pp. 27–35.
31. Dong, L.; Wei, F.; Tan, C.; Tang, D.; Zhou, M.; Xu, K. Adaptive Recursive Neural Network for Target-Dependent Twitter Sentiment Classification. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Baltimore, MD, USA, 22–27 June 2014; Association for Computational Linguistics: Dublin, Ireland, 2014; pp. 49–54.
32. Li, X.; Bing, L.; Lam, W.; Shi, B. Transformation Networks for Target-Oriented Sentiment Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; Association for Computational Linguistics: Dublin, Ireland, 2018; pp. 946–956.
33. Xu, H.; Liu, S.; Wang, W.; Deng, L. RAG-TCGCN: Aspect Sentiment Analysis Based on Residual Attention Gating and Three-Channel Graph Convolutional Networks. *Appl. Sci.* **2022**, *12*, 12108. [CrossRef]

applied sciences

MDPI

Article

Integration of Multi-Branch GCNs Enhancing Aspect Sentiment Triplet Extraction

Xuefeng Shi [1,2,3], Min Hu [1,2,3,*], Jiawen Deng [4], Fuji Ren [5,*], Piao Shi [1,2,3] and Jiaoyun Yang [1,2,4]

[1] School of Computer and Information, Hefei University of Technology, Hefei 230601, China
[2] Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology, Hefei 230601, China
[3] Key Laboratory of Affective Computing and Advanced Intelligent Machine, Hefei University of Technology, Hefei 230601, China
[4] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[5] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
* Correspondence: jsjxhumin@hfut.edu.cn (M.H.); renfuji@uestc.edu.cn (F.R.)

Abstract: Aspect Sentiment Triplet Extraction (ASTE) is a complex and challenging task in Natural Language Processing (NLP). It aims to extract the triplet of aspect term, opinion term, and their associated sentiment polarity, which is a more fine-grained study in Aspect Based Sentiment Analysis. Furthermore, there have been a large number of approaches being proposed to handle this relevant task. However, existing methods for ASTE suffer from powerless interactions between different sources of textual features, and they usually exert an equal impact on each type of feature, which is quite unreasonable while building contextual representation. Therefore, in this paper, we propose a novel Multi-Branch GCN (MBGCN)-based ASTE model to solve this problem. Specifically, our model first generates the enhanced semantic features via the structure-biased BERT, which takes the position of tokens into the transformation of self-attention. Then, a biaffine attention module is utilized to further obtain the specific semantic feature maps. In addition, to enhance the dependency among words in the sentence, four types of linguistic relations are defined, namely part-of-speech combination, syntactic dependency type, tree-based distance, and relative position distance of each word pair, which are further embedded as adjacent matrices. Then, the widely used Graph Convolutional Network (GCN) module is utilized to complete the work of integrating the semantic feature and linguistic feature, which is operated on four types of dependency relations repeatedly. Additionally, an effective refining strategy is employed to detect whether word pairs match or not, which is conducted after the operation of each branch GCN. At last, a shallow interaction layer is designed to achieve the final textual representation by fusing the four branch features with different weights. To validate the effectiveness of MBGCNs, extensive experiments have been conducted on four public and available datasets. Furthermore, the results demonstrate the effectiveness and robustness of MBGCNs, which obviously outperform state-of-the-art approaches.

Keywords: ASTE; biaffine attention; structure-biased BERT; GCN; linguistic feature

Citation: Shi, X.; Hu, M.; Deng, J.; Ren, F.; Shi, P.; Yang, J. Integration of Multi-Branch GCNs Enhancing Aspect Sentiment Triplet Extraction. *Appl. Sci.* **2023**, *13*, 4345. https://doi.org/10.3390/app13074345

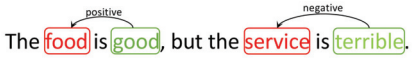
Academic Editors: Xiangjie Kong, Wei Wang and Han Liu

Received: 20 February 2023
Revised: 25 March 2023
Accepted: 27 March 2023
Published: 29 March 2023

1. Introduction

Recently, a tremendous advance has been achieved in the development of social media platforms, which largely encourage people to express their emotional states online [1,2]. Furthermore, it has become popular to publish users' comments or opinions about services and products on specific electronic platforms in a timely manner. These perspectives expressed directly by consumers are extremely important for merchants to improve their service while in a dealing. Thus, how to extract the exact aspect terms, opinion terms, and their corresponding sentiment from a specific sentence is a significant Natural Language Processing (NLP) subtask [3–5]. The recent developing task, Aspect-Based Sentiment

Analysis (ABSA), aims to mine the explicit or implicit sentiment information about the opinion terms with regard to the specific aspect terms, which implements sentiment analysis about consumers' reviews effectively. Generally, the ABSA task contains seven fundamental subtasks (Figure 1), which are Aspect Term Extraction (ATE) [6], Aspect Term Extraction and Sentiment Classification (AESC) [7], Opinion Term Extraction (OTE) [6], Aspect-Based Sentiment Classification (ABSC) [8], Aspect-Oriented Opinion Term Extraction (AOE) [9], Pair Extraction (PE) [10], and Aspect Sentiment Triplets Extraction (ASTE) [11]. In particular, as the fine-grained subtask in ABSA, the ASTE task takes aspect terms, opinion terms, and sentiment polarities into consideration simultaneously, which is challenging but significant. For example, as shown in Figure 1, the review "The food is good, but the service is terrible". contains two triplets, (food, good, positive) and (service, terrible, negative). Unlike the other subtasks, such triplets extracted by the ASTE task can better reflect multiple emotional factors (aspect, opinion, sentiment) from the user reviews and are more proper for practical application scenarios.



Tasks	Definitions	Targets
ATE	aims to extract aspect terms from the text.	food, service
AESC	aims to extract aspect terms from the text, and classifies its sentiment polarity.	(food, positive), (service, negative)
OTE	aims to extract opinion terms from the text.	good, terrible
ABSC	aims to predict the sentiment polarity with respect to the specific aspect terms.	food => positive, service => negative
AOE	aims to extract opinion terms according to aspect terms.	food => good, service => terrible
PE	aims to extract aspect and opinion terms simultaneously.	(food, good), (service, terrible)
ASTE	aims to complete the extraction of aspect and opinion terms and the prediction of sentiment polarity simultaneously.	(food, good, positive), (service, terrible, negative)

Figure 1. Example of the task categories in (ABSA) Aspect-Based Sentiment Analysis.

In previous studies, the pipeline manner is widely applied in the approaches to ASTE. Peng et al. [12] first introduced the ASTE task and extracted the triplet {aspect, opinion, sentiment} via utilizing a pipeline method, which contains a two-stage framework. The first stage provided predictions about aspect, opinion, and sentiment, respectively. Furthermore, the second stage was designed to pair up the predictions achieved from the first stage and output triplets. However, the interactions among them were totally ignored, and the potential error was propagated between these two stages [13,14]. To take the dependencies among the multiple subtasks into consideration, the multi-turn machine reading comprehension (MRC) manner [15,16] was utilized to jointly train multiple subtasks together, and it has achieved significant results. In addition, the fashion of end-to-end [17,18] also attracts many researchers' attentions, which is constructed based on the new tagging scheme.

Although the paradigm of the framework is important to enhance the performance of the ASTE task, the effective utilization of various linguistic relations between words is also decisive to the task's success [19]. Specifically, the syntactic dependency tree is widely used to present the structure of a sentence, which tends to depict the syntactic relations among words. Zhao et al. [20] adopted the dependency tree as the support to capture relations between aspect and opinion terms. Furthermore, the work [21] directly employed an interactive attention mechanism to integrate syntactic and semantic relations between words. In addition, the contribution of part-of-speech categories to ASTE is also noticed, which straightly impacts the semantic representation of sentences. Except for the

dependency tree, relative position also largely influences the expression of the sentence. Xu et al. [22] applied a position-aware tagging scheme to mark the relative position between words in a sentence. Furthermore, the semantic features in this work are represented by Long Short-term Memory (LSTM) with the pre-trained Glove, which cannot handle contextual ambiguity comprehensively. Moreover, the tree-based distance and relative position distance of each word pair in the sentence also contribute a lot to the improvement in the ASTE task [23], and the utilization of Bidirectional Encoder Representation from Transformers (BERT) can largely enhance the feature representation from the semantic perspective. However, although significant progress has been achieved by previous studies, there are still remaining limitations: the effective optimization of semantic features is not enough, and the powerful utilization of multi-type textual features is unsolved yet.

To address these two problems, motivated by the impressive performance achieved by BERT, we propose a novel BERT- and Graph Convolutional Network-based (GCN-based) model Multi-branch Graph Convolutional Network (MBGCN) for the ASTE task. In detail, in our model, to evacuate the potential capability of BERT and obtain a more exquisite contextual representation, a structure-biased BERT [24] is firstly utilized as the semantic feature encoder. Subsequently, depending on the generated representations, aspect-oriented and opinion-oriented feature maps are extracted by two multi-layer perceptions (MLP). Then, before incorporating other relations of words, a biaffine attention module is applied to unify the aspect-oriented and opinion-oriented semantic features effectively. Unlike fusing textual features via a single GCN, an MBGCN employs four branch GCNs to integrate semantic representation with syntactic dependency type, part-of-speech combination, tree-based distance, and relative position distance among each word pair, respectively. Through the complementary of these four branches, a more precise textual representation is achieved. Finally, a shallow interaction strategy is designed to complete the work of information fusion before the triplet decoding layer. To validate the effectiveness of the MBGCN, a series of experiments are conducted on four widely used and available datasets. The experimental results prove that MBGCNs can efficiently deal with the complex relations among sentences and outperform the state-of-the-art (SOTA) ASTE approaches.

The main contributions of this work can be summarized as follows:

- We propose a framework MBGCN to extract the aspect, opinion, and sentiment triplet from review sentences in an end-to-end fashion, which can avoid error propagation among different subtasks;
- We utilize a structure-biased BERT to improve the ability to extract abundant contextual information, which provides rich textual features for subsequent task-oriented operations;
- Our proposed MBGCN adopts four branch GCNs to integrate the semantic feature with four types of linguistic relations, including syntactic dependency type, part-of-speech combination, tree-based distance, and relative position distance of each word pair. Furthermore, a shallow interaction layer is introduced to output the final textual representation;
- The extensive experiments conducted on multiple ASTE datasets prove that the proposed MBGCN outperforms the mentioned SOTA baselines.

The remainder of this article is organized as follows. In Section 2, we present a brief overview of the development of ABSA, previous research about ASTE, and the application of GCNs. The proposed framework MBGCN is introduced in detail in Section 3. In Section 4, we provide detailed experimental studies and performance analyses. Finally, Section 5 provides a conclusion of this study and an outlook for future work.

2. Related Works

In the past decade, fine-grained sentiment analysis and opinion extraction have been attractive research in the NLP community, and have firmly attracted many researchers' attentions. In this section, we will first briefly review the development of the ABSA task.

Secondly, a succinct summary of existing approaches for ASTE will be introduced. Lastly, the application of GCNs in ASTE will be shortly summarized.

2.1. Aspect-Based Sentiment Analysis

ABSA is a fine-grained task that aims to recognize the explicit or implicit sentiment information in a given sentence [25–27]. Normally, a sentence usually includes several aspect terms and opinion terms simultaneously, which means multiple sentiment expressions are contained in it. Specifically, with the development of e-commerce, this situation usually happens in the reviews of products and services, which are published on online platforms [28,29]. Through mining the opinions from these reviews, the merchants can learn the real and direct requests from consumers about their services. Thus, many efforts have been contributed to this task since it was proposed. Additionally, we can categorize the existing ABSA approaches into three types: the lexicon-based method [30], machine learning method [31], and deep learning method [32]. In traditional methods, the performance of the ABSA task largely depends on feature engineering, such as bag-of-words [33] and part-of-speech [23]. Although impressive performance has been achieved by traditional methods, the cost of handcrafted features is unbearable for human experts. Currently, the rapid development of deep learning promotes the improvement in contextual representation, which also encourages the progress of ABSA tasks straightforwardly [34,35]. In deep learning methods, they usually fine-tune the pre-trained language model (PLM) with the specific training data to generate task-oriented feature maps. As a representative of PLM, BERT makes a remarkable impression on vast NLP researchers with its outstanding ability to model contextual information. Thus, it is also utilized as a backbone in our proposed model for the ASTE task.

2.2. ASTE Methods

As a subtask of sentiment analysis, ASTE has been studied by many NLP researchers after being proposed [36,37], and aims to extract aspect terms, opinion terms, and the corresponding sentiment polarity in a sentence, simultaneously. From the above investigation, it has been known that the pipeline manner method proposed by [12] had an error propagation problem between different subtasks. However, the methods with an end-to-end manner can avoid this problem with their unique architecture. Chen et al. [11] decomposed the ASTE task into three subtasks: target tagging, opinion tagging, and sentiment tagging. Furthermore, a new target-aware tagging scheme was used to identify the correspondences between opinion targets and the whole sentence. In addition, span-level features also contribute a lot to the ASTE task. Chen et al. [38] proposed a joint training framework to process all potential entities as independent spans, and the related representations of the spans were utilized to classify their corresponding sentiment polarities. Moreover, to reduce the cost of sequence tagging, a tagging-free solution was proposed by Mukherjee et al. [39]. In the method, an encoder–decoder architecture with a pointer network-based decoding framework was introduced, which effectively captured the interactions between the aspects and opinions by considering the whole detected spans in predicting sentiment polarity. To prove the simple span-based method is also effective for ASTE, Xu et al. [40] proposed a three-layers framework, which consisted of a BERT-based encoding layer, a span representation layer, and an aspect–sentiment–opinion prediction layer. This work verified that the performance of the model for ASTE was impacted by explicit local context information largely. Through the above summary, it is obviously learned that the approaches with the end-to-end manner contribute a lot to the ASTE task, and it is essential to pay more attention to the research of effectively utilizing the relations among words in a sentence. Thus, in this work, we propose a novel model to integrate five kinds of words' relations together to enhance the performance of ASTE.

2.3. Application of GCN in ASTE

In (ABSA) Aspect-Based Sentiment Analysis tasks, the syntax dependency tree plays an important role in catching the key feature from the review sentence [41–43]. Furthermore, it is well known that the regular method GCN is popular in handling dependency graphs in previous works. Regarding ASTE, GCNs are also used widely to fuse different sources of information. As mentioned above, Shi et al. [21] employed a GCN to enhance the interaction between syntactic and semantic features. To fully exploit the potential information implied in syntactic and semantic features, the work [18] also integrated semantic and syntactic representations through a GCN module, which preserved the sequential information and enhanced the linguistic representation, simultaneously. Moreover, to overcome the problem of many aspect terms to one opinion term or one aspect term to many opinion terms, Li et al. [44] combined a GCN with a base encoder to build the span representations, which included both aspect terms and opinion terms. In [45], a GCN was also employed to model the graph based on the concatenated representations of aspects terms and opinion terms. Thus, it is quite clear that GCNs are extremely important in enhancing the feature representations in the ASTE task. Motivated by their impressive ability, we also process the work of feature fusion under the guidance of the GCN in this paper.

Conclusively, as aforementioned, ASTE is a difficult and challenging subtask in ABSA, which attracts a lot of researchers' attentions. In this paper, inspired by the existing works which apply BERT and GCNs to NLP tasks, we propose a novel model MBGCN to process semantic feature, syntactic dependency type, part-of-speech combination, tree-based distance, and relative position distance, simultaneously.

3. Framework of MBGCN

In this section, the detailed framework of the MBGCN is described. Firstly, the definition of the ASTE task is introduced briefly. Then, the mechanism of feature generation through the backbone structure-biased BERT is depicted, and this step is utilized to generate semantic features. After that, multi-branch GCNs are employed to integrate semantic features with the other four types of linguistic feature representations. Lastly, the shallow interaction, output layer, and training are introduced shortly. Additionally, the overall architecture of the MBGCN is described in detail in Figure 2.

3.1. Task Formulation

Given a sentence with a sequence of words $\mathbf{X} = \{w_1, w_2, \ldots, w_n\}$ as input, where n is the number of words, the goal of the ASTE task is to extract and output a set of triplets $\{(a, o, s)_k\}_{k=1}^{m}$, where a, o, and s are the aspect term, opinion term, and the corresponding sentiment polarity, respectively, and m is the number of triplets. Concretely, the aspect a can be decomposed into two or more elements, i.e., (a_b, a_e), where b and e mean the start and end positions. The opinion o can be decomposed as (o_b, o_e) similarly. Furthermore, s is selected from the set (position, netural, negative) to represent the sentiment polarity of the corresponding opinion term on the aspect term. For the sentence shown in Figure 1, the triplets are collected as (food, good, positive) and (service, terrible, negative).

Specifically, to make the target of our ASTE task more explicit, ten types of relations between words in a review are defined, which are collected in Table 1. Similarly, the mentioned relations also can be seen as the labels, and these labels are introduced to present the relations in the word pairs, which are also the eventual predictions of our MBGCN.

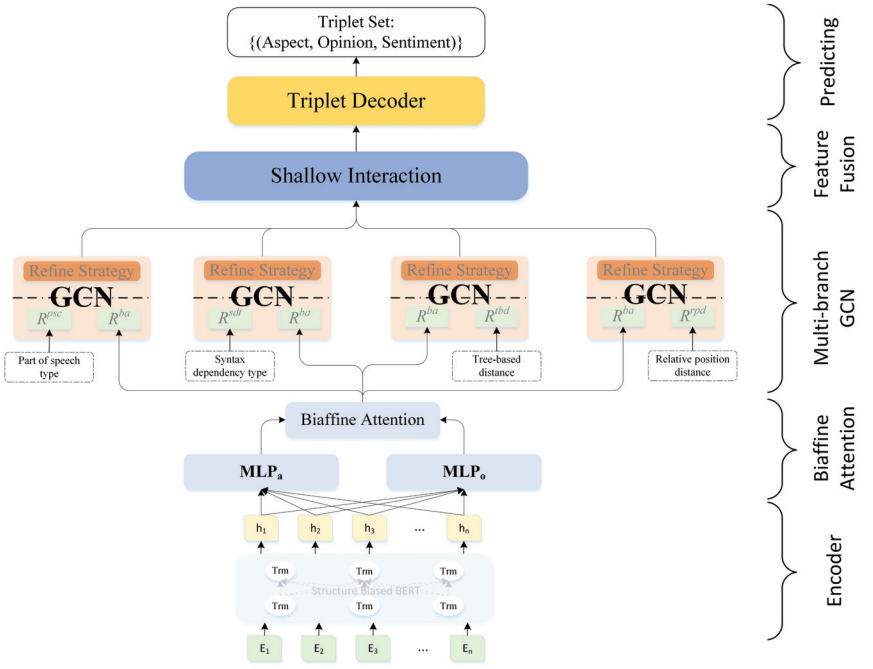


Figure 2. The overall framework of the proposed Multi-branched Graph Convolutional Network (MBGCN). $[E_1, E_2, E_3, \ldots, E_n]$ is the input vector **E** of self-attention (Equations (1) and (2)).

Table 1. The definitions of our defined relations.

Items	Relation	Definition
1	B-A	beginning of aspect term.
2	I-A	inside of aspect term.
3	A	aspect term.
4	B-O	beginning of opinion term.
5	I-O	inside of opinion term.
6	O	opinion term.
7	POS	sentiment polarity is positive.
8	NEU	sentiment polarity is neutral.
9	NEG	sentiment polarity is negative.
10	N	belong to no aforementioned relations.

3.2. Embedding via Structure-Biased BERT

As aforementioned, BERT has an impressive performance in modeling contextual representation in various NLP tasks [46–49]. Therefore, in our proposed Multi-branches Graph Convolutional Netwrok (MBGCN),we also utilize it to generate the semantic features by the version of the `bert-uncased-base`. To be precise, before feeding the review **X** into the MBGCN, the input is always formulated in three formats: segment embedding X_s, position embedding X_p, and tokens embedding X_t. Then, these three aspects of embedding are summarized as the input to the selective feature generator, which is shown in Equation (1),

$$\mathbf{E} = X_s + X_p + X_t, \tag{1}$$

where $\mathbf{E} = [E_1, E_2, E_3, \ldots, E_n]$ is the input of self-attention (Equation (2)). In detail, BERT is a PLM with the structure of a stacked transformer, which has 12 transformer layers in total. Furthermore, in each transformer layer, the feature representations are trans-

formed by multi-head self-attention with a residual structure (Figure 3a). Furthermore, this transformation can be formulated as follows:

$$h^0 = LN(\mathbf{E}), \tag{2}$$

$$\hat{h}^l = LN(h^{l-1} + MHSA(h^{l-1})), \tag{3}$$

$$h^l = LN(\hat{h}^l + FFN(\hat{h}^l)), \tag{4}$$

where $l \in [1, 12]$ is the l-th layer transformer, and h^0 means the input embedding of BERT, which is built from $\mathbf{E}$ with a liner function. The outputs of 12-layer transformers are denoted as $[h^1, h^2, \ldots, h^{12}]$. FFN includes two linear functions with a $ReLU$ activation function between. $MHSA$ is the core of the transformer, which has a stacked structure with 12 heads of self-attention. Thus, we can formulate the architecture of attention as follows:

$$\hat{h}^l_{Mj} = softmax(e^l_j)(h^{l-1}\mathbf{W}_V), \tag{5}$$

$$e^l_j = \frac{h^{l-1}\mathbf{W}_Q(h^{l-1}\mathbf{W}_K)^\top}{\sqrt{d}}), \tag{6}$$

$$\hat{h}^l_M = \sum_{j=1}^{N} \hat{h}^l_{Mj}, \tag{7}$$

where parameters $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ are the learnable weights for query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$, and d is the head dimensionality. $\hat{h}^l_{Mj}$ is the single attention, and $\hat{h}^l_M$ denotes the sum of N heads attention (MHSA).

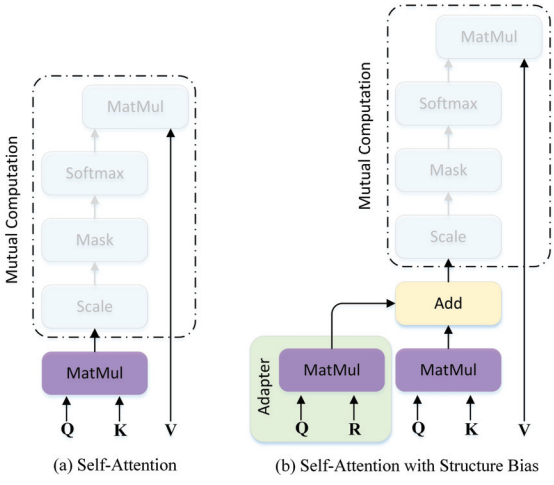


Figure 3. The mechanism of structure bias utilized in BERT. Furthermore, **Q**, **K**, and **V** denote the query vector, key vector, and value vector, respectively, which are standardized inputs for the transformer module. **R** indicates the relation distance embedding (Equation (8)).

Inspired by the structure-biased BERT utilized in [24,50], we also introduce it into our MBGCN for generating more informative feature maps. In the optimized approach, self-attention is re-constructed by inserting the relative distance or the dependency between words. Furthermore, the effectiveness of this modification has been obviously proven by

the NLP task [51]. Thus, we describe this change in our model as Equation (8), which can be implemented in Equation (6) directly. Additionally, the procedure is depicted below:

$$\begin{aligned} e_j^l &= \frac{h^{l-1}\mathbf{W}_Q(h^{l-1}\mathbf{W}_K + \mathbb{R}^{l-1})^\top}{\sqrt{d}} \\ &= \underbrace{\frac{h^{l-1}\mathbf{W}_Q(h^{l-1}\mathbf{W}_K)^\top}{\sqrt{d}})}_{\text{Raw}} + \underbrace{\frac{h^{l-1}\mathbf{W}_Q(\mathbb{R}^{l-1})^\top}{\sqrt{d}}}_{\text{Bias}}, \end{aligned} \tag{8}$$

where $\mathbf{R}^{l-1} \in \mathcal{R}^{k\times k}$ indicates the relative distance embedding between the word pairs of the k-th sentence in $(l-1)$-th transformer layer. Note that each dependency embedding is independent from one layer to another layer, but it can be transformed across different heads as an entirety. Additionally, the sketch of the difference between raw self-attention (a) and biased self-attention (b) is shown in Figure 3.

With the backbone encoder of structure-biased BERT, the semantic features h^l is obtained, which provides more accurate contextual information to the module of biaffine attention.

3.3. Biaffine Attention

Biaffine attention has been proven to have the ability to capture the relationship among the different words or word pairs [23,52]. Thus, in this paper, we also apply it to predict the relation probability of word pairs in a sentence. To present the process of biaffine attention, the hidden states h_ς and h_τ of w_ς and w_τ in $\mathbf{X}$ are extracted from h^l. With the aforementioned $\mathbf{MLP_a}$ and $\mathbf{MLP_o}$, the aspect-specific feature h_ς^a (Equation (9)) and opinion-specific feature h_τ^o (Equation (10)) are obtained, which are adopted into the processing of biaffine attention directly.

$$h_\varsigma^a = \mathbf{MLP_a}(h_\varsigma), \tag{9}$$

$$h_\tau^o = \mathbf{MLP_o}(h_\tau). \tag{10}$$

and the transformation of biaffine attention can be formulated as

$$g_{\varsigma,\tau} = h_\varsigma^{a\top}\mathbf{U}_o h_\tau^o + \mathbf{U}_a(h_\varsigma^a \oplus h_\tau^o) + \mathbf{b}, \tag{11}$$

$$R_{\varsigma,\tau,\xi} = \frac{\mathbf{exp}(g_{\varsigma,\tau,\xi})}{\sum_{\xi=1}^{\Xi}\mathbf{exp}(g_{\varsigma,\tau,\xi})}, \tag{12}$$

where $\mathbf{U}_o$, $\mathbf{U}_a$, and $\mathbf{b}$ are the trainable weights and biases, and $\oplus$ denotes the operation of concatenation. The relations between w_ς and w_τ are modeled as $R_{\varsigma,\tau} \in \mathbb{R}^{1\times\Xi}$. Ξ is the number of relation types. Furthermore, we use R^{ba} to represent the relations obtained in this manner in the following sections.

With the aspect-oriented and opinion-oriented processing of biaffine attention, the probability of relation R^{ba} between the word pairs in a sentence can be modeled effectively. Furthermore, this relation will be integrated with the other four types of linguistic features via GCNs adequately.

3.4. Multi-Branch GCN

Except for encoding text as semantic feature maps, it also can be represented in the linguistic feature types. Furthermore, the most widely utilized type is the syntax dependency graph, where the feature is formed in a graph $G = (V, E)$. V is the vertex (i.e., node or word), and E is the edge (i.e., dependency or syntactic relation) between two nodes. Generally, we usually denote this kind of relation through a matrix, namely adjacent matrix A. $A_{\tau,\varsigma} = 1$ if the relation between w_ς and w_τ exists, and $A_{\tau,\varsigma} = 0$ otherwise. In addition, in this paper, we also introduce three other types of linguistic features for each word pair to enhance the contextual representation of the sentence, which are the part-of-speech combination, tree-based distance, and relative position distance (Figure 4). Before

feeding them into the GCN, they are all encoded in the fashion of adjacency matrices. Then, these four types of linguistic features are integrated with semantic features, respectively. For instance, we apply the GCN to integrate the R^{ba} with R^{sdt} encoded from syntactic dependency type tensor E_{sdt}, and the process is depicted as follows:

$$R^{sdt}_{\tau,\varsigma} = \sigma(E_{sdt}), \tag{13}$$

$$H^{f1}_{\tau,\varsigma} = f((R^{ba}_{\tau,\varsigma} \oplus R^{sdt}_{\tau,\varsigma})H^{Bert}_{\tau,\varsigma}), \tag{14}$$

where H^{Bert} is obtained from the original contextual representation h^l through a dense layer and a $ReLU$ activation layer; σ is the function $softmax$. Furthermore, $f(\cdot)$ is an average pooling function applied on the node hidden representations of all channels. To make the extracted relations more accurate, a refining strategy is employed to enhance the relations among words, which can be described as

$$R^{F}_{\tau,\varsigma} = h^{l}_{\tau,\varsigma} \oplus E^{sdt}_{\tau,\varsigma}, \tag{15}$$

$$\hat{F}^{G(\tau,\varsigma)}_{ba+sdt} = R^{F}_{\tau,\varsigma} \oplus R^{F}_{\varsigma,\varsigma} \oplus R^{F}_{\tau,\tau} \oplus H^{f1}_{\varsigma} \oplus H^{f1}_{\tau}, \tag{16}$$

$$F^{G(\tau,\varsigma)}_{ba+sdt} = \sigma(LN(\hat{F}^{G(\tau,\varsigma)}_{ba+sdt})), \tag{17}$$

we use $\oplus$ to concatenate contextual representation $h^l_{\tau,\varsigma}$ and syntactic dependency type tensor $E^{sdt}_{\tau,\varsigma}$. Furthermore, in Equation (16), $R^F_{\tau,\tau}$ and $R^F_{\varsigma,\varsigma}$ are the main diagonal and vice diagonal, which are used to refine the representation $\hat{F}^{G(\tau,\varsigma)}_{ba+sdt}$. Finally, with the operations of a linear layer and a softmax layer, the distribution of probabilities on ten defined relations between τ and ς is obtained, which is denoted as $F^{G(\tau,\varsigma)}_{ba+sdt}$.

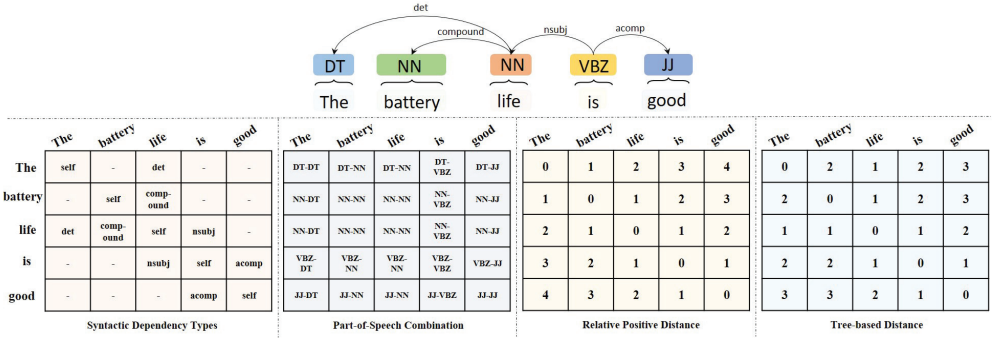


Figure 4. The example of four mentioned types of dependency relations among words in reviews.

Similarly, we integrate $R^{ba}_{\tau,\varsigma}$ with $E^{psc}_{\tau,\varsigma}$, $E^{tbd}_{\tau,\varsigma}$, and $E^{rpd}_{\tau,\varsigma}$ via different branches of the GCN to obtain the refined feature representations $F^{G(\tau,\varsigma)}_{ba+psc}$, $F^{G(\tau,\varsigma)}_{ba+tbd}$ and $F^{G(\tau,\varsigma)}_{ba+rpd}$, respectively. Through the operations described in this part, we enhance the contextual feature $R^{ba}_{\tau,\varsigma}$ with four types linguistic features, respectively.

3.5. Shallow Interaction and Output Layer

To further enhance the performance of our MBGCN, we apply a shallow interaction layer to fuse the four types of integrated feature representations, which can be depicted as follows:

$$T^{F}_{\tau,\varsigma} = [\alpha, \beta, \gamma, \mu] \cdot [F^{G(\tau,\varsigma)}_{ba+sdt}, F^{G(\tau,\varsigma)}_{ba+psc}, F^{G(\tau,\varsigma)}_{ba+tbd}, F^{G(\tau,\varsigma)}_{ba+rpd}]^{\top}, \tag{18}$$

where α, β, γ and μ are manually selective hyper-parameters to control the weights of different feature representations. Furthermore, $\top$ is the transposition operation for the

related matrix. With this layer, the MBGCN achieves the final textual representation fused from four branches of the GCN, which take five types of textual features into consideration simultaneously.

3.6. Training

Generally, the deep learning models are always optimized by minimizing a loss function, and cross entropy is usually applied to complete this work. Without simply applying cross entropy in the proposed MBGCN, due to various contextual information involved, it is necessary to take these into the final fine tuning. For instance, the separated loss $\mathcal{L}_{ba}$ to measure the influence of R^{ba} is modeled as

$$\mathcal{L}_{ba} = -\sum_{\varsigma}^{n}\sum_{\tau}^{n}\sum_{\xi\in\Xi}\mathbb{I}(y_{\varsigma,\tau}=\xi)log(g_{\varsigma,\tau|\xi}), \tag{19}$$

where $\mathbb{I}(\cdot)$ is the indicator function, $y_{\varsigma,\tau}$ is the ground truth relation of word pair (w_ς, w_τ). Furthermore, Ξ denotes the whole relations set. With a similar operation, the other four separated linguistic features' losses $\mathcal{L}_{psc}$, $\mathcal{L}_{sdt}$, $\mathcal{L}_{tbd}$, and $\mathcal{L}_{rpd}$ are all obtained likewise. Thus, with the prediction, the final loss function $\mathcal{L}$ in the paper is designed as

$$\mathcal{L} = \mathcal{L}_{T^F} + \rho\mathcal{L}_{ba} + \kappa(\mathcal{L}_{psc} + \mathcal{L}_{sdt} + \mathcal{L}_{tbd} + \mathcal{L}_{rpd}), \tag{20}$$

$$\mathcal{L}_{T^F} = \mathbb{I}(Y=\Xi)log(T^F|\Xi) \tag{21}$$

where ρ and κ are the manual hyper-parameters to control the influence of each part on the final loss function. Through this manner, our MBGCN can adjust its fine-tuning from six aspects simultaneously.

4. Experiments and Discussion

In this section, the results of the conducted experiments are depicted in Tables and Figures, and the corresponding analyses are also provided in detail. We first introduce the widely used ASTE datasets and the related settings of experiments. The detailed experimental results are clearly shown in the analysis secondly.

4.1. Datasets

In this paper, extensive experiments are conducted on four benchmarks, namely Laptop14, Restaurant14, Restaurant15, and Restaurant16, which are public and available for ASTE tasks. Furthermore, these four datasets are all collected from the SemEval ABSA challenges [53–55]. It's worth noting that these four datasets are revised by Wu et al. [56] and Xu et al. [22] for ASTE tasks, respectively, which are denoted as $\mathcal{V}_1$ and $\mathcal{V}_2$ in this paper. Moreover, the statistics for these two versions of datasets are shown in Table 2.

Table 2. Statistics of two groups of experiment datasets.

Datasets		Laptop14		Restaurant14		Restaurant15		Restaurant16	
		#S	#T	#S	#T	#S	#T	#S	#T
$\mathcal{V}_1$	train	899	1452	1259	2356	603	1038	863	1421
	dev	225	383	315	580	151	239	216	348
	test	332	547	493	1008	325	493	328	525
$\mathcal{V}_2$	train	906	1460	1266	2338	605	1013	857	1394
	dev	219	346	310	577	148	249	210	339
	test	328	543	492	994	322	485	326	514

Note: #S denotes the number of sentences; #T means the number of triplets contained in the datasets.

4.2. Experimental Setup

To conduct the extensive experiments successfully, we use the BERT [57] with structure bias as our review encoder, and it consists of 12 transformer layers, where 12 heads self-attention are included in each layer. Furthermore, the size of the hidden state in self-attention is 768. In addition, the total number of the model's parameters is approximately 110 M. The optimizer AdamW is employed to optimize the training process, where the learning rate is set as 2×10^{-5}. The dropout is set to 0.5. Moreover, we train our model with 100 epochs with a batch size of 8. The hyper-parameters α, β, γ, and μ of fusion work in Equation (18) are set as 0.625, 0.125, 0.125 and 0.125, respectively, and the parameters ρ and κ to control the weights of each loss in Equation (16) are set as 0.1 and 0.01, respectively. In addition, the experiments are conducted on a system on NVIDIA GeForce RTX 3080Ti with 12GB of graphics memory. To validate our MBGCN effectively, the widely and popularly used evaluations of Precision (P), Recall (R), and macro-F1 (F1) are employed to present the performance of the proposed approach on four benchmarks.

4.3. Baselines

Specifically, to demonstrate the validity of the proposed MBGCN, we make comparisons with several existing SOTA methods designed for ASTE tasks, which are shown as follows:

- GTS-BERT [56] proposes an end-to-end tagging scheme, Grid Tagging Scheme (GTS) with cooperation with BERT, to address the extraction task;
- GTS-CNN [56] is the Grid Tagging Scheme (GTS) that cooperates with CNN;
- GTS-BiLSTM [56] is the Grid Tagging Scheme (GTS) that cooperates with BiLSTM;
- S^3E^2 [18] exploits the syntactic and semantic relationships between word pairs in a sentence by a graph-sequence dual representation and modeling paradigm for the ASTE task;
- Peng-two-stage+IOG [56] is the combination of Peng-two-stage [12] and IOG [58];
- Peng-two-stage [12] is a two-stage pipeline model. It extracts both aspect–sentiment pairs and opinion terms in the first stage, and pairs the extraction results into triplets in the second stage;
- OTE-MTL [59] treats the ABSA task as an opinion triplet extraction work, and jointly extracts aspect terms, opinion terms, and parses their sentiment via a multi-task learning framework;
- JET-BERT [22] builds a joint model to extract the triplets using a position-aware tagging approach, which is capable of jointly extracting aspect terms, opinion terms, and their sentiment together;
- BMRC [16] transforms the ASTE task into a Multi-Turn Machine Reading Comprehension (MTMRC) task, and three types of queries are devised to handle the related inputs;
- EMC-GCN [23] transforms the sentence into a multi-channel graph by treating words and edges as nodes and edges, respectively, while ten types of relations for ASTE are defined;
- MuG-Bert [24] proposes an approach, Multi-task learning with Grid decoding (MuG), to integrate the multi-task learning framework with grid triplets decoding from GTS;
- UniASTE$_{BERT}$ [11] proposes an end-to-end method that decomposes ASTE into three subtasks, namely target tagging, opinion tagging, and sentiment tagging. Furthermore, a target-aware tagging scheme is introduced to identify the correspondences between opinion targets and opinion expressions;
- Dual-MRC [15] solves the ASTE task via constructing two machine reading comprehension problems, and trains two BERT-MRC models jointly with parameters sharing.

4.4. Main Results

In this subsection, we report the main results of ASTE tasks in Table 3 for version $\mathcal{V}_1$ and Table 4 for version $\mathcal{V}_2$, respectively. According to the results reported in these two Tables, two observations can be concluded and stated as follows.

First, the performances of the PLM-based approaches on ASTE are much better than the normal word2vector-based models. Furthermore, this is quite clear in the comparisons

among GTS-BERT, GTS-CNN, and GTS-BiLSTM. Observing from Table 3, it is obvious that our proposed MBGCN acquires the optimal performance when compared with the previously mentioned SOTA baselines. To be precise, for experimental results in four benchmarks, our MBGCN achieves 72.33%, 57.46%, 59.57%, and 70.43% on the main indicator F1, respectively; while compared with the best baseline EMC-GCN†, the proposed MBGCN obtains 1.13% (72.33–71.20), 0.92% (57.46–56.54), 1.53% (59.57–58.04), and 1.40% (70.42–69.03) improvements on F1 in the four datasets, respectively, and it achieves the new SOTA. This observation from the comparison indicates the effectiveness of our proposed model with the multi-branch framework.

Second, even within the comparison with other BERT-based approaches, the MBGCN also enhances the ASTE performance through its excellent contextual understanding. Notably, in Table 4, the experimental results conducted on $\mathcal{V}_2$ are collected clearly. For the vital evaluation F1, the performances of the MBGCN are improved to 71.37%, 58.89%, 63.07%, and 67.34% in four corpora, respectively, which outperform the mentioned baselines obviously and are only a little worse than EMC-GCN† on F1 in Restaurant14. Specifically, there is a dramatic increase in F1 in Restaurant15, which is nearly 3.46% (63.07–59.61). Furthermore, in Laptop14 and Restaurant16, the MBGCN also achieves 0.58% (58.89–58.31) and 0.31% (67.34–66.74) increases on the F1 indicator. This can be viewed as direct evidence to support the usefulness of the combination of structure-biased BERT and multi-branch GCNs.

Conclusively, as the results show, structure-biased BERT-based multi-branch GCNs can further boost the performance of ASTE tasks, which is beneficial in excavating the semantic and syntactic information in reviews comprehensively.

Table 3. The performance of Multi-branches Graph Convolutional Network (MBGCN) on $\mathcal{V}_1$.

Models	Restaurant14			Laptop14			Restaurant15			Restaurant16		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Peng-two-stage+IOG© [56]	58.89	60.41	59.64	48.62	45.52	47.02	51.70	46.04	48.71	59.25	58.09	58.67
GTS-CNN© [56]	70.79	61.71	65.94	55.93	47.52	51.38	60.09	53.57	56.64	62.63	66.98	64.73
GTS-BiLSTM© [56]	67.28	61.91	64.49	59.42	45.13	51.30	63.26	50.71	56.29	66.07	65.05	65.56
GTS-BERT© [56]	70.92	69.49	70.20	57.52	51.92	54.58	59.29	58.07	58.67	68.58	66.60	67.58
S^3E^2© [18]	69.08	64.55	66.74	59.43	46.23	52.01	61.06	56.44	58.66	71.08	63.13	66.87
Dual-MRC© [15]	-	-	70.32	-	-	55.58	-	-	57.21	-	-	67.40
EMC-GCN † [23]	70.92	71.49	71.20	58.96	54.31	56.54	54.99	61.46	58.04	65.74	72.66	69.03
MBGCN	72.89	71.79	**72.33**	57.30	57.62	**57.46**	60.76	58.42	**59.57**	71.68	69.22	**70.43**

Note: The "†" denotes that we reproduce the models using released code with original parameters on the dataset. The "©" denotes the results are referred from the original paper. The "-" denotes not mentioned in original paper. And the bold format denotes the optimal performance.

Table 4. The performance of MBGCN on $\mathcal{V}_2$. The "§" means the results are retrieved from [8]. The "¶" denotes the results are retrieved from [23].

Models	Restaurant14			Laptop14			Restaurant15			Restaurant16		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Peng-two-stage § [12]	43.24	63.66	51.46	37.38	50.38	42.87	48.07	57.51	52.32	46.96	64.24	54.21
OTE-MTL ¶ [59]	62.00	55.97	58.71	49.53	39.22	43.42	56.37	40.94	47.13	62.88	52.10	56.96
JET-BERT § [22]	70.56	55.94	62.40	55.39	47.33	51.04	64.45	51.96	57.53	70.42	58.37	63.83
BMRC ¶ [16]	75.61	61.77	67.99	70.55	48.98	57.82	68.51	53.40	60.02	71.20	61.08	65.75
EMC-GCN † [23]	70.35	73.14	**71.72**	61.48	55.45	58.31	56.33	63.30	59.61	62.46	72.32	67.03
MuG-BERT© [24]	68.40	67.64	68.00	58.30	52.21	55.06	60.65	54.12	57.10	66.26	67.39	66.74
UniASTE$^{©}_{BERT}$ [11]	72.14	66.30	69.09	62.24	51.77	56.51	64.83	54.31	59.06	69.06	65.53	67.22
MBGCN	67.92	75.18	71.37	59.96	57.86	**58.89**	62.25	63.92	**63.07**	63.76	71.35	**67.34**

Note: The "†" denotes that we reproduce the models using released code with original parameters on the dataset. The "©" denotes the results are referred from the original paper. And the bold format denotes the optimal performance.

4.5. Ablation Study

To further validate the effectiveness of each component in the MBGCN, we conduct ablation experiments and answer the following questions:

- Is the contribution of each linguistic feature equal?
- Does the structure-biased BERT promote the performance of the MBGCN on the ASTE task?

4.5.1. Effect of Each Linguistic Feature

We first validate whether each type of linguistic feature is equal to improve the performance of the MBGCN in modeling the textual representation. Accordingly, a single branch GCN is constructed to integrate semantic features achieved by structure-biased BERT with a single linguistic feature. Moreover, the experimental results are shown in Figures 5 and 6.

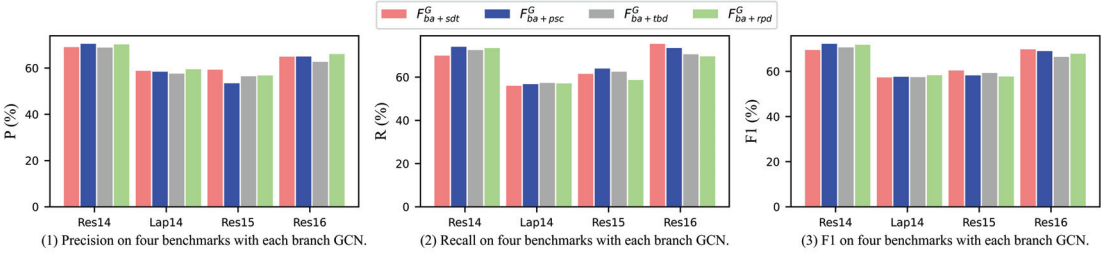


Figure 5. The Results of Ablation Study on $\mathcal{V}_1$. Res14 means Restaurant14, Lap14 means Laptop14, Res15 denotes Restaurant15, and Res16 denotes Restaurant16.

Above all, we compare the effectiveness of the four types of linguistic features on version $\mathcal{V}_1$ of the four benchmarks, respectively. From Figure 5, we can observe that the approach utilizing F^G_{ba+psc} achieves the optimal performance in the experimental results, in which the representation is implemented by semantic feature and part-of-speech combination R^{psc} only. This suggests that linguistic feature R^{psc} can largely enhance the textual representation of semantic features generated by structure-biased BERT. Conversely, the effectiveness of F^G_{ba+tbd} is slightly worse than the observation of the performance of the ablation experiments, but it still contributes to improving the model's capability to extract triplets. In addition, F^G_{ba+sdt} and F^G_{ba+rpd} both have a significant impact on improving the performance of the proposed model. Additionally, extensive experiments are conducted on version $\mathcal{V}_2$ of four datasets, and the results are presented in Figure 6. Observing from the figure, the same conclusion can be obtained from the experimental results based on F^G_{ba+psc} and F^G_{ba+tbd}. Therefore, we believe that the triplets extraction task benefits from the cooperation of all four GCN branches directly.

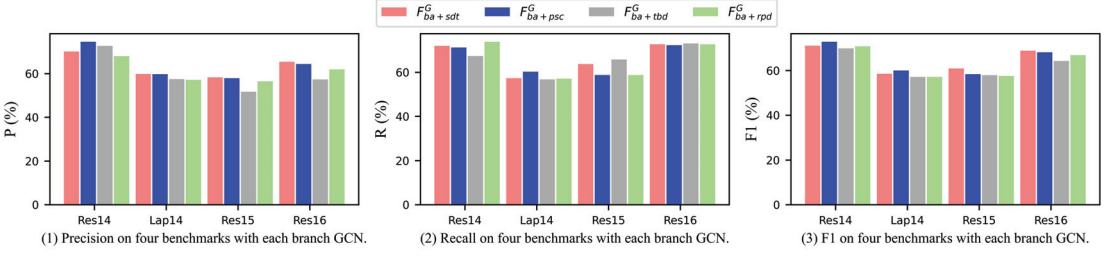


Figure 6. The Results of Ablation Study on $\mathcal{V}_2$. Res14 means Restaurant14, Lap14 means Laptop14, Res15 denotes Restaurant15, and Res16 denotes Restaurant16.

4.5.2. Effect of Adapter BERT

To validate the influence of structure-biased BERT on textual semantic representation, extensive experiments are conducted on the aforementioned datasets. Table 5 shows the results on version $\mathcal{V}_1$ and $\mathcal{V}_2$ of the four datasets. We can see that structure-biased BERT strengthens the performance of the proposed model in extracting triplets on three datasets (i.e., Restaurant14, Restaurant15, and Restaurant16). In particular, for $\mathcal{V}_1$, the model without structure bias only achieves 71.66%, 58.55%, and 68.52% of F1 on Restaurant14, Restaurant15 and Restaurant16, respectively, which are obviously worse than structure-biased BERT based MBGCN. Furthermore, when it comes to version $\mathcal{V}_2$, the method based on structure-biased BERT also achieves higher F1 scores on three datasets, which are Restaurant14, Laptop14 and Restaurant15, respectively. Furthermore, the corresponding improvements are 1.53% (71.37–69.84), 0.29% (58.89–58.70), and 3.23% (63.07–59.84). Conclusively, the above description indicates that the employment of structure-biased BERT can extract more abundant textual semantic features in the current ASTE task.

Table 5. The contribution of adapter to MBGCN for ASTE task on $\mathcal{V}_1$ and $\mathcal{V}_2$.

Versions	Models	Restaurant14			Laptop14			Restaurant15			Restaurant16		
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
$\mathcal{V}_1$	MBGCN	72.89	71.79	**72.33**	57.30	57.62	57.46	60.76	58.42	**59.57**	71.68	69.22	**70.43**
	w/o Structure bias	70.38	72.99	71.66	60.86	54.50	**57.50**	56.76	60.45	58.55	64.53	73.04	68.52
$\mathcal{V}_2$	MBGCN	67.92	75.18	**71.37**	59.96	57.86	**58.89**	62.25	63.92	**63.07**	63.76	71.35	67.34
	w/o Structure bias	69.49	70.19	69.84	59.77	57.67	58.70	58.88	60.83	59.84	65.36	71.74	**68.40**

Note: The "w/o" denotes the abbreviation for without.

4.6. Case Study

To further analyze the role of each linguistic feature in our task, two samples are selected and visualized by attention weights on each word, and they are expressed through Figure 7. As shown in the figure, each row means the visualization of the representation obtained by the single branch GCN, and each column denotes the visualization of each word presented by the four branch GCN. From Figure 7, we can conclude two observations related to the core idea of the proposed model. First, it is obvious that the attention of each branch of the GCN is attracted by the different words in the sentence. For example, in Figure 7b, the key word in branch F^G_{ba+sdt} is "is", while "is" is the last word in sorted attention sequence from F^G_{ba+psc}, and F^G_{ba+psc} gives a heavy attention weight to "this". The same conclusion can be summarized from F^G_{ba+tbd} and F^G_{ba+rpd}. Second, we find that if one branch misses the specific word, such as "is" in F^G_{ba+psc} and F^G_{ba+rpd} in Figure 7b, another branch of the GCN would provide a higher attention weight on this word, such as F^G_{ba+sdt} and F^G_{ba+tbd}. Furthermore, this phenomenon directly corresponds to the core of feature integration. In addition, Figure 7a also provides the same information to us about the attention distributions via a four branch GCN. For this case, our proposed MBGCN completes its work of fusing various branch features to enhance the textual representation and finally improve the performance of the ASTE task.

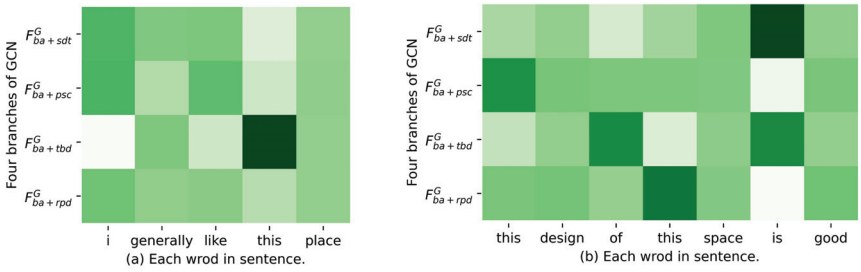


Figure 7. Attention distribution on each word in samples for case study.

4.7. Attempts via Prompt Learning

Prompt learning is the process of creating a prompt format to guide the training of the model on the downstream tasks [60]. Furthermore, from the investigation of previous research, we learn that creating intuitive templates based on human introspection is the most widely used method that has been adopted in many studies [61–63]. In addition to the mentioned strategies, we also tried to exploit the usefulness of prompt learning in our designed experiments. Following the core idea of prompt learning, the comprehensive prompt template in this task is designed as "the targets are aspect, opinion, sentiment". Thus, the input to the model is remodeled as {REVIEW, the targets are aspect, opinion, sentiment.}, which directly tells the PLM the exact target of the current task.

Furthermore, the relative experiments are conducted both on version $\mathcal{V}_1$ and $\mathcal{V}_2$, and the results are collected in Table 6. From the observations, first, we find that the improvement in the model's performance on $\mathcal{V}_1$ is confined to Laptop14 and Restaurant15, and different declines happen to the experimental results in Restaurant14 and Restaurant16. In other words, the effectiveness of prompt learning is limited for current version $\mathcal{V}_1$ under the framework of our proposed MBGCN. Second, while observing the experimental results on $\mathcal{V}_2$ shown in Table 6, we can learn that the approach with prompts outperforms the baseline MBGCN on three benchmarks clearly. However, it fails in the experiments conducted in Laptop14, which suggests that the optimized model by current prompts is not sensitive to the reviews in Laptop14. Finally, we can conclude that prompt learning with the aforementioned template can improve the model's capability in modeling textual representation in some aspects, but it is not the most proper manner for the current designed framework, which means a lot of effort is essential to improve the performance of prompt learning in the ASTE task.

Table 6. The contribution of prompt learning to MBGCN for ASTE task on $\mathcal{V}_1$ and $\mathcal{V}_2$.

Versions	Models	Restaurant14			Laptop14			Restaurant15			Restaurant16		
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
$\mathcal{V}_1$	MBGCN	72.89	71.79	**72.33**	57.30	57.62	57.46	60.76	58.42	59.57	71.68	69.22	**70.43**
	w/ Prompts	73.27	70.18	71.69	56.35	59.45	**57.86**	58.32	64.71	**61.35**	64.11	68.64	66.30
$\mathcal{V}_2$	MBGCN	67.92	75.18	71.37	59.96	57.86	**58.89**	62.25	63.92	63.07	63.76	71.35	67.34
	w/ Prompts	73.63	71.31	**72.46**	58.93	56.75	57.82	64.68	62.68	**63.67**	65.96	72.90	**69.26**

Note: The "w/" denotes the abbreviation of with.

5. Conclusions

In this work, we propose an end-to-end model MBGCN for the ASTE task, which processes Aspect Term Extraction, Opinion Term Extraction, and sentiment polarity prediction in a sentence, simultaneously. For modeling the textual semantic feature more accurately, an optimized attention module is inserted into BERT, namely structure-biased BERT, which is employed to enhance the representation of the specific sentence. In addition,

to emphasize the key features in the generated representation, biaffine attention is utilized to absorb the crucial components from both aspect-oriented and opinion-oriented feature maps. Furthermore, a novel fusion architecture with a multi-branch GCN is proposed to integrate the semantic feature with the linguistic feature. In this part, through each branch GCN, attentive semantic representation is integrated with syntactic dependency types, part-of-speech combination, relative positive distance, and tree-based distance, respectively. Eventually, four branch features are synthesized as an entirety via a designed shallow interaction layer. To validate the effectiveness of our proposed model, we conduct extensive experiments on the benchmark datasets, and the results show that the MBGCN achieves SOTA performances.

Although outstanding performances were achieved by our proposed MBGCN, several limitations still exist, which are the working aims of our future study. First, the working mechanism of prompt learning should be optimized to be more proper for our current task. Second, a more robust integration strategy is essential in our future study for feature fusion.

Author Contributions: Conceptualization, X.S., M.H. and F.R.; methodology, X.S. and F.R.; validation, X.S. and M.H.; formal analysis, X.S.; investigation, X.S., P.S. and J.Y.; writing—original draft preparation, X.S.; writing—review and editing, X.S., J.D. and F.R.; visualization, X.S. and M.H.; supervision, F.R. and M.H.; project administration, F.R. and M.H.; funding acquisition, M.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant 62176084 and Grant 62176083, and in part by the Fundamental Research Funds for the Central Universities of China under Grant PA2022GDSK0066 and Grant PA2022GDSK0068.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: TheASTE-data-V1 dataset is publicly available at https://github.com/NJUNLP/GTS (accessed on 10 October 2022). Furthermore, the ASTE-data-V2 dataset is publicly available at https://github.com/xuuuluuu/SemEval-Triplet-data (accessed on 13 September 2022). The datasets were revised by Wu et al. [56] and Xu et al. [22], which were cited in this research work for the ASTE task simultaneously.

Conflicts of Interest: The authors declare no conflict of interest.

Reference

1. Deng, J.; Ren, F. Multi-label Emotion Detection via Emotion-Specified Feature Extraction and Emotion Correlation Learning. *IEEE Trans. Affect. Comput.* **2020**, *9*, 162018–162034. [CrossRef]
2. Liang, B.; Su, H.; Gui, L.; Cambria, E.; Xu, R. Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks. *Knowl.-Based Syst.* **2022**, *235*, 107643. [CrossRef]
3. Zhao, A.; Yu, Y. Knowledge-enabled BERT for aspect-based sentiment analysis. *Knowl.-Based Syst.* **2021**, *227*, 107220. [CrossRef]
4. Bie, Y.; Yang, Y. A multitask multiview neural network for end-to-end aspect-based sentiment analysis. *Big Data Min. Anal.* **2021**, *4*, 195–207. [CrossRef]
5. Zhang, Y.; Du, J.; Ma, X.; Wen, H.; Fortino, G. Aspect-based sentiment analysis for user reviews. *Cogn. Comput.* **2021**, *13*, 1114–1127. [CrossRef]
6. Zhao, H.; Huang, L.; Zhang, R.; Lu, Q.; Xue, H. Spanmlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 3239–3248.
7. Akhtar, M.S.; Garg, T.; Ekbal, A. Multi-task learning for aspect term extraction and aspect sentiment classification. *Neurocomputing* **2020**, *398*, 247–256. [CrossRef]
8. Xu, Q.; Zhu, L.; Dai, T.; Yan, C. Aspect-based sentiment classification with multi-attention network. *Neurocomputing* **2020**, *388*, 135–143. [CrossRef]
9. Ying, C.; Wu, Z.; Dai, X.; Huang, S.; Chen, J. Opinion transmission network for jointly improving aspect-oriented opinion words extraction and sentiment classification. In Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing, Zhengzhou, China, 14–18 October 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 629–640.
10. Li, Z.; Li, Q.; Zou, X.; Ren, J. Causality extraction based on self-attentive BiLSTM-CRF with transferred embeddings. *Neurocomputing* **2021**, *423*, 207–219. [CrossRef]

11. Chen, F.; Yang, Z.; Huang, Y. A multi-task learning framework for end-to-end aspect sentiment triplet extraction. *Neurocomputing* **2022**, *479*, 12–21. [CrossRef]
12. Peng, H.; Xu, L.; Bing, L.; Huang, F.; Lu, W.; Si, L. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8600–8607.
13. Xu, L.; Chia, Y.K.; Bing, L. Learning Span-Level Interactions for Aspect Sentiment Triplet Extraction. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Virtual, 1–6 August 2021; pp. 4755–4766.
14. Yu Bai Jian, S.; Nayak, T.; Majumder, N.; Poria, S. Aspect sentiment triplet extraction using reinforcement learning. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, 17–21 October 2021; pp. 3603–3607.
15. Mao, Y.; Shen, Y.; Yu, C.; Cai, L. A joint training dual-mrc framework for aspect based sentiment analysis. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 13543–13551.
16. Chen, S.; Wang, Y.; Liu, J.; Wang, Y. Bidirectional machine reading comprehension for aspect sentiment triplet extraction. In Proceedings of the AAAI Conference On Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 12666–12674.
17. Yan, H.; Dai, J.; Ji, T.; Qiu, X.; Zhang, Z. A Unified Generative Framework for Aspect-based Sentiment Analysis. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Virtual, 1–6 August 2021; pp. 2416–2429.
18. Chen, Z.; Huang, H.; Liu, B.; Shi, X.; Jin, H. Semantic and Syntactic Enhanced Aspect Sentiment Triplet Extraction. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online, 1–6 August 2021; pp. 1474–1483.
19. Wu, S.; Li, B.; Xie, D.; Teng, C.; Ji, D. Neural transition model for aspect-based sentiment triplet extraction with triplet memory. *Neurocomputing* **2021**, *463*, 45–58. [CrossRef]
20. Zhao, Y.; Meng, K.; Liu, G.; Du, J.; Zhu, H. A Multi-Task Dual-Tree Network for Aspect Sentiment Triplet Extraction. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; pp. 7065–7074.
21. Shi, L.; Han, D.; Han, J.; Qiao, B.; Wu, G. Dependency graph enhanced interactive attention network for aspect sentiment triplet extraction. *Neurocomputing* **2022**, *507*, 315–324. [CrossRef]
22. Xu, L.; Li, H.; Lu, W.; Bing, L. Position-Aware Tagging for Aspect Sentiment Triplet Extraction. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Virtual, 16–20 November 2020; pp. 2339–2349.
23. Chen, H.; Zhai, Z.; Feng, F.; Li, R.; Wang, X. Enhanced Multi-Channel Graph Convolutional Network for Aspect Sentiment Triplet Extraction. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 2974–2985.
24. Zhang, C.; Ren, L.; Ma, F.; Wang, J.; Wu, W.; Song, D. Structural Bias for Aspect Sentiment Triplet Extraction. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; pp. 6736–6745.
25. Tang, F.; Fu, L.; Yao, B.; Xu, W. Aspect based fine-grained sentiment analysis for online reviews. *Inf. Sci.* **2019**, *488*, 190–204. [CrossRef]
26. Xiao, L.; Xue, Y.; Wang, H.; Hu, X.; Gu, D.; Zhu, Y. Exploring fine-grained syntactic information for aspect-based sentiment classification with dual graph neural networks. *Neurocomputing* **2022**, *471*, 48–59. [CrossRef]
27. Consoli, S.; Barbaglia, L.; Manzan, S. Fine-grained, aspect-based sentiment analysis on economic and financial lexicon. *Knowl.-Based Syst.* **2022**, *247*, 108781. [CrossRef]
28. Phan, M.H.; Ogunbona, P.O. Modelling context and syntactical features for aspect-based sentiment analysis. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 3211–3220.
29. Wang, K.; Shen, W.; Yang, Y.; Quan, X.; Wang, R. Relational Graph Attention Network for Aspect-based Sentiment Analysis. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 3229–3238.
30. Zhang, B.; Xu, D.; Zhang, H.; Li, M. STCS lexicon: Spectral-clustering-based topic-specific Chinese sentiment lexicon construction for social networks. *IEEE Trans. Comput. Soc. Syst.* **2019**, *6*, 1180–1189. [CrossRef]
31. Kiritchenko, S.; Zhu, X.; Cherry, C.; Mohammad, S. NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, 23–24 August 2014; pp. 437–442.
32. Dai, J.; Yan, H.; Sun, T.; Liu, P.; Qiu, X. Does syntax matter? A strong baseline for Aspect-based Sentiment Analysis with RoBERTa. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Mexico City, Mexico, 6–11 June 2021; pp. 1816–1829.
33. Zhang, Y.; Jin, R.; Zhou, Z.H. Understanding bag-of-words model: A statistical framework. *Int. J. Mach. Learn. Cybern.* **2010**, *1*, 43–52. [CrossRef]
34. Tang, H.; Ji, D.; Li, C.; Zhou, Q. Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6578–6588.

35. Xiao, Z.; Wu, J.; Chen, Q.; Deng, C. BERT4GCN: Using BERT Intermediate Layers to Augment GCN for Aspect-based Sentiment Classification. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online, 7–11 November 2021; pp. 9193–9200.
36. Zhang, Z.; Zuo, Y.; Wu, J. Aspect Sentiment Triplet Extraction: A Seq2Seq Approach with Span Copy Enhanced Dual Decoder. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2022**, *30*, 2729–2742. [CrossRef]
37. Dai, D.; Chen, T.; Xia, S.; Wang, G.; Chen, Z. Double embedding and bidirectional sentiment dependence detector for aspect sentiment triplet extraction. *Knowl.-Based Syst.* **2022**, *253*, 109506. [CrossRef]
38. Chen, Y.; Zhang, Z.; Zhou, G.; Sun, X.; Chen, K. Span-based dual-decoder framework for aspect sentiment triplet extraction. *Neurocomputing* **2022**, *492*, 211–221. [CrossRef]
39. Mukherjee, R.; Nayak, T.; Butala, Y.; Bhattacharya, S.; Goyal, P. PASTE: A Tagging-Free Decoding Framework Using Pointer Networks for Aspect Sentiment Triplet Extraction. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online, 7–11 November 2021; pp. 9279–9291.
40. Xu, K.; Li, F.; Xie, D.; Ji, D. Revisiting Aspect-Sentiment-Opinion Triplet Extraction: Detailed Analyses Towards a Simple and Effective Span-Based Model. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2022**, *30*, 2918–2927. [CrossRef]
41. Zhu, X.; Zhu, L.; Guo, J.; Liang, S.; Dietze, S. GL-GCN: Global and local dependency guided graph convolutional networks for aspect-based sentiment classification. *Expert Syst. Appl.* **2021**, *186*, 115712. [CrossRef]
42. Cai, H.; Tu, Y.; Zhou, X.; Yu, J.; Xia, R. Aspect-category based sentiment analysis with hierarchical graph convolutional network. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 833–843.
43. Feng, S.; Wang, B.; Yang, Z.; Ouyang, J. Aspect-based sentiment analysis with attention-assisted graph and variational sentence representation. *Knowl.-Based Syst.* **2022**, *258*, 109975. [CrossRef]
44. Li, Y.; Lin, Y.; Lin, Y.; Chang, L.; Zhang, H. A span-sharing joint extraction framework for harvesting aspect sentiment triplets. *Knowl.-Based Syst.* **2022**, *242*, 108366. [CrossRef]
45. Fei, H.; Ren, Y.; Zhang, Y.; Ji, D. Nonautoregressive Encoder-Decoder Neural Framework for End-to-End Aspect-Based Sentiment Triplet Extraction. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–13. early access. [CrossRef]
46. Xu, H.; Shu, L.; Philip, S.Y.; Liu, B. Understanding Pre-trained BERT for Aspect-based Sentiment Analysis. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 244–250.
47. Wu, Z.; Ong, D.C. Context-guided bert for targeted aspect-based sentiment analysis. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021; Volume 35, pp. 14094–14102.
48. Zhu, L.; Xu, Y.; Zhu, Z.; Bao, Y.; Kong, X. Fine-Grained Sentiment-Controlled Text Generation Approach Based on Pre-Trained Language Model. *Appl. Sci.* **2022**, *13*, 264. [CrossRef]
49. Mutinda, J.; Mwangi, W.; Okeyo, G. Sentiment Analysis of Text Reviews Using Lexicon-Enhanced Bert Embedding (LeBERT) Model with Convolutional Neural Network. *Appl. Sci.* **2023**, *13*, 1445. [CrossRef]
50. Shaw, P.; Uszkoreit, J.; Vaswani, A. Self-Attention with Relative Position Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; Volume 2 (Short Papers), pp. 464–468.
51. Wang, B.; Shin, R.; Liu, X.; Polozov, O.; Richardson, M. RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7567–7578.
52. Dozat, T.; Manning, C.D. Deep Biaffine Attention for Neural Dependency Parsing. *arXiv* **2016**, arXiv:1611.01734.
53. Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Association for Computational Linguistics, Dublin, Ireland, 23–24 August 2014; pp. 27–35. [CrossRef]
54. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Manandhar, S.; Androutsopoulos, I. SemEval-2015 Task 12: Aspect Based Sentiment Analysis. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Co, USA, 4–5 June 2015; pp. 486–495. [CrossRef]
55. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; AL-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; De Clercq, O.; et al. SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), Association for Computational Linguistics, San Diego, CA, USA, 16–17 June 2016; pp. 19–30. . [CrossRef]
56. Wu, Z.; Ying, C.; Zhao, F.; Fan, Z.; Dai, X.; Xia, R. Grid Tagging Scheme for Aspect-oriented Fine-grained Opinion Extraction. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Punta Cana, Dominican Republic, 8–12 November 2020; pp. 2576–2585.
57. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, USA, 2–7 June 2019; Volume 1 (Long and Short Papers), pp. 4171–4186.
58. Fan, Z.; Wu, Z.; Dai, X.Y.; Huang, S.; Chen, J. Target-oriented Opinion Words Extraction with Target-fused Neural Sequence Labeling. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics:

Human Language Technologies, Volume 1 (Long and Short Papers); Association for Computational Linguistics: Minneapolis, MN, USA, 2–7 June 2019; pp. 2509–2518. [CrossRef]
59. Zhang, C.; Li, Q.; Song, D.; Wang, B. A Multi-task Learning Framework for Opinion Triplet Extraction. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Punta Cana, Dominican Republic, 8–12 November 2020; pp. 819–828.
60. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv* **2021**, arXiv:2107.13586.
61. Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P.; Bakhtin, A.; Wu, Y.; Miller, A. Language Models as Knowledge Bases? In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November2019; pp. 2463–2473.
62. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
63. Schick, T.; Schütze, H. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Kiev, Ukraine, 19–23 April 2021; pp. 255–269.

applied sciences

MDPI

Article

Sentiment Analysis of Comment Texts on Online Courses Based on Hierarchical Attention Mechanism

Baohua Su [1,2] and Jun Peng [2,*]

[1] College of Chinese Language and Culture, Jinan University, Guangzhou 510632, China; subaohua@hwy.jnu.edu.cn
[2] School of Education, Research Institute of Macau Education Development, City University of Macau, Macau 999078, China
* Correspondence: 4588775@163.com

Abstract: With information technology pushing the development of intelligent teaching environments, the online teaching platform emerges timely around the globe, and how to accurately evaluate the effect of the "any-time and anywhere" teacher–student interaction and learning has become one of the hotspots of today's education research. Bullet chatting in online courses is one of the most important ways of interaction between teachers and students. The feedback from the students can help teachers improve their teaching methods, adjust teaching content, and schedule in time so as to improve the quality of their teaching. How to automatically identify the sentiment polarity in the comment text through deep machine learning has also become a key issue to be automatically processed in online course teaching. The traditional single-layer attention mechanism only enhances certain sentimentally intense words, so we proposed a sentiment analysis method based on a hierarchical attention mechanism that we called HAN. Firstly, we use CNN and LSTM to extract local and global information, gate mechanisms are used for extracting sentiment words, and the hierarchical attention mechanism is then used to weigh the different sentiment features, with the original information added to the attention mechanism concentration to prevent the loss of information. Experiments are conducted on China Universities MOOC and Tencent Classroom comment data sets; both accuracy and F1 are improved compared to the baseline, and the validity of the model is verified.

Keywords: review text for online courses; sentiment analysis; attention mechanism; gating mechanism

Citation: Su, B.; Peng, J. Sentiment Analysis of Comment Texts on Online Courses Based on Hierarchical Attention Mechanism. *Appl. Sci.* **2023**, *13*, 4204. https://doi.org/10.3390/app13074204

Academic Editor: João M. F. Rodrigues

Received: 2 March 2023
Revised: 23 March 2023
Accepted: 24 March 2023
Published: 26 March 2023

1. Introduction

In recent years, network technologies, such as the Internet, the Internet of things, and big data, have developed rapidly, and network platforms for e-commerce, social communication, and education are emerging timely. These platforms have not only enriched our daily life but also changed our ways of working, studying, and living. The sentiment comment texts on the network platform reflect people's opinions on something. Thus, how to effectively use these opinions has become an important factor in improving service quality. In education, many countries have shifted their offline teaching to online teaching due to the global COVID-19 pandemic [1,2]. Compared with the traditional offline classroom, online education has the advantages of lower costs, flexible forms, and fewer geographical restrictions [3,4]. Its promotion and application increase the equity of higher education, realize knowledge sharing, improve the effectiveness and efficiency of decision-making, and make higher education more open [5]. In order to further evaluate the quality of teaching and strengthen the interaction between teachers and students, a large number of teaching platforms, such as China Universities MOOC and Tencent Classroom, have provided the bullet chatting function. The bullet chatting imbued with sentiment information plays an important role in the teaching process. Through students' feedback, teachers can know what points students are weak in. School administrators can dynamically adjust the knowledge points, teaching plans, teaching objectives, and teaching staff structure of the courses

based on the sentiment analysis of comment texts. Therefore, how to leverage useful information from comment text with sentiment information has become one of the hot research directions in natural language processing [6].

Sentiment analysis is used to judge the sentiment polarity (positive, neutral, or negative) of reviews. Since Pang et al. studied the sentiment analysis of film reviews, sentiment analysis technology has been widely used in the business community [7]. As an emerging educational approach in the era of information technology, online courses have attracted many educators and learners around the world with their advantages of spanning time and space and flexible learning methods. Comments, as the most direct way of interactive feedback in online courses, are of great significance in improving the quality of teaching, reducing the dropout rate, and promoting the sustainable development of online courses [8–10]. So, sentiment analysis is also very important in the field of education, but very few researchers do sentiment analysis in online course reviews, and even public data sets on this are very scarce.

There are three main methods of sentiment analysis: sentiment analysis based on sentiment dictionaries and rules, sentiment analysis based on traditional machine learning, and sentiment analysis based on deep learning [11]. Soe et al. further calculated sentiment scores to achieve the purpose of analyzing students' emotions through a part-of-speech tagging analyzer and vocabulary resources [12]. The second type of sentiment analysis method recognizes sentiment through constructing features artificially and using naïve Bayes, maximum entropy, and support vector machine and other classifiers. The accuracy of this method depends entirely on the construction of features and the selection of classifiers, while most of the current research mainly focuses on the former. Therefore, the quality of feature selection largely determines the accuracy of the experimental results. Feature construction faces not only the problems of large workload and sparsity of features but also the common problem of domain adaptability.

With the development of deep learning and the improvement in text representation methods based on deep learning, many researchers began to study the application of deep learning to deal with text sentiment analysis. Represented by RNN, LSTM, and other classical neural networks, deep learning-based sentiment analysis methods can not only solve the shortcomings of traditional machine learning but also have significant classification effects. CNN can obtain the local information of a text, whereas recurrent neural networks such as LSTM can obtain the global information of a text. On the one hand, sequence-based neural networks such as LSTM have been restricted by the sequence length and computational memory. Attention mechanisms, on the other hand, could alleviate this problem since it allows modeling of the dependency output sequence without considering the distance between texts [13–15]. As a result, there are some sentiment analysis methods that use classical neural networks combined with an attention mechanism. Yang et al. [16] and Liu et al. [17] shows that the combination of an attention mechanism and LSTM can improve the accuracy of the model. The single-layer attention mechanism tends to focus on the words with strong sentiment expression and ignore the words with weak sentiment expression and opposite polarity, leading to the misjudgment of sentiment polarity. Take a real comment, for example:

> " 互联网时代, 教师个人知识与在线资源连线, 现在的问题不是资源太少, 而是资源太多, 良莠混杂, 无从选择。该课程讲解了教师个人知识管理的体系架构, 资源分类与统筹管理的方法, 对教师理清个人知识体系, 提高工作学习效率大有裨益. (In the Internet era, teachers' personal knowledge is connected with online resources. The problem now is not that there are too few resources, but that there are too many, and it is hard to choose from these resources as they are of mixed qualities. This course explains the system structure of teachers' personal knowledge management, the classification of resources and integrated approaches to management, which is of great benefit for teachers to clarify their personal knowledge system and improve their work and learning efficiency.)"

As the single-layer attention mechanism often focuses on the words with strong sentiment expressions, such as " 资源太少 (too few resources)," " 良莠混杂 (mixed resources differ in quality)", and " 无从选择 (there is no way to choose)", resulting in the misjudgment of sentiment polarity. Therefore, this paper proposes the use of a hierarchical attention mechanism to deal with the sentiment analysis of online course reviews. Though CNN and LSTM are often combined for text sentiment analysis, such as [18,19], there is still a lack of effective ways to take advantage of useful information (e.g., historical, global, and local information) extracted by them. Therefore, this paper uses the gate mechanism to further select useful local information. The significance of this study may be declared as follows:

First, this study indicated that the single-layer attention mechanism cannot accurately identify the sentiment words that are useful for global information. When human beings know that they need to carry out sentiment analysis task of short texts, they will first pay attention to the sentiment words in the sentence and then read the sentence from the beginning to the end to judge which sentiment words are more important, and to obtain the sentiment polarity of the sentence. According to this and the human's way of sentiment analysis of reading text, this study designs a hierarchical interactive attention mechanism, obtains the local features of sentences through CNN, and obtains the global information and the temporal features of sentences by using LSTM. Then the gate mechanism filters the local sentiment information. At the same time, the local sentiment word information extracted by CNN can enrich the hidden layer representation extracted by LSTM, and then the sentiment polarity of the sentence can be obtained after the information is weighted by the hierarchical attention mechanism.

Second, in the design of the attention mechanism, this study preserves original information through the connection way of residuals. The experiment shows that the hierarchical attention mechanism is effective in the sentiment analysis of online course reviews.

2. Related Work

Based on deep learning, there are two major categories of sentiment analysis models: graph-based models and sequence-based models.

The TextGCN model proposed by Yao et al. was the first time to use GCN in text classification (sentiment analysis) [20]. Two graphs were employed by that study as effective tools. The one named PMI was used to construct the relationship between words, and another named TF-IDF was used to construct the relationship between documents and words, and then the text category was obtained by the classifier. Then, Ragesh et al. [21] and Galke et al. [22] developed HeteGCN, which combined features of predictive text and TextGCN; It means the adjacency matrix was split into word documents and word submatrices, and the representations of different layers were fused as needed. Subsequently, HyperGAT was brought forward by Ding et al., from which an edge can connect multiple vertices [23]. So the text information was transformed into a hypergraph between nodes and edges, and the information between each layer was aggregated by dual attention. At last, tensorGCN was presented by Liu et al. [24]. This model constructed multiple graphs to describe semantic, syntactic, and contextual information and improved the effect of text classification through learning intra-graph propagation and inter-graph propagation.

Some studies have found that in recent years, most of the new methods for sentiment analysis (text classification) are based on GCN, while transformer-based sequence models are rare in the literature [22]. However, much empirical evidence shows that transformer-based sequence models outperform GCN-based methods. So here is a look at some sequence-based text classification methods. After obtaining the representation of each word, Kim embedded the word into CNN to obtain the sentiment polarity of the text [25]. Through the experimental results of a large number of data sets, he proved the ability of CNN on the task of text classification. After obtaining the text representation, Liu et al. used an RNN to classify the sentiment of the comment text [26]. Wang et al. proved that LSTM could achieve better experimental results than traditional RNNs in

tweet sentiment analysis through experiments on tweet datasets [27]. After acquiring the word representation, the RNNs acquire the phrase representation and the sentence representation in order according to the syntactic structure. Huang et al. used a two-layer LSTM to classify the sentiment of tweets and believed that the sentiment polarity of the current tweet was largely related to the previous and subsequent tweets [28]. If the sentiment polarity is judged by the current tweet alone, the system would be deceived by its irony and other language expressions. Therefore, the hidden layer state of the current tweet should be input into a higher-level LSTM to obtain the current tweet representation containing context information and, finally, obtain the sentiment polarity distribution of the current tweet through the classifier. Yang et al. used the attention mechanism to aggregate word information to obtain sentence information, then they used the second layer attention mechanism to aggregate sentence information, in order to obtain the overall sentiment polarity in the discourse-level sentiment analysis, which fully proved the importance of an attention mechanism in sentiment analysis [16]. Vaswani et al. proposed the transformer model, which once again proved the importance of an attention mechanism in text classification [13]. Since the invention of BERT in 2018, there has been a lot of research on sentiment analysis based on BERT [29]. In Order to solve the negative effect of mask in BERT, XLNet uses an autoregressive language model instead of an autoencoding language model and introduces a double-stream self-attention mechanism and transformer-xl [30]. Compared with BERT, XLNet achieves better experimental results. ERNIE uses the same coding structure as BERT, but the author thinks that the random mask mechanism in BERT ignores the semantic relationship to some extent, so the original mask is split into three parts, the first part retains the original random mask, the second part masks the entity word as a whole. The last part is to mask the phrase as a whole. Compared with ERNIE, ERNIE 2.0 proposes three types of unsupervised tasks, which provide the model with a better representation ability of sentences, grammar, and semantics [31]. The performance and advantages of some methods on data sets are summarized in Table 1.

Table 1. Comparison with some methods.

Model	Data (acc)						Advantages
	SST-2	20NG	R8	R52	Ohsumed	MR	
Text GCN	-	0.863	0.970	0.935	0.683	0.767	A heterogeneous graph based on text and words is constructed, and the semi-supervised classification of text can be performed on GCN
HeteGCN	-	0.846	0.972	0.939	0.638	0.756	Reduce the complexity of TextGCN
HyperGAT	-	0.862	0.970	0.950	0.699	0.783	Capturing higher-order interactions between words while improving computational efficiency
TensorGCN	-	0.877	0.980	0.951	0.701	0.780	Rich multi-subgraph feature representation
LSTM	-	0.754	0.961	0.905	0.511	0.773	More effective way to process sequence data
BERT	0.928	-	-	-	-	-	The vector representation is rich, which overcomes the gradient problem of LSTM when solving long sequence data
ROBERTa	0.937	-	-	-	-	-	Raining models with larger corpora and sequences, dynamic MASK mechanism
XL-net	0.971	-	-	-	-	-	Autoregressive training method to overcome the shortcomings of bert
ernie	0.935	-	-	-	-	-	Taking advantages of The lexical, syntactic and knowledge information, large-scale text corpora and KGs to train an augmented language representation model

"-" indicates that the original paper was not tested on this data set.

3. Model Building

A segment with a length of n is given when we analyze the sentiment of online course reviews. Additionally, the sentiment polarity expressed by the bullet screen is judged by

analyzing the review. After obtaining the comment sentence S , each word in the sentence needs to be vectorized. In this study, each word in the sentence S is randomly initialized. That is $S_v = \left[v^1, v^2, \cdots, v^n\right] \epsilon \mathbb{R}^{n \times d_w}$, in which d_w is the size of the word vector dimension.

3.1. Model Construction Process

In order to obtain the local sentiment features of reviews, we use CNN to obtain the local features of sentences (i.e., $H_C = \left[h_c^1, h_c^2, \cdots, h_c^m\right] \epsilon$) as shown in Figure 1. Then, to further extract the hidden features of the text, LSTM is used to extract the hidden information of the comment text. Finally, the context hidden state (i.e., $H_L = \left[h_l^1, h_l^2, \cdots, h_l^n\right] \epsilon \mathbb{R}^{n \times d_h}$) is extracted by LSTM. After that, a gate mechanism is used to screen the important sentiment information of H_C and H_L , as follows.

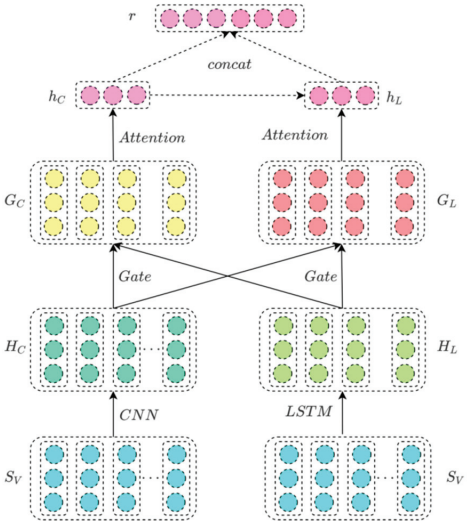


Figure 1. Overall model structure.

First, average pooling needs to be performed for the hidden state of the sentence H_L extracted by LSTM and the local emotion information H_C, as shown in Formula (1) and Formula (2), respectively. The LSTM hidden layer vector representation $h_g^L \in \mathbb{R}^{d_h}$ is obtained. Additionally, a local aggregate information vector representation $h_g^C \in \mathbb{R}^{d_c}$ is obtained.

$$h_g^L = \sum_{i=1}^{n} H_L / n \tag{1}$$

$$h_g^C = \sum_{i=1}^{m} H_c / m \tag{2}$$

Then the local emotion information extracted from CNN needs to be filtered through the gate mechanism by using h_g^L , and the specific gate mechanism calculations are shown in Formulas (3)–(5).

$$T_C = relu\left(H_C W_C + W_g h_g^L \times w_g\right) \tag{3}$$

$$E^i = tanh\left(H_C^i W_E\right) \tag{4}$$

$$G_C^i = E^i T_C \tag{5}$$

In these calculations, $W_C \in \mathbb{R}^{d_c \times d_c}$, $W_g \in \mathbb{R}^{m \times d_h}$, $W_g \in \mathbb{R}^{d_C}$ are parameter vectors; $relu$, $tanh$ are activation functions; and $W_E \in \mathbb{R}^{d_c \times m}$, $G_C^i \in \mathbb{R}^{d_c}$ are global features. A selective representation of the gate mechanism, $G_C \in \mathbb{R}^{m \times d_c}$, can be obtained after expressing the chosen i vector of H_c through the gate mechanism.

$G_L \in \mathbb{R}^{n \times d_h}$ can be obtained in the same way. After G_C is obtained, information is aggregated by the attention mechanism, as shown in Formulas (6) and (7).

$$\alpha = softmax\left(G_C w_\alpha + H_C w_C\right) \tag{6}$$

$$h_c = \sum_{i=1}^{m} \alpha^i G_c^i \tag{7}$$

$w_\alpha \in \mathbb{R}^{d_c}$, $w_C \in \mathbb{R}^{d_c}$ are parameter vectors, and H_C is the feature information extracted by the original CNN. After obtaining the selection information G of the gate mechanism, the original information H_C is further added when the attention mechanism coefficients are weighted to avoid the loss of original information. Finally, the vector $h_C \in \mathbb{R}^{d_h}$ is weighted by Formula (7).

The single-layer attention mechanism can only focus on the strong sentiment words in the sentiment expression while ignoring the most important words in sentiment analysis. In order to highlight the importance of different words in sentiment analysis, this study proposes to use a multi-layer attention mechanism to focus on the importance of different words in sentences; therefore, weighted by the first layer of attention mechanism h_c. After that, to improve the accuracy of sentiment analysis by using text information, the information of G_L is further weighted by h_c, as shown in Formulas (8)–(10).

$$\gamma\left(h_L^i, h_c\right) = tanh\left(h_L^i W_L h_c^T\right) \tag{8}$$

$$\beta_i = \frac{exp\left(\gamma\left(h_t^i, h_c\right)\right)}{\sum_{j=1}^{n} exp\left(\gamma\left(h_t^i, h_c\right)\right)} \tag{9}$$

$$h_t = \sum_{i=1}^{n} \beta_i h_t^i \tag{10}$$

$h_L^i \in \mathbb{R}^{d_h}$ is the i vector of G_L, $tanh$ is the activation function, $W_L \in \mathbb{R}^{d_h \times d_C}$ is parameter matrix, and h_c^T is the transpose of the h_c vector. $\gamma\left(h_L^i, h_c\right)$ is obtained by Formula (8), and $\gamma\left(h_L^i, h_c\right)$ is the attention mechanism coefficient between h_L^i and h_c. β_i is the attention mechanism coefficient between h_L^i and h_c after they are normalized. Finally, $h_L \in \mathbb{R}^{d_h}$ is obtained by the weighted summation of Formula (10).

$r \in \mathbb{R}^{d_c + d_h}$ can be obtained by splicing h_c and h_L, and it is the final representation of a sentence containing sentiment information. Finally, the sentiment polarity of the sentence is obtained by the $softmax$ classifier, as shown in Formulas (11) and (12).

$$x = tanh\left(W_r r + b_r\right) \tag{11}$$

$$y_i = \frac{exp\left(x_i\right)}{\sum_{j=1}^{C} exp\left(x_j\right)} \tag{12}$$

$W_r \in \mathbb{R}^{C \times (d_c + d_h)}$ is a parameter matrix, $b_r \in \mathbb{R}^C$ is a bias vector, and C is the total number of sentiment categories. Two data sets are used in this study, and the total number of sentiment classifications of one data set is two, positive and negative. The total number of sentiment classifications in the other data set is three, and they are positive, neutral, and negative.

3.2. Model Training

In order to use backward propagation to iteratively update all the parameter matrices and bias vectors proposed above, this study uses the cross-entropy and regularization of all the sentence classification results of the training set as the loss function, and the formulas are shown in (13) and (14).

$$J = -\sum_{i=1}^{C} g_i log y_i + \lambda_r \left(\sum_{\theta \in \Theta} \theta^2\right) \tag{13}$$

$$\Theta = \Theta - \lambda_l \frac{\partial J(\Theta)}{\partial \Theta} \tag{14}$$

g_i is the true sentiment distribution in the reviews, y_i is the prediction of the sentiment polarity of the review by the model, Θ is the set of all parameters, λ_r is the parameter of the L_2 regularization, and λ_l is the learning rate of the update parameter.

4. Experimental Process

In order to verify the effect of the proposed multi-layer attention mechanism of sentiment analysis model, this study conducted experiments, including data sets, evaluation indicators, and hyperparameter settings.

The average accuracy and F1 are calculated as shown in Formulas (15)–(18).

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

$$precision = \frac{TP}{TP + FP} \tag{16}$$

$$recall = \frac{TP}{TP + FN} \tag{17}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{18}$$

where *TP* means that the real label is a positive example and the predicted label is also a positive example; *TN* means that the real label is a negative example and the predicted label is also a negative example; *FP* means that the true label is a negative example, and the predicted labels are positive examples; *FN* means that real labels are positive examples and predicted labels are negative examples. The average accuracy represents the proportion of the correct prediction to all data, precision represents the accuracy of the positive prediction, recall represents the proportion of the correct prediction to all positive examples, and F1 takes into account both accuracy and recall, which is also a commonly used measurement standard.

4.1. Experimental Setup

Data sets: The data sets used in this study are two open online course review data sets, namely China University MOOC reviews [32] and MOOC and Ke reviews [25]. China University MOOC is jointly launched by NetEase Youdao and Higher Education Press and carries more than 10,000 open courses and more than 1400 national quality courses. It cooperates with 803 universities and is the largest Chinese MOOC platform [26]. Ke reviews comes from Tencent Classroom. Tencent Classroom is a comprehensive online lifelong learning platform launched by Tencent. It gathers a large number of high-quality educational institutions and famous teachers and offers many online quality courses [25]. Such as vocational training, civil service examination, TOEFL and IELTS, certification and grading examination, oral English, etc. Both MOOC and Tencent Classroom in Chinese universities contain a large number of users and rich classroom reviews. Sentiment analysis of classroom comments can master students' sentiment tendencies and help to carry out targeted classroom improvement, which can improve teaching quality to a certain extent. The MOOC data set contains 11,293 reviews on Chinese online courses from MOOCs, and the affective polarity of the data is divided into positive and negative, with 6164 positive reviews and 5129 negative reviews. MOOC and Ke reviews collected 1808 online course reviews from MOOC and Ke reviews, and the affective polarity of the data set was classified as positive, neutral, and negative. Among them, there are 817 reviews on positive sentiment polarity, 750 reviews on neutral sentiment polarity, and 241 reviews on negative sentiment polarity. In this study, the two data sets were mixed. For the MOOC data set, 80% of them were set as the training set, 10% as the validation set, and 10% as the test set.

Due to the small number of MOOC and KE data sets, 80% of them are set as the training set, and the rest are set as the test set. The specific distribution of the two data sets is shown in Table 2. In addition, experiments on the common data set R8 were conducted. The total number of data in R8 is 7674, among which the number of data in the training set is 5482, the number of data in the test set is 2189, and the classification category is 8.

Table 2. Data set distribution.

Dataset	Positive	Neutral	Negative
MOOC-Train	4609	0	4068
MOOC-Val	598	0	531
MOOC-Test	600	0	630
MOOC and Ke-Train	639	609	200
MOOC and Ke-Test	180	141	41

Evaluation index: Average accuracy is used to measure the performance of the sentiment analysis model of online classroom reviews based on a hierarchical attention mechanism, and F1 is used to evaluate it on MOOC.

Hyperparameters: We use a single Chinese character as a word. Words are initialized through random vector initialization; the dimension of the word vector is 300, the dimension of the LSTM hidden layer is also set to 300, and the number of LSTM layers is set to 2. The convolution windows of CNN are 2, 3, and 4, and the number of convolution sums is set to 256. For the words not in the dictionary, the values are randomly taken between -0.1 and 0.1. The initial values of all the parameter matrices and vectors are randomly chosen between -0.1 and 0.1. The initial value of the bias is set to 0. To adjust the parameters, the optimizer used in this study was Adam, the learning rate was set to 0.01, and the dropout was set to 0.5 to prevent overfitting.

4.2. Experimental Results and Analysis

The results of the comparison between the seven baselines and the HAG are shown in the following Table 3.

Table 3. Comparative experimental results.

Methods	MOOC (acc and F1)	MOOC and Ke (acc)	R8
CNN	0.903 and 0.911	0.453	95.34
LSTM	0.912 and 0.918	0.461	96.09
LSTM-Attention	0.932 and 0.920	0.472	96.59
BERT	0.932 and 0.940	0.495	98.03
RoBERTa	0.934 and 0.945	0.496	98.23
ERNIE	0.937 and 0.936	0.496	98.04
HAN	0.940 and 0.938	0.499	97.65

It can be seen from Table 2 that the experimental results of all models on the MOOC and MOOC and Ke data sets are quite different for four reasons: First, the data of MOOC is relatively large compared with the data of MOOC and Ke, and the number of data in MOOC is 11,036, while the number of data in MOOC and Ke is 1810. Secondly, the MOOC and Ke data set is divided into positive and negative categories, while the reviews in MOOC and Ke are divided into positive, neutral, and negative categories, which increases the difficulty of classification to a certain extent. Then, the proportion of positive and negative MOOCs relative to the MOOC and Ke dataset is more balanced. Finally, the review data in MOOC is clean, while the MOOC and Ke data set is noisy, which limits the accuracy of sentiment classification of MOOC and Ke data to some extent. In these three data sets, the accuracy of CNN is lower than in other models. CNN plays a significant role in extracting local features and can obtain sentiment word information to a certain

extent. LSTM can learn long-term dependencies and extract the sequence information of text effectively. Compared with CNN, the accuracy of LSTM on MOOC. MOOC, Ke, and R8 are increased by 0.9%, 0.8%, and 0.75%, respectively, and the accuracy of F1 is increased by 0.07%. The accuracy of the model is improved by the attention mechanism. BERT is also a classic model in sentiment analysis. BERT uses a multi-head attention mechanism to give the output of the attention layer, which contains the coding representation information in different subspaces, thus enhancing the expressive power of the model. Since the use of BERT for emotion analysis, various improvement methods based on BERT have been proposed. RoBERTa uses a dynamic masking mechanism and abandons that NSP (Next Sentence Predict) task; compared with BERT, RoBERTa performs slightly better on both datasets. Ernie also adjusts the mask mechanism in BERT. In the sentiment analysis of Chinese online courses, Ernie can identify the importance of words better than BERT. HAN is inferior to BERT-based models (i.e., BERT, Roberta, and Ernie) on long news texts of R8. This is partly due to the randomly initialized representation of HAN. Moreover, for the long news texts, BERT could alleviate the problem of vanishing gradients. However, in the class comments, HAN has achieved the best experimental results. HAN first uses CNN to extract local sentiment information and then uses the gate mechanism to filter the local sentiment information through the overall text information obtained by LSTM. At the same time, it uses the local sentiment information extracted by CNN to enrich the sequence information extracted by LSTM. Then through the weighting of the hierarchical attention mechanism, the sentiment tendency experiment of online classroom network reviews is obtained, and the experimental results once again prove the effectiveness of the HAN proposed in this study.

4.3. Case Study

To test the reliability of the model, a visual comparison between the weight of the final attention mechanism and the weight of the attention mechanism in baseline LSTM-Attention was made. The sample is derived from the comments in real online courses, such as Figure 2. As shown, the top half is the weight coefficient of the proposed model, and the bottom half is the weight of the attention mechanism of LSTM-Attention. The darker color means the greater weight of the attention mechanism and vice versa.

> “她们的许许多多的创意都很值得我学习, 但会自责, 同样的课程, 为什么差距这么大。 (Most of their creative ideas are well worth learning, but we will reproach ourselves for the same course with a great difference.)”

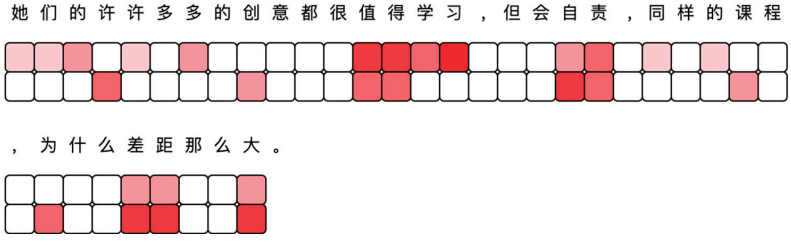


Figure 2. Example for the comments of real online course.

It seems that in Figure 2, the words “值得学习 (worth learning)”, “自责 (reproach ourselves)”, and “差距 (difference)” possess larger attention mechanism weights. However, when taking a closer look, it can be noticed that the overall emotional tendency of the sample comment is positive. Meanwhile, negative words such as “自责” and “差距大 (a great difference)” would interfere with the result. In this case, The LSTM-Attention method cannot distinguish the effect of these words on the overall sentiment analysis, and the words with strong emotional intensity were given higher weights. The model proposed in this

study strengthened the weight of " 值得学习" and correspondingly reduced the weight of " 自责", " 大", and other words, which verified the effectiveness of the model.

5. Conclusions

Using deep learning technology and starting from the sentiment analysis of online course review text, this study proposes a method to analyze the comments on online courses based on a hierarchical attention mechanism. The method enriches the extracted information by using CNN to extract local sentiment information and LSTM to obtain the hidden representation of the text. Then the global sentiment information extracted by the CNN and the global information extracted by the LSTM are screened by gate mechanism, respectively. The hierarchical attention mechanism reduces the influence of noise on affective polarity judgments, and the interference of the words with strong sentiment information to the model judgment is also reduced. This study proves the reliability of the HAN through three data sets. In the future, we will use more information to enrich the embedded representation of words, such as adding some speech information, sentiment information, and location information. In addition, in the construction of data sets, we will collect high-quality attribute-level online course reviews to ensure the accuracy of online courses' information feedback and promote the courses' deep interaction.

Author Contributions: Conceptualization, B.S. and J.P.; methodology, B.S.; software, J.P.; validation, B.S. and J.P.; formal analysis, B.S.; investigation, B.S.; resources, B.S. and J.P.; data curation, B.S. and J.P.; writing—original draft preparation, B.S.; writing—review and editing, B.S. and J.P.; visualization, B.S.; supervision, J.P.; project administration, B.S. and J.P.; funding acquisition, J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by the 2021 Higher Education Fund of the Macao SAR Government. (Project name: Development and Effectiveness Assessment of Higher Education Online Courses in Macao. Project No: HSS-CITYU-2021-07).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study can be obtained from the corresponding author upon request.

Conflicts of Interest: The authors declare that there are no conflict of interest regarding the publication of this paper.

References

1. Tarkar, P. Impact of COVID-19 pandemic on education system. *Int. J. Adv. Sci. Technol.* **2020**, *29*, 3812–3814.
2. Zhou, L.; Wu, S.; Zhou, M.; Li, F. 'School's out, but class' on', the largest online education in the world today: Taking China's practical exploration during The COVID-19 epidemic prevention and control as an example. *Best Evid. Chin. Edu.* **2020**, *4*, 501–519. [CrossRef]
3. Cao, R.; Xu, S.; Wang, X. Digitalization Leads the Future of Global Higher Education—Summary of the Main Session of the 2022 World MOOC and Online Education Conference. *China Educ. Informatiz.* **2023**, *29*, 82–95. [CrossRef]
4. Wang, X.; Guo, S. Practice and Enlightenment of Online and Offline Integrated Teaching in Tsinghua University. *Mod. Educ. Technol.* **2022**, *32*, 106–112.
5. Global MOOC and Online Education Alliance. Trends, Stages and Changes of Digitalization of Higher Education: An Excerpt from Infinite Possibilities: Report on the Development of Digitalization of World Higher Education. *China Educ. Informatiz.* **2023**, *29*, 3–8. [CrossRef]
6. Feng, C.; Li, H.; Zhao, H.; Xue, Y.; Tang, J. Attribute level sentiment analysis based on hierarchical attention mechanism and gate mechanism. *Chin. J. Inf. Technol.* **2021**, *35*, 128–136.
7. Pang, B.; Lee, L.; Vaithyanathan, S. Thumbs up? Sentiment classification using machine learning techniques. *arXiv* **2002**, arXiv:cs/0205070.
8. Wang, L.; Hu, G.; Zhou, T. Semantic analysis of learners' sentiment tendencies on online MOOC education. *Sustainability* **2018**, *10*, 1921. [CrossRef]
9. Mite-Baidal, K.; Delgado-Vera, C.; Solís-Avilés, E.; Espinoza, A.H.; Ortiz-Zambrano, J.; Varela-Tapia, E. Sentiment analysis in education domain: A systematic literature review. In Proceedings of the Technologies and Innovation: 4th International Conference, CITI 2018, Guayaquil, Ecuador, 6–9 November 2018; Springer International Publishing: Cham, Switzerland, 2018; pp. 285–297.

10. Pan, F.; Zhang, H.; Dong, J.; Shou, Z. Sentiment analysis of Chinese online course reviews based on efficient Transformer. *Comput. Sci.* **2021**, *48*, 264–269.
11. Xu, G.; Meng, Y.; Qiu, X.; Yu, Z.; Wu, X. Sentiment analysis of comment texts based on BiLSTM. *IEEE Access* **2019**, *7*, 51522–51532. [CrossRef]
12. Soe, N.; Soe, P.T. Domain oriented aspect detection for student feedback system. In Proceedings of the 2019 International Conference on Advanced Information Technologies (ICAIT), Yangon, Myanmar, 6–7 November 2019; pp. 90–95.
13. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
14. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
15. Kim, Y.; Denton, C.; Hoang, L.; Rush, A.M. Structured attention networks. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
16. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. Hierarchical attention networks for document classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1480–1489.
17. Liu, Z.; Zhou, W.; Li, H. AB-LSTM: Attention-based bidirectional LSTM model for scene text detection. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2019**, *15*, 107. [CrossRef]
18. Zhang, J.; Li, Y.; Tian, J.; Li, T. LSTM-CNN Hybrid Model for Text Classification. In Proceedings of the 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 October 2018; pp. 1675–1680. [CrossRef]
19. She, X.; Zhang, D. Text classification based on hybrid CNN-LSTM hybrid model. In Proceedings of the 2018 11th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 8–9 December 2018; Volume 2, pp. 185–189.
20. Yao, L.; Mao, C.; Luo, Y. Graph convolutional networks for text classification. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 7370–7377. [CrossRef]
21. Ragesh, R.; Sellamanickam, S.; Iyer, A.; Bairi, R.; Lingam, V. Hetegcn: Heterogeneous graph convolutional networks for text classification. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, 8–12 March 2021; pp. 860–868.
22. Galke, L.; Scherp, A. Bag-of-words vs. graph vs. sequence in text classification: Questioning the necessity of text-graphs and the surprising strength of a wide MLP. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; pp. 4038–4051.
23. Ding, K.; Wang, J.; Li, J.; Li, D.; Liu, H. Be more with less: Hypergraph attention networks for inductive text classification. *arXiv* **2020**, arXiv:2011.00387.
24. Liu, X.; You, X.; Zhang, X.; Wu, J.; Lv, P. Tensor graph convolutional networks for text classification. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 8409–8416. [CrossRef]
25. Kim, Y. Convolutional neural networks for sentence classification. *arXiv* **2014**, arXiv:1408.5882.
26. Liu, P.; Qiu, X.; Huang, X. Recurrent neural network for text classification with multi-task learning. In Proceedings of the 25th International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016; pp. 2873–2879.
27. Wang, X.; Liu, Y.; Sun, C.J.; Wang, B.; Wang, X. Predicting polarities of tweets by composing word embeddings with long short-term memory. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, 26–31 July 2015; pp. 1343–1353.
28. Huang, M.; Cao, Y.; Dong, C. Modeling rich contexts for sentiment classification with lstm. *arXiv* **2016**, arXiv:1605.01478.
29. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
30. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Generalized autoregressive pretraining for language understanding. *arXiv* **2019**, arXiv:1906.08237.
31. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Tian, H.; Wu, H.; Wang, H. Ernie 2.0: A continual pre-training framework for language understanding. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 8968–8975. [CrossRef]
32. Barbosa, L.; Feng, J. Robust sentiment detection on twitter from biased and noisy data. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Beijing, China, 23–27 August 2010; pp. 36–44.

applied sciences

MDPI

Article

A Manifold-Level Hybrid Deep Learning Approach for Sentiment Classification Using an Autoregressive Model

Roop Ranjan [1], Dilkeshwar Pandey [2], Ashok Kumar Rai [3], Pawan Singh [4], Ankit Vidyarthi [5], Deepak Gupta [6], Puranam Revanth Kumar [7] and Sachi Nandan Mohanty [8,*]

[1] Department of Computer Science and Engineering, KIPM College of Engineering and Technology, Gorakhpur 273209, India
[2] Department of Computer Science and Engineering, Krishna Institute of Engineering and Technology, Ghaziabad 201206, India
[3] Department of Computer Science and Engineering, Madan Mohan Malaviya University of Technology, Gorakhpur 273016, India
[4] Department of Computer Science and Engineering, Amity School of Engineering and Technology Lucknow, Amity University Uttar Pradesh, Noida 201301, India
[5] Department of Computer Science and Engineering & IT, Jaypee Institute of Information Technology, Noida 201309, India
[6] Department of Computer Science and Engineering, Maharaja Agrasen Institute of Technology, Delhi 110086, India
[7] Department of Electronics and Communication Engineering, IcfaiTech (Faculty of Science and Technology), IFHE University, Hyderabad 500029, India
[8] School of Computer Science & Engineering (SCOPE), VIT-AP University, Amaravati 522237, India
* Correspondence: sachinandan09@gmail.com

Abstract: With the recent expansion of social media in the form of social networks, online portals, and microblogs, users have generated a vast number of opinions, reviews, ratings, and feedback. Businesses, governments, and individuals benefit greatly from this information. While this information is intended to be informative, a large portion of it necessitates the use of text mining and sentiment analysis models. It is a matter of concern that reviews on social media lack text context semantics. A model for sentiment classification for customer reviews based on manifold dimensions and manifold modeling is presented to fully exploit the sentiment data provided in reviews and handle the issue of the absence of text context semantics. This paper uses a deep learning framework to model review texts using two dimensions of language texts and ideogrammatic icons and three levels of documents, sentences, and words for a text context semantic analysis review that enhances the precision of the sentiment categorization process. Observations from the experiments show that the proposed model outperforms the current sentiment categorization techniques by more than 8.86%, with an average accuracy rate of 97.30%.

Keywords: autoregressive model; customer reviews; deep learning; emotion analysis; optimized classification

Citation: Ranjan, R.; Pandey, D.; Rai, A.K.; Singh, P.; Vidyarthi, A.; Gupta, D.; Revanth Kumar, P.; Mohanty, S.N. A Manifold-Level Hybrid Deep Learning Approach for Sentiment Classification Using an Autoregressive Model. *Appl. Sci.* **2023**, *13*, 3091. https://doi.org/10.3390/app13053091

Academic Editors: Xiangjie Kong, Wei Wang and Han Liu

Received: 23 January 2023
Revised: 10 February 2023
Accepted: 11 February 2023
Published: 27 February 2023

1. Introduction

With easy access to the web, people now interact with brands and products in a whole new way. Whether with physical products or online services, people can share their opinions and reviews immediately on various platforms over the Internet. The world has transformed dramatically as a result of current advancements. Analyzing this large volume of consumer reviews will be helpful for consumers in making an informed decision about a product or service. In social network analyses, the sentiment analysis is an effective method for extracting user thoughts and determining a single user's sentiments. Social media, with its rich sentiments, has developed into a valuable resource for businesses and governments to understand the opinions and sentiments of online users [1]. For instance,

users of Twitter and other social media platforms routinely send out a lot of quick text messages with emoticons to communicate their opinions about various subjects. A textual sentiment analysis (SA) is not just a theoretical approach; it has applications in a variety of fields, including finance [2], education [3], health [4], and other areas.

Machine learning models have drawn a lot of attention recently. Traditional machine learning models almost universally use a two-step procedure. First, some manually created features from the papers are extracted. In a later stage, the features are sent to a classifier that performs predictions. The hand-crafted elements include the bag of words (BoW). Support vector machines (SVM), naive Bayes, gradient boosting trees, random forests, and the hidden Markov model (HMM) are some of the most used classification algorithms. There are various drawbacks to the two-step procedure. To achieve good performance relying on hand-crafted features, this necessitates time-consuming feature engineering and analysis phases. Furthermore, it is challenging to apply the strategy to new positions because it depends on domain expertise for feature creation.

Regarding mobile applications, the majority of apps can freely downloaded and a wide range of possibilities are accessible for a given sort of app, meaning sentiment analyses are made even more challenging. Users usually consult reviews or advice from other users before making decisions. App store owners can use the reviews to increase in the search ranks and catch fraud, while developers can use them to extract feedback (such as features, complaints, and privacy problems) [5]. Manual analyses are quite challenging due to the rapidly increasing volume of reviews (including false and spam reviews). As a result, app reviews have been rated in various ways throughout the last few years, from general exploratory research to categorization, feature extraction, review filtering, and summarizing. Furthermore, evaluations frequently include user opinions, which can be viewed as additional useful meta-data.

To alleviate the restrictions caused by the usage of hand-crafted features, neural techniques have been investigated. These techniques do not require hand-crafted features since they use a machine learning model that converts text into a low-dimensional vector of features. An LSA (latent semantic analysis) was proposed by Dumais et al. [6] in 1989 and was one of the earliest embedding models. An LSA is a trained linear model with 200,000 words and fewer than 1 million parameters. The first neural language model was put forth by Bengio et al. [7] in 2001, and the model worked on a feed-forward neural network that had been trained using 14 million words. The reason they are rarely used is that these early embedding models outperform conventional models with hand-crafted variables. A range of NLP tasks quickly gained popularity for a collection of word2vec models [8] that Google released in 2013, which were trained on 6 billion words. Using Google's Transformer [9], a fresh NN architecture, in 2018, embedding models were produced by OpenAI. For text-generating projects, their original model, GPT [10], is now extensively used. The same year, Google created BERT [7], a bidirectional Transformer-based system. BERT, which includes 340 million parameters and 3.3 billion words of training data, is currently the most advanced embedding model. It is possible for convolutional neural networks (CNN) [8] to learn local responses from spatial or temporal data, but not sequential correlations. Short-term dependencies in a sequence of data can be handled by recurrent neural networks (RNNs) [9], but long-term relationships are a problem for these networks.

To overcome the constraints of the existing systems in evaluating user sentiments for a certain service or product, a unique methodology based on deep learning utilizing XLNet has been developed. The existing sentiment categorization systems have two issues with handling missing context semantics in text:

i. The existing studies primarily use language symbol information in texts to classify sentiments. Only a few research have looked at sentiment data with punctuation marks in the dataset. The issue of text context semantics can be resolved with the aid of punctuation symbols that include sentimental information;

ii. The majority of the ongoing research is focused on the extraction of emotional characterizations and the modeling of textual material at the document level. On the other hand, studies rarely take into consideration doing other levels of text content, such as words or phrases. To overcome the lack of text context semantics in social media assessments, sentiment information can be efficiently collected from many levels via the extraction of sentiment features and by modeling texts from various levels.

Given the above issues in existing models for sentiment classification, a model named the manifold and multi-level sentiment modeling method (MFMLSC) is proposed. Therefore, the main contributions of this work are as follows:

i. Based on two dimensions, language symbols and emoticon symbols, the manifold sentiment classification method (MFSC) is proposed. In this approach, the problem of text context semantics missing in text reviews is tackled using the word, sentence, and document levels;
ii. The multi-dimensional sentiment classification method (MDSC) uses two symbol types, i.e., emoticons symbols and linguistic symbols. This approach is used to tackle the problem of missing context information from texts, which plays a significant role in obtaining hidden information from sentiments;
iii. Based on the effectiveness of these two models, the final model is proposed as the multi-fold and multi-level sentiment modeling method (MFMLSC)
iv. The proposed model is implemented on three different datasets of Google Pay, Phonpe, and Paytm mobile app reviews. Additionally, the proposed model is validated on the IMDB benchmark dataset.

The rest of the sections are organized as follows. Section 2 discusses the related work. Section 3 provides details and describes the workings of the proposed model. In Section 4, various settings and evaluation parameters are discussed. In Section 5, a summary and the conclusions are presented.

2. Related Work

This section provides a comprehensive review of the recent studies, along with recommended methodologies for addressing sentiment analysis challenges based on word embedding and deep learning (DL) techniques. Next, the state-of-the-art literature is addressed, with a focus on sentiment analyses in different areas.

Over the last two decades, the classification of user sentiments has attracted an increasing number of scholars and yielded a large number of research findings [10]. The classical machine learning and deep learning methods for classifying emotions mostly depend on supervised learning. The challenge is that natural language processing relies on efficient word embedding. By thoroughly training the global word–word co-occurrence of statistical data from the corpus, Mikolov et al. [11] and Pennington [12] first revealed that word vectors are learned through an RNN. As seen in [13], the final global vector (GloVe) has an intriguing linear substructure in the word vector space. Tang et al. [14] offered three models that took into account the text's emotional propensity and learned word embeddings with the sentiment. Word2Vec embedding was used in [15] to perform a sentiment analysis on reviews received from the Indonesian website Traveloka. It is estimated that their model is 91.9% accurate. The authors of [16] presented a monitoring system based on DL and ontology to aid the traveling process. Fuzzy ontologies and Word2vec embeddings were utilized to construct the suggested system's feature extraction module; the BiLSTM model was then used to classify the input text. According to Facebook, TripAdvisor, and Twitter data, the proposed technique was tested and found to be 84% accurate in its predictions.

A multi-layer architecture for customer evaluation approaches (such as word embedding and compositional vector models) was proposed in [17]. A back-propagation technique was used to train the network and provide weights for the various aspects of the design once it had been integrated into a neural network. GloVe-DCNN, a brand-new device featuring a variety of sentimental qualities, was introduced in [18]. Word embedding,

n-grams, and the polarity score properties of sentiment words were used to create a deep CNN. The authors of [19–21] developed a document representation system using the fuzzy bag of words paradigm (FBoW). An enhanced FBoW model that replaces the initial hard planning module with the Word2vec approach using fuzzy mapping was developed by replacing the original module with the Word2vec embedding. To determine the degree of similarity between words and clusters in seven different real-world document datasets, the researchers used three different approaches.

For the identification and condition analysis of traffic accidents, the authors of another study proposed a system based on using ontology with LDA (OLDA) and a BiLSTM network [22]. OLDA was employed in the proposed system to extract data and label texts. As a result, classifiers such as FastText and BiLSTM are employed. This system was more accurate than the previous one. In another study, BiLSTMs were used to gather data on the long-term reliance on word and sentence locations [23]. A CNN and BiLSTM were combined in the suggested hybrid strategy. LSTM outputs from sentence classification are applied to the multi-channel CNN to produce n-gram features. To find ADRs (adverse drug reactions) in electronic medical data, the authors of [24] suggested using a deep learning approach (EHRs). The proposed approach used the joint AB-LSTM model and embeddings based on lemmas to locate ADRs. The proposed technique had an F-measure of 73.3% on the EHR dataset. The combined model, for example, outperformed previous models that used a stack of CNNs and LSTM deep learning models, as shown in [25]. The dataset representation of Word2Vec is preferable to Word2Seq. Sentiment-based and dictionary-based representations of texts are some of the ways that texts are encoded. For extracting sentence features, the CNN model is paired with three attention methods. They concluded that the proposed CNN models were the most effective of all the models considered.

According to Hameed and Garcia-Zapirain [26], the accuracy of the BiLSTM approach was 85.8% on the IMDB Movie Review and SST2 (Stanford Sentiment Treebank) datasets [27]. The authors demonstrated that the BiLSTM method is both more efficient and suitable for sentiment analysis problems. Word2Vec, LSTM, RNN, and CNN methods were utilized by Xu and colleagues [28] to extract emotions from Chinese hotel reviews. The model with the highest F-score, 92%, was the BiLSTM method.

Some researchers have proposed hybrid deep learning-based models to improve accuracy, such as the LSTM-CNN grid-search (GS) approach for Amazon and IMDB reviews [29]. The authors utilized a grid-search technique and compared it to CNN, LSTM, CNN–LSTM, and other approaches. Their model outperformed several baseline models with an overall accuracy of 96%. In a similar study, the researchers [30] used Amazon reviews to model topics before using a CNN to identify views. The authors stated that their proposed approach improved the accuracy by 6 to 20% in comparison with the established methods.

Further studies were conducted on the more efficient embedding approach, BERT, and its derivatives in enhancing the analysis of sentiments for user reviews. The authors of [31] employed BERTCNN to improve a sentiment analysis for commodities reviews, with the results stating that the BERT-CNN (F1-score of 84.3%) outperforms the BERT (82%) and CNN (84.3%) (70.9%) approaches. Similarly, in [32] the SenBERT-CNN (sentiment BERT-CNN) was proposed for analyzing the feedback for JD.com, a mobile phone supplier, by merging the BERT and CNN approaches to obtain deep characteristics of the dataset. When the LSTM, BERT, and CNN approaches were compared, the authors found that BERT-CNN worked the best, with a score or 95.7%. In [33], on the other hand, a dataset from Drugs.com was used to develop neural network models for predicting reviews of drugs. On a scale from 0 to 9, patients' levels of happiness were given scores between 0 and 9. The authors tested many neural network models, including the BERT-LSTM model, with the following methods: 10-class and 3-class compressed forms of the dataset. The results showed that the BERT-LSTM model was the best-suited for the 3-class setup, even though it took a very long time to train. Others examples include [34], who used BERT to train different NN models on a dataset of movie reviews. The results showed that BERT

was the most accurate, while [35] used BERT to analyze Twitter sentiments by turning jargon into plain text for BERT training.

Additionally, in [36], the authors suggested a deep learning model using BERT for ADE (adverse drug effect) retrieval and detection to find pharmacological side effects. As a classifier and retrieval tool, the proposed model utilized sentence structure feature embeddings and BERT. Furthermore, in [37], the authors developed a method for extracting medical relations that relied on a pre-trained technique and a mechanism of fine-tuning rather than manual labeling. For feature extraction, the suggested method combined the BERT architecture with one-dimensional convolutional neural networks (1D-CNNs). The suggested method was tested on three datasets: the BioCreative V chemical relation corpus of illness, a classical Chinese literature dataset, and the i2b2 2012 temporal relation challenge dataset, and F1 score values of 0.7156, 0.8982, and 0.7085, respectively, were obtained. It was proposed by Ma et al. [38] that an enhanced version of Sentic LSTM be used for a joint task that combined the target-dependent detection of aspects and targeted aspect-based polarity classification. In another study, Sentic LSTM was developed by Ma et al. for the explicit integration of explicit and implicit information. By refining pre-trained word vectors with scores of sentiment intensity provided by sentiment lexicons, Gu et al. [39] presented a word vector refinement method that improved each word vector and performed better in the sentiment analysis. Hashida et al. [40] created a hybrid paradigm of multi-channel decentralized representation for textual data.

Various pre-trained language models, such as ELMo [41], BERT [42], and GPT [43], have recently demonstrated effective performance. Various Transformer-based language models such as BERT [42], robustly optimized BERT pre-training approach (RoBERTa) [44], and a lite BERT for self-supervised learning language representations (ALBERT) [45], have recently obtained the highest performance in many NLP tasks. Transformer's bidirectional encoder representation is known as BERT. Position embedding and word embedding are included in BERT's inputs. BERT's feature representation layers, unlike those of 1D-CNN and LSTM, rely on both left and right context information. A more advanced embedding technique, known as BERT, was also found to be useful in improving the sentiment analysis of reviews. Another study [46] examined the sentiment analysis performance of the SVM, multi-nomial naive Bayes, LSTM, and BERT approaches. Stemming, tokenization, lemmatization, and punctuation removal were among the preprocessing techniques used. The dataset includes 1.6 million tweets classified as good or negative. The study determined that BERT's performance was the best, with an accuracy rate of 85.4%. Two deep learning algorithms were created by the authors of [47] for the analysis of sentiments in multi-lingual social media text. During Pakistan's 2018 general election, Twitter was used to gather data. 80% of the dataset was used for training and 20% for testing. The XLM-RoBERTa and multi-lingual BERT (mBERT) from Transformer approaches were studied for their performance in this regard (XLM-R). The mBERT learning rate was set to 2×10^{-5}, and the XLM-R learning rate was set to 2×10^{-6} during the hyperparameter tweaking. Furthermore, mBERT had a precision rate of 69%, while XLM-R had a precision rate of 71%, according to the results of the trial. Using a deep bidirectional long short-term memory (DBLSTM) approach, in [48] the sentiments of Tamil tweets were analyzed. The dataset contains 1500 tweets categorized as either positive, negative, or neutral. The data were cleaned and pre-trained using the Word2Vec model before being represented using the DBLSTM word embedding approach. Furthermore, 80% of the dataset was utilized for training and 20% for testing. The DBLSTM approach was shown to be 86.2% accurate in the research. In a recent study [49], the authors proposed an adversarial strategy for handling the domain shift problem. The adversarial meaning stems from the parallel structure designed between the loss function on training samples and that on test samples. Using a projector and classifier, they presented a theoretical analysis of several benchmark datasets. In [50], the researchers performed a survey on an aspect-based sentiment analysis (ASBA). The authors showed a comparison of several techniques used in the ASBA.

In recent years, numerous studies have presented deep-learning-based sentiment assessments, each with its own set of characteristics and performance results. The traditional method for sentiment analyses is suitable for dealing with the categorization of small-scale texts. In the face of huge amounts of data, the analytical efficiency is low, and locating sentiment information is challenging. In recent years, deep learning approaches have demonstrated promising accuracy and efficiency in textual data sentiment classification. With the advent of Transformer-based pre-trained representations, the accuracy and efficacy have increased dramatically. Consequently, this study investigates and proposes a unique sentiment classification model based on the deep learning technique and XLNet's autoregressive pre-trained model.

3. Proposed Model

The proposed model primarily consists of two major components. Manifold emotion modeling is a technique that incorporates three different components: words, sentences, and documents. The second method makes use of language and punctuation marks to model multi-dimensional sentiments in two dimensions. Each word in the dataset is broken up into its unique phrase by using emoticons as separators. Through the practice of regarding emoticons and linguistic markings as unrecognized words, every sentence is segmented utilizing the word segmentation methodology that is currently in use. A technique for modeling the emotions associated with textual material is presented with three levels: word, phrase, and document. A multi-dimensional technique for classifying sentiment is given for modeling the text content using two dimensions: language-based symbols and emoji symbols at the word and sentence level.

The multi-fold with multi-level modeling results are inputs into the multi-level perception network using the pre-trained autoregressive word representation model XLNet to produce the final sentiment classification results (Figure 1). The algorithm of the proposed model is shown as Algorithm 1.

The proposed model is divided into four modules. The module-wise discussions of the proposed model are presented below.

Algorithm 1: Multi-Fold Dimensional Modeling Method for Sentiment Classification

```
input: IDocument
output: IDocumentDVector
initialization of the XLNet and Dual-LSTM models
IDocumentSVector = []
for each sentence in IDocument:
for each W_word, emoji in sentence:
WVector = BERT(W_word)
L_languageWVector = XLNet (L_language)
P_emoticonsWVector = XLNet (P_emoticons)
sentence WVector = [WVector, emoticon WVector]
SVector = Attention(Dual-LSTM(S WVector))
L_languageSVector = L_languageWVector
sentence SVector = [SVector, L_languageSVector]
IDocumentSVector += sentence SVector
IDocumentDVector = Attention(Dual-LSTM(IDocumentSVector))
```

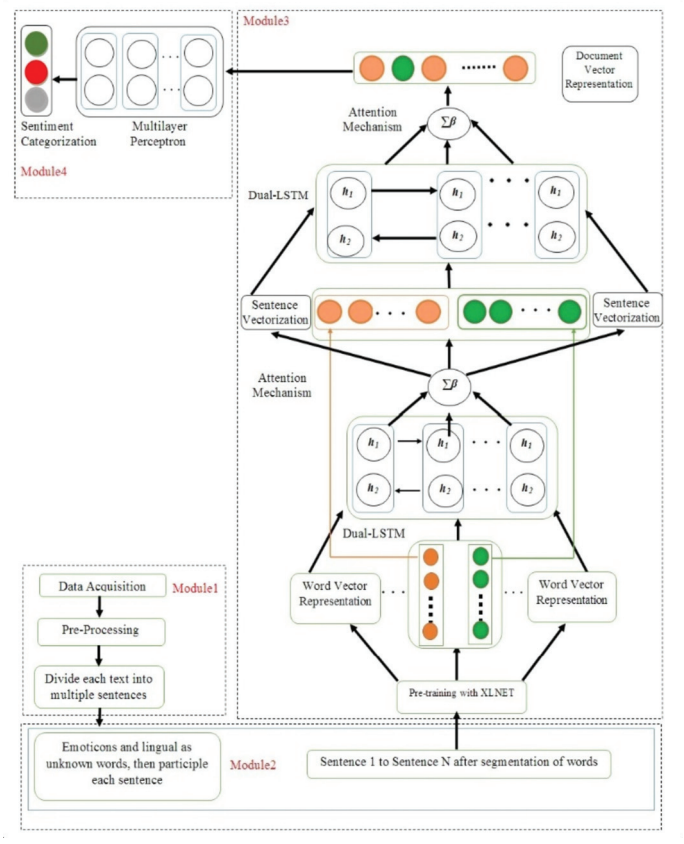


Figure 1. The proposed model.

3.1. Pre-Processing

The goal of the pre-processing phase is to remove all extraneous words from the corpus. The following are the major stages of the pre-processing phase:

i. Using the WordPiece tokenization paradigm, each word in the social input text is tokenized and can be broken into several sub-words;
ii. The Natural Language Toolkit (NLTK) removes stop words (is, the, a, etc.);
iii. Slang is converted to more formal forms;
iv. By eliminating texts that include indentations or by employing a widely unused set of suffixes and indentations, such as "-ing" or "pre-," one can restore extracted words to the word stem format using a rule-based stemmer technique;
v. Lemmatization removes inflection endings and returns words to the dictionary format. The proposed approach utilizes the NLTK suffix-dropping algorithm for stemming and lemmatization to improve the lexical context and analysis;
vi. Uppercase characters are converted to lowercase characters and repeated characters to their generic form;
vii. Spelling corrections are made using the Levenshtein distance and by selecting misspelled keywords.

Punctuation marks are used to divide cleaned and pre-processed texts into sentences. Punctuation is a collection of symbols that control and clarify the contents of various texts. Punctuation serves to clarify the meanings of texts by connecting or separating words, phrases, and clauses. As a result, punctuation is used to transform words into sentences.

XLNet

XLNet is a novel NLP pretraining approach that produces cutting-edge outcomes on several NLP tasks. Autoregressive (AR) language modeling and autoencoding (AE) are two pretraining aims for pretraining neural networks used in transfer learning NLP that have been proven effective. While avoiding the limitations of the two types of language pretraining objectives (AR and AE), XLNet incorporates concepts from both.

3.2. Multi-Fold Sentiment Modeling Method (MFSC)

The majority of the current research focuses on document-level text content modeling and sentiment feature extraction, with minimal attention paid to the interaction and correlation among sentences in the document. Between successive sentences in the text, there are evident progressive (forward) and adversative (reverse) linkages, as well as clear correlation and reciprocal influences between terms. As a result, the technique is suggested here for multi-fold sentiment modeling. The extraction of sentiment features and modeling content of text at several levels, such as words, phrases, and documents, helps address the lack of context semantics in dataset texts.

The multi-fold sentiment modeling method has three stages, the (i) word, (ii) sentence, and (iii) document levels. In the first fold of words, the input is the outcome of the segmentation 'of sentences. The outcome of this process is the representation of the word vector for the given sentences. In the second fold, i.e., the sentence level, the input for the model is the representation of vectorized words of the given set of sentences, and the outcome is the representation of vectorized sentences from the set of sentences. The multi-dimensional sentiment model is described in detail in the next section. The vectorized collection of several sentences is provided as the input in the document fold, and the result is the vectorized document.

The specifics at the document level are listed below.

i. Based on the grammatical rules and conjunctions between sentences, two types of relations are obtained: forward relations and reverse relations;
ii. The attention-based network is provided with prior knowledge of the following two types of relationships between sentences. Sentences with a reverse connection should have opposing sentiment polarities as much as is feasible. Sentences with forwarding relationships should have uniform sentiment polarity as much as is feasible. An attention-based system at the sentence level that is based on relationship constraints between sentences is provided here. This mechanism takes into account the two different sorts of linkages that exist between sentences. In the research, the attention-based method utilizes the attention formula at the phrase level;
iii. The vectorized text of every phrase is provided as the input for the dual-LSTM network based on the limitations of the attention-based mechanism, and the vectorized view of the given document is collected.

An output for sentiment categorization is generated by a multi-layer perception network using the representation of a vectorized document that has been obtained. Equation (1) provides a definition of the sentiment classification function that is based on multi-fold and multi-dimensional sentiment modeling:

$$\min_{x} \sum_{j=1}^{M} \left(x^T y_j - z_j \right)^2 + \partial_1 x_1 + \partial_2 \sum_{j=1}^{M} \sum_{k \neq j} S_{jk} (\omega_j - \omega_k)^2 + \partial_3 \sum_{j=1}^{M} \sum_{k \neq j} P_{jk} (\mu_j - \mu_k)^2 \quad (1)$$

Here, the total number of texts is represented by M, which represents the model of the sentiment classification; y_j is the representation of the vector of the jth text and z_j is the sentimental orientation of the jth text; ω_j and ω_k is the factor of attention for the word level; μ_j and μ_k is the factor of attention for the sentence level; S_{jk} is the factor of similarity of sentiment text j and sentiment phrase k; P_{jk} is the similarity factor of sentence j and sentence k; ∂_1, ∂_2, and ∂_3 represent the various hyperparameters.

3.3. Multi-Dimensional Sentiment Classification Method (MDSC)

The primary actions involved in multi-dimensional sentiment modeling at the level of individual words are discussed below:

(1) Since emoji and linguistic data provide information about sentiments, the dataset that contains emoji and linguistic symbols is used as the input to the language model, i.e., pre-training XLNet;
(2) Emojis and linguistic symbols are processed in the same way as sentiment words when a pre-trained model is used to model information available on social networks. This leads to the creation of the linguistic symbol word vector as well as the emoticons symbol word vector. This combination produces a multi-dimensional representation of the text's emotions.

The following are the primary steps in the multi-dimensional sentiment modeling at the sentence level:

i. The attention network provides prior knowledge of sentimental words. An approach based on word-level attention on the dictionary of sentiment restriction is provided, with the attention coefficients of sentiment-related words being as similar as possible. The attention formula is based on the attention formula at the word level;
ii. Vectorized words of language symbols and emoji symbols are given as inputs to a dual-LSTM network integrated with attention; the output is received as the vector of sentences of language symbols;
iii. The vectorized words of the emoji symbols are taken as outputs as the vectors of sentences of the emoji symbols directly;
iv. Combining the obtained sentence vectors of language symbols with emoticon symbols yields the sentence vectors.

The detailed mechanism of sub-modules is discussed below.

3.4. Sentiment Classification Using Multi-Layer Perceptron

The document vector representation is fed into a multi-level perceptron. The following parameter settings shown in Table 1 are used in obtaining optimized performance during sentiment classification. These parameters are obtained by performing several experiments with different parameters.

Table 1. Parameter settings for the MLP.

Parameters	Values
Optimization function	*sgd* (Stochastic Gradient Descent)
Batch-Size	64
Learning rate	0.03
Number of iterations	20
Activation Function	ReLu
Epochs	50

Using the above parameters in Table 1, the multi-layer perceptron (as shown in Figure 2) goes through the learning process and the output class labels are obtained using the below process, the MLP learning Procedure, as shown in Figure 3.

i. Using forward propagation, the data from the input layers are transmitted to the output layer;
ii. The error is calculated based on the received output (the difference between the predicted outcome and the achieved outcome);

iii. The error is back-propagated and its derivatives are obtained concerning all weights in the network, then the model is updated.

These three steps are repeated over multiple epochs to learn the ideal weights. Finally, the output is achieved through a threshold function to obtain the predicted class labels.

The error, i.e., the mean square error, is calculated using the following equation:

$$\Delta_w(t) = -\in \frac{dE}{dw_{(t)}} + \propto \Delta_w(t-1) \tag{2}$$

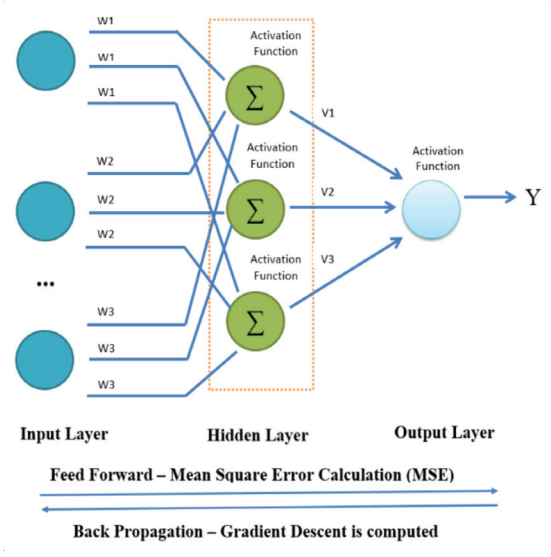


Figure 2. The multi-layer perceptron.

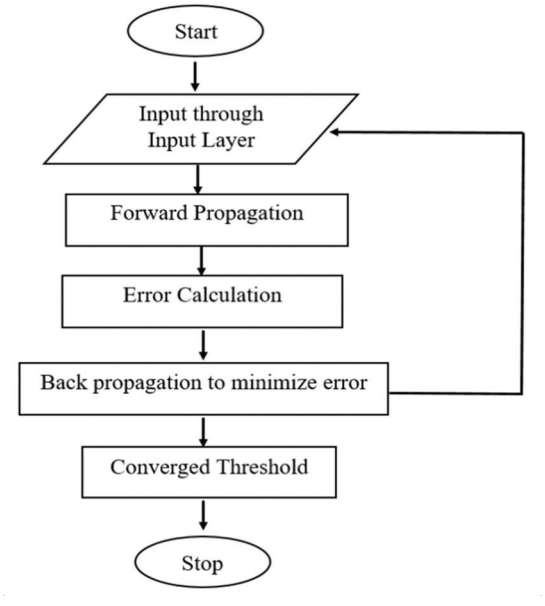


Figure 3. Learning process of the MLP.

Here, $\Delta_w(t)$ is the gradient of the current iteration, $\in$ is the bias, dE is the error in each iteration, the weight vector is represented by $dw_{(t)}$, $\propto$ represents the learning rate, and the gradient of the previous iteration is denoted by $\Delta_w(t-1)$.

This process continues until each input–output pair's gradient has converged, which means the freshly computed gradient has not changed more than the set convergence threshold since the previous iteration. Here, the network updates are performed incrementally.

4. Results and Discussion

4.1. Data Acquisition

Using the Google Play Scraper package with Python APIs, the dataset for three popular UPI mobile payment apps were collected. The three payment apps were GooglePay, PhonePe, and Paytm. Google Play Scraper offers Python APIs for crawling the Google Play Store without external dependencies. The details of the dataset obtained are as shown in Table 2. Here, we considered only positive and negative reviews, while neutral reviews were not considered.

Table 2. Datasets.

Dataset	Total Reviews	Positive	Negative
GooglePay	45,597	20,975	24,622
PhonePe	43,209	17,715	25,494
PayTM	47,932	33,073	14,859

In this process, the equations are numbered consecutively, with equation numbers shown in parentheses flush with the right margin of the column, as in (1). First, use the equation editor to create the equation. Then, select the "Equation" markup style. Press the tab key and write the equation number in parentheses. To make your equations more compact, you may use the solidus (/),exp function, or appropriate exponents. Use parentheses to avoid ambiguities in denominators. Punctuate equations when they are part of a sentence, as in:

$$B_p + H_2 = 40. \tag{3}$$

4.2. Data Augmentation

A balanced dataset facilitates the establishment of unambiguous decision limits for every class and enables models for the classification of data more precisely in any classification task. Any unbalanced dataset can be converted to a balanced one using data augmentation techniques, guaranteeing that the dataset is consistent across labels. The algorithm is named SMOTE [51], and is a commonly used data augmentation approach that may be used for any dataset without any influence on predictions based on a particular label. SMOTE samples the class with a minority with the help of a k-nearest neighbours classifier; it selects samples close to the feature space and generates synthesized data points. In this study, we use SMOTE to balance the dataset in terms of the labels and performs an evaluation.

4.3. Performance Measurement

To assess how well the suggested model works, an accuracy matrix is computed. For positive sentiment classification, true positive and false positive variables are identified. For negative sentiment classification, the true negative and true positive variables are defined as shown in Table 3.

Table 3. The accuracy parameters.

	Positive Class	**Negative Class**
Identification of Positive Class	X_1 = True Positive	Y_1 = False Positive
Identification of negative Class	X_2 = False Negative	Y_2 = True Negative

Using the parameters in Table 3, the following equation is defined to assess the accuracy of the proposed model:

$$\text{Accuracy}(Z) = \frac{X_1 + X_2}{Y_1 + Y_2 + X_1 + X_2} \tag{4}$$

4.4. Performance Evaluation

For a clear view of and simplicity in the graphical representations, the models are termed hereafter as shown in Table 4.

Table 4. The models and their aliases.

Models	**Alias**
CNN with Word2Vec	**MO-01**
BiLSTM with Word2Vec	**MO-02**
CNN with BERT	**MO-03**
BILSTM with BERT	**MO-04**
MFSC with CNN and Word2Vec	**MO-05**
MFSCwith CNN and BERT	**MO-06**
MFSC with BiLSTM and Word2Vec	**MO-07**
MFSCwith BiLSTM and BERT	**MO-08**
MFSCwith XLNet	**MO-09**

A hyperparameter is a value for a parameter that is used to influence the learning process. Different hyperparameters are tuned for optimized performance accuracy. Comprehensive experiments are performed using several hyperparameters, such as the embedding type, activation function, and dropout.

The deep learning methods CNN and BiLSTM with different word embedding methods, i.e., Word2Vec and BERT, are tested on different hyperparameters. The proposed model is also tuned with several hyperparameters. The hyperparameter tuning process is performed with different embedding combinations on 200, 300, and 400 words and with learning rates ranging from 0.01 to 0.10. The observations of these experiments are shown in Tables 5 and 6.

The above Table 5 provides the performance accuracy rates of different models with an embedding size of 200 with dropout from 0.01 to 0.10. All models M01, M02, M03, M04, M05, M06, M07, M08, and M09 are tested using this combination. It can be observed that the proposed model achieves the highest classification accuracy rate of 96.62% using a dropout rate of 0.10 for dataset 1.

For dataset 2, the highest accuracy can be observed for the dropout of 0.04 with 95.95% accuracy. At the same time, 96.36% accuracy is obtained for dataset 3 at a dropout rate of 0.04. The accuracy rates of the other models vary depending on the different dropout values. Overall, the proposed model shows the highest performance in terms of classification accuracy as compared to the other eight models.

Table 5. The performance accuracy (%) for an embedding size of 200.

Dropout = 0.01				**Dropout = 0.02**				**Dropout = 0.03**			
Models	**Dataset 1**	**Dataset 2**	**Dataset 3**	**Models**	**Dataset 1**	**Dataset 2**	**Dataset 3**	**Models**	**Dataset 1**	**Dataset 2**	**Dataset 3**
M01	61.66	60.35	62.33	M01	65.36	61.33	64.12	M01	66.19	64.32	62.15
M02	66.36	65.21	66.33	M02	61.32	64.22	62.14	M02	63.20	62.55	61.32
M03	70.66	68.55	68.32	M03	68.55	69.56	68.22	M03	64.32	66.25	65.32
M04	72.33	74.25	70.25	M04	71.42	72.22	70.65	M04	70.62	69.32	71.25
M05	81.65	81.56	83.22	M05	80.62	82.65	81.24	M05	81.55	84.12	83.85
M06	84.11	80.35	81.25	M06	82.15	81.25	83.36	M06	88.85	83.54	84.98
M07	86.32	84.25	83.22	M07	87.65	84.26	84.11	M07	91.65	89.55	89.99
M08	88.35	87.15	85.25	M08	89.22	88.95	88.01	M08	85.95	86.32	84.62
M09	93.28	91.56	92.36	M09	92.69	90.33	91.56	M09	95.62	95.05	95.99
(a)				(b)				(c)			
Dropout = 0.04				Dropout = 0.05				Dropout = 0.06			
Models	Dataset 1	Dataset 2	Dataset 3	Models	Dataset 1	Dataset 2	Dataset 3	Models	Dataset 1	Dataset 2	Dataset 3
M01	70.15	69.66	70.69	M01	68.35	64.69	66.35	M01	64.20	60.17	62.96
M02	72.66	73.21	73.65	M02	69.36	68.32	67.35	M02	61.32	62.65	60.98
M03	71.15	75.11	76.02	M03	61.25	62.35	61.22	M03	65.21	67.32	67.06
M04	77.62	78.65	77.12	M04	69.36	70.32	68.33	M04	70.26	71.06	69.49
M05	82.15	83.62	84.12	M05	84.63	83.98	84.05	M05	85.77	84.12	83.85
M06	86.66	85.95	86.01	M06	82.65	81.63	80.62	M06	85.19	83.54	84.98
M07	90.65	90.36	91.65	M07	91.65	90.61	89.63	M07	91.20	89.55	89.99
M08	92.15	91.62	92.99	M08	86.63	87.65	89.65	M08	87.97	86.32	84.62
M09	96.33	95.95	96.36	M09	94.32	95.62	93.64	M09	95.22	95.05	95.99
(d)				(e)				(f)			
Dropout = 0.07				Dropout = 0.08				Dropout = 0.09			
Models	Dataset 1	Dataset 2	Dataset 3	Models	Dataset 1	Dataset 2	Dataset 3	Models	Dataset 1	Dataset 2	Dataset 3
M01	65.32	64.12	63.25	M01	66.30	62.27	65.06	M01	62.10	63.43	61.76
M02	64.15	64.32	66.21	M02	68.74	67.70	66.73	M02	66.41	66.58	68.47
M03	61.15	62.35	65.32	M03	72.00	75.96	76.87	M03	60.60	61.70	60.57
M04	72.55	71.65	72.36	M04	69.07	69.87	68.30	M04	75.80	81.65	75.61
M05	83.15	84.13	85.65	M05	84.10	85.08	86.60	M05	85.08	86.06	87.58
M06	80.75	81.73	83.25	M06	88.31	87.60	87.66	M06	83.81	82.79	81.78
M07	85.82	86.80	88.32	M07	88.59	85.20	85.05	M07	93.91	92.87	91.89
M08	82.75	86.32	85.25	M08	90.16	89.89	88.95	M08	89.98	91.00	93.00
M09	90.72	91.70	93.22	M09	93.63	91.27	92.50	M09	93.97	92.65	93.29
(g)				(h)				(i)			
Dropout = 0.10											
Models	Dataset 1	Dataset 2	Dataset 3								
M01	65.95	84.32	63.88								
M02	69.93	68.89	67.92								
M03	67.41	88.65	69.26								
M04	83.89	84.87	86.39								
M05	79.83	81.30	81.80								
M06	87.47	88.45	89.97								
M07	87.57	88.59	90.59								
M08	91.63	90.35	93.32								
M09	96.62	94.32	95.32								
(j)											

Table 6. The performance accuracy (%) for an embedding size of 300.

Dropout = 0.01				**Dropout = 0.02**				**Dropout = 0.03**			
Models	**Dataset 1**	**Dataset 2**	**Dataset 3**	**Models**	**Dataset 1**	**Dataset 2**	**Dataset 3**	**Models**	**Dataset 1**	**Dataset 2**	**Dataset 3**
M01	67.36	64.32	68.62	M01	71.22	70.39	70.84	M01	72.32	72.32	72.32
M02	70.35	72.15	72.44	M02	72.51	72.87	72.42	M02	66.36	65.32	67.33
M03	74.22	73.12	71.56	M03	74.84	75.48	75.88	M03	61.63	62.36	64.36
M04	63.00	83.65	82.22	M04	76.48	77.51	75.98	M04	74.33	73.66	72.35
M05	95.65	91.59	94.22	M05	91.12	90.65	93.32	M05	84.36	83.22	86.35
M06	86.31	84.32	87.35	M06	89.25	88.65	85.65	M06	88.25	87.56	86.32
M07	92.56	93.32	91.21	M07	90.32	92.35	90.36	M07	90.65	91.63	91.54
M08	84.56	85.12	85.58	M08	93.65	94.62	92.65	M08	89.99	87.25	88.63
M09	92.36	94.25	91.35	M09	94.56	95.65	93.65	M09	96.32	94.32	95.33
(a)				(b)				(c)			
Dropout = 0.04				Dropout = 0.05				Dropout = 0.06			
Models	Dataset 1	Dataset 2	Dataset 3	Models	Dataset 1	Dataset 2	Dataset 3	Models	Dataset 1	Dataset 2	Dataset 3
M01	72.36	71.53	71.98	M01	70.51	67.47	71.77	M01	69.35	67.32	68.35
M02	73.65	74.01	73.56	M02	73.50	75.30	75.59	M02	72.12	71.15	73.65
M03	75.98	76.62	77.02	M03	77.37	76.27	74.71	M03	75.65	74.36	76.32
M04	77.62	78.65	77.12	M04	81.32	84.36	83.32	M04	84.35	81.36	82.35
M05	84.92	85.63	86.01	M05	85.21	86.07	84.14	M05	86.32	87.18	85.25
M06	87.16	87.9	88.1	M06	83.21	84.21	85.00	M06	84.32	85.32	86.11
M07	91.21	91.56	92.01	M07	89.14	90.41	88.25	M07	90.25	91.52	89.36
M08	93.63	94.01	93.9	M08	87.42	86.04	85.21	M08	88.53	87.15	86.32
M09	**97.23**	**97.65**	**97.01**	M09	95.14	93.21	94.77	M09	96.25	94.32	95.88
(d)				(e)				(f)			
Dropout = 0.07				Dropout = 0.08				Dropout = 0.09			
Models	Dataset 1	Dataset 2	Dataset 3	Models	Dataset 1	Dataset 2	Dataset 3	Models	Dataset 1	Dataset 2	Dataset 3
M01	73.83	70.79	75.09	M01	72.16	71.33	71.78	M01	72.90	71.93	74.43
M02	74.84	75.20	74.75	M02	72.88	74.68	74.97	M02	77.10	77.46	77.01
M03	76.16	75.06	73.50	M03	76.83	77.47	77.87	M03	76.72	75.62	74.06
M04	80.07	83.11	82.07	M04	83.16	80.17	81.16	M04	83.32	81.65	85.32
M05	86.37	87.23	85.30	M05	87.32	88.18	86.25	M05	86.33	84.12	85.22
M06	84.37	85.37	86.16	M06	88.81	89.55	89.75	M06	84.21	86.32	82.55
M07	90.30	91.57	89.41	M07	91.26	93.29	91.30	M07	91.56	90.21	91.24
M08	88.58	87.20	86.37	M08	94.59	95.56	93.59	M08	92.56	91.25	91.11
M09	96.30	94.37	95.93	M09	95.50	96.59	94.59	M09	95.62	94.12	95.22
(g)				(h)				(i)			
Dropout = 0.10											
Models	Dataset 1	Dataset 2	Dataset 3								
M01	74.46	84.32	75.72								
M02	74.07	75.87	76.16								
M03	77.85	88.65	78.52								
M04	85.14	82.93	84.03								
M05	87.53	88.39	86.46								
M06	85.53	86.53	87.32								
M07	91.46	92.73	90.57								
M08	89.74	88.36	87.53								
M09	97.46	95.53	97.09								
(j)											

Table 6 shows the classification accuracy performance for the embedding size of 300 and with dropout rates ranging from 0.01 to 0.10. As per the observations for the above figure, it is clear that none of the models shows consistent performance. For example, model M01 shows an accuracy rate of 67.36% for dataset 1, but for dataset 2 the accuracy decreases to 64.32%, and again the model achieves a higher accuracy rate of 68.62% for dataset 3, with a dropout rate of 0.01. Model M02 achieves its highest accuracy rate of 77.46% for dataset 2 with a dropout rate of 0.09, whereas the lowest accuracy rate of 67.33% is achieved with a dropout rate of 0.04. The observations from the experiments with an embedding size of 300 and dropout rate of 0.03 indicate that this combination with other hyperparameters has shown consistent performance for all models.

Table 7 shows the accuracy performance for the embedding size of 400 and with dropout rates ranging from 0.01 to 0.10. The observations show that except for the proposed model, none of the models show consistency.

Table 7. The performance accuracy (%) for an embedding size of 400.

Dropout = 0.01				**Dropout = 0.02**				**Dropout = 0.03**			
Models	**Dataset 1**	**Dataset 2**	**Dataset 3**	**Models**	**Dataset 1**	**Dataset 2**	**Dataset 3**	**Models**	**Dataset 1**	**Dataset 2**	**Dataset 3**
M01	64.32	66.33	65.24	M01	66.55	67.36	68.22	M01	62.35	66.35	64.21
M02	66.55	64.32	62.33	M02	68.36	67.21	69.36	M02	66.32	67.24	65.32
M03	68.36	68.32	70.56	M03	70.22	71.56	72.32	M03	70.25	69.68	71.56
M04	70.25	71.52	72.22	M04	72.36	73.32	71.35	M04	74.65	73.22	74.01
M05	74.36	76.32	72.52	M05	75.62	78.32	74.22	M05	76.32	77.25	74.35
M06	76.32	78.25	77.85	M06	77.55	75.22	76.32	M06	81.65	82.54	80.26
M07	84.66	85.65	83.26	M07	79.65	80.25	91.56	M07	84.68	85.10	86.32
M08	89.56	88.32	90.23	M08	84.32	82.35	83.77	M08	89.62	90.21	91.25
M09	94.35	94.56	93.26	M09	90.21	91.36	91.55	M09	94.56	95.21	94.96
(a)				(b)				(c)			
Dropout = 0.04				Dropout = 0.05				Dropout = 0.06			
Models	Dataset 1	Dataset 2	Dataset 3	Models	Dataset 1	Dataset 2	Dataset 3	Models	Dataset 1	Dataset 2	Dataset 3
M01	67.87	68.68	69.54	M01	69.25	66.32	68.32	M01	66.32	68.21	65.22
M02	70.76	68.53	66.54	M02	72.65	73.26	71.25	M02	68.21	69.32	70.21
M03	73.54	74.88	75.64	M03	74.32	74.21	74.88	M03	73.32	70.54	72.25
M04	78.53	79.65	77.52	M04	77.36	78.65	79.32	M04	76.95	75.32	74.55
M05	81.32	82.52	93.56	M05	81.54	80.32	81.01	M05	81.65	80.32	84.32
M06	86.32	86.21	86.55	M06	84.56	85.65	86.32	M06	86.32	87.21	85.32
M07	88.25	87.36	89.32	M07	88.32	87.36	90.32	M07	88.51	89.32	88.81
M08	91.52	91.98	92.65	M08	92.65	93.25	91.35	M08	91.56	93.35	92.80
M09	94.23	95.88	96.21	M09	92.54	94.36	95.21	M09	96.21	95.18	94.21
(d)				(e)				(f)			
Dropout = 0.07				Dropout = 0.08				Dropout = 0.09			
Models	Dataset 1	Dataset 2	Dataset 3	Models	Dataset 1	Dataset 2	Dataset 3	Models	Dataset 1	Dataset 2	Dataset 3
M01	69.21	70.25	68.11	M01	65.32	67.21	66.25	M01	70.50	71.61	72.50
M02	70.22	71.56	72.54	M02	68.22	69.01	70.15	M02	73.27	72.36	75.60
M03	71.32	70.41	73.65	M03	70.21	71.15	69.32	M03	75.79	74.50	76.90
M04	76.32	79.25	78.22	M04	72.54	71.25	73.65	M04	80.00	80.70	78.90
M05	78.00	79.15	77.25	M05	78.65	79.35	77.55	M05	83.19	81.97	82.66
M06	80.21	81.56	83.32	M06	81.36	83.35	82.65	M06	87.64	87.53	87.87
M07	83.55	84.32	85.11	M07	85.65	84.32	86.35	M07	90.23	89.34	91.30
M08	88.55	89.56	88.36	M08	89.32	90.35	90.99	M08	94.88	96.67	96.12
M09	92.25	93.36	93.35	M09	94.21	92.25	91.36	M09	94.75	96.57	97.42
(g)				(h)				(i)			

Table 7. *Cont.*

Dropout = 0.10			
M01	67.25	68.32	69.11
M02	70.22	71.66	69.21
M03	72.35	74.31	71.33
M04	77.55	76.32	79.65
M05	80.32	79.55	80.11
M06	83.35	84.22	85.21
M07	88.66	87.32	86.21
M08	90.32	90.11	90.56
M09	94.21	96.21	91.56
(j)			

Table 8 above shows the average performance accuracy of each model for the three datasets. The average accuracy is measured on dropout rates ranging from 0.01 to 0.10 for an embedding size of 200. Model M01 exhibits the lowest accuracy rate of 61.45% for the 0.01 dropout rate and the highest average accuracy rate of 71.38% for the 0.10 dropout rate. Model M02 has the lowest average accuracy rate of 61.65% for the dropout rate of 0.06 and the highest average accuracy rate of 73.17% for the dropout rate of 0.04. For models M03, M04, M05, and M06, the lowest observed performance results are 60.96% for a dropout rate of 0.09, 69.08% for a 0.08 dropout rate, 80.98% for a dropout rate of 0.10, and 81.90% for a dropout rate of 0.01, respectively. The highest accuracy rates achieved for these models are 75.11% for M03 using a dropout rate of 0.10, 85.05% for M04 on a dropout rate of 0.10, and 86.24% for M05 using a dropout rate of 0.09, while for M06, the highest average accuracy can be observed for a dropout rate of 0.10, with 88.63%. The highest average performance rate for model M07 can be observed for a dropout rate of 0.09%, with an accuracy rate of 92.89%, whereas the lowest average accuracy rate of 84.60% can be observed with a floor dropout rate of 0.01%. The performance of the proposed model is the highest among all models, with the lowest average accuracy rate of 91.53% for a dropout rate of 0.02, whereas the highest accuracy rate of 96.21% can be observed for a dropout rate of 0.04. In Table 8, the observations clearly show that the proposed model performs much better and is more consistent for all dropout rates as compared to the other eight models.

Table 8. Average classification accuracy (%) results for an embedding size of 200.

	Dropout									
Models	**0.01**	**0.02**	**0.03**	**0.04**	**0.05**	**0.06**	**0.07**	**0.08**	**0.09**	**0.1**
M01	61.45	63.60	64.22	70.17	66.46	62.44	64.23	64.54	62.43	71.38
M02	65.97	62.56	62.36	73.17	68.34	61.65	64.89	67.72	67.15	68.91
M03	69.18	68.78	65.30	74.09	61.61	66.53	62.94	74.94	60.96	75.11
M04	72.28	71.43	70.40	77.80	69.34	70.27	72.19	69.08	75.44	85.05
M05	82.14	81.50	83.17	83.30	84.22	84.58	84.31	85.26	86.24	80.98
M06	81.90	82.25	85.79	86.21	81.63	84.57	81.91	87.86	82.79	88.63
M07	84.60	85.34	90.40	90.89	90.63	90.25	86.98	86.28	92.89	88.92
M08	86.92	88.73	85.63	92.25	87.98	86.30	84.77	89.67	91.33	91.77
M09	92.40	91.53	95.55	96.21	94.53	95.42	91.88	92.47	94.18	95.42

Table 9 shows the comparative observations of all models with dropout rates of 0.01 to 0.10 for an embedding size of 300. Again, the observations show that none of the models achieve better performance than the proposed model. For an embedding size of 300, all the models show much better performance as compared to the embedding size of 200. Model M01 shows the lowest average accuracy rate of 66.77%, which is 5.32% more than that of the embedding size of 200. The highest performance rate for model M01 is 78.17% for a dropout rate of 0.1, which is again much better than the performance of model M01, which is just 71.38% for the embedding size of 200. Model M02 has the lowest average accuracy rate of 66.34% for the dropout rate of 0.04. The highest performance accuracy rate for M02 of 77.19% can be observed for the dropout rate of 0.09. For the dropout rate of 0.01, an exceptional case can be identified for model M05, which shown better performance than model M09, with an average accuracy rate of 93.82%, while the proposed model shows a 92.65% average accuracy rate. The overall observations in Table 9 show that except for model M09, none of the models are consistent, but the proposed model M09 shows clear and consistent performance, with the highest average accuracy rate of 97.3% for the dropout rate of 0.03 and embedding size of 300.

Table 9. Average classification accuracy (%) results for an embedding size of 300.

	Dropout									
Models	**0.01**	**0.02**	**0.03**	**0.04**	**0.05**	**0.06**	**0.07**	**0.08**	**0.09**	**0.1**
M01	66.77	70.82	72.32	71.96	69.92	68.34	73.24	71.76	73.09	78.17
M02	71.65	72.60	66.34	73.74	74.80	72.31	74.93	74.18	77.19	75.37
M03	72.97	75.40	62.78	76.54	76.12	75.44	74.91	77.39	75.47	81.67
M04	76.29	76.66	73.45	77.80	83.00	82.69	81.75	81.50	85.00	84.03
M05	93.82	91.70	84.64	85.52	85.14	86.25	86.30	87.25	88.23	87.46
M06	85.99	87.85	87.38	87.72	84.14	85.25	85.30	89.37	85.30	86.46
M07	92.36	91.01	91.27	91.59	89.27	90.38	90.43	91.95	91.53	91.59
M08	85.09	93.64	88.62	93.85	86.22	87.33	87.38	94.58	89.57	88.54
M09	92.65	94.62	95.32	97.30	94.80	95.91	95.53	95.56	94.45	97.12

For the embedding size of 400 and using different dropout rates ranging from 0.01 to 0.10, the average classification accuracy results are shown in Table 10. As far as the performance is considered, the same trend can also be observed here, showing that the proposed model M09 outperforms the other models but these embedding and dropout combinations do not achieve the highest and most consistent performance for all models as well as the proposed model. The proposed model shows better performance than the other models, but these hyperparameter combinations do not achieve the best performance.

Figure 4a–c depict the average performance accuracy results for all of the models for the three datasets. Figure 4a shows the average performance accuracy results for an embedding size of 200 and with dropout rates ranging from 0.01 to 0.10. Figure 4b shows the average performance accuracy results for an embedding size of 300 and with the dropout rates ranging from 0.01–0.10. Figure 4c shows the average performance accuracy results for an embedding size of 400 and with the dropout rates ranging from 0.01 to 0.10. The experimental findings for the three datasets demonstrate that the proposed model shows effective and efficient performance over the other models, and except for very few combinations of hyperparameters, the models do not show consistent performance results.

Table 10. Average classification accuracy (%) results for an embedding size of 400.

	Dropout									
Models	**0.01**	**0.02**	**0.03**	**0.04**	**0.05**	**0.06**	**0.07**	**0.08**	**0.09**	**0.1**
M01	65.30	67.38	64.30	68.70	67.96	66.58	69.19	66.26	71.54	68.23
M02	64.40	68.31	66.29	68.61	72.39	69.25	71.44	69.13	73.74	70.36
M03	69.08	71.37	70.50	74.69	74.47	72.04	71.79	70.23	75.73	72.66
M04	71.33	72.34	73.96	78.57	78.44	75.61	77.93	72.48	79.87	77.84
M05	74.40	76.05	75.97	85.80	80.96	82.10	78.13	78.52	82.61	79.99
M06	77.47	76.36	81.48	86.36	85.51	86.28	81.70	82.45	87.68	84.26
M07	84.52	83.82	85.37	88.31	88.67	88.88	84.33	85.44	90.29	87.40
M08	89.37	83.48	90.36	92.05	92.42	92.57	88.82	90.22	95.89	90.33
M09	94.06	91.04	94.91	95.44	94.04	95.20	92.99	92.61	96.25	93.99

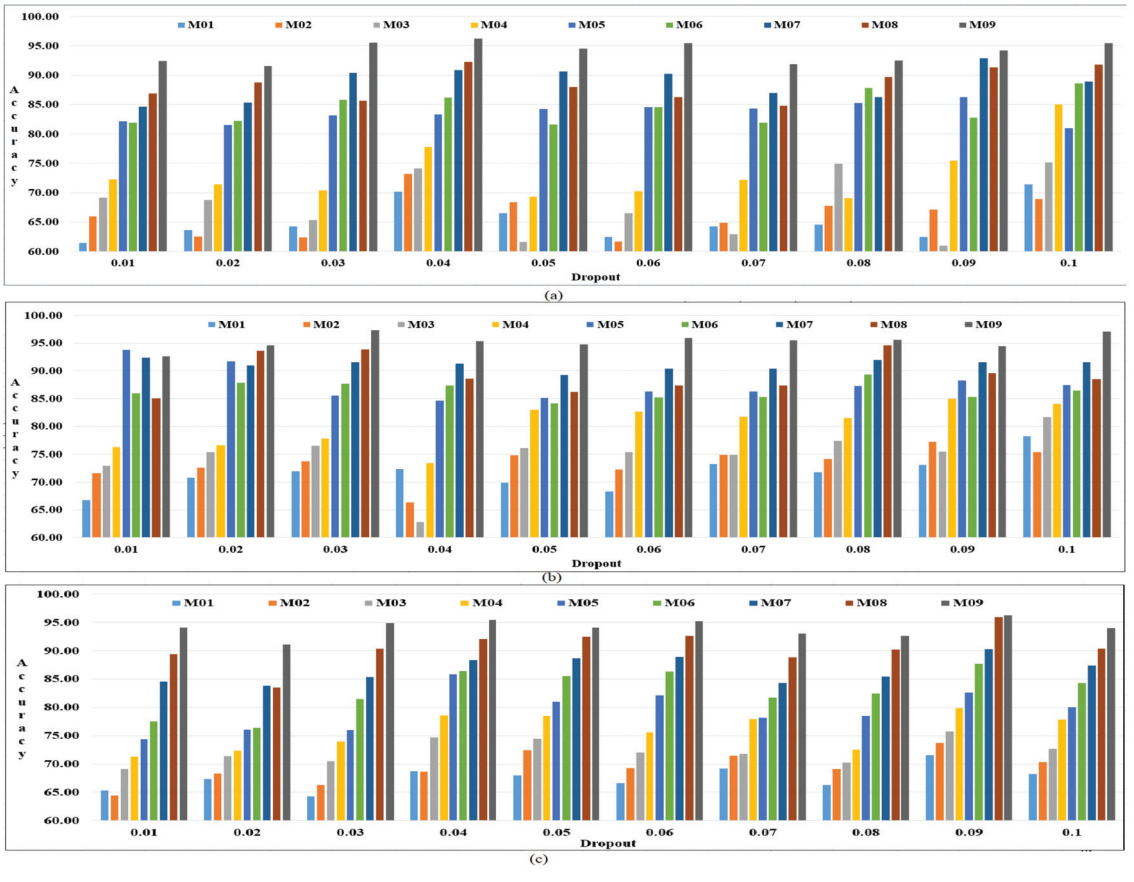


Figure 4. Average accuracy performance results for different embedding sizes: (**a**) embedding size of 200; (**b**) embedding size of 300; (**c**) embedding size of 400.

Out of all the models under consideration, and particularly as compared models M01 and M02, when Word2Vec is applied with CNN and BiLSTM, respectively, the response of the model is very poor. If BERT is used in place of Word2Vec then some improvement can be observed in inaccuracy, which shows the effectiveness of the BERT model in text classification. The BERT model shows its supremacy over the Word2Vec model, with improvements of 5% to 10% for sentiment classification. Models M05, M06, M07, and M08 also show improvements, but the proposed model shows the highest and most consistent performance for all datasets for the embedding size of 300 and dropout rate of 0.03. Since this combination showed consistent performance for other models, the embedding size 300 and dropout rate of 0.03 were implemented on all datasets for all models to conduct further experiments, as shown in Table 11.

Table 11. Hyperparameters settings.

Models	Techniques	Embedding	Activation	Embedding Size	Dropout	Optimizer	Epochs	Filters
M01	*CNN*	Word2Vec	ReLu	300	0.03	*sgd*	50	512
M02	*BiLSTM*	Word2Vec	ReLu	300	0.03	*sgd*	50	-
M03	*CNN*	BERT	ReLu	300	0.03	*sgd*	50	512
M04	*BILSTM*	BERT	ReLu	300	0.03	*sgd*	50	-
M05	MFMLSC	Word2Vec	ReLu	300	0.03	*sgd*	50	-
M06	MFMLSC	Word2Vec	ReLu	300	0.03	*sgd*	50	-
M07	MFMLSC	BERT	ReLu	300	0.03	*sgd*	50	-
M08	MFMLSC	BERT	ReLu	300	0.03	*sgd*	50	-
M09	MFMLSC	XLNet	ReLu	300	0.03	*sgd*	50	-

4.5. Evaluation of Multi-Fold Model of Sentiment Classification (MFSC)

To investigate the performance of a sentiment classification approach that relies solely on multi-dimensional sentiment modeling, the performance of the proposed multi-fold sentiment modeling method with XLNet (MFSC) shown in Table 12 and Figure 5 is compared with a CNN with Word2Vec, BiLSTM with Word2Vec, CNN with BERT, and BILSTM with BERT. The methods are discussed below.

CNN with Word2Vec: Firstly, Word2Vec is used to initialize the vectorized word, following which CNN is applied to extract the features of the sentiments from the dataset, and finally a fully connected network is used for sentiment classification of the social media text.

BiLSTM with Word2Vec: In this instance, Word2Vec is applied to achieve the word vectors, then BiLSTM is implemented for extraction of the sentiment characteristics of a given dataset, and finally a fully connected network is used for implement sentiment classification of the dataset.

CNN with BERT: The initialization of the word vector is accomplished with the help of BERT, then the CNN is applied for extraction of the sentiment features of the dataset, and finally a fully connected network is used for sentiment classification of the dataset.

BILSTM with BERT: Here, BERT is utilized to initialize the vector of words, followed by the BiLSTM technique being used for extraction of the sentiment features of the dataset, then in the last phase a fully connected network is used to implement sentiment classification of a dataset.

Table 12. Sentiment classification accuracy (%) results using the MFSC model.

Methods	Dataset 1	Dataset 2	Dataset 3	Average
M01	72.36	71.53	71.98	71.96
M02	73.65	74.01	73.56	73.74
M03	75.98	76.62	77.02	76.54
M04	77.62	78.65	77.12	77.80
M05	83.65	82.98	83.12	83.25
M06	85.61	84.52	86.32	85.48
M07	89.56	89.98	90.32	89.95
M08	91.96	92.05	92.25	92.09
M09	**94.32**	**94.1**	**95.01**	**94.48**

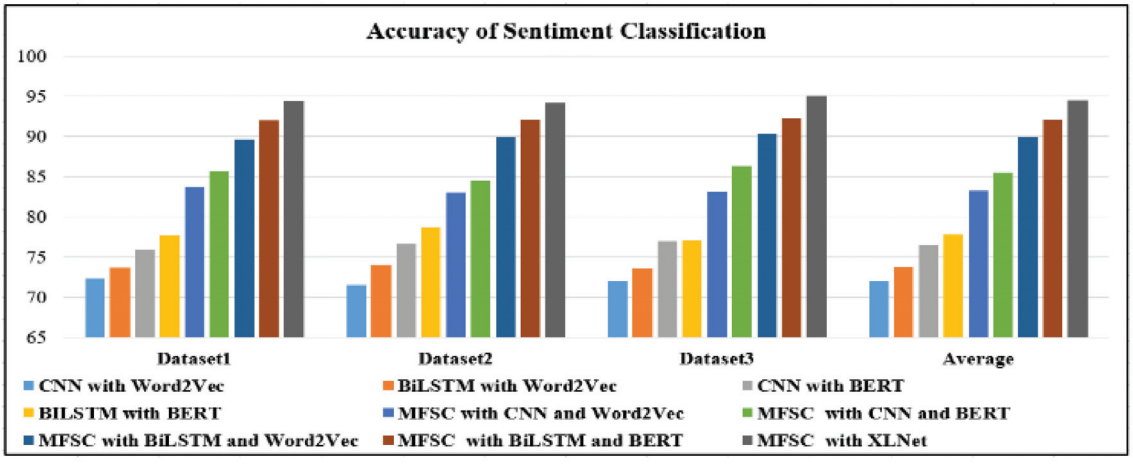


Figure 5. Graphical representation of the performance results with the MFSC model.

MFSM with CNN and Word2Vec: The Word2Vec, CNN, and MFSM approaches are used to classify sentiments. To begin, emoji-based symbols are treated as language symbols in a social media text. Next, Word2Vec is implemented to for the initialization of the word vector, and the CNN extracts sentiment characteristics from the dataset. Finally, the sentiment categorization approach is accomplished through a completely connected network.

MFSM with CNN and BERT: the BERT, CNN, and MFSM approaches are used to create a sentiment classification system. To begin, both language symbols and emoticon symbols are handled in datasets in the same manner as language symbols. Next, BERT is used for the initialization of the word vector, and the CNN is implemented to extract the emotional components of the dataset. Finally, the sentiment categorization approach is accomplished through a completely connected network.

MFSM with BiLSTM and Word2Vec: The Word2Vec, BiLSTM, and MFSM-based sentiment categorization approaches are used. To begin, all symbols in a dataset, including language symbols and emoticon symbols, are regarded as language symbols. The vector of the word is then initialized using Word2Vec, and the BiLSTM model extracts features of sentiments from the dataset. Finally, the sentiment categorization approach is accomplished through a completely connected network.

MFSM with BiLSTM and BERT: This is a sentiment categorization approach based on the BERT, BiLSTM, and MFSM models. To begin, in the dataset, language symbols and emoticon symbols are both treated as language symbols. The BiLSTM model collects sentiment characteristics from the dataset after initializing the word vector with BERT. Finally, a completely connected network is used to achieve sentiment categorization.

4.6. Evaluation of Multi-Level Model of Sentiment Classification (MLSC)

In the second phase of the performance evaluation of the proposed model, the evaluation is conducted only with the multi-dimension model of sentiment classification (MLSC). The MDSC model with XLNet is compared with the CNN with Word2Vec, BiLSTM with Word2Vec, CNN with BERT, and BILSTM with BERT approaches, as shown in Table 13 and Figure 6. In addition to these models, the MDSC model is also implemented with the abovementioned techniques.

Table 13. Sentiment classification accuracy results using the MLSC.

Methods	Dataset 1	Dataset 2	Dataset 3	Average
M01	72.36	71.53	71.98	71.96
M02	73.65	74.01	73.56	73.74
M03	75.98	76.62	77.02	76.54
M04	77.62	78.65	77.12	77.80
M05	83.65	84.21	84.32	84.06
M06	86.33	85.98	86.1	86.14
M07	89.32	88.75	89.1	89.06
M08	92.62	92.78	91.92	92.44
M09	**95.51**	**95.32**	**95.98**	**95.60**

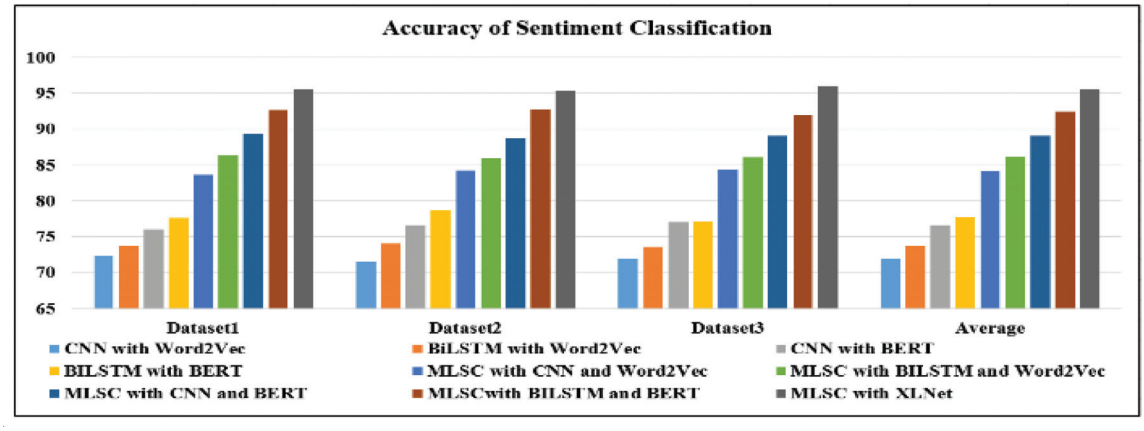


Figure 6. Graphical representation of the performance results with the MLSC.

MLSC with CNN and Word2Vec: The classification of sentiments is accomplished with the assistance of the Word2Vec, CNN, and MLSM models. Initially, the vectorized word is populated with the help of Word2Vec, and then with a CNN-based attention mechanism, the emotional characteristics of the dataset are retrieved from different levels of words,

sentences, and phrases. Lastly, the completely linked network is used to implement the sentiment classification.

MLSC with BILSTM and Word2Vec: This is a Word2Vec, BiLSTM, and MDSC-based sentiment categorization algorithm. Here, Word2Vec is used to initialize the word vector, and then BiLSTM is used to extract sentiment features of the dataset from different levels of words and sentences using an attention mechanism. Finally, the completely linked network is used for sentiment classification in the given dataset.

MLSC with CNN and BERT: This is a BERT, CNN, and MDSC-based sentiment classification approach. The word vector's initialization is achieved using BERT, and then the CNN is utilized to extract the sentiment features of the dataset from different levels, as discussed using an attention mechanism. Finally, the completely linked network is used for the sentiment classification of the dataset.

MLSC with BILSTM and BERT: This is a BERT, BiLSTM, and MDSC-based sentiment classification approach. BERT is used to initialize the word vector, and then BiLSTM is utilized to extract the sentiment features of the dataset from the given levels using an attention mechanism. In the final phase, using a fully interconnected computer network, the dataset classification process is carried out

4.7. Assessment of Multi-Fold and Multi-Level Modeling of Sentiment Method (MFMLSC)

To assess our method's overall performance, the performance results in terms of the multi-fold and multi-level classification for the sentiment method are compared with the methods discussed in the previous section.

As shown in Figure 7 and Table 14, the proposed model achieves the maximum performance as compared to the other deep learning models that use combinations of different deep learning and word embedding models. For the embedding size of 300 and dropout rate of 0.03, the proposed MFMLSC shows the highest accuracy rates during sentiment classification, with scores of 97.23%, 97.65%, and 97.01% for datset 1, dataset 2, and datset 3, respectively. The proposed model outperforms the other models, with an average accuracy rate of 97.30%.

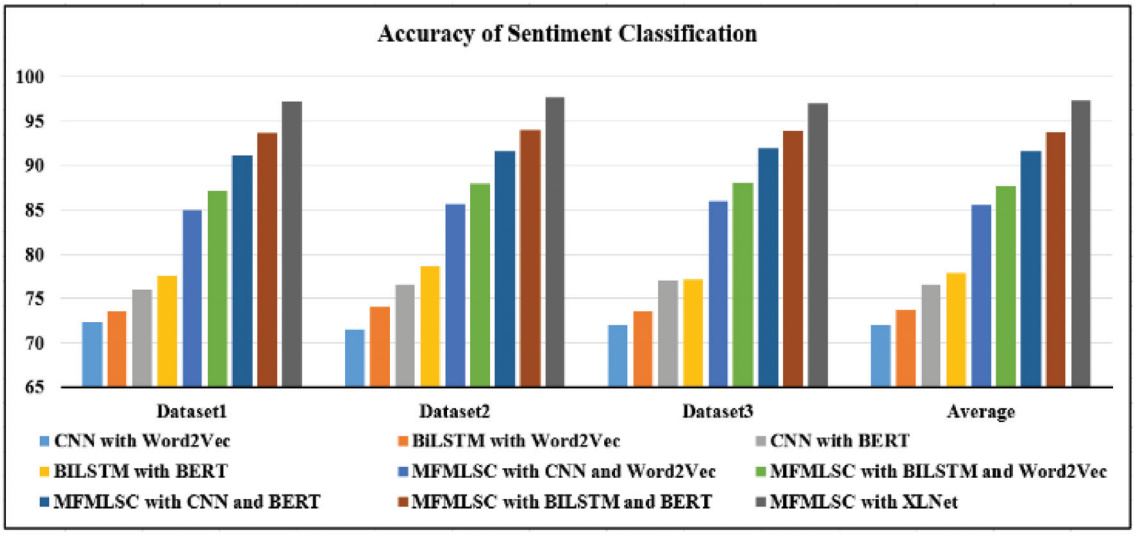


Figure 7. Graphical representation of the performance with the MFMLSC.

Table 14. Accuracy results for sentiment classification using the MFMLSC.

Methods	Dataset 1	Dataset 2	Dataset 3	Average
M01	72.36	71.53	71.98	71.96
M02	73.65	74.01	73.56	73.74
M03	75.98	76.62	77.02	76.54
M04	77.62	78.65	77.12	77.80
M05	84.92	85.63	86.01	85.52
M06	87.16	87.9	88.1	87.72
M07	91.21	91.56	92.01	91.59
M08	93.63	94.01	93.9	93.85
M09	**97.23**	**97.65**	**97.01**	**97.30**

5. Conclusions

We observed that the autoregressive-based model for sentiment classification that uses the pre-trained word vector XLNet showed the greatest classification accuracy, with an average of 97.30% accuracy for all datasets. The proposed model solved the problem of the lack of semantic information in reviews, which affects the accuracy during classification. The experimental findings demonstrated that when compared to the current methods, our method significantly increases the accuracy of the sentiment classification process for social media datasets.

Author Contributions: Methodology, R.R., D.P., A.K.R. and P.S.; Software, R.R., A.K.R. and P.S.; Validation, D.P. and A.K.R.; Formal analysis, P.S., A.V. and D.G.; Investigation, A.V. and D.G.; Resources, P.R.K.; Data curation, P.R.K.; Writing—original draft, P.R.K.; Writing—review & editing, S.N.M.; Visualization, S.N.M.; Supervision, S.N.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Article processing charges supported by School of Computer Science & Engineering(SCOPE), VIT-AP University, Amaravati, Andhra Pradesh, India.

Conflicts of Interest: The authors declare no competing interests.

References

1. Vicario, M.D.; Vivaldo, G.; Bessi, A.; Zollo, F.; Scala, A.; Caldarelli, G.; Quattrociocchi, W. Echo Chambers: Emotional Contagion and Group Polarization on Facebook. *Sci. Rep.* **2016**, *6*, 37825. [CrossRef] [PubMed]
2. Kazameini, A.; Fatehi, S.; Mehta, Y.; Eetemadi, S.; Cambria, E. Personality trait detection using bagged SVM over BERT word embedding ensembles. *arXiv* **2020**, arXiv:2010.01309.
3. Genc-Nayebi, N.; Abran, A. A systematic literature review: Opinion mining studies from mobile app store user reviews. *J. Syst. Softw.* **2017**, *125*, 207–219. [CrossRef]
4. Katarya, R. A review: Predicting the performance of students using machine learning classification techniques. In Proceedings of the 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 12–14 December 2019; pp. 36–41.
5. Ahmad, H.; Asghar, M.Z.; Alotaibi, F.M.; Hameed, I.A. Applying deep learning technique for depression classification in social media text. *J. Med. Imag. Health Informat.* **2020**, *10*, 2446–2451. [CrossRef]
6. Deerwester, S.; Dumais, S.T.; Furnas, G.W.; Landauer, T.K.; Harshman, R. Indexing by latent semantic analysis. *J. Am. Soc. Inf. Sci.* **1990**, *41*, 391–407. [CrossRef]

7. Bengio, Y.; Ducharme, R.; Vincent, P.; Jauvin, C. A neural probabilistic language model. *J. Mach. Learn. Res.* **2013**, *3*, 1137–1155.
8. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 3–6 December 2012; pp. 1097–1105.
9. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2017; pp. 5998–6008.
10. Zhang, L.; Wang, S.; Liu, B. Deep learning for sentiment analysis: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1253. [CrossRef]
11. Levy, O.; Goldberg, Y.; Dagan, I. Improving distributional similarity with lessons learned from word embeddings. *Trans. Assoc. Comput. Linguist.* **2015**, *3*, 211–225. [CrossRef]
12. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Volume 14, pp. 1532–1543.
13. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. 2013. Available online: https://arxiv.org/abs/1301.3781 (accessed on 7 September 2022).
14. Tang, D.Y.; Wei, F.; Yang, N.; Zhou, M.; Liu, T.; Qin, B. Learning sentiment-specific word embedding for Twitter sentiment classification. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, 22 June 2014; pp. 1555–1565.
15. Turney, P.D. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; pp. 417–424.
16. Ali, F.; El-Sappagh, S.; Kwak, D. Fuzzy ontology and LSTM-based text mining: A transportation network monitoring system for assisting travel. *Sensors* **2019**, *19*, 234. [CrossRef]
17. Pham, D.-H.; Le, A.-C. Learning multiple layers of knowledge representation for aspect based sentiment analysis. *Data Knowl. Eng.* **2014**, *114*, 26–39. [CrossRef]
18. Jianqiang, Z.; Xiaolin, G.; Xuejun, Z. Deep convolution neural networks for twitter sentiment analysis. *IEEE Access* **2018**, *6*, 23253–23260. [CrossRef]
19. Zhao, R.; Mao, K. Fuzzy bag-of-words model for document representation. *IEEE Trans. Fuzzy Syst.* **2018**, *26*, 794–804. [CrossRef]
20. Sharma, N.; Mangla, M.; Mohanty, S.N. Supervised Learning Techniques for Sentiment Analysis. In *Emerging Technologies in Data Mining and Information Security*; Dutta, P., Chakrabarti, S., Bhattacharya, A., Dutta, S., Shahnaz, C., Eds.; Lecture Notes in Networks and Systems; Springer: Singapore, 2023; Volume 490. [CrossRef]
21. Chandra, S.; Gourisaria, M.K.; Harshvardhan, G.M.; Rautaray, S.S.; Pandey, M.; Mohanty, S.N. Semantic Analysis of Sentiments through Web-Mined Twitter Corpus. In Proceedings of the International Semantic Intelligence Conference 2021 (ISIC 2021), New Delhi, India, 25–27 February 2021; CEUR Workshop Proceedings 2786, CEUR-WS.org 202; pp. 122–135.
22. Ali, F.; Ali, A.; Imran, M.; Naqvi, R.A.; Siddiqi, M.H.; Kwak, K.-S. Traffic accident detection and condition analysis based on social networking data. *Accid. Anal. Prev.* **2021**, *151*, 105973. [CrossRef] [PubMed]
23. Guo, Y.; Li, W.; Jin, C.; Duan, Y.; Wu, S. An integrated neural model for sentence classification. In Proceedings of the 2018 Chinese Control and Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 6268–6273.
24. Dandala, B.; Joopudi, V.; Devarakonda, M. Adverse drug events detection in clinical notes by jointly modeling entities and relations using neural networks. *Drug Saf.* **2019**, *42*, 135–146. [CrossRef]
25. Feizollah, A.; Ainin, S.; Anuar, N.B.; Abdullah, N.A.B.; Hazim, M. Halal products on Twitter: Data extraction and sentiment analysis usingstack of deep learning algorithms. *IEEE Access* **2019**, *7*, 83354–83362. [CrossRef]
26. Zhang, Z.; Zou, Y.; Gan, C. Textual sentiment analysis via three different attention convolutional neural networks and cross-modality consistent regression. *Neurocomputing* **2018**, *275*, 1407–1415. [CrossRef]
27. Hameed, Z.; Garcia-Zapirain, B. Sentiment classification using a single-layered BiLSTM model. *IEEE Access* **2021**, *8*, 73992–74001. [CrossRef]
28. Xu, G.; Meng, Y.; Qiu, X.; Yu, Z.; Wu, X. Sentiment analysis of comment texts based on BiLSTM. *IEEE Access* **2019**, *7*, 51522–51532. [CrossRef]
29. Priyadarshini, I.; Cotton, C. A novel LSTM–CNN–grid search-based deep neural network for sentiment analysis. *J. Supercomput.* **2021**, *77*, 13911–13932. [CrossRef]
30. Mandhula, T.; Pabboju, S.; Gugalotu, N. Predicting the customer's opinion on amazon products using selective memory architecture-based convolutional neural network. *J. Supercomput.* **2019**, *76*, 5923–5947. [CrossRef]
31. Dong, J.; He, F.; Guo, Y.; Zhang, H. A commodity review sentiment analysis based on BERTCNN model. In Proceedings of the 5th International Conference on Computer And Communication Systems (ICCCS), Shanghai, China, 15–18 May 2020; pp. 143–147. [CrossRef]
32. Wu, F.; Shi, Z.; Dong, Z.; Pang, C.; Zhang, B. Sentiment analysis of online product reviews based on SenBERT-CNN. In Proceedings of the 2020 International Conference on Machine Learning and Cybernetics (ICMLC), Adelaide, Australia, 2 December 2020; pp. 229–234. [CrossRef]
33. Colón-Ruiz, C.; Segura-Bedmar, I. Comparing deep learning architectures for sentiment analysis on drug reviews. *J. Biomed. Inform.* **2020**, *110*, 103539. [CrossRef] [PubMed]

34. Munikar, M.; Shakya, S.; Shrestha, A. Fine-grained sentiment classification using BERT. In Proceedings of the 2019 Artificial Intelligence for Transforming Business and Society (AITB), Kathmandu, Nepal, 5 November 2019; IEEE: Piscataway, NJ, USA, 2019.
35. Pota, M.; Ventura, M.; Catelli, R.; Esposito, M. An effective BERT-based pipeline for twitter sentiment analysis: A case study in ITALIAN. *Sensors* **2021**, *21*, 133. [CrossRef] [PubMed]
36. Fan, B.; Fan, W.; Smith, C.; Garner, H.S. Adverse drug event detection and extraction from open data: A deep learning approach. *Inf. Process. Manage.* **2020**, *57*, 102131. [CrossRef]
37. Chen, T.; Wu, M.; Li, H. A general approach for improving deep learning-based medical relation extraction using a pre-trained model and fine-tuning. *Database* **2019**, *2019*, baz116. [CrossRef] [PubMed]
38. Ma, Y.; Peng, H.; Khan, T.; Cambria, E.; Hussain, A. Sentic LSTM: A hybrid network for targeted aspect-based sentiment analysis. *Cognit. Comput.* **2018**, *10*, 639–650. [CrossRef]
39. Gu, S.; Zhang, L.; Hou, Y.; Song, Y. A position-aware bidirectional attention network for aspect-level sentiment analysis. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 774–784.
40. Hashida, S.; Tamura, K.; Sakai, T. Classifying sightseeing tweets using convolutional neural networks with multi-channel distributed representation. In Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 7–10 October 2018; pp. 178–183.
41. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, LO, USA, 1–6 June 2018; Volume 1, pp. 2227–2237.
42. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the North American Chapter of the Association for Computational Linguistics, Minneapolis, Minnesota, 2–7 June 2019; pp. 4171–4186.
43. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018. Available online: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/languageunderstandingpaper.pdf (accessed on 15 September 2022).
44. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
45. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *arXiv* **2019**, arXiv:1909.11942.
46. Dhola, K.; Saradva, M. A comparative evaluation of traditional machine learning and deep learning classification techniques for sentiment analysis. In Proceedings of the 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 28–29 January 2021; pp. 932–936.
47. Younas, A.; Nasim, R.; Ali, S.; Wang, G.; Qi, F. Sentiment analysis of code-mixed Roman Urdu-English social media text using deep learning approaches. In Proceedings of the 2020 IEEE 23rd International Conference on Computational Science and Engineering (CSE), Guangzhou, China, 29 December 2020–1 January 2021; pp. 66–71.
48. Anbukkarasi, S.; Varadhaganapathy, S. Analyzing sentiment in Tamil tweets using deep neural network. In Proceedings of the 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 11–13 March 2020; pp. 449–453.
49. Youfa, L.; Du, B.; Ni, F. Adversarial strategy for transductive zero-shot learning. *Inform. Sci.* **2021**, *578*, 750–761. [CrossRef]
50. Brauwers, G.; Frasincar, F. A Survey on Aspect-Based Sentiment Classification. *ACM Comput. Surv.* **2022**, *55*, 37. [CrossRef]
51. Anish, M.; Ali, M. Investigating the Performance of Smote for Class Imbalanced Learning: A Case Study of Credit Scoring Datasets. *Eur. Sci. J.* **2017**, *13*, 340–353, November 2017 edition.

applied sciences

MDPI

Article

Sentiment Analysis of Text Reviews Using Lexicon-Enhanced Bert Embedding (LeBERT) Model with Convolutional Neural Network

James Mutinda [1,*], Waweru Mwangi [2] and George Okeyo [3]

[1] Department of Research and Consultancy, Kenya School of Government-Embu Campus, Embu P.O. Box 402-60100, Kenya
[2] Department of Computing, Jomo Kenyatta University of Agriculture & Technology, Nairobi P.O. Box 62000-00200, Kenya
[3] College of Engineering, Carnegie Mellon University Africa, Kigali Innovation City, Bumbogo, Kigali BP 6150, Rwanda
* Correspondence: james.mutinda@ksg.ac.ke

Abstract: Sentiment analysis has become an important area of research in natural language processing. This technique has a wide range of applications, such as comprehending user preferences in ecommerce feedback portals, politics, and in governance. However, accurate sentiment analysis requires robust text representation techniques that can convert words into precise vectors that represent the input text. There are two categories of text representation techniques: lexicon-based techniques and machine learning-based techniques. From research, both techniques have limitations. For instance, pre-trained word embeddings, such as Word2Vec, Glove, and bidirectional encoder representations from transformers (BERT), generate vectors by considering word distances, similarities, and occurrences ignoring other aspects such as word sentiment orientation. Aiming at such limitations, this paper presents a sentiment classification model (named LeBERT) combining sentiment lexicon, N-grams, BERT, and CNN. In the model, sentiment lexicon, N-grams, and BERT are used to vectorize words selected from a section of the input text. CNN is used as the deep neural network classifier for feature mapping and giving the output sentiment class. The proposed model is evaluated on three public datasets, namely, Amazon products' reviews, Imbd movies' reviews, and Yelp restaurants' reviews datasets. Accuracy, precision, and F-measure are used as the model performance metrics. The experimental results indicate that the proposed LeBERT model outperforms the existing state-of-the-art models, with a F-measure score of 88.73% in binary sentiment classification.

Keywords: natural language processing; word embeddings; BERT; sentiment analysis; convolutional neural network; sentiment lexicon

Citation: Mutinda, J.; Mwangi, W.; Okeyo, G. Sentiment Analysis of Text Reviews Using Lexicon-Enhanced Bert Embedding (LeBERT) Model with Convolutional Neural Network. *Appl. Sci.* **2023**, *13*, 1445. https://doi.org/10.3390/app13031445

Academic Editors: Xiangjie Kong, Wei Wang and Han Liu

Received: 24 December 2022
Revised: 12 January 2023
Accepted: 13 January 2023
Published: 21 January 2023

1. Introduction

Recently, social media platforms have created opportunities for businesses and organizations to obtain feedback from their customers and clients through reviews in the form of user-generated posts. Such posts are availed through social media and worldwide web in form of blogs, which contain data in text, audio, visual, or a combination of the three modes. Specifically, social media text data are characterized by short sentences, which are unstructured, semi-structured, and normally full of colloquial language, making it messy, difficult, and time consuming to build its vector representations and sentiment classification [1–4]. However, through sentiment analysis (SA), one of the big data analytics techniques, the text data can provide insightful business information [4]. Sentiment analysis is the process of classifying texts into predetermined opinion classes [3], which can be performed at document level, sentence level, or word level. Sentence level SA is a text classification task that assigns short texts (sentences) to predefined sentiment or opinion

classes. Sentiment analysis of social media data is a sentence level SA task since most posts are very short usually less than forty (40) words. Currently, few tools can perform sentiment analysis of social media text data effectively [4]. This is attributed to the nature of social media texts, which are unstructured, making it difficult to extract the right features at the text representation phase. According to Zhiying Jiang et al. [1], text representation is the second phase in sentiment analysis after text data preprocessing. In this phase, documents or sentences are converted into numeric vectors that represent the texts by use of vector space models (VSM).

Conversion of text to vector representation is the cornerstone of text classification models [5]. The accuracy and efficiency of sentiment analysis is dependent on whether or not the word vector is representative of the text [5–7]. From the literature, there are two widely used text-vector representation techniques: (1) natural language processing (NLP) techniques based on bag of words, part of speech (POS) tags, and sentiment lexicons [1,8–10]; (2) deep learning-based automated vector representation approaches such as word embeddings [11–13]. Word Embedding is one of the most useful deep learning methods used for constructing vector representations of words and documents in text classification tasks. This is because of their abilities to capture the syntactic and semantic relations among words [14]. Word embeddings models are based on deep learning Word2Vec [15], global vectors (Glove) [16], FastText [17], and bidirectional encoder representations from transformers (BERT) model [18]. Although these word embeddings methods are very effective compared to conventional NLP-based methods [19,20], they have some limitations and thus need improvement. For instance, effective training and vector representation of words and word embeddings require a very large corpus. Due to these limitations, researchers use pre-trained word embeddings for transfer learning, which may not correspond well with their data, especially small-sized datasets [21]. Further, the pre-trained word embeddings vectors do not consider the context of the word or other characteristics of the word, such as semantic orientation of the word. Existing NLP techniques, such as sentiment lexicon, POS tags, and word positions, can be used to improve performance of sentiment analysis models based on word embeddings [14].

In this paper, we propose a deep learning-based sentiment analysis model for user reviews, which combines sentiment lexicon, N-grams, and BERT word embeddings. In the model, we combine pre-trained word embeddings with sentiment lexicon to generate word representation for sentiment analysis. A text (review, sentence, or a document) is treated as a collection of word N-grams, and a sentiment lexicon is used to identify a section (N-grams) of the text where a sentiment may be found. BERT pre-trained word embeddings are then used to build vector representation of the text. In addition to solving the aforementioned limitations of word embeddings, our model reduces high feature dimensionality and computational costs brought by building word vectors from the entire text. We evaluate the proposed approach on the Yelp datasets in which the experimental results show that the model improves accuracy of pre-trained word embeddings. The main contribution of this paper, therefore, is to advance utilization of BERT pre-trained word embeddings model for sentiment analysis. We noted that BERT is one of the state-of-the-art models for building word vectors for NLP tasks, such as sentiment analysis. The novelty of the proposed model is the use of sentiment lexicon with N-grams to identify a section of input text, such as a review where sentiment is likely to be found. This approach proactively reduces feature dimensions of word vectors in the embedding layer of deep learning models such as CNN.

The rest of the paper is organized as follows. Section 2 presents related work. The proposed approach is described in Section 3. Section 4 describes the experimental procedures carried out. The result and discussion are presented in Section 5. Finally, Section 6 concludes the paper and recommends future work in this area of study.

2. Related Work

Sentiment analysis (SA) is a branch in NLP, which utilizes text mining and related technologies to classify subjective text into classes of opinions, emotions, or any other category. Vector representation of text is a very important task in sentiment analysis since it determines the accuracy and efficiency of the developed SA models [5]. Recently, there are many studies that have used lexicon-based techniques, pre-trained word embeddings, NLP techniques, and deep learning models in vector representation and generally in SA. In Section 2.1, current research in lexicon-based techniques, N-grams, and NLP is discussed, whereas in Section 2.2, research in pre-trained word embeddings and deep learning models is discussed.

2.1. Lexicon-Based Techniques, N-Grams and Natural Language Processing

The lexicon-based techniques use a dictionary of words labeled with their sentiment orientations. In such techniques, a piece of text is converted into a bag of words whose sentiment orientations are summarized or aggregated to classify the text. This technique is simple, but it is mostly dependent on manual labeling of the text [22]. Baharudin and Khan [23] suggested that sentence structure and contextual information are important for sentiment orientation and classification. In their work, each term in the sentence was assigned a sentiment score from the Sent WordNet lexicon. The overall classification of the sentence is the sum total score of the individual scores of the terms in the sentence. While the approach is interesting, one of the limitations of this approach is that words can be of the same orientation, but negating one another, thus giving the wrong sentiment classification. The main improvements of lexicon-based techniques involve using lexicon labeled words as input to machine learning classifiers. Mudinas et al. [24] combined lexicon-based approach and support vector machine. In their method, they generated word sentiment labels and used them as input to the SVM classifier. Seyed et al. [14] used several lexicons to assign lexicon vectors to words in a text, which they referred to as Lexicon2Vec (L2V). They combined their vector with Word2Vec and PoS2vec to obtain a hybrid vector representation. Generally, little research has been performed on combining lexicon-based methods and deep learning architectures. Huang et al. [25] proposed a sentiment analysis model of online reviews, which they referred to as polymerization topic sentiment model (PTSM). In their model, they used lexicon dictionary to extract sentiment information from online reviews. Although their model performed well with the support vector machine, they did not test their model with deep learning classifiers or word embedding algorithms. However, they recommended use of lexicon-based methods to solve the over-fitting problem of sentiment analysis models and to filter unnecessary information

Generation of word N-grams is another important NLP technique applicable in sentiment analysis. In text classification, word-grams are used to generate word co-occurrence patterns and vectors for machine learning classifiers. N-gram NLP models are widely used due to their simplicity and effectiveness [26]. However, they do not consider the information encapsulated in the sequence of the words. For instance, words could be negating one another in a sentence or having different meaning in different context. Kumar et al. [27], in their recent research on use of N-grams in text representation, used bi-grams and tri-grams to extract features from text data. Their work yielded promising results, which is an indication that N-grams can be utilized for effective text representation. They proposed a big data analytics framework for sentiment analysis and classification using intelligent cognitive inspired computing. In their model, they used fuzzy cognitive maps as classifiers. In our research, we advance this work by investigating use of hybrid NLP techniques, including N-grams and sentiment lexicon. They also recommended future research on deep learning architecture, an area which is also being explored in this research work. We do so by seeking to combine pre-trained word embeddings with sentiment lexicon and N-grams.

2.2. Word Embeddings-Based Techniques and Deep Learning Models

Recently, word embeddings-based vector representation techniques are playing an important role in natural language processing [28]. According to Mikolov [29], research in word embedding feature selection gained momentum in 2013. The main word embeddings algorithms are Word2Vec [15], Glove [16] and FastText [17,30], which are used to convert words to vectors. Recently, bidirectional encoder representations from the transformers (BERT) model [18] has received much attention due to its bidirectional and attention mechanisms. Consequently, use of BERT embedding-based models outperforms other models, thus showing remarkable performance in sentiment analysis tasks [31,32]. Word embeddings are better than the normal bag of words representation, since they cater for synonyms and produce vectors with lower dimensionality than the bag of words [14,15]. Garg [33] did research on word embeddings and established that Word2Vec embeddings performed better than the other word-embedding algorithms. Currently, most researchers use pre-trained word embeddings vectors as inputs of machine learning classifiers in their sentiment analysis research since they are more accurate and compatible with deep learning neural networks [22]. However, pre-trained word embeddings ignore sentiment orientation of words and their context, hence affecting sentiment classification accuracy [14,28]. This is because they use word distances and synonyms to calculate word vectors.

Kim [34] studied use of pre-trained Word2Vec vectors as inputs to convolutional neural networks and improved their performance by hyper parameter tuning of the CNN model. Wang et al. [35] used pre-trained Glove vectors as inputs for attention-based LSTM models for aspect-level sentiment analysis. Liu et al. [21] used pre-trained Word2Vec in idiom recommendation model in essay writing. Liu et al. [36] used pre trained Word2Vec model and improved them for cross-domain classification by extending the vector to include domain information. Recently D'Silva and Sharma [37] used FastText pre-trained word embeddings and neural networks to classify Konkani texts. Hu et al. [38] used BERT to integrate mental features and short text vector to improve topic classification and false detection in short text. Although their work showed better performance, they did not compare their proposal with other word embedding models. They also suggested more research to be performed on application of BERT in other contexts of text classification. Prottasha et al. [31] did a study to compare Word2Vec, Glove, FastText, and BERT. They demonstrated that transformer architectures, such as BERT models, are the state-of-the-art models for text representation and play a crucial role in sentiment analysis. The superiority of BERT is that it can read series of words in either direction, unlike other word embedding algorithms. Further, BERT employs the attention mechanism of the transformer that assigns a word its vector, depending on the surrounding words. This mechanism enhances the semantic representation of the target text. However, the series of input words to be read by the BERT algorithm maintains the entire words of the target text. We propose that the performance of BERT algorithm can be enhanced by focusing the input series to a few words, which contain sentiment information and their neighbours of the target text. This can be guided by utilization of sentiment lexicon and word N-grams. In a recent study [13], the researchers investigated a text representation technique using sentiment lexicon and N-grams where a Lexicon-pointed hybrid N-gram feature extraction model (LeNFEM) was proposed and investigated. A three-word N-gram was identified, which contains a sentiment word by use of a sentiment lexicon. The N-gram was then expanded to form a hybrid vector containing words, POS tags, and sentiments. Although this is a novel text representation technique, a proposal was put forth on investigation of how the approach could be applied with deep learning models, including word embeddings. In this paper, we extend on this work and present a text representation technique named lexicon selected-BERT embedding (LeBERT) Model. The model combines sentiment lexicon and BERT word embeddings via word N-grams for sentiment classification.

Based on the related work discussed, we observe that existing deep learning models for sentiment analysis generate text representation vectors using word embeddings. We also noted that the BERT model is one of the state-of-the-art embedding models. Thus,

any study on improving it advances sentiment analysis and natural language processing research. With this objective, this study suggested and investigated combination of BERT word embedding model, sentiment lexicon, and N-grams. The novelty of the proposed LeBERT model is that the sentiment lexicon is utilized to identify a section of a text (sentence or a document) where sentiment information is domiciled, and the BERT algorithm is used to build word vectors from that section only. In Section 3, we present and describe the details of the proposed model.

3. The Proposed LeBERT Model

In deep learning, the BERT model is one of the current word embeddings and text representation models under study for sentiment analysis. BERT, unlike other word embedding algorithms, can effectively read series of words in either direction of the input text, and since it uses the attention mechanism to assign a word, its vector depends on the surrounding words, and it is efficient in word vectorization [39]. Although BERT considers the context of a word when assigning the vector, it does so for all the words in the input text, which leads to a resultant vector with high dimensionality. Second, word vectors built from BERT do not contain semantic information, which is critical in sentiment classification. Compared with BERT, the sentiment lexicon can be used to identify sentiment words in a text and assign specific sentiment polarity to the words. However, sentiment lexicon cannot generate representative word vectors, hence leading to high data sparseness. Thus, to improve sentiment classification, this paper proposes the LeBERT model, which combines sentiment lexicon, N-grams, and BERT algorithms.

The design idea of the LeBERT model is to first use N-grams to split the input text into sections, and then use a sentiment lexicon to identify a section or sections that contain a sentiment word. It is worth noting that text reviews, such as social media posts, contain short text, and characteristically, semantic features in short texts are concentrated in a certain part [39]. Thus, extracting features from such parts will lead to efficient and effective text representation. The words of the identified section(s) are then converted into a vector by BERT. The output word vector is then used as the input into a CNN model with a fully connected layer where features from the vector are obtained. The features extracted are then integrated by the dense output layer, and finally the sentiment class of the text is performed by a SoftMax classifier. The architecture of the proposed LeBERT model is shown in Figure 1.

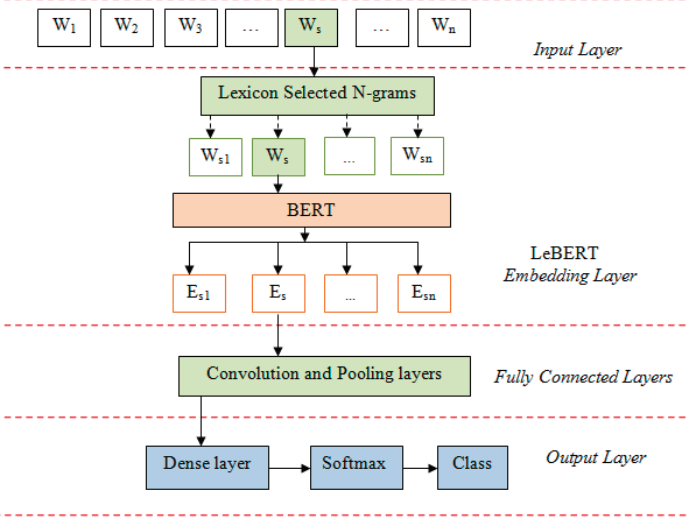


Figure 1. Architecture of the proposed sentiment analysis model.

As shown in Figure 1, the sentiment lexicon, N-grams, and BERT algorithm are used in the embedding layer to build the word vector. The overall sentiment analysis model using the LeBERT model is presented in Figure 2.

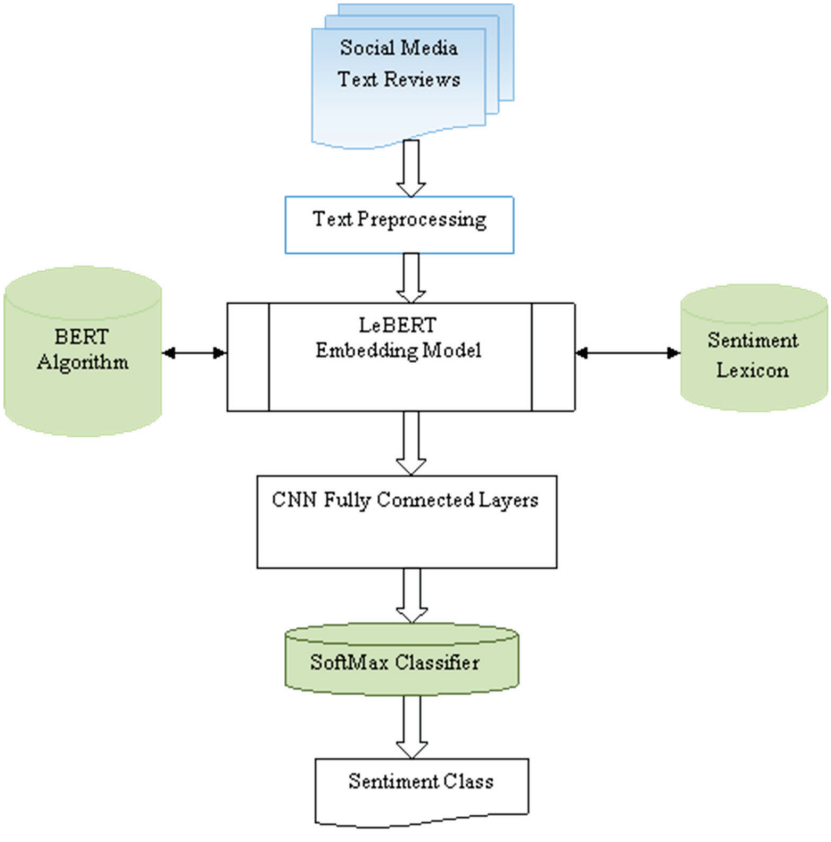


Figure 2. Sentiment analysis model using the LeBERT model.

3.1. LeBERT Embedding

There are currently two common methods used to build text vectors for sentiment analysis: word-embedding based methods or lexicon-based methods. In our proposed model, we sought to utilize both methods through N-grams. The sentiment lexicon is used to identify word N-grams containing a sentiment word, and then the vector from the N-gram words using BERT word embedding model is used.

To build the vector, we first generate word N-grams from the sentences. A N-gram is a combination of words from a sentence, which forms a Markovian process. Normally, this is used to predict the next word in a sequence of words. Further, Markovian process also generates co-occurrence of words, which is a key aspect in influencing sentiment in a text. In this case, we use N-gram sequences to partition a sentence into various sections that represent the entire text, such as an online review or a sentence. This is because N-grams present co-occurrence of words in a text in a more comprehensive manner than mere bag of words (BoW). The size of the partition depends on the value of N.

For instance, if we consider a sentence S given as:

$$S = \{w1, w2, w3, w4, w5, \ldots, \ldots, \ldots wn\} \quad (1)$$

where, w_i are words.

For various values of N, we have;

N = 1, the set of N-grams $N_1 = \{w_1, w_2, w_3, \ldots, w_n\}$

N = 2, the set of N-grams $N_2 = \{w_1\_w_2, w_2\_w_3, w_3\_w_4, \ldots, w_{n-1}\_w_n\}$

N = 3, the set of N-grams $N_3 = \{w_1\_w_2\_w_3, w_2\_w_3\_w_4, w_3\_w_4\_w_5, \ldots, w_{n-2}\_w_{n-1}\_w_n\}$

The fundamental idea is that, with the set of N-grams, it is possible to select a section of the entire input text. This ensures that we use the most significant words when building text vectors for sentiment analysis. Once the N-gram(s) are identified from the text, it is then reverted to a bag of words. Each word is then converted into a vector using the BERT word-embedding algorithm.

3.2. The LeBERT Embedding Algorithm

Let L: sentiment lexicon; C: corpus of subjective user reviews (R_i); $\boldsymbol{V_i}$: vector representation of a subjective review (R_i); W_t: sentiment term; W_1: the first word neighboring the sentiment term; and W_2: the second word neighboring the sentiment term.

We define the text vector, $\boldsymbol{v_i}$, of a subjective review, R_i, as the vector originating from a selected section of the review S_i using sentiment lexicon and BERT word embedding model (***Be***). The algorithm listing of the sentence vector representation generation is presented in Algorithm 1.

Algorithm 1 Contextualized Text Vector Generation

Inputs:
$R_i = \{w_1, w_2, \ldots \ldots, w_n)$, input review containing n words
L = sentiment lexicon
Be = BERT word-embedding model
Output: Contextualized Text Vector ($\boldsymbol{v_i}$), representing the subjective user review
START
Set the N-gram value to N = 3
FOR each review ($R_i \epsilon C$) with n word tokens
PRINT the word trigrams;
Call the sentiment lexicon (L)
FOR each trigram check for a sentiment word;
IF a trigram contains a sentiment word **THEN**
PRINT the trigram words (w_1, w_t, w_2)
ELSE delete the trigram
ENDIF
END
Generate section vector(·)
FOR Each word (w_1, w_t, and w_2) in the trigram
READ (w_i) into gag of words ($\boldsymbol{B_{wi}}$)
Call the pre-trained word-embedding (***Be***)
Calculate the word vector ($\boldsymbol{w}_{vi}$)
END
Update vector $\boldsymbol{V_i}$:‹$\boldsymbol{w_{v1}}$ **and** $\boldsymbol{w_{vt}}$ **and** $\boldsymbol{w_{v2}}$›
END
Return Vector $\boldsymbol{V_i}$.

3.3. The CNN Layer

The CNN deep learning model is used as the classifier, which uses the resultant vector from LeBERT embedding as input and gives the sentiment class as the output. CNNs are specialized types of artificial neural networks, which are capable of outperforming the common machine learning algorithms in supervised learning tasks. CNNs' main function is to identify and learn the information characteristic patterns through the use of convolution layers and thus facilitate classification of the objects. The CNN model is presented in Figure 3. Using the convolution kernels (windows) and the nonlinear function (filter), feature maps are obtained. A pooling operation is then applied on the feature maps to

select the optimal features. The dense output layer then classifies the optimal features using softmax activation function (which uses probability) into a positive or a negative class.

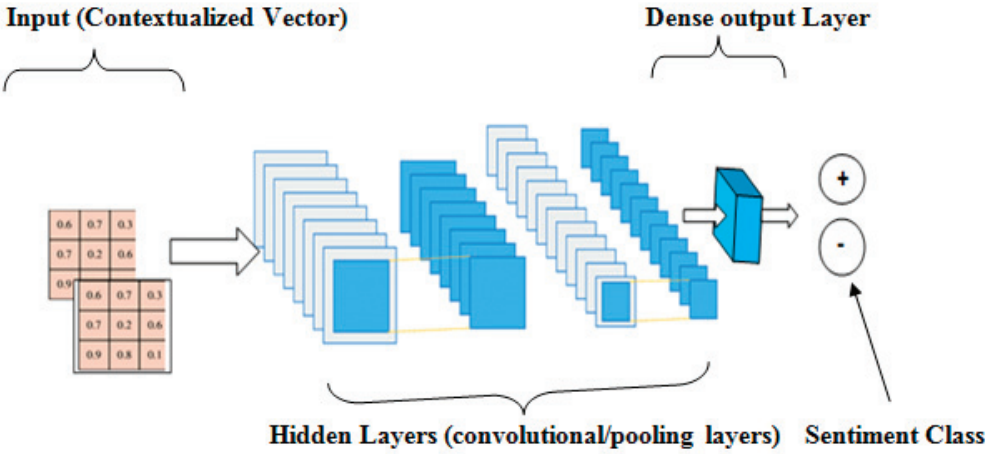


Figure 3. The CNN model.

4. Experiments

This section describes the dataset used; the experiments set up was carried out to evaluate the performance of the proposed model. The tools and techniques used in model formulation and evaluation are also discussed.

4.1. Dataset

In order to evaluate the effectiveness of the proposed model, the experiments were carried using a dataset complied from three public datasets. The dataset contains three world datasets including: Amazon products' reviews dataset, with 70,000 reviews, Imbd dataset, with 50,000 movie reviews, and Yelp dataset, with 300,000 restaurants' reviews. In the experiments, we used 3000 reviews, as compiled by Kotzias et al. [40] and published in a machine learning repository. For each website, Kotzias et al. [40] randomly sampled 500 positive and 500 negative tweets, which were clearly positive and negative.

4.2. Experiment Setup

The reviews presented in the dataset were cleaned of non-English words and pre-processed. Tokenization, N-grams generation, text vector building, and designing of the CNN layers was conducted using python programming language in the virtual laboratory (Google Colaboratory). The obtained vector was split into two subsets, 80% of the dataset was used for training the CNN model, and the other 20% was used for evaluating the classification performance. Since the dataset contained multiple sentences (reviews), pooled output was used in the BERT embedding. The rectified linear unit (RELU) was used as the activation function, with 100 neurons for the hidden fully connected layer. The output dense layer was set up with two (2) neurons since the texts were to be classified into two classes. Softmax was used as the activation function, which was in line with the text classification problem at hand. In the study, we used 50-dimensional Glove word embeddings trained on Google News, 250-dimensional Word2Vec embeddings trained on Wikipedia, and 128-dimensional BERT embeddings trained on English Wikipedia corpus. In the experiments, we used tensor flow tools to prepare the data and build our proposed model. Among the training set, a small potion (100) of the reviews was used for validation. In Section 4.3, we present the model parameters of the designed CNN model.

4.3. Model Parameters BERT, Glove, and Word2Vec Pre-Trained Word Embeddings

The model parameters for the BERT word embeddings were as shown in Table 1.

Table 1. Model Parameters for BERT word embedding.

Layer (Type)	Output Shape	Parameters
Keras Layer	(None, 128)	4,385,921
Dense Hidden layers	(None, 16)	2064
Dense Output Layer	(None, 1)	17
Total Parameters 4,388,002 Trainable Parameters: 4,388,001 Non-trainable Parameters: 1		

From Table 1, the Keras layer represents the shape of embedding and the preprocessor used for the BERT model. In the experiment, the BERT word embeddings were initialized using small BERT due to limitations of computation resources. Consequently, the dimension of the word embedding was set to 128 and appropriate preprocessor for the BERT was set. Glove and Word2Vec word embeddings of 50 and 250 dimensions, respectively, were used as baseline models, and their parameters were set as shown in Tables 2 and 3.

Table 2. Model Parameters for 50-diemsional Glove word embeddings.

Layer (Type)	Output Shape	Parameters
Keras Layer	(None, 50)	48,190,600
Dense Hidden layers	(None, 16)	816
Dense Output Layer	(None, 1)	17
Total Parameters 48,191,433 Trainable Parameters: 48,191,433 Non-trainable Parameters: 0		

Table 3. Model Parameters for 250-dimensional Word2Vec word embeddings.

Layer (Type)	Output Shape	Parameters
Keras Layer	(None, 250)	252,343,750
Dense Hidden layers	(None, 16)	4016
Dense Output Layer	(None, 1)	17
Total Parameters 48,191,433 Trainable Parameters: 48,191,433 Non-trainable Parameters: 0		

From Tables 2 and 3, The Keras layer represents the input layer in which the input vector was obtained using the Glove and Word2Vec word embeddings. The shape of the Keras layer was determined by the dimensions of the word embeddings. The dense output layer is for binary classification of the input text into positive or negative sentiment.

4.4. Model Performance Evaluation

To verify the effectiveness of the proposed model, a 2 by 2 contingency matrix that shows the number of correctly predicted positive reviews (TP), true negative reviews (TN), false positive reviews (FP), and false negative reviews [41] was used, as shown in Table 4.

Table 4. Contingency Table.

	Classified as Positive	Classified as Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Four model evaluation metrics were selected: accuracy, precision, recall, and F-measure. From Table 4, we calculated the metrics, as discussed and presented in Equations (2)–(5).

Accuracy is the ratio of the correctly classified predictions to the total sum of predictions. It is given as;

$$Accuracy = \frac{TP + TN}{(TP + FN + FP + TN)} \tag{2}$$

Precision is the ratio of accurately classified data to the total data classified in the class. It is given as;

$$Precision = \frac{TP}{(TP + FP)} \tag{3}$$

Recall is the ratio of accurately classified data to the actual data in the class. It is given as;

$$Recall = \frac{TP}{(TP + FN)} \tag{4}$$

F-measure is the mean of precision and recall. It is given as;

$$\mathrm{F-measure} = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{(\mathrm{Precision} + \mathrm{Recall})} \tag{5}$$

5. Results and Discussion

This section describes the results obtained from the experiments. We first sought to test the effect of using sentiment lexicon on the input text data and the vector. We compared the shape of Yelp dataset (restaurants reviews) before and after using the sentiment lexicon. Table 5 presents the details of the text data.

Table 5. Details of the text data before and after using sentiment lexicon.

Text Data Item	Before Using the Sentiment Lexicon	After Using the Sentiment Lexicon
Characters(no spaces)	46,744	14,182
Characters(with spaces)	56,616	19,212
Number of words	10,863	2989
Number of paragraphs	996	996
Average Number of words per Post/paragraph	11	3

From Table 5, it was evident that application of sentiment lexicon to select a section of the input text significantly reduced the size of input text. Although the number of posts or paragraphs remained the same, the shape of the input text changed from 11 rows to 3 rows, which, in turn, would reduce the computation time for the model. We then designed and performed experiments with deep learning CNN to evaluate how the LeBERT embedding model would perform in sentiment analysis.

5.1. Ablation Study on Effect of Size of N-Grams on LeBERT Model

In order to verify the effectiveness of using LeBERT model as the embedding layer to generate word vectors, we first did an experiment to study the effect of the size of N-grams on the LeBERT model with CNN. In the experiment, the restaurant reviews datasets were used. The experimental results of N = 1,2, 3 and all words were as shown in Table 6.

Table 6. Sentiment classification results of various sizes of N-grams with the LeBERT model.

N-Grams	LeBERT-CNN			
	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)
N = 1	65.02	65.02	65.15	65.08
N = 2	79.45	79.50	80.04	79.77
N = 3	88.20	88.45	89.01	**88.73**
N = 4	87.65	87.65	87.80	87.72
All words	84.00	84.00	84.20	84.10

For N = 1, it implies that, for each sentence, only one word was used, which was chosen by the sentiment lexicon. The results indicate a low performance since one word cannot represent the sentiment of the entire text. The highest model performance was obtained at N = 3. As shown in Table 6, we generated N-grams up to N = 4 due to computational resources. The category of 'All words' implies that the sentiment lexicon was not applied on the input text to select some words, hence, this reverts to the original BERT model. The results indicated that N = 3 is an ideal size of N-gram for the proposed model. Section 5.2 presents the performance results of the model in comparison to the baseline models in the three datasets.

5.2. Comparison of LeBERT Model Performance with Baseline Models

The experiment was carried out to validate the performance of the proposed LeBERT model in terms of accuracy, recall, precision, and F-measure of the CNN on the three discussed datasets. Glove and Word2Vec were used as baseline word embedding models. In this experiment, tri-grams (N = 3) were used. Tables 7–9 show the performance results on restaurants reviews, movie reviews, and product reviews datasets, respectively.

Table 7. Sentiment classification prediction under Yelp dataset (restaurant reviews).

Embedding Model	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)
Glove	78.50	78.56	78.70	78.63
Le-Glove	81.50	82.00	83.01	82.50
Word2Vec	75.50	75.50	75.80	75.65
Le-Word2Vec	82.40	82.45	83.15	82.80
BERT	84.00	84.00	84.20	84.10
LeBERT(our Model)	88.20	88.45	89.01	**88.73**

Table 8. Sentiment classification prediction under IMDB dataset (movie reviews).

Embedding Model	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)
Glove	79.50	79.50	80.10	79.80
Le-Glove	82.50	82.70	83.25	82.97
Word2Vec	77.45	77.46	78.01	77.73
LeWord2Vec	83.00	83.02	83.42	83.22
BERT	84.01	84.08	84.63	84.35
LeBERT (our Model)	86.10	86.71	87.00	**86.85**

The presented tables indicate the comparative results between the pre-trained word embeddings, with and without the proposed fusion with sentiment lexicon. Generally, the proposed LeBERT model performs better compared to the baseline word embeddings models. Accuracy is considered to be a good performance evaluation metric when the classes are balanced [41]. Since, in our experiments all the three datasets exhibited balanced classes, we compared accuracy of the model with the various approaches for the three datasets. The results obtained were as shown in Figure 4.

Table 9. Sentiment classification prediction under Amazon dataset (products reviews).

Embedding Model	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)
Glove	79.00	79.00	79.65	79.32
Le-Glove	79.60	80.00	80.45	80.22
Word2Vec	79.50	79.50	80.25	79.87
Le-Word2Vec	81.50	81.50	82.05	81.77
BERT	81.72	81.75	82.04	81.89
LeBERT	82.40	82.40	82.64	**82.52**

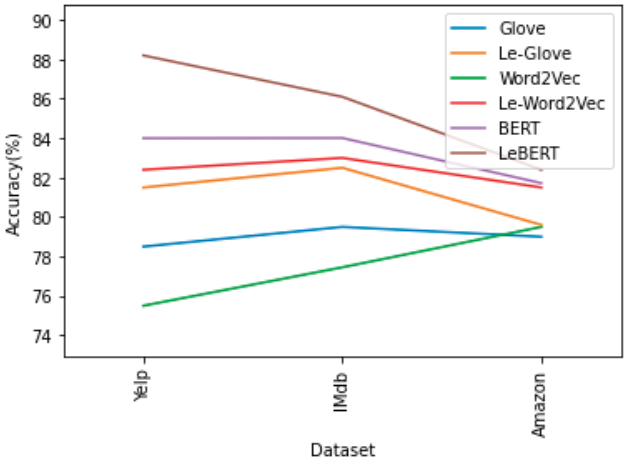


Figure 4. Sentiment prediction accuracy using various embedding models.

From Figure 4, our proposed model (LeBERT) had the highest accuracy in all datasets, with relatively lower accuracy on Amazon's product reviews dataset. This could be attributed to the fact that the reviews referred to various products, and thus the sentiment terms varied from one product to another.

6. Conclusions

Sentiment analysis of social media reviews is a difficult task due to sparsity and high dimensionality of word vectors representing the text. Use of sentiment lexicon and word embedding algorithms can improve sentiment analysis models for text reviews. In this context, we proposed a sentiment analysis model, named LeBERT, based on sentiment lexicon, N-grams, BERT word embedding, and CNN. In the model, a section of a document or a sentence where sentiment information can be highly found is selected using sentiment lexicon and word N-grams, and then the words are vectorized using the BERT word embedding algorithm. A CNN classifier is then used to classify the input vector into a sentiment class. To validate the performance of the proposed LeBERT model, original Word2Vec, Glove, and BERT word embeddings were used as baseline models on three benchmark sentiment datasets. From the experimental results, use of sentiment lexicon significantly reduces the dimension of the input vector, thus improving efficiency of sentiment analysis models. Secondly, integration of sentiment lexicon and N-grams with BERT embedding algorithm yields a better representative word vector, hence increasing the predictive performance of the resultant sentiment analysis model. The results also indicated that sentiment lexicon with BERT (through LeBERT model) outperformed other word embedding algorithms.

This paper had some limitations. The designed model utilized convolutional neural network (CNN) only. In the future, the LeBERT embedding model could be implemented and evaluated in other neural networks, such as long short-term memory (LSTM). Our

proposed model was tested and found to be effective in binary sentiment classification, where sentiment lexicon was used. It would be interesting to evaluate the model on other text classification tasks where other types of lexicons are used.

Author Contributions: Conceptualization, J.M., W.M. and G.O.; Data curation, J.M.; Formal analysis, G.O. and W.M.; Investigation, J.M., G.O. and W.M.; Methodology, J.M., G.O. and W.M.; Supervision, G.O. and W.M.; Writing—original draft, J.M.; Writing—review and editing, G.O. and W.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used in the experiments is publicly available from https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences accessed on 6 October 2022. The dataset was compiled by Kotzias et al. [40], which is cited in this research work.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Jiang, Z.; Gao, B.; He, Y.; Han, Y.; Doyle, P.; Zhu, Q. Text Classification Using Novel Term Weighting Scheme-Based Improved TF-IDF for Internet Media Reports. *Math. Probl. Eng.* **2021**, *2021*, 6619088. [CrossRef]
2. Onan, A.; Üniversitesi, I.K. Ensemble of Classifiers and Term Weighting Schemes for Sentiment Analysis in Turkish. *Sci. Res. Commun.* **2021**. [CrossRef]
3. Kalarani, P.; Selva, B.S. An overview on research challenges in opinion mining and sentiment analysis. *Int. J. Innov. Res. Comput. Commun. Eng.* **2015**, *3*, 1–6.
4. Yang, J.; Xiu, P.; Sun, L.; Ying, L.; Muthu, B. Social media data analytics for business decision making system to competitive analysis. *Inf. Process. Manag.* **2021**, *59*, 102751. [CrossRef]
5. Rao, L. Sentiment Analysis of English Text with Multilevel Features. *Sci. Program.* **2022**. [CrossRef]
6. Onan, A.; Korukoğlu, S. A feature selection model based on genetic rank aggregation for text sentiment classification. *J. Inf. Sci.* **2016**, *43*, 25–38. [CrossRef]
7. Bhadane, C.; Dalal, H.; Doshi, H. Sentiment Analysis: Measuring Opinions. *Procedia Comput. Sci.* **2015**, *45*, 808–814. [CrossRef]
8. Mozetič, I.; Grčar, M.; Smailović, J. Multilingual Twitter Sentiment Classification: The Role of Human Annotators. *PLoS ONE* **2016**, *11*, e0155036. [CrossRef]
9. Li, B.; Guoyong, Y. Improvement of TF-IDF Algorithm based on Hadoop Framework. In Proceedings of the 2nd International Conference on Computer Application and System Modeling, Taiyuan, China, 27–29 July 2012; pp. 391–393.
10. Ankit, N.S. An Ensemble Classification System for Twitter Sentiment Analysis. *Procedia Comput. Sci.* **2018**, *132*, 937–946. [CrossRef]
11. Ahuja, R.; Chug, A.; Kohli, S.; Gupta, S.; Ahuja, P. The Impact of Features Extraction on the Sentiment Analysis. *Procedia Comput. Sci.* **2019**, *152*, 341–348. [CrossRef]
12. Rao, G.; Huang, Z.F.; Cong, Q. LSTM with sentence representations for document level sentiment classification. *Neurocomputing* **2018**, *308*, 49–57. [CrossRef]
13. Mutinda, J.; Mwangi, W.; Okeyo, G. Lexicon-pointed hybrid N-gram Features Extraction Model (LeNFEM) for sentence level sentiment analysis. *Eng. Rep.* **2021**, *3*, e12374. [CrossRef]
14. Rezaeinia, S.M.; Rahmani, R.; Ghodsi, A.; Veisi, H. Sentiment analysis based on improved pre-trained word embeddings. *Expert Syst. Appl.* **2019**, *117*, 139–147. [CrossRef]
15. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781. [CrossRef]
16. Pennington, J.; Socher, R.; Manning, C. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 26–28 October 2014; pp. 1532–1543.
17. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
18. Kenton, J.D.M.W.C.; Toutanova, L.K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the NaacL-HLT, Minneapolis, Minnesota, 2 June 2019; pp. 4171–4186.
19. Sharma, A.K.; Chaurasiaa, S.; Srivastavaa, D.K. Sentimental Short Sentences Classification by Using CNN Deep Learning Model with Fine Tuned Word2Vec. *Procedia Comput. Sci.* **2020**, *167*, 1139–1147. [CrossRef]
20. Dashtipour, K.; Gogate, M.; Adeel, A.; Larijani, H.; Hussain, A. Sentiment Analysis of Persian Movie Reviews Using Deep Learning. *Entropy* **2021**, *23*, 596. [CrossRef]

21. Liu, Y.; Liu, B.; Shan, L.; Wang, X. Modelling context with neural networks for recommending idioms in essay writing. *Neurocomputing* **2018**, *275*, 2287–2293. [CrossRef]
22. Giatsoglou, M.; Vozalis, M.G.; Diamantaras, K.; Vakali, A.; Sarigiannidis, G.; Chatzisavvas, K.C. Sentiment analysis leveraging emotions and word embeddings. *Expert Syst. Appl.* **2017**, *69*, 214–224. [CrossRef]
23. Baharudin, B.; Khan, A. Sentiment Classification Using Sentence-level Semantic Orientation of Opinion Terms from Blogs. In Proceedings of the 2011 National Postgraduate Conference, Perak, Malaysia, 19–20 September 2011; IEEE: Piscataway, NJ, USA, 2011. [CrossRef]
24. Mudinas, A.; Zhang, D.; Levene, M. Combining lexicon and learning based approaches for concept-level sentiment analysis. In Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining, Beijing, China, 12 August 2012; pp. 1–8.
25. Huang, L.; Dou, Z.; Hu, Y.; Huang, R. Textual Analysis for Online Reviews: A Polymerization Topic Sentiment Model. *IEEE Access* **2019**, *7*, 91940–91945. [CrossRef]
26. Fotis, A.; Dimitrios, T.; John, V.; Theodora, V. Using N-Gram Graphs for Sentiment Analysis: An Extended Study on Twitter. In Proceedings of the 2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService), Oxford, UK, 29 March–1 April 2016; pp. 44–51.
27. Jain, D.K.; Boyapati, P.; Venkatesh, J.; Prakash, M. An Intelligent Cognitive-Inspired Computing with Big Data Analytics Framework for Sentiment Analysis and Classification. *Inf. Process. Manag.* **2022**, *59*, 102758. [CrossRef]
28. Araque, O.; Corcuera-Platas, I.; Sánchez-Rada, J.; Iglesias, C. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Syst. Appl.* **2017**, *77*, 236–246. [CrossRef]
29. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, CA, USA, 5–10 December 2013; pp. 3111–3119.
30. Chandrasekaran, G.; Nguyen, T.N.; Hemanth, J.H. Multimodal sentimental analysis for social media applications: A comprehensive review. WIREs Data Min. *Knowl. Discov.* **2021**, *11*, e1415.
31. Prottasha, N.J.; Sami, A.A.; Kowsher; Murad, S.A.; Bairagi, A.K.; Masud, M.; Baz, M. Transfer Learning for Sentiment Analysis Using BERT Based Supervised Fine-Tuning. *Sensors* **2022**, *22*, 4157. [CrossRef] [PubMed]
32. Jain, P.K.; Quamer, W.; Saravanan, V.; Pamula, R. Employing BERT-DCNN with sentic knowledge base for social media sentiment analysis. *J. Ambient. Intell. Humaniz. Comput.* **2022**, 1–13. [CrossRef]
33. Garg, S.B.; Subrahmanyam, V.V. Sentiment Analysis: Choosing the Right Word Embedding for Deep Learning Model. In *Advanced Computing and Intelligent Technologies*; Lecture Notes in Networks and Systems; Bianchini, M., Piuri, V., Das, S., Shaw, R.N., Eds.; Springer: Singapore, 2022; Volume 218. [CrossRef]
34. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 26–28 October 2014; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1746–1751.
35. Wang, Y.; Huang, M.; Zhu, X.; Zhao, L. Attention-based LSTM for Aspect-level Sentiment Classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, 1–5 November 2016; pp. 606–615.
36. Liu, J.; Zheng, S.; Xu, G.; Lin, M. Cross-domain sentiment aware word embeddings for review sentiment analysis. *Int. J. Mach. Learn. Cybern.* **2021**, *12*, 343–354. [CrossRef]
37. D'Silva, J.; Sharma, U. Automatic text summarization of konkani texts using pre-trained word embeddings and deep learning. *Int. J. Electr. Comput. Eng. (IJECE)* **2022**, *12*, 1990–2000. [CrossRef]
38. Hu, Y.; Ding, J.; Dou, Z.; Chang, H. Short-Text Classification Detector: A Bert-Based Mental Approach. *Comput. Intell. Neurosci.* **2022**. [CrossRef]
39. Yang, H. Network Public Opinion Risk Prediction and Judgment Based on Deep Learning: A Model of Text Sentiment Analysis. *Comput. Intell. Neurosci.* **2022**, *2022*. [CrossRef]
40. Kotzias, D.; Denil, M.; de Freitas, N.; Smyth, P. From Group to Individual Labels Using Deep Features. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 15 August 2015; Association for Computing Machinery: New York, NY, USA; pp. 597–606. [CrossRef]
41. Singh, K.N.; Devi, S.D.; Devi, H.M.; Mahanta, A.K. A novel approach for dimension reduction using word embedding: An enhanced text classification approach. *Int. J. Inf. Manag. Data Insights* **2022**, *2*, 100061. [CrossRef]

applied sciences

MDPI

Article

Fine-Grained Sentiment-Controlled Text Generation Approach Based on Pre-Trained Language Model

Linan Zhu, Yifei Xu, Zhechao Zhu, Yinwei Bao and Xiangjie Kong *

College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China
* Correspondence: xjkong@ieee.org

Abstract: Sentiment-controlled text generation aims to generate texts according to the given sentiment. However, most of the existing studies focus only on the document- or sentence-level sentiment control, leaving a gap for finer-grained control over the content of generated results. Fine-grained control allows a generated review to express different opinions toward multiple aspects. Some previous works attempted to generate reviews conditioned on aspect-level sentiments, but they usually suffer from low adaptability and the lack of an annotated dataset. To alleviate these problems, we propose a novel pre-trained extended generative model that can dynamically refer to the prompt sentiment, together with an auxiliary classifier that extracts the fine-grained sentiments from the unannotated sentences, thus we conducted training on both annotated and unannotated datasets. We also propose a query-hint mechanism to further guide the generation process toward the aspect-level sentiments at every time step. Experimental results from real-world datasets demonstrated that our model has excellent adaptability in generating aspect-level sentiment-controllable review texts with high sentiment coverage and stable quality since, on both datasets, our model steadily outperforms other baseline models in the metrics of BLEU-4, METETOR, and ROUGE-L etc. The limitation of this work is that we only focus on fine-grained sentiments that are explicitly expressed. Moreover, the implicitly expressed fine-grained sentiment-controllable text generation will be an important puzzle for future work.

Keywords: artificial intelligence; natural language processing; controllable text generation; review generation; pre-trained language model; fine-grained sentiment

Citation: Zhu, L.; Xu, Y.; Zhu, Z.; Bao, Y.; Kong, X. Fine-Grained Sentiment-Controlled Text Generation Approach Based on Pre-Trained Language Model. *Appl. Sci.* **2023**, *13*, 264. https://doi.org/10.3390/app13010264

Academic Editor: Antonio Moreno

Received: 15 November 2022
Revised: 11 December 2022
Accepted: 20 December 2022
Published: 26 December 2022

1. Introduction

In recent years, Transformer-based pre-trained language models (LMs) have greatly improved the state-of-the-art of natural language processing tasks as well as natural language generation (NLG). Large-scale autoregressive Transformer models [1] that leverage large amounts of unannotated data and a simple log-likelihood training objective have achieved remarkable results in many text-generation tasks, such as machine translation, text summarization, and text style transfer. Meanwhile, for other real-world text-generation applications, such as review generation and essay writing, users prefer the generated text to be more controllable. However, since the LMs are trained on unannotated data, controlling attributes of generated text becomes difficult without modifying the model architecture to allow for extra input attributes or fine-tuning with attribute-specific data [2,3]. Therefore, some approaches, such as Plug-and-Play-Language-Models (PPLM) [4], control generated text through attribute models without changing the architecture or weights of pre-trained LMs. These models usually regard controllable text generation as generating tasks conditioned on the attributes, such as topic and sentiment at the sentence- or document-level, leaving a gap for finer-grained (e.g., aspect-level) control over the content of generated texts.

The fine-grained sentiment-conditioned text-generation task aims to automatically generate a highly relevant statement when given a series of fine-grained sentiments (e.g.,

aspect-opinion, aspect-sentiment) as input. Zang and Wan [5] first introduced the aspect-sentiment information to perform aspect-level sentiment-controllable review generation. They conducted conditional training by adopting a supervised method requiring a large dataset annotated with sentence-level aspect-sentiment labels. However, very few datasets provide such sufficient fine-grained labels, and it is also labor-intensive and time-consuming to conduct annotation on all data instances. Chen et al. [6] proposed a mutual learning framework leveraging large unlabeled data through interactive learning between the generator and the classifier. Besides the aspect-sentiment, aspect-opinion pairs also express aspect-level sentiment information. Therefore, inspired by them, in this work, we introduce the aspect-opinion information into the fine-grained sentiment-controllable text generation.

The aspect-opinion pairs represent the fine-grained sentiments that could be expressed within a review sentence, where the aspect term refers to the target of an opinion, and the opinion term refers to the sentimental words that describe the aspect term. For example, in the sentence of Figure 1, ("hotdog", "better") is an aspect-opinion pair, where "hotdog" is an aspect term, and "better" is an opinion term, together they form the backbone of fine-grained sentiment in the review text. Therefore, the aspect-opinion conditioned generation task aims to generate a review text X that correctly contains the sentiment information from n non-repeated aspect-opinion pairs $(\mathrm{a}, \mathrm{o})_{1:n}$. Most previous works [5,7,8] used the aspect-polarity pairs rather than the aspect-opinion pairs, and they used a straightforward data-to-text modeling approach, which is much more difficult due to the discrete and sparsity of the input data. To tackle this problem, relying on the natural characteristics of aspect-opinion pairs directly presented in sentences, our approach proposed a query-hint mechanism as a dynamic prompt strategy to guide the generation direction. Furthermore, in order to guarantee the quality of the generated results, in the generator, we incorporate a GPT-2 345M model [9] as the "super generator," then by extending this state-of-the-art model with our proposed query-hint mechanism and our sentiment control loss function to guide the generating process toward the given controlling information. Moreover, to further enhance the generator's performance, with the assistance of a classifier by extracting the fine-grained sentiments, we leveraged a large unlabeled dataset to train the generator. The experimental results demonstrate the effectiveness of these components.

Sentence:
Their hotdog is better compared with tasteless bread.
Aspect-Opinion Pairs:
{(hotdog, better), (bread, tasteless)}

Figure 1. An illustrative example of how the aspect-opinion pairs are expressed in a review sentence. The terms highlighted in red and blue are aspect terms and opinion terms, respectively.

Our Contributions:

- We propose our conditional generative model by extending a pre-trained state-of-the-art Transformer-based generative model with our introduced query-hint mechanism and sentiment control loss function to further guide the text generation at a finer-grained level.
- To better model a text-to-text schema, we introduce the aspect-opinion pair as the fine-grained sentiment unit to control the constrained text generation.
- Through employing an auxiliary classifier, we leverage a large unannotated dataset to re-train and fine-tune an end-to-end conditioned text generative model.

The remainder of this paper is organized as follows. Section 2 discusses the related works in controlled text generation, including the review generation and the aspect-level sentiment-controlled generation, which is less studied. Section 3 introduces our proposed approach that achieved finer-grained sentiment control in generation. In Section 4, the experimental settings are detailed, and evaluation metrics and results are also discussed to

demonstrate the validity of our approach. Finally, we conclude this work in Section 5 while discussing future work.

2. Related Work

2.1. Controlled Text Generation

Recently, there has been many studies that aim to generate text conditioned on input attributes with neural networks. Some of the earlier efforts have studied this controlled text generation by training a conditional generative model [10,11] while fine-tuning pre-trained models with Reinforcement Learning (RL) [3] and training a Generative Adversarial Network [12] have also shown inspiring results. The Conditional-Transformer-Language (CTRL) model [2] is a recent approach that trains a language model conditioned on a variety of control codes (e.g., "Reviews" and "Legal" control the model to generate reviews and legal texts, respectively), which prepended meta-data to the text during generation. Although it uses a GPT-2-like architecture to generate high-quality text, the result is at the cost of fixing the control codes and training a very large model. PPLM [4] composed a pre-trained LM with attribute controllers guiding text generation toward the desired attribute. At the same time, its flexible design allows it to control the generating process through relatively small "pluggable" attribute models while keeping parameters in the LM fixed. Chan et al. [13] incorporated a pre-trained GPT-2 model with a Content-Conditioner (CoCon) to control the generated text under the guidance of target text content. Yu et al. [14] proposed a simple and flexible method, infusing attribute representations into a pre-trained unconditional LM without changing the LM parameters to achieve sentiment- and topic-controlled generation. Different from our fine-grained sentiment-controlled text-generation (FSCTG) task, these works focus on sentence-based sentiment and topic control in text generation. In the FSCTG task, the text-generation process is controlled by a series of fine-grained sentiments (e.g., aspect-opinion or aspect-sentiment).

2.2. Review Generation

Review generation [7,15], a generation task aiming to automatically generate review text, is a related area that generates reviews conditioned on the given information. While most of the previous approaches [7,8] have framed review generation as A2T (Attribute-to-Text problem), leaving a gap between attributes (e.g., user, product, and rating) and linguistic data. To tackle this problem, Kim et al. [16] proposed AT2T (Attribute-matched-Text-to-Text) by augmenting inductive biases of attributes with matching reference reviews to learn the rich representations of attributes.

2.3. Aspect-Level Sentiment Control

Nevertheless, most of these works only focus on sentence-level sentiments and ignore the aspect-level sentiment control, and very few researchers have studied generating reviews from fine-grained sentiments due to the lack of announced data. Zang and Wan [5] gave the first attempt to generate reviews from aspect-sentiment scores, which requires the reviews with sentence-level aspect-sentiment score annotations. This makes it impractical in real-world applications due to the lack of labeled data. To tackle this problem, Chen et al. [6] proposed a semi-supervised aspect-level sentiment-controllable review generation method, under their proposed mutual learning framework with the assistance of a classifier, it can take advantage of large-scale unlabeled data to achieve aspect-level sentiment control in review generation with few labeled data. Fei et al. [17] combined fine-grained sentiment classification and generation tasks as a joint dual learning system, strengthening the mutual connection of both tasks. To overcome the defect of sparsity and discrete nature brought by the input data in the data-to-text scheme, Yuan et al. [18] proposed a hierarchical template-transformer (HTT); they split the generation task into two corresponding pipeline subtasks, i.e., opinion phrase generation and review composition, which were jointly trained on the HTT. Although in different ways, they all trained an efficient end-to-end generative model. However, they did not attempt to dynamically adjust the attention weights during the

model's generation process since some contents (e.g., the completion of sentiment words generation) are informative to the global generation and need to be notified.

3. Method

In this section, we introduce our fine-grained sentiment-controllable text-generation task together with a conditional generative model named **A**spect-**l**evel **Se**ntiment **Cond**itioner (**AlSeCond**), which was trained with both labeled and unlabeled data to learn a fine-grained sentiment review generator with the assistance of a classifier.

First, we give the formalization of our fine-grained sentiment-controllable text-generation task. Specifically, given the fine-grained sentiment units (i.e., aspect-polarities or aspect-opinions) as the input s, the model generates a target text X that covers the input sentiments. As a straightforward approach, as other studies have used [5,7,8], the data-to-text modeling can be much more difficult when compared with the text-to-text modeling due to the discrete and sparsity of the input data [17]. Therefore, in this work, we consider a translation of this task to the text-to-text formulation. More conveniently, given aspect and polarity, it is effortless to retrieve opinion phrases from aspect sentiment triplets (AST [19], i.e., the triplet of aspect, opinion, and sentiment polarity) extracted from the review text. This work, therefore, set $s = \{(a_1, o_1), (a_2, o_2), \ldots, (a_n, o_n)\}$ and aims to generate a review text X comprising m words ($X = \{x_1, x_2, \ldots, x_m\}$), which presents each aspect phrase a_i and its corresponding opinion phrase o_i ($i \in \{1, 2, \ldots, n\}$) properly.

In this task, we have a labeled dataset L and an unlabeled dataset U. In the labeled dataset L, each labeled datum $\ell \in L$ comprises a review text and a list of aspect-opinion phrase pairs s, i.e., $\ell = \langle X, s \rangle$, while in the unlabeled dataset U, each $u \in U$ only contains a review text, i.e., $u = \langle X \rangle$.

In the following subsections, we first introduce our main framework for how to train a generator on both labeled and unlabeled datasets. Then, we explain our generator and classifier in detail.

3.1. Main Framework

To make full use of both the limited labeled dataset and the large unlabeled dataset, inspired by Chen et al. [6], in the case of a text generator G, our proposed method additionally employs a sentiment classifier C, which is incorporated to extract all aspect sentiment triplets (aspect, opinion, polarity) in each sentence through a sequence-labeling schema, thus yielding pseudo labels for the unlabeled dataset. We assume that the generator can enhance itself by leveraging a large dataset with pseudo labels predicted by the classifier.

In order to benefit from both the data size of the unlabeled dataset and the correctness of the labeled dataset, we train our model sequentially using these two datasets. Specifically, as shown in Figure 2, following Chen et al. [6], we adopt three steps to make full use of the large unlabeled dataset:

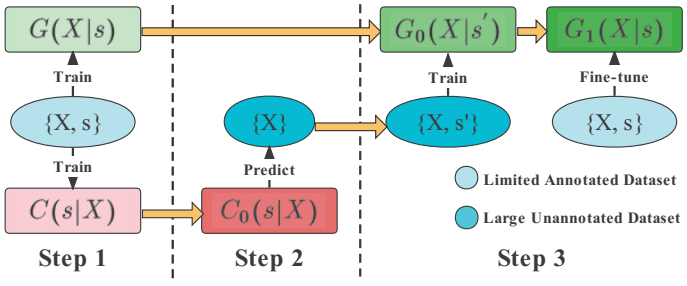


Figure 2. Illustration of the training steps for the generator and classifier. Note that "X", "s", "G", and "C" represent the review text, fine-grained sentiment, generator, and classifier, respectively.

- **Step 1:** We train both our generator and classifier on a limited labeled dataset to get *G0* and *C0*, respectively.
- **Step 2:** The *C0* is then used to extract the fine-grained sentiments in the large unlabeled dataset, thus yielding the pseudo labels for the next step's training.
- **Step 3:** Again, the generator is trained on the unlabeled dataset that is attached with pseudo labels. Finally, the generator is fine-tuned with the labeled dataset (used in Step 1) to receive the final generator *G1*.

As a result, we obtain an enhanced generator *G1* trained on both the limited labeled dataset and the large unlabeled dataset.

3.2. Generator

Unconditional language models (LMs) are trained on the huge amount of unlabeled text data to optimize the probability of $p(x_i|x_1{:}x_{i-1})$ in an auto-regressive manner [20,21] where x_i is the next token and $x_1{:}x_{i-1}$ are the previous tokens. While in the controlled text generation, the conditional distribution $p(x_i|a, x_1{:}x_{i-1})$ is optimized, where a is the attribute for the model to control the generation.

To make use of the LM pre-trained with large unlabeled datasets, we need to infuse attribute a into the unconditional distribution $p(x_i|x_1{:}x_{i-1})$. What is more, the pre-trained Transformer-based language model GPT-2 [9] has demonstrated remarkable natural text generation in an auto-regressive manner in recent years. Thereby, to improve the generated texts' quality, our generative model incorporates a pre-trained GPT-2 model as the "super-generator," and we further use the fine-grained sentiment infusion blocks, which are stacked in the AlSeCond to extend this pre-trained state-of-the-art language model's decoder blocks.

Essentially, the GPT-2 model is stacked with numerous Transformer-Decoder blocks, each consisting of layer normalization [22], multi-head self-attention [1], and position-wise feed-forward operations. Therefore, our AlSeCond blocks extend this kind of decoder block and incorporate a sentiment infusion operation together with our proposed query-hint mechanism to conditionally infuse the fine-grained sentiments into the next-token prediction process.

The sentiment infusion operation is performed inside the AlSeCond's blocks. Figure 3 briefly illustrated how our AlSeCond model works. Specifically, the target fine-grained sentiment pairs **s0** are appended sequentially as a prompt to the head of the regular sequence **s1** to form the **S**. This special appended sequence **S** is then encoded to h ($h = [h^0; h^1]$, h^0, h^1 is the hidden representation of **s0** and **s1**, respectively) through numerous AlSeCond blocks, thus h_t^1 self-attends to the hidden states of the regular sequence h^1 for previous t time steps and, further, all time steps of the fine-grained sentiment pairs h^0. Therefore, the sentiment representation h^0 is infused into the intermediate representation h^1 to control the next token logits (o) and hence the generation process.

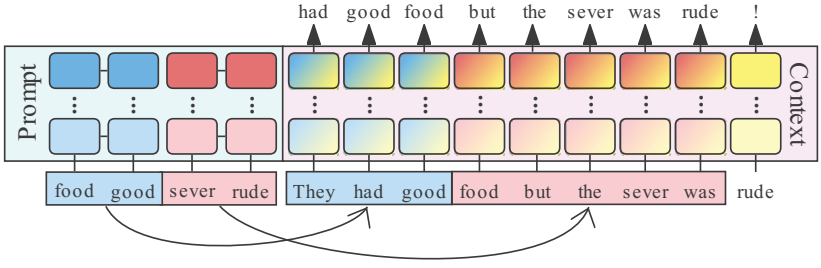


Figure 3. Illustration of how the AlSeCond model works. The curved arrow indicates where the sentiment unit should be hinted to the review sequence. The gradient color in the square indicates that this step is affected by the query-hint mechanism with prompt values brought by it.

Our AlSeCond's block (illustrated in the pink block in Figure 4) is a special Transformer-Decoder block that incorporates our proposed query-hint mechanism to guide the controlled generation process. Specifically, for fine-grained sentiment-appended hidden states, $h = [h^0; h^1]$ (h^0 and h^1 are the hidden states for the sentiment and regular sequence, respectively.), its key, value, and a special hinted query matrix ($K, V, Q' \in R^{(l_s+t)\times d}$, l_s, t is the length of the appended sentiments and regular sequence, respectively) are computed to perform a query-hinted self-attention. Furthermore, during the computation of the hinted query (Q') matrix, we infuse $K^0 \in R^{l_s \times d}$, the sentiments' part of K, into $Q^1 \in R^{t\times d}$ at their corresponding time step as the query-hint:

$$\begin{aligned} Q &= [Q^0; Q^1] = h \times W_q^T, \quad K = [K^0; K^1] = h \times W_k^T \\ Q' &= [Q^0; Q'^1], \qquad\qquad Q'^1 = f_{hint}(K^0, Q^1) \times W_{q'}^T \end{aligned} \tag{1}$$

$$f_{hint}(K^0, Q^1) = Q^1 + M_h \times \begin{bmatrix} Mean(K_{0:l_1}) \\ Mean(K_{l_1:l_2}) \\ \cdots \\ Mean(K_{l_{n-1}:l_n}) \end{bmatrix}$$

where $f_{hint}(\cdot)$ is our proposed function, it strategically allocated the sentiments' representation to Q^1 as the query-hint information, and $M_h \in R^{t\times n}$ is an adjacency matrix, representing which sentiment pair should be hinted for each time step in Q^1, and n is the number of sentiment pairs, l_a ($a \in \{1, 2, \ldots, n\}$) is the end index of the a-th sentiment pair in **S**. As a result, we guide the text generation by infusing the sentiment information into the generation process through the query-hinted self-attention operation.

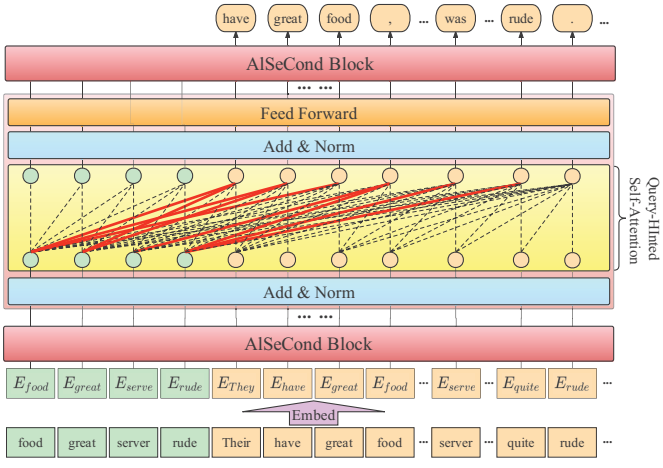


Figure 4. Architecture of the generator. This model is stacked with 24 AlSeCond blocks with the same structure. The dashed lines in the block represent the general attention, while the red solid lines represent the attention that is hinted at with prompt key values.

3.3. Query-Hint Mechanism

Since the distance from the prompt and the next-token prediction correlates negatively with the prompt's influence [23], which makes it difficult to use a prompt to guide a non-adjacent piece of text, especially when the generation time step is far away from the prompt. In other words, prompt and regular sentences share equal importance, which is inadequate for prompt-based generative models because the prompt tokens propagate less dominant information to the next-token prediction as the sequence expands. Our idea is similar to Xia et al. [24], where the actual importance of information from different sentiment units is unequal to each token in a sentence, so they need to be attended to differently. Therefore,

as mentioned in Section 3.2, we introduced a query-hint mechanism to further remind each generation time step about the following content. The main idea of this mechanism is to let the generation process understand what text to generate in order to catch the next sentiment text.

Specifically, for each general sentiment pair, its aspect and opinion phrases have their own corresponding subsequence to provide query-hints. As shown in Figure 5 (e.g., 1 to 1), a sentiment pair's member starts query-hint at the beginning of the sentence or the end step of the previous sentiment pair and closes before its own full-presenting. The hinted steps form a "hint-unit" (framed in the red dotted line in Figure 5).

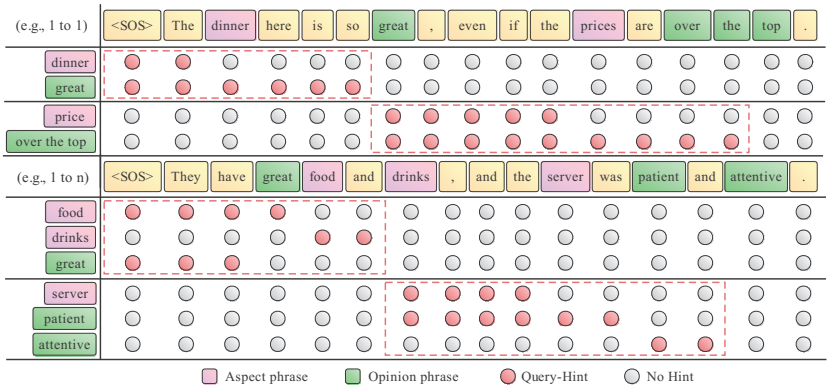


Figure 5. Strategy of the query-hint mechanism, this illustration demonstrates two different instances of query-hint strategy, i.e., "1 to 1" and "1 to n," which correspond to the one-to-one and one-to-many situations for aspect-opinion pairs, respectively.

In the source sentences, however, there are also some sentiment pairs that share the same phrase either in aspect or opinion (e.g., (food, great), (drinks, great)). Therefore, in order to make query-hint consistent in the training and generation process, given n sentiment pairs that share the same aspect/opinion phrase, their query-hints are merged into one "hint-unit". As shown in Figure 5 (e.g., 1 to n), within the "hint-unit", each aspect/opinion phrase gives the query-hint sequentially.

Although our proposed strategy of query-hint in the training process is almost identical to the generation process, there is still a slight difference between them. During the training process, the corresponding time steps in the sentence are provided with query-hint according to the position of each sentiment information presented in the sentence. While in the generation process, since the part of the sentence that has not been generated is unknown, query-hint should be allocated according to the generated part of the sentence.

3.4. Loss Functions

Generation loss function: through an LM training objective, we train our conditional generative model with the general generating loss term conditioned on previous $x_{:t-1}$ and input sentiment information s:

$$\mathcal{L}_G = -\sum_t log[p(x_t{}'|s, x_{:t-1})]_{I^x(x_t)} \tag{2}$$

where $x_t{}'$ is the predicted token at time step t. $I^x(\cdot)$ is the index function of a vector.

Sentiment control loss function: To encourage the generator to output texts incorporating the input sentiment information (phrases), we train the generator additional with our proposed sentiment-control loss function. The main idea of this loss function is to maximize the probability value of the one with the highest probability in terms of given aspect/opinion word from all the next-word predictions of a sentence. Specifically, for

every aspect phrase a and opinion phrase o presented in the source text, the training loss is defined as:

$$\begin{aligned}
&\mathcal{L}_{Senti} = \mathcal{L}_a + \mathcal{L}_o \\
&\mathcal{L}_a = -\sum_a \sum_t log[Q(x', Mask_{a,t})]_{I^x(x_{a,t})} \\
&\mathcal{L}_o = -\sum_o \sum_t log[Q(x', Mask_{o,t})]_{I^x(x_{o,t})} \\
&Q(x, Mask) = Mask \odot p_{max}(x) + (1 \oplus Mask) \times \phi_{mean} \\
&p_{max}(x) = MaxPooling([p(x_1|s, x_{:0}); p(x_2|s, x_{:1}); \cdots; p(x_t|s, x_{:t-1})])
\end{aligned} \tag{3}$$

where $\mathcal{L}_a$ and $\mathcal{L}_o$ are the losses for aspect and opinion term inclusion, respectively. $Mask_{a,t/o,t}$ is a one-hot vector with the size of $\mathcal{V}$ (vocabulary size), and only the element in the index of a_t/o_t is 1. ϕ_{mean} is a hyper-parameter controlling how much the prediction of aspect/opinion terms should be enhanced. $p_{max}(\cdot)$ is a max-pooling operation with a kernel size of $l_t \times 1$ (l_t is the length of the target text). $\odot$ and $\oplus$ represent the element-wise product and XOR, respectively.

As a result, our final loss function comprehensively considers the loss of generation quality and the loss of sentiment control:

$$\mathcal{L}_{total} = \lambda_G \mathcal{L}_G + \lambda_{Senti} \mathcal{L}_{Senti} \tag{4}$$

where λ values are hyper-parameters controlling how much the loss terms dominate the training.

3.5. Classifier

In this section, first, we give the task definition of Aspect Opinion Pair Extraction (AOPE), then we briefly introduced the model architecture of our sentiment classifier **C**.

The task of AOPE aims to extract aspect terms and their corresponding opinion terms as pairs [25–27]. This task can be defined as follows: Given a sentence with m words $X = \{x_1, x_2, \ldots, x_m\}$, the goal of this task is to extract all aspect-opinion pairs $\tau = \{(a, o)_n\}_{n=1}^{|\tau|}$ from X, where $\{(a, o)_n\}$ is an aspect-opinion pair presented in X and the notations a and o denote an aspect term and an opinion term, respectively.

For the overall architecture of our classifier, the two-dimensional interaction-based multi-task learning framework (2D-IMLF) is shown in Figure 6. Given an input sentence, two highly related works of the extraction task (aspect term extraction and opinion term extraction) are adopted to learn aspect-related and opinion-related features, respectively. Then, to capture different interactive features of aspect terms and opinion terms, a 2D interactive representation is obtained by tensor composition. Finally, the classifier model regards the AOPE task as a grid tagging problem and in the end, obtains the final results by applying a decoding algorithm [28].

As shown in Figure 6, we first use a group of CNN layers to encode the input sentence and get their hidden state:

$$\begin{aligned}
&H_k^c = Conv1D_k(X) \\
&H_*^c = [H_1^c; H_2^c; \ldots; H_k^c] \\
&H^c = Conv1D_3(Conv1D_5(H_*^c))
\end{aligned} \tag{5}$$

where $k \in \{1, 2, 3, \ldots\}$ represents the kernel size of an 1D-CNN. Then, a Bi-LSTM layer together with multi-head self-attention is incorporated to extract the context information from the sentences:

$$\begin{aligned}
&H_t^l = BiLSTM(H_{t-1}^l, H_t^c) \\
&H_c = MultiHeadAttention(H^l)
\end{aligned} \tag{6}$$

Afterward, we concatenate the hidden state H_c with their transferring state H_c^T to get a grid-formed feature. We then obtain the prediction probabilities of P_a^c and P_o^c for aspect and opinion terms, respectively, from the final logits P:

$$\begin{aligned} \hat{O}_c &= [H_c; H_c^T] \\ P &= Linear(\hat{O}_c) \end{aligned} \tag{7}$$

Finally, by using a grid-formed tagging schema [28], we can easily obtain a series of aspect-opinion pairs.

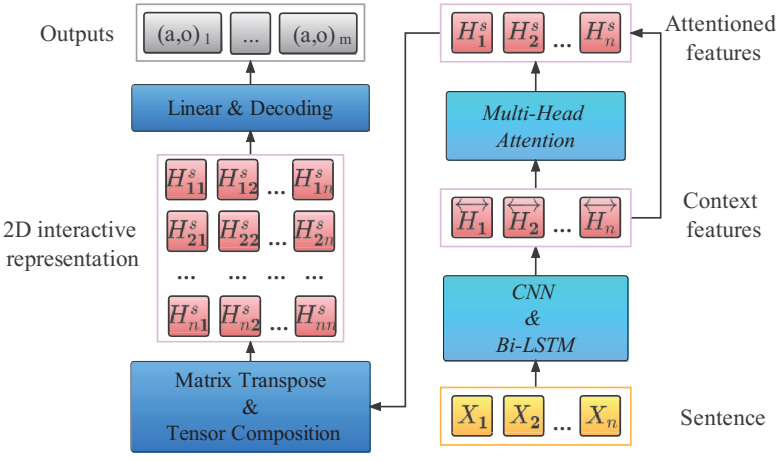


Figure 6. Architecture of the classifier. This model incorporates 2D interaction representation and grid-formed tagging schema [28] to extract all aspect and opinion phrases in a sentence.

4. Experiments

In this section, we first introduce datasets and settings in our experiment and then report the evaluation metrics and results.

4.1. Dataset and Settings

We conduct experiments on three real-world datasets, two labeled and one unlabeled; the statistics of the datasets are reported in Table 1. Moreover, the experimental settings are also listed in this subsection.

Table 1. Statistics of the labeled and unlabeled datasets. Note that "Val" is short for "Validation", the ASTE-Data-V2-Rest is labeled with aspect, opinion, and polarity, while the MAMS-ASTA is labeled with only aspect and polarity.

Dataset		#Instance	#Positive	#Neutral	#Negative	Sentiment Form
ASTE-Data-V2-Rest	Train	2728	3490	241	1014	Aspect-Opinion-Polarity
	Val	668	841	76	248	
	Test	1140	1497	120	376	
MAMS-ASTA	Train	4297	3380	5042	2764	Aspect-Polarity
	Val	500	403	604	325	
	Test	500	400	607	329	
Yelp	-	1,160,546	-	-	-	-

4.1.1. Labeled Dataset

We conduct experiments of aspect-opinion and aspect-polarity pairs of conditioned controllable text generation on English restaurant reviews with ASTE-Data-V2 from Xu et al. [29] and MAMS-ASTA from Jiang et al. [30], respectively.

ASTE-Data-V2 (https://github.com/xuuuluuu/SemEval-Triplet-data, accessed on accessed on 18 May 2022): From Xu et al. [29], is originally from SemEval Challenges [31–33], and contains both aspect and opinion labels in each review datum. Specifically, we union the 14Rest, 15Rest, and 16Rest included in the ASTE-Data-V2 as our labeled dataset.

MAMS-ASTA: From MAMS (https://github.com/siat-nlp/MAMS-for-ABSA, accessed on accessed on 14 May 2022) (Multi-Aspect Multi-Sentiment), ref. [30] is an aspect-level sentiment-labeled dataset. Wherein, each datum instance in MAMS-ASTA is labeled with at least two aspects and different sentiment polarities, while no opinion term is labeled. Therefore, by using our classifier to retrieve opinion phrases according to the original pairs of aspect-polarity, we also conduct aspect-level sentiment-controllable text generation on MAMS-ASTA.

4.1.2. Unlabeled Dataset

To ensure that the training data are in the relevant review domain, we use Yelp's review dataset (https://www.kaggle.com/yelp-dataset/yelp-dataset, accessed on accessed on 18 May 2022) as the unlabeled dataset and filter out the sentences with a length greater than 150. Unlike the labeled datasets, the Yelp dataset did not contain fine-grained sentiment labels. Therefore, we only use the sentences in the unlabeled data and discard other items, including user information.

4.1.3. Experimental Settings

Generator: In the experiment, we train our AlSeCond model that extends from a pre-trained GPT-2 medium 345M model [9]. The AlSeCond's blocks clone the GPT-2 Transformer blocks' parameters and settings. To ensure the generator can compute the probability of (and also generate) any string, we apply Byte Pair Encoding (BPE) [34] for the inputs. The max generating length was set to 32. We tune the λ_G together with λ_{senti} to 1 and 8, respectively. Adam [35] is used for optimization, while the batch size is set to 16, and the learning rate is set to 5×10^{-5}. During the period of G0, the generator is trained with the labeled and pseudo-labeled dataset for 4 and 2 epochs, respectively. In the following G1, the generator is fine-tuned with the labeled dataset for 24 epochs. We apply the above steps to train our model on an RTX A4000 GPU for 20 h. Furthermore, the above steps are also applied to train other baseline models. We ran our model and all baselines five times to average the scores.

Classifier: Following GTS [28], we combine a 300-dimension domain-general embedding from pre-trained GloVe [36] and a 100-dimension domain-specific embedding trained with fastText [37] to initialize double word embeddings. We use Adam as the optimizer, and the learning rate is 5×10^{-4}. The batch size and dropout rate are set to 32 and 0.5, respectively. The number of hidden units in Bi-LSTM is set to 128.

4.2. Baselines

We compare with six baselines. **PPLM** [4] incorporates an attribute model BoW (bag of words) to steer a pre-trained GPT-2 model toward increasing the generating probability of the target words. In this baseline, the BoW is formed with the words contained in the target sentiment pairs. For **HTT** [18], we omit the process of opinion phrase generation and only use its results (i.e., sentiment pairs) to compose the review. Through prepending the task description before the input text, the state-of-the-art text-to-text model **T5** [38] is pre-trained with a multi-task objective. Following this schema, we append the sentiment pairs into the prompt, thus forming: "*generate a sentence with* a_1 *is* $o_1, \ldots,$ a_n *is* o_n.", and fine-tune the model with the target sentence. Its coverage of the input sentiment pairs in the baselines serves as an upper bound. Moreover, we also fine-tune **UniLM** [39], **UniLM-v2** [40], and

BERT-Gen [40] in a similar sequence-to-sequence fashion with both the large unlabeled dataset and the limited labeled dataset.

4.3. Generated Quality Evaluation

To study the performance of these models in a diversified manner, we conduct evaluations on both the quality and sentiment coverage of the generated text.

4.3.1. Fluency and Diversity Evaluation

We conduct a fluency evaluation on the generated texts with some automatic metrics: BLEU [41], ROUGE [42], and METEOR [43], which compare the similarity between the generated text and ground truth based on n-gram matching. Moreover, the diversity of generations is also an important indicator. We measure diversity for the generated results with Dist-1,-2,-3 [44] scores and Self-Bleu [45].

Table 2 shows the fluency and diversity evaluation results by the automatic evaluations. From the results, we can observe that: (1) Compared with baseline models, our AlSeCond model extended from the GPT-2 achieves better performance in fluency evaluations. (2) Comparing results in diversity metrics, it can be observed that our AlSeCond model performs much better than the rest of the baselines in the MAMS-ASTA dataset, which means the results generated by our model are less like the template-generated text than that generated by other models.

Table 2. Results for the fluency and diversity evaluation. Note that "↑" means the higher the better, "↓" means the lower the better, "w/o" means "no".

Dataset	Models	BLEU-3 (↑)	BLEU-4 (↑)	METETOR (↑)	ROUGE-L (↑)	Self-Bleu-4(↓)	Dist-1 (↑)	Dist-2 (↑)	Dist-3 (↑)
ASTE-Data-V2	PPLM	0.196	0.032	14.078	13.827	**7.939**	0.0841	0.4102	0.7180
	HTT	13.100	7.656	34.899	42.544	42.664	0.0525	0.2356	0.4113
	T5-base	21.246	13.216	29.007	41.092	22.580	0.1621	0.4725	0.6101
	T5-large	24.747	16.462	29.986	43.614	23.045	**0.1721**	0.4658	0.5934
	UniLM	33.093	27.486	46.808	52.582	20.334	0.1489	0.4961	0.6663
	BERT-Gen	32.693	28.050	45.223	45.162	24.149	0.1450	0.4957	0.6411
	UniLM-v2	32.159	27.525	45.107	44.514	22.830	0.1451	0.5060	0.6553
	AlSeCond	**40.453**	**34.611**	55.127	**63.720**	15.972	0.1610	**0.5439**	0.7073
	⌊ w/o sentiment loss	37.961	32.190	**55.699**	62.911	16.195	0.1552	0.5301	0.7028
	⌊ w/o query-hint	34.305	29.080	55.391	61.237	14.442	0.1551	0.5431	**0.7264**
	⌊ w/o unlabeled dataset	29.085	26.387	42.601	48.213	21.727	0.1444	0.4942	0.6628
MAMS-ASTA	HTT	2.279	0.412	17.193	23.197	51.373	0.0602	0.2271	0.4003
	T5-base	3.653	1.479	14.400	24.181	27.671	0.1299	0.3761	0.5541
	T5-large	4.212	1.767	15.180	25.828	27.626	0.1418	0.3761	0.5591
	UniLM	3.178	1.251	18.833	23.872	37.890	0.1032	0.3211	0.4878
	BERT-Gen	4.003	1.605	17.751	24.162	28.284	0.1284	0.4024	0.5778
	UniLM-v2	3.898	1.559	17.757	23.999	27.858	0.1255	0.3989	0.5796
	AlSeCond	**5.159**	**2.113**	19.736	**31.738**	13.714	**0.1627**	0.5085	0.6811
	⌊ w/o sentiment loss	4.944	1.999	**23.734**	31.302	14.112	0.1477	0.4978	0.7171
	⌊ w/o query-hint	4.208	1.635	23.661	29.497	**10.835**	0.1604	**0.5538**	**0.7653**
	⌊ w/o unlabeled dataset	3.458	1.026	20.761	28.924	15.787	0.1478	0.4728	0.6627

4.3.2. Sentiment Evaluation

As to measure the quality of sentiment containment in the generated sentence and indicate whether the input sentiments are correctly expressed in the generated text, we employ two metrics: **Coverage** (Cov.), just like in Lin et al. [46], which is the average rate of input sentiment pairs presented in the generated texts. This metric includes Cov-a, Cov-o, and Cov-ao, representing the presenting rate of aspect, opinion, and aspect-opinion pairs, respectively. **Accuracy** (Acc.) is a rate indicating how many fine-grained sentiments are accurately expressed in the sentence, and it is evaluated by the external sentiment classifier [30] trained on MAMS-ASTA.

Table 3 shows the results of sentiment coverage and accuracy for generated texts. It is worth noting that for a linguistically complicated sentence, its aspect-level sentiments are more difficult to be correctly predicted by the external classifier than a relatively simple sentence, so its sentiment accuracy may be lower than the actual situation. What is more, T5's original seq2seq architecture allows it to generate texts that highly correspond to

the input sequences. Hence its coverage and accuracy scores serve as an upper bound, although its generated results' syntax is relatively simple and repetitive.

Table 3. Results for the sentiment evaluation. Note that Accuracy (Acc.) is a rate indicating how many fine-grained sentiments are accurately expressed in the sentence, and it is automatically evaluated by an external classifier.

Dataset	Models	Cov-a	Cov-o	Cov-ao	Acc.
ASTE-Data-V2	PPLM	0.3597	0.3642	0.1094	0.1761
	HTT	0.7689	0.7773	0.6050	0.6328
	T5-base	0.9563	0.9764	0.9403	0.7812
	T5-large	0.9633	0.9839	0.9508	**0.7948**
	UniLM	0.9513	0.9568	0.9182	0.7450
	BERT-Gen	0.9352	0.9343	0.8886	0.7521
	UniLM-v2	0.9438	0.9488	0.9087	0.7475
	AlSeCond	**0.9824**	**0.9849**	**0.9734**	0.7771
	⌊ w/o sentiment loss	0.9633	0.9649	0.9468	0.7683
	⌊ w/o query-hint	0.9412	0.9313	0.8966	0.7443
	⌊ w/o unlabeled dataset	0.8158	0.8841	0.7556	0.6306
MAMS-ASTA	HTT	0.7203	0.5123	0.3800	0.4532
	T5-base	0.9610	0.9147	0.9042	0.5734
	T5-large	0.9738	0.9453	0.9416	0.5698
	UniLM	0.9251	0.7821	0.7590	0.5883
	BERT-Gen	0.9438	0.8009	0.7807	0.6048
	UniLM-v2	0.9341	0.7515	0.7305	**0.6310**
	AlSeCond	**0.9798**	**0.9588**	**0.9558**	0.6267
	⌊ w/o sentiment loss	0.9318	0.8952	0.8825	0.6050
	⌊ w/o query-hint	0.8338	0.6811	0.6257	0.5447
	⌊ w/o unlabeled dataset	0.7829	0.7095	0.6325	0.5157

Comparing the above metrics results for all models on different datasets, we can observe that our model has stable advantages over both ASTE-Data-V2 and MAMS-ASTA, which indicates that our AlSeCond model has stronger adaptability. Additionally, Figure 7 presents the learning curves for fine-tuning all models with the labeled dataset, which also demonstrates the strong capabilities of our model compared to baselines.

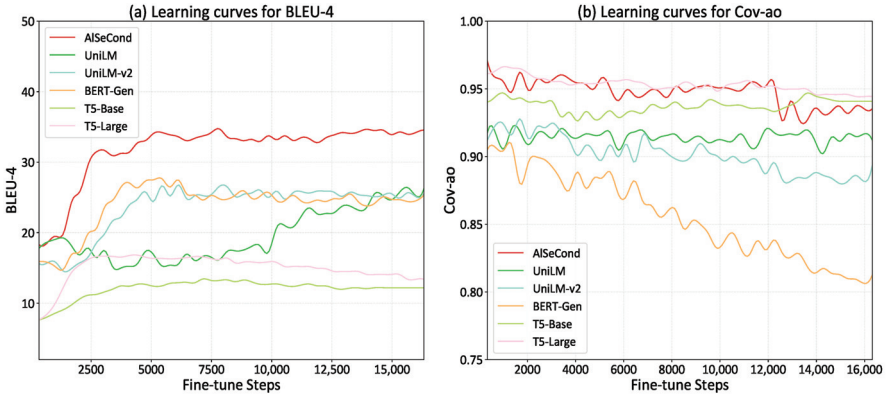


Figure 7. Learning curves for fine-tuning models with the labeled dataset. (**a**) illustrated the learning curves for BLEU-4 changing with fine-tuning steps. (**b**) illustrated the learning curves for Cov-ao changing with fine-tuning steps.

4.4. Case Study

Figure 8 presents some generated cases from AlSeCond, HTT, T5, UniLM, BERT-Gen, and UniLM-v2. From the cases, we found that: AlSeCond tends to generate more linguistically complicated sentences, while the other baselines are more likely to focus on generating review texts that simply express the input information and less on the complexity of the expressions and the syntaxes.

Aspect-level Sentiments: {wait staff - friendly, meal - great}
AlSeCond: the wait staff is very friendly and will take great care of you, if you end up getting a great meal, they 'll even throw in some dessert.
HTT: the wait staff is very friendly and you get a great deal for a great meal at a great price and a great value.
T5-Large: wait staff was friendly and the meal was great.
UniLM: The wait staff is friendly and you always have a great meal and always leave feeling satisfied.
BERT-Gen: the wait staff is very friendly and always has a great meal.
UniLM-v2: wait staff is friendly and we have always had a great meal!

Aspect-level Sentiments: {hostess - kind, hostess - gracious}
AlSeCond: It's always a delight to have greeted by a kind and gracious hostess.
HTT: the hostess was very kind, gracious, and made us feel at home with a smile.
T5-Large: the hostess was kind and gracious.
UniLM: The hostess was very kind and gracious.
BERT-Gen: the hostess is very kind and gracious.
UniLM-v2: our hostess and all of the people helping her were kind and gracious.

Aspect-level Sentiments: {atmosphere - cozy, service - horrible}
AlSeCond: When I sat down at the bar the atmosphere was cozy but service was horrible.
HTT: the atmosphere is very cozy, but the service is horrible, i would never go there again.
T5-Large: the atmosphere is cozy, but the service is horrible.
UniLM: The atmosphere is very cozy but the service is horrible.
BERT-Gen: cozy atmosphere and horrible service.
UniLM-v2: cozy atmosphere but horrible service

Aspect-level Sentiments: {place - hidden away, place - worth}
AlSeCond: The place is a little hidden away, but it is worth finding.
HTT: this place is a must if you want to try it, it's a must and worth it.
T5-Large: hidden away and worth every penny.
UniLM: This place is so hidden away but is def worth an hour drive.
BERT-Gen: this place is hidden away but is def worth it.
UniLM-v2: this place is a bit hidden away but is worth it.

Figure 8. Generated samples from the generative models. Red phrases represent the aspect-level sentiment formed by aspect-opinion pairs.

5. Conclusions and Future Work

In this paper, we propose a fine-grained sentiment-controllable text-generation method based on the pre-trained language model and the auxiliary sentiment classifier that utilizes both the labeled and unlabeled dataset to reach the aspect-level sentiment control in text generation. Our proposed query-hint mechanism and fine-grained sentiment control loss function have greatly enhanced the generator in controlling the sentiment during the text-generating process. Experiments on real-world datasets have demonstrated our generator's ability to generate aspect-level sentiment-controllable review statements with high quality and diverse syntax.

For future work, we will explore the controllable text generation for implicitly expressed fine-grained sentiments (e.g., in this sentence: *"We had to constantly ask the waiter to top up water glasses."*, the reviewer had a negative opinion of the waiter although there is no related opinion phrase in the sentence.), since the query-hint mechanism proposed in this paper is only effective for explicitly expressed fine-grained sentiments.

Author Contributions: Conceptualization, L.Z. and Y.X.; methodology, Y.X.; software, Y.X.; validation, Y.X. and Z.Z.; formal analysis, Y.B.; investigation, L.Z. and Y.B.; resources, Y.X. and X.K.; data curation, Y.X. and X.K.; writing—original draft preparation, Y.X. and Y.B.; writing—review and editing, Y.X. and Y.B.; visualization, Y.X. and Z.Z.; supervision, L.Z. and X.K.; project administration, Y.X.; funding acquisition, L.Z. and X.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China (No. 62176234, 62072409).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The source code used to generate the results shown in this paper is available at https://github.com/ashooha0/Alsecond, accessed on 1 November 2022. The dataset, attached with pseudo labels by our classifier, is available at: https://drive.google.com/file/d/1HjyTLBBlyAOn_pphWC6VjgWQ2HPZglAp/view?usp=share_link, accessed on 15 November 2022. The ASTE-data-V2 dataset is available at https://github.com/xuuuluuu/SemEval-Triplet-data, accessed on accessed on 18 May 2022, and the MAMS dataset is available at https://github.com/siat-nlp/MAMS-for-ABSA, accessed on accessed on 14 May 2022.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LMs	Language Models
NLG	Natural Language Generation
GPT	Generative Pre-Training
PPLM	Plug and Play Language Model
CTRL	Conditional-Transformer-Language
CoCon	Content-Conditioner
FSCTG	Fine-grained Sentiment-Controlled Text Generation
A2T	Attribute-to-Text
AT2T	Attribute-matched-Text-to-Text
HTT	Hierarchical Template-Transformer
AlSeCond	Aspect-level Sentiment Conditioner
AOPE	Aspect Opinion Pair Extraction
2D-IMLF	Two-Dimensional Interaction-Based Multi-task Learning Framework
CNN	Convolutional Neural Networks
Bi-LSTM	Bidirectional Long Short-Term Memory
Val	Validation
AST	Aspect Sentiment Triplet
ASTE	Aspect Sentiment Triplet Extraction
MAMS-ASTA	Multi-Aspect Multi-Sentiment Aspect-Term Sentiment Analysis
BPE	Byte Pair Encoding
GTS	Grid Tagging Scheme
GloVe	Global Vectors
UniLM	Unified Language Model
BERT	Bidirectional Encoder Representations from Transformer
BLEU	Bilingual Evaluation Understudy
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
METEOR	Metric for Evaluation of Translation with Explicit Ordering
Dist	Distinct
Cov	Coverage
Acc.	Accuracy

References

1. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
2. Keskar, N.S.; McCann, B.; Varshney, L.R.; Xiong, C.; Socher, R. CTRL: A Conditional Transformer Language Model for Controllable Generation. *arXiv* **2019**, arXiv:1909.05858.
3. Ziegler, D.M.; Stiennon, N.; Wu, J.; Brown, T.B.; Radford, A.; Amodei, D.; Christiano, P.F.; Irving, G. Fine-Tuning Language Models from Human Preferences. *arXiv* **2019**, arXiv:1909.08593.
4. Dathathri, S.; Madotto, A.; Lan, J.; Hung, J.; Frank, E.; Molino, P.; Yosinski, J.; Liu, R. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. *arXiv* **2019**, arXiv:1912.02164.
5. Zang, H.; Wan, X. Towards Automatic Generation of Product Reviews from Aspect-Sentiment Scores. In Proceedings of the International Conference on Natural Language Generation, Santiago de Compostela, Spain, 4–7 September 2017.
6. Chen, H.; Lin, Y.; Qi, F.; Hu, J.; Li, P.; Zhou, J.; Sun, M. Aspect-Level Sentiment-Controllable Review Generation with Mutual Learning Framework. In Proceedings of the National Conference on Artificial Intelligence, Online, 2–9 February 2021.
7. Dong, L.; Huang, S.; Wei, F.; Lapata, M.; Zhou, M.; Xu, K. Learning to Generate Product Reviews from Attributes. In Proceedings of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 3–7 April 2017.
8. Sharma, V.; Sharma, H.; Bishnu, A.; Patel, L. Cyclegen: Cyclic consistency based product review generator from attributes. In Proceedings of the International Conference on Natural Language Generation, Tilburg, The Netherlands, 5–8 November 2018.
9. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
10. Kikuchi, Y.; Neubig, G.; Sasano, R.; Takamura, H.; Okumura, M. Controlling Output Length in Neural Encoder-Decoders. *arXiv* **2016**, arXiv:1609.09552.
11. Ficler, J.; Goldberg, Y. Controlling Linguistic Style Aspects in Neural Language Generation. *arXiv* **2017**, arXiv:1707.02633.
12. Yu, L.; Zhang, W.; Wang, J.; Yu, Y. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In Proceedings of the National Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016.
13. Chan, A.T.S.; Ong, Y.S.; Pung, B.T.W.; Zhang, A.; Fu, J. CoCon: A Self-Supervised Approach for Controlled Text Generation. *arXiv* **2020**, arXiv:2006.03535.
14. Yu, D.; Yu, Z.; Sagae, K. Attribute Alignment: Controlling Text Generation from Pre-trained Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021, Online, 7–11 November 2021; pp. 2251–2268.
15. Lipton, Z.C.; Vikram, S.; McAuley, J. Generative Concatenative Nets Jointly Learn to Write and Classify Reviews. *arXiv* **2015**, arXiv:1511.03683.
16. Kim, J.; Choi, S.; Amplayo, R.K.; Hwang, S-w. Retrieval-Augmented Controllable Review Generation. In Proceedings of the International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020.
17. Fei, H.; Li, C.; Ji, D.; Li, F. Mutual disentanglement learning for joint fine-grained sentiment classification and controllable text generation. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 1555–1565.
18. Yuan, L.; Zhang, X.; Yu, L.C. Hierarchical template transformer for fine-grained sentiment controllable generation. *Inf. Process. Manag.* **2022**, *59*, 103048. [CrossRef]
19. Peng, H.; Xu, L.; Bing, L.; Huang, F.; Lu, W.; Si, L. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8600–8607.
20. Manning, C.D.; Schütze, H. *Foundations of Statistical Natural Language Processing*; MIT Press: Cambridge, MA, USA, 1999.
21. Bengio, Y.; Ducharme, R.; Vincent, P. A Neural Probabilistic Language Model. In Proceedings of the Advances in Neural Information Processing Systems, Denver, CO, USA, 29 November–4 December 2000.
22. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer Normalization. *arXiv* **2016**, arXiv:1607.06450.
23. Zou, X.; Yin, D.; Zhong, Q.; Ding, M.; Yang, Z.; Tang, J. Controllable Generation from Pre-trained Language Models via Inverse Prompting. In Proceedings of the Knowledge Discovery and Data Mining, Virtual, 14–18 August 2021.
24. Xia, F.; Wang, L.; Tang, T.; Chen, X.; Kong, X.; Oatley, G.; King, I. CenGCN: Centralized Convolutional Networks with Vertex Imbalance for Scale-Free Graphs. *IEEE Trans. Knowl. Data Eng.* **2022**. [CrossRef]
25. Zhao, H.; Huang, L.; Zhang, R.; Lu, Q.; Xue, H. SpanMlt: A Span-based Multi-Task Learning Framework for Pair-wise Aspect and Opinion Terms Extraction. In Proceedings of the Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020.
26. Chen, S.; Liu, J.; Wang, Y.; Zhang, W.; Chi, Z. Synchronous Double-channel Recurrent Network for Aspect-Opinion Pair Extraction. In Proceedings of the Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020.
27. Zhu, L.; Xu, M.; Bao, Y.; Xu, Y.; Kong, X. Deep learning for aspect-based sentiment analysis: A review. *PeerJ Comput. Sci.* **2022**, *8*, e1044. [CrossRef] [PubMed]
28. Wu, Z.; Ying, C.; Zhao, F.; Fan, Z.; Dai, X.; Xia, R. Grid Tagging Scheme for End-to-End Fine-grained Opinion Extraction. In Proceedings of the EMNLP (Findings), Online, 16–20 November 2020.
29. Xu, L.; Li, H.; Lu, W.; Bing, L. Position-Aware Tagging for Aspect Sentiment Triplet Extraction. *arXiv* **2020**, arXiv:2010.02609.

30. Jiang, Q.; Chen, L.; Xu, R.; Ao, X.; Yang, M. A Challenge Dataset and Effective Models for Aspect-Based Sentiment Analysis. In Proceedings of the Empirical Methods in Natural Language Processing, Hong Kong, China, 3–7 November 2019.
31. Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In Proceedings of the International Conference on Computational Linguistics, Dublin, Ireland, 23–29 August 2014.
32. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Manandhar, S.; Androutsopoulos, I. SemEval-2015 Task 12: Aspect Based Sentiment Analysis. In Proceedings of the North American Chapter of the Association for Computational Linguistics, Denver, CO, USA, 31 May–5 June 2015.
33. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; AL-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; Clercq, O.D.; et al. SemEval-2016 task 5: Aspect based sentiment analysis. In Proceedings of the North American Chapter of the Association for Computational Linguistics, San Diego, CA, USA, 12–17 June 2016.
34. Sennrich, R.; Haddow, B.; Birch, A. Neural Machine Translation of Rare Words with Subword Units. In Proceedings of the Meeting of the Association for Computational Linguistics, Beijing, China, 26–31 July 2015.
35. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
36. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global Vectors for Word Representation. In Proceedings of the Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014.
37. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
38. Liu, P.J.; Matena, M.; Lee, K.; Roberts, A.; Zhou, Y.; Shazeer, N.; Raffel, C.; Narang, S.; Li, W. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
39. Dong, L.; Wang, Y.; Wei, F.; Zhou, M.; Yang, N.; Gao, J.; Hon, H.W.; Liu, X.; Wang, W. Unified Language Model Pre-training for Natural Language Understanding and Generation. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
40. Piao, S.; Dong, L.; Wang, Y.; Wei, F.; Zhou, M.; Yang, N.; Gao, J.; Hon, H.W.; Bao, H.; Liu, X.; et al. UniLMv2: Pseudo-Masked Language Models for Unified Language Model Pre-Training. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020.
41. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002.
42. Lin, C.Y. ROUGE: A Package for Automatic Evaluation of Summaries. In Proceedings of the Meeting of the Association for Computational Linguistics, Barcelona, Spain, 21–26 July 2004.
43. Lavie, A.; Agarwal, A. METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In Proceedings of the Workshop on Statistical Machine Translation, Prague, Czech Republic, 23 June 2007.
44. Brockett, C.; Dolan, B.; Galley, M.; Gao, J.; Li, J. A Diversity-Promoting Objective Function for Neural Conversation Models. In Proceedings of the North American Chapter of the Association for Computational Linguistics, Denver, CO, USA, 31 May–5 June 2015.
45. Zhu, Y.; Lu, S.; Zheng, L.; Guo, J.; Zhang, W.; Wang, J.; Yu, Y. Texygen: A Benchmarking Platform for Text Generation Models. In Proceedings of the International Acm Sigir Conference on Research and Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018.
46. Lin, B.Y.; Zhou, W.; Shen, M.; Zhou, P.; Bhagavatula, C.; Choi, Y.; Ren, X. CommonGen: A Constrained Text Generation Challenge for Generative Commonsense Reasoning. *arXiv* **2019**, arXiv:1911.03705.

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

Applied Sciences Editorial Office
E-mail: applsci@mdpi.com
www.mdpi.com/journal/applsci

Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-1824-2