applied sciences

MDPI

Article

Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism

Beakcheol Jang, Myeonghwi Kim, Gaspard Harerimana, Sang-ug Kang and Jong Wook Kim *

Department of Computer Science, Sangmyung University, Seoul 03016, Korea; bjang@smu.ac.kr (B.J.); nopkgogo3@gmail.com (M.K.); gharelim@alumni.cmu.edu (G.H.); sukang@smu.ac.kr (S.-u.K.)

* Correspondence: jkim@smu.ac.kr

Received: 20 July 2020; Accepted: 18 August 2020; Published: 24 August 2020

Abstract: There is a need to extract meaningful information from big data, classify it into different categories, and predict end-user behavior or emotions. Large amounts of data are generated from various sources such as social media and websites. Text classification is a representative research topic in the field of natural-language processing that categorizes unstructured text data into meaningful categorical classes. The long short-term memory (LSTM) model and the convolutional neural network for sentence classification produce accurate results and have been recently used in various natural-language processing (NLP) tasks. Convolutional neural network (CNN) models use convolutional layers and maximum pooling or max-overtime pooling layers to extract higher-level features, while LSTM models can capture long-term dependencies between word sequences hence are better used for text classification. However, even with the hybrid approach that leverages the powers of these two deep-learning models, the number of features to remember for classification remains huge, hence hindering the training process. In this study, we propose an attention-based Bi-LSTM+CNN hybrid model that capitalize on the advantages of LSTM and CNN with an additional attention mechanism. We trained the model using the Internet Movie Database (IMDB) movie review data to evaluate the performance of the proposed model, and the test results showed that the proposed hybrid attention Bi-LSTM+CNN model produces more accurate classification results, as well as higher recall and F1 scores, than individual multi-layer perceptron (MLP), CNN or LSTM models as well as the hybrid models.

Keywords: text classification; CNN; Bi-LSTM; attention mechanism

1. Introduction

There is an unprecedented deluge of text data due to increased internet use, resulting in generation of text data from various sources such as social media and websites. Text data are unstructured and contain natural-language constructs, making it difficult to infer an intended message from the data. This has led to increased research into the use of deep learning for natural-language-based sentiment classification and natural-language inference.

Natural-language-based sentiment classification has a wide range of applications, such as movie review classification, subjective and objective sentence classification, and text classification technology [1]. Traditional text classification methods are dictionary-based and basic machine learning methods and have been recently replaced by more powerful deep learning methods, such as sequence-based long-term short memory (LSTM) [2] and, more recently, the convolution neural network (CNN) method [3].

LSTM is an improved recurring neural network (RNN) architecture that uses a gating mechanism consisting of an input gate, forget gate, and output gate [4]. These gates help determine whether data in the previous state should be retained or forgotten in the current state. Hence, the gating mechanism helps the LSTM address the issue of long-term information preservation and the vanishing gradient problem encountered by traditional RNNs [5]. The LSTM's powerful ability to extract advanced text information plays an important role in text classification. The scope of application of LSTMs has expanded rapidly in recent years, and many researchers have proposed many ways to revamp LSTMs to further improve their accuracy [6].

To further increase the sentiment classification accuracy of unstructured text, Yoon Kim [7] proposed a 1D CNN approach for text classification. CNN is more accurate than LSTM, especially during the feature extraction step. 1D CNN processes text as a one-dimensional image and a 1D CNN is used to capture the latent associations between neighboring words, in contrast with LSTMs, which process each word in a sequential pattern [8].

In this study, to address the individual weaknesses and leverage the distinct advantages of LSTM and CNN, we propose a Bi-LSTM+CNN hybrid model that classifies text using an Internet Movie Database (IMDB) movie review dataset. To further improve the accuracy and reduce the number of learnable parameters the model is boosted by an attention mechanism. The model uses a CNN to extract features from different locations in a sentence, thereby shrinking the amount of input features. The LSTM is used to extract contextual information from the features obtained from the convolutional layer. The attention mechanism uses bias alignment over the inputs and assigns weights to input components that are highly correlated with classification, hence further decreasing the number of parameters to be learned during training. The attention mechanism also enhances the weight distribution for variable-length sequences. We also use weights that were pretrained using Word2vec with a large corpus, which ensured the model had higher accuracy. The accuracy of the model was evaluated in terms of precision, recall, and F1-score.

The first set of results were obtained by fixing the data size and evaluating the performance of the proposed model per each training epoch. The accuracy is 0.8874 for CNN, 0.8940 for LSTM, 0.7129 for multi-layer perceptron (MLP), 0.8906 for the hybrid model, and the proposed model 0.9141. As for the F1 score, CNN achieved 0.8875, 0.8816 for LSTM and 0.7708 for MLP, the hybrid model achieved 0.8887 while the proposed model achieved the highest score of 0.9018. The second set of results were obtained with variable data size and the proposed model was superior in terms of accuracy and F1 score. It is evident from the two sets of results that the proposed model outperformed the existing CNN, LSTM, MLP, and hybrid model baselines.

The remainder of this paper is organized as follows. A review of related research is presented in Section 2. The architecture and details of the proposed model are described in Section 3. Experiments are presented in Section 4, the results are covered in Section 5, the analysis is in Section 6 and future works in Section 7. Finally, conclusions are provided in Section 8.

2. Background and Related Research

2.1. Deep Learning Model

2.1.1. Word2vec

Word2vec is a popular sequence embedding method that transforms natural-language into distributed vector representations [9]. It can capture contextual word-to-word relationships in a multidimensional space and has been widely used as a preliminary step for predictive models in semantic and information retrieval tasks. Figure 1 describes the Word2vec process, which involves two distinct components: Continuous Bag of Words (CBOW) and skip-gram [10]. The CBOW component infers the target word when given the context words, while the skip-gram component infers the context words when given an input word [11].

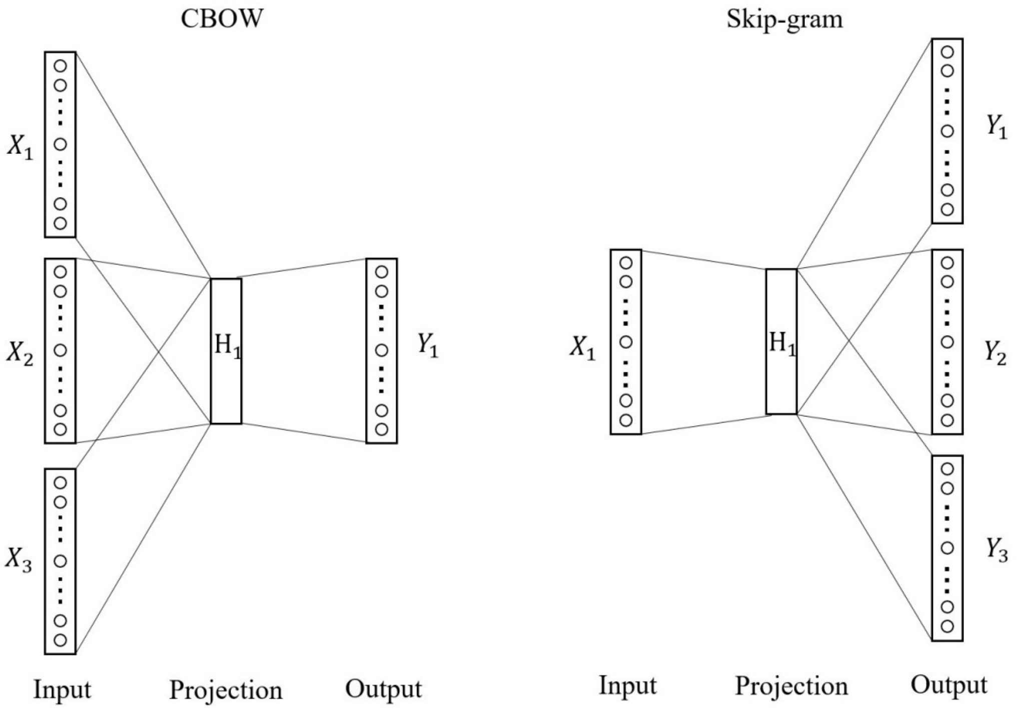

Figure 1. Word2vec model (Continuous Bag of Words (CBOW) and Skip-Ngram).

2.1.2. 1D Convolutional Neural Networks

Early 2D CNNs have been widely used in image processing, but they have only recently been used in text classification tasks and have outperformed sequence-based approaches [12]. The CNN constructs a feature map through a chain of convolutions and pooling using a convolutional layer, and a subsampling layer (or maximum pooling layer). With the 1D CNN, the convolution layer uses a 1D cross-correlation operation that involves a sliding convolution window with variable-size kernels over the input text starting from left to right [13]. It uses a max-overtime pooling layer that consists of a 1D global maximum pooling layer, which reduces the number of features needed to encode the text. Figure 2 shows the basic architecture of the 1D CNN [14].

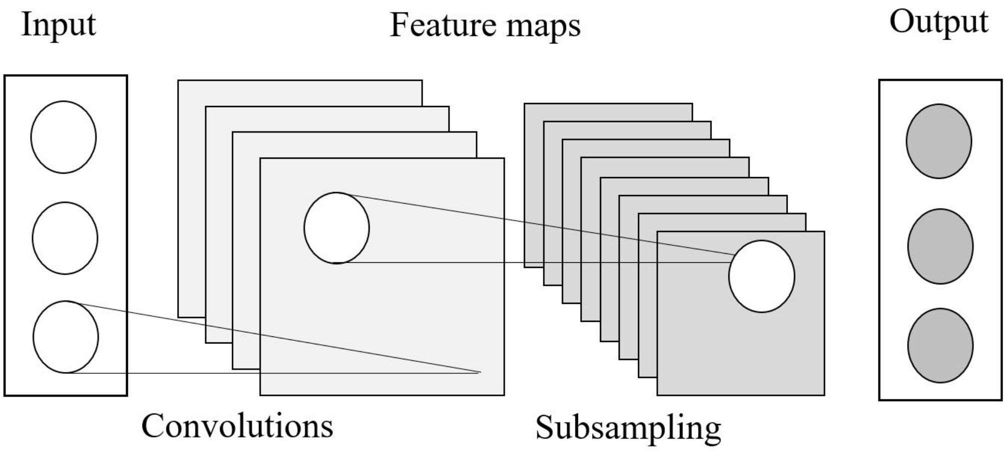

Figure 2. Structure of the 1D convolutional neural network (CNN) for text classification.

2.1.3. Bi-LSTM

The Bi-LSTM neural network is composed of LSTM units that operate in both directions to incorporate past and future context information. Bi-LSTM can learn long-term dependencies without retaining duplicate context information [15]. Therefore, it has demonstrated excellent performance for sequential modeling problems and is widely used for text classification. Unlike the LSTM network, the Bi-LSTM network has two parallel layers that propagate in two directions with forward and reverse passes to capture dependencies in two contexts [16,17]. The structure of the Bi-LSTM network is shown in Figure 3.

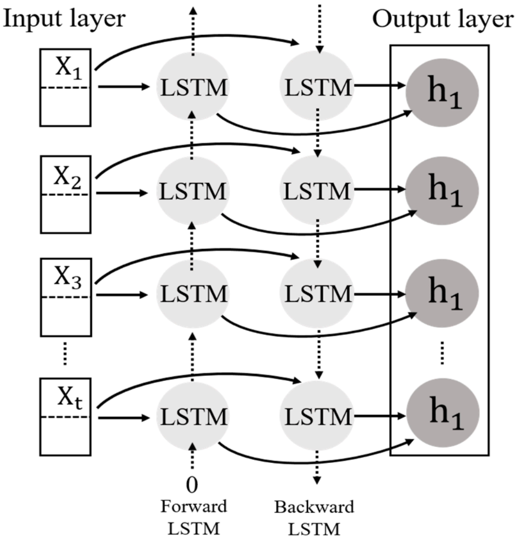

Figure 3. The Bi-long short-term memory (LSTM) process used to capture sequential features. The last hidden layer of the LSTM is extracted as features representing the text.

2.1.4. Attention Mechanism

The RNN-based seq2seq model has two major problems. Since all the information cannot be compressed in one fixed-size vector, there can be instances of information loss, and a vanishing gradient problem [18]. Hence the accuracy deteriorates with the increased length of the input. As an alternative, an attention mechanism has been proposed as an emerging technique for increasing the prediction accuracy, although this decreases when the input sequence becomes long [19]. That is, the decoder predicts the output word in every time step, and the entire input sentence from the encoder is referenced once again. However, not all the input sentences are referenced equally; the part of the input word that is related to the word to be predicted at that time receives more attention [20]. The attention mechanism gives initial weights to each input, and these weights are updated during training according to the correlation between each input and the final prediction.

2.2. Related Research

Melamud et al. [21] proposed a B-LSTM neural network architecture based on Word2vec's CBOW architecture. The main goal of the model was to efficiently learn the general context embedding function for variable-length sentence contexts around the target word, making contextual expression very simple. This resulted in cutting-edge results in sentence completion, vocabulary replacement, and word sense clarification, outperforming other techniques of general contextual representation of average word embedding. Another key model was proposed for medical texts by Zagreb [22]; they developed a model for understanding complex, specific medical structures as medical text contains some of the most specialized information that only field experts can understand. The proposed embedding model is generated in a specialized general corpus with or without features handcrafted by medical experts. Xiao et al. [23] proposed a text classification model based on Word2vec and LSTM to efficiently classify text in the security field. They used a pre-trained Word2vec model to overcome the high dimensionality suffered by traditional methods. Finally, by training the LSTM classification model, the study extracted text functions and performed patent text classification in the security field. The results show that the model can provide classification of 93.48% for these patent texts. This laid a solid foundation for further research and effective use of patents.

To increase the classification accuracy, CNN and LSTM have been combined in some studies. The LSTM model and a CNN were used for a variety of natural-language processing (NLP) tasks with surprising and effective results. Rehman et al. [24] proposed a hybrid model that uses a very deep CNN and LSTM to overcome previous challenges for sentiment analysis. The proposed model also uses dropout techniques, normalization techniques, and rectified linear units to increase the prediction accuracy. Results show that the proposed hybrid CNN-LSTM model outperforms conventional deep learning and machine learning technologies in terms of precision, recall, F1-score, and accuracy. The study in [25] proposed a text classification model called CNN-COIF-LSTM and experiments that included eight variants, showed that the combination of CNN and LSTM without an activation function or a variant thereof exhibited higher accuracy. Wang et al. [26] proposed a local CNN-LSTM model with a tree structure to measure the steps in emotional analysis. Unlike conventional CNNs, where the entire text is considered as the input, which is divided into multiple regions and useful emotional information is extracted from each region and weighted according to that region; the proposed regional CNN uses a portion of the text as a region. By combining CNN and LSTM, the classification accuracy is further increased because this combination considers both local (regional) information in sentences and long-distance dependencies between sentences. Novel hybrid CNN-LSTM models have been proposed and achieved good results than the previous ones. She et al. [27] have proposed a novel hybrid approach that leverages the capabilities of CNN to extract local features and addresses its inherent weakness to express long-term contextual features. The model also tries to tackle the natural weaknesses of LSTM which processes information sequentially and hence making it a worst feature extractor. The hybrid model achieved better results compared to counterparts' models, but the results were not interesting as compared with the models that use attention mechanism. Salur et al. [28] proposed a novel hybrid model that combines different word embedding (Word2Vec, FastText, and character-level embedding) with various learning approaches (LSTM, Gated recurrent unit (GRU), Bi-LSTM, CNN). The model used CNN and LSTM for feature extraction. Various other studies such as [29] have used this hybrid approach but with the absence of an attention mechanism, the results have been not improved. More recently the attention models have been introduced and achieved state-of-the-art results. Dong et al. [30] proposed a method that use a self-interaction attention approach and combined it with label embedding. The model leverages the novelty of BERT (bidirectional encoder representation from transformers) [31] for text extraction. The approach uses the joint embedding of words and labels to improve the classification accuracy due to self-interaction attention mechanism. However, the BERT model has been found to be hard to train especially for big text. Traditional machine learning approaches like decision trees and K Nearest Neighbor (KNN) have also performed well in certain classification scenarios. Joulin et al. [32] have trained a FastText model using more than one billion words and classified a half a million sentences in 312K classes. This seemingly traditional approach outperformed some of the deep learning approaches. However, with the reported fast performance the authors remained uncertain if sentiment classification can be used as the right candidate to compare such shallow models against deep learning approaches which have much higher representational power.

3. The Proposed Model

LSTMs have received considerable research attention, and various studies that leverage them for sequence-based sentiment classification have produced great results. Various gating mechanisms used by LSTMs help them track long-term dependencies and tackle the inherent drawbacks of vanishing gradients, especially when longer sequences are used as input. However, the existing LSTMs fail to extract the context information of future tokens and lack the ability to extract local context information. The accuracy of the LSTM is further hampered by its inability to recognize different relationships between parts of a document. To tackle this drawback, a 1D CNN has been proposed. In this study, we leverage the unique advantages of LSTM and CNN, and we propose a hybrid model that uses CNN for text feature extraction and a Bi-LSTM component with an attention mechanism for sentiment classification. The overall objective of this study is to capture the complex associations between adjacent words to increase the classification accuracy. Figure 4 summarizes the steps in the proposed model.

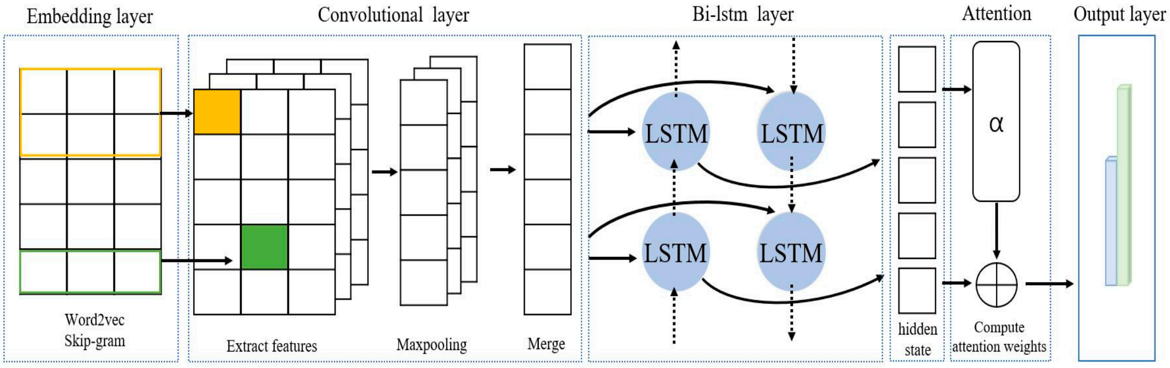

Figure 4. The proposed model architecture.

3.1. Sequence Embedding Layer

The text was preprocessed before the data was passed to the model. This includes removing whitespace and meaningless words, converting other forms of words to approximations, and removing duplicate words. The preprocessed dataset provides a unique and meaningful sequence of words, and every word has a unique identification. The embedding layer learns the distributed representation of each preprocessed input token. Representation of the tokens reflect the latent relationships between words that are likely to appear in the same context. We used the pretrained vectors that were trained with Word2vec's skip-gram model.

3.2. 1D CNN

The role of the convolution process is to extract features from the input text. Convolutional layers are used to extract low-level semantic features from the original text and to reduce the number of dimensions as opposed to Bi-LSTM, which process the text as sequential data. In this study, we use several 1D convolution kernels to perform convolution over the input vectors.

Equation (1) defines a vector of sequential text obtained by concatenating embedding vectors of the component words:

$$X_{1:T} = [x_1, x_2, x_3, x_4, \cdots x_T], \tag{1}$$

where T is the number of tokens in the text. To capture the inherent features using a 1D CNN, convolution kernels with varying sizes are applied to $X_{1:T}$ to capture the unigram, bigram, and trigram features of the text. During the tth convolution, when a window of d words that stretches from $t : t + d$ is taken as an input, the convolution process generates features for that window as follows:

$$h_{d,t} = \tan\mathrm{h}(W_d x_{t:t+d-1} + b_d), \tag{2}$$

where $x_{t:t+d-1}$ are the embedding vectors of the words in the window, W_d is the learnable weights matrix, and b_d is the bias. Since each filter must be applied to various regions of the text, the feature map of the filter with convolution size d is:

$$h_d = [h_{d1}, h_{d2}, h_{d3}, h_{d4}, \cdots x_{T-d+1}]. \tag{3}$$

The key advantage of using convolution kernels with diverse widths is that the hidden correlations between several adjacent words can be captured. The most important aspect of using a CNN for textual feature extraction is reducing the number of trainable parameters during feature learning, which is achieved using max-overtime pooling [18]. The input is acted upon by numerous convolution channels, and each distinct channel consists of values on diverse timesteps. Hence, during max-overtime pooling, the output from each convolution channel will be the largest value of all timesteps in that channel.

For each convolution kernel, we apply max-overtime pooling to the feature maps with convolution size d to obtain

$$p_d = Max^t(h_{d1}, h_{d2}, h_{d3}, h_{d4}, \cdots x_{T-d+1}), \tag{4}$$

To obtain the final feature map of the window, we concatenate p_d for each filter size d = 1, 2, 3 and extract the unigram, bigram, and trigram hidden features:

$$h_d = [p_1, p_2, p_3]. \tag{5}$$

The advantage of using the CNN over the LSTM is that it reduces the number of dimensions in the input features to be fed to a sentiment classifier or a natural inference prediction model that must follow the feature extraction step.

3.3. Bi-LSTM Attention Layer

The main classification component of our model was built on an attention-based Bi-LSTM. Although the CNN shrinks the input features to be used for prediction, the correlation between each word and final classification is not the same for all input words. In this study, we want to leverage both advantages of CNN and Bi-LSTM. The Bi-LSTM is used to effectively encode long-distance word dependencies.

The Bi-LSTM process is depicted in Figure 4. It receives the features generated by the CNN stage and generates features by extracting the last hidden layer. The Bi-LSTM has access to both the preceding and subsequent contextual information, and the information obtained by Bi-LSTM can be regarded as two different textual representations. The features obtained from the CNN S$_{ij}$ are fed to a Bi-LSTM model that produces a representation of the sequence. This final feature representation is fed to an attention layer that chooses which features are highly correlated for final classification. For a sentence such as "Though he is sick he is happy" to be classified as having positive sentiment, the attention mechanism gives more weight to "happy" and less weight to "sick" To show that the sickness (though negative occurrence) is not affecting the "happy" sentiment. With that process the attention mechanism increases the prediction accuracy considerably and reduces the number of learnable weights needed for the prediction. The proposed model's attention mechanism uses the Bahdanau attention with attention scores as follows:

$$score(querry,\ key) \ = V^T tanh\Big(W_{1key} + W_{2query}\Big) \tag{6}$$

4. Experiment

In this section, we present the experiments conducted for evaluating the performance of the proposed model. The experimental results are compared with existing state-of-the-art deep learning models. Two experiments were conducted with the number of epochs and data size taken as variables. The data used in the experiment, settings, and evaluation methods are described below.

4.1. Dataset

The experiment uses the IMDB movie review dataset to train the classification model. The IMDB movie review dataset contains binary labeled reviews that are tagged with positive and negative sentiments about movies. The first experiment used an educational test set with 20,000 entries, and the second experiment compares classification results as the number of data was increased from 5000 to 15,000. Table 1 shows the details of the dataset used in the experiment.

Table 1. Data set.

Total	Training	Test	Category	Length (Max)	Words (Avg)	Vocabulary_Size
50,000	25,000	25,000	2	500	237.71	10,000

4.2. Parameter Set

The word vector used in this experiment was embedded using skip-gram in Word2vec, and the embedding size was 500. The batch size of all data sets was set to 128 and the dropout ratio was set to 0.2. Other parameter settings are given in Table 2.

Table 2. Parameter setting.

Word Embedding	Embedding Size	Dropout	Batch Size	Optimizer	Regularizer
skip-gram	500	0.2	128	Adam	L2

4.3. Performance Evaluation Metrics

The models used in the experiment were evaluated with four evaluation metrics: accuracy, F1-score, precision, and recall. These parameters are defined as:

$$\text{Precision} = \frac{Ture\ Positive}{(True\ Positive\ +\ False\ Positive)} \tag{7}$$

$$\text{Recall} = \frac{Ture\ Positive}{(True\ Positive\ +\ False\ Negative)} \tag{8}$$

$$\text{F1 Score} = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \tag{9}$$

In Equations (7) and (8), true positive (TP) is the number of data classified as positive among the data tagged as positive, and true negative is the number of data classified as negative among the data tagged as negative. False negative (FN) is the number of data classified as negative but were tagged in the dataset as positive, and false positive (FP) is the number of data classified as positive but were tagged in the dataset as negative. Recall is the proportion of documents classified as positive by the model among all actual positively tagged data. Precision is the percentage of documents positively classified by the model as positive, and F1-score is the average of recall and precision.

5. Results

Figure 5 shows the performance of CNN, LSTM, MLP, hybrid combined with CNN+Bi-LSTM and the proposed model that change as the number of epochs increases when the data size is fixed at 20 k. Figure 5a shows the accuracy. In Figure 5a, the CNN model shows the rate of change of decreasing accuracy up to epoch 10, and the accuracy is maintained after epoch 10. Next, the LSTM model increases the accuracy up to epoch 15, and after that, a stable rate of change is observed. Next, the hybrid model increases the accuracy up to 3 epochs and has remained unchanged since then. The MLP model shows that it maintains a low accuracy of 0.7. Finally, the proposed model shows a sudden increase in accuracy up to epoch 12, and the accuracy remains stable after epoch 15. Figure 5b shows the F1 score. CNN, LSTM, hybrid, and the proposed model showed similar patterns as in the accuracy, and only the MLP model showed that the rate of change did not stabilize.

Figure 6 shows the performance of the model changing as the number of epochs is fixed at 10 and the data size is increased from 5 k to 15 k. Figure 6a,b show the accuracy and F1 score, respectively. In Figure 6a, the hybrid model and the proposed model show that the accuracy increases as the data size increases. The CNN model shows a certain level of accuracy after the data size is 11 k.

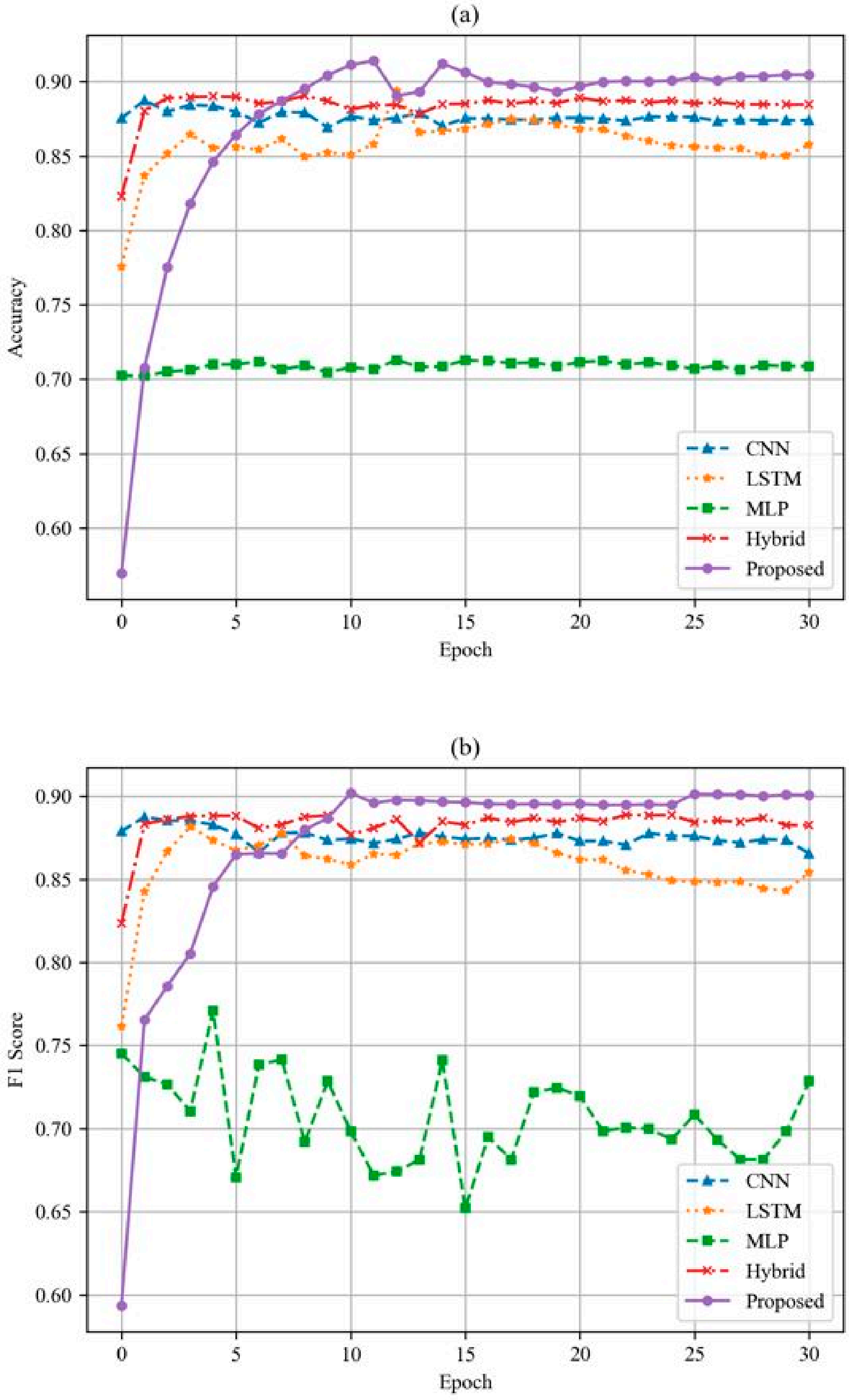

Figure 5. Model performance as the number epochs increases: (**a**) Accuracy; (**b**) F1 Score.

And the LSTM model shows stable accuracy after 10 k. In the case of the MLP model, it shows that the accuracy between 0.70 and 0.75 is maintained regardless of the data size. Next, in Figure 6b, the models other than MLP show a similar pattern to the accuracy graph, and in the case of MLP, the range of change is very large and it does not show a stable value.

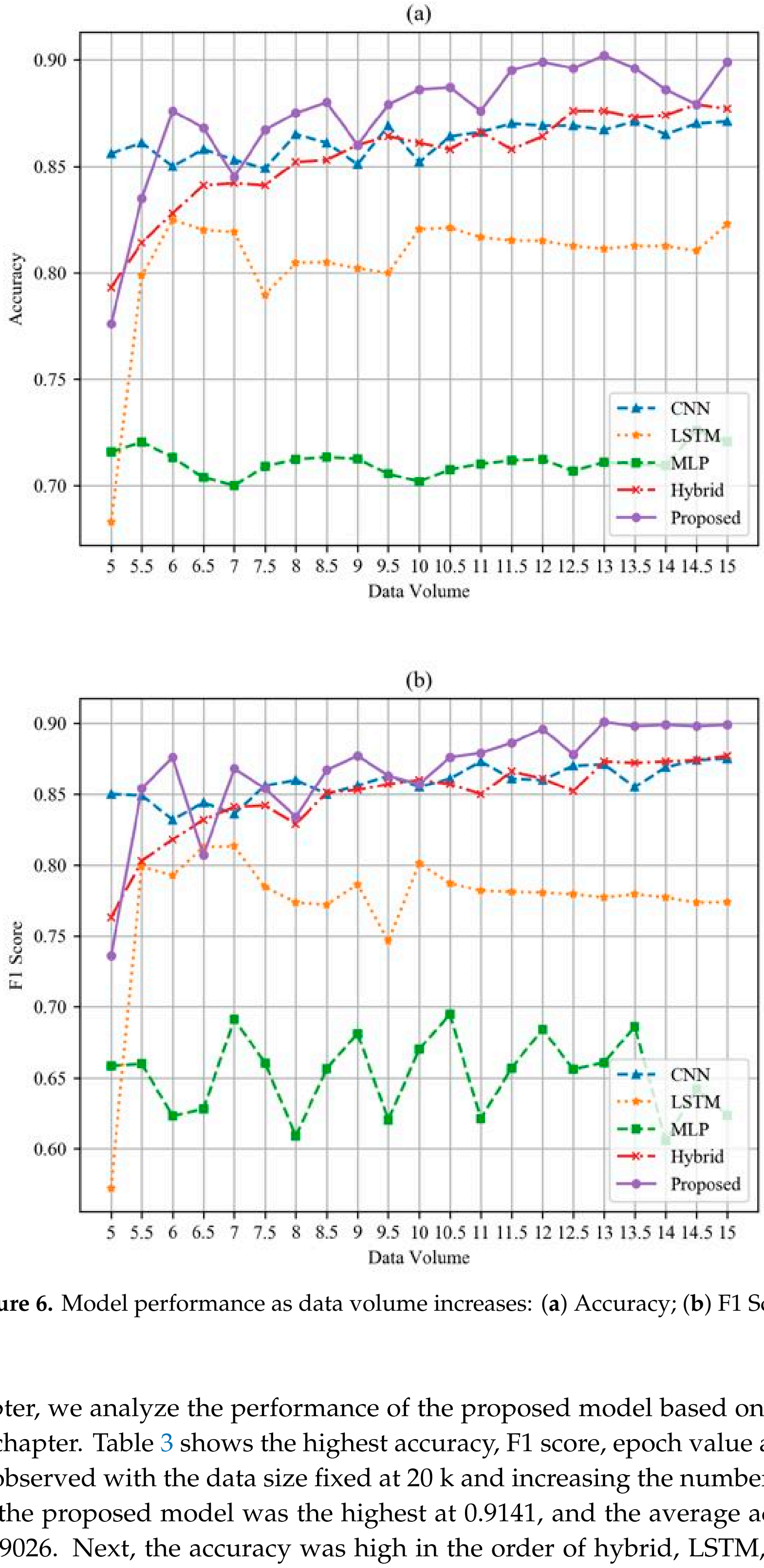

Figure 6. Model performance as data volume increases: (**a**) Accuracy; (**b**) F1 Score.

6. Analysis

In this chapter, we analyze the performance of the proposed model based on the experiments in the previous chapter. Table 3 shows the highest accuracy, F1 score, epoch value and training data for each model observed with the data size fixed at 20 k and increasing the number of epochs. First, the accuracy of the proposed model was the highest at 0.9141, and the average accuracy was also the highest at 0.9026. Next, the accuracy was high in the order of hybrid, LSTM, CNN, and MLP. The epoch value for obtaining the optimal accuracy of each model was 10 for cnn, 14 for LSTM, 5 for MLP, and 15 epochs for hybrid 5, which took the longest learning time. The F1 score also showed the best performance of the proposed model at 0.9018, followed by hybrid, CNN, LSTM, and MLP. The proposed model for F1 score also showed the best performance at 0.9018.

Table 3. Experimental results according to optimal accuracy, F1 score.

MODEL	CNN	LSTM	MLP	Hybrid	Proposed
Accuracy(max)	0.8874	0.8940	0.7129	0.8906	0.9141
Accuracy(avg)	0.8839	0.8579	0.7078	0.8763	0.9026
Data volume	20 k	20 k	20 k	20 k	20 k
Epoch	2	13	13	8	12
F1 Score	0.8875	0.8816	0.7708	0.8887	0.9018
Recall	0.8947	0.9504	0.8562	0.8919	0.9057
Precision	0.8818	0.8281	0.7081	0.8872	0.8975
Data volume	20 k	20 k	20 k	20 k	20 k
Epoch	2	4	5	23	10

Next, Table 4 shows the results of the experiment conducted while fixing the number of epochs and increasing the data size from 5 k to 15 k. When the data size was 13 k, the proposed model had the highest accuracy at 0.902, followed by the hybrid model at 0.879, CNN model showed the same accuracy as 0.871, LSMT model 0.825, and MLP model 0.726. In terms of average accuracy, the proposed model was the highest at 0.874, followed by CNN, hybrid, LSTM, and MLP. The F1 score also showed the highest performance at 0.901 when the data size was 13 k.

Table 4. Experimental results according to optimal data volume.

MODEL	CNN	LSTM	MLP	Hybrid	Proposed
Accuracy(max)	0.871	0.825	0.726	0.879	0.902
Accuracy(avg)	0.862	0.805	0.711	0.855	0.874
Data volume	9.5 k	6.5 k	14 k	12.5 k	13 k
Epoch	10	10	10	10	10
F1 Score	0.875	0.813	0.694	0.877	0.901
Recall	0.887	0.799	0.890	0.886	0.905
Precision	0.868	0.830	0.696	0.874	0.901
Data volume	11 k	7 k	7.5 k	12.5 k	13 k
Epoch	10	10	10	10	10

The model proposed in this paper has demonstrated improved performance compared to the existing models, and the accuracy increases as the size of the data increases and the number of training increases. This approach addresses the data-loss and long-term dependency problems which affects the existing models especially when the data size becomes high. The only disadvantage of the current model is that it requires more training data and training time than the existing baselines. Even with this limitation, it can be effective in classification that requires a lot of training data, such as sentiment analysis, which is a representative field in text classification in recent years, and text classification for specialized fields (which require more inferences). In addition, in the case of MLP used in the experiment, the F1 score was not optimized, and it can be confirmed that the accuracy is significantly lower than that of other models. This is because the data set is a sentiment classification representing positives and negatives. It is judged that the learning has not been completed because the rules for classification are not clear. It is evident that the existing MLP for multi-classification is not suitable for sentiment analysis, which is a topic in recent text classification.

7. Future Works

Recently, text classification has been applied in specialized fields such as sentiment analysis and clinical [33–35], law [36,37], healthcare [38–40], and business marketing [41,42]. These fields require much more beyond simple sentiment classification. Hence as our future research, we aim to explore more on more representative extensions of sentiment classification to other analytic fields. We aim to obtain enough training data to apply to the model using other innovative approaches such as transfer learning. We also aim at increasing the classification labels for multi-class prediction. The current proposed model consists of a combination of existing models, so its limitations are clear, and to solve this problem, new techniques or designs of other architectures remain our priority in the future works.

8. Conclusions

In this study, we proposed a Bi-LSTM+CNN attention hybrid model for text classification. Evaluation experiments were performed using IMDB movie review data, and the proposed model exhibits higher performance than the existing CNN, LSTM, MLP, hybrid models. The proposed model achieved higher accuracy which increased as the size of training data and the number of training epochs increase. This can provide an alternative solution to the long-term dependency problem in existing models and the data-loss problem that occurs as the size of training data increases. Recently, the field of text classification research has focused on extracting accurate semantics and features from special fields (e.g., medical, engineering, and emotional) and areas requiring specialized knowledge, rather than simply increasing accuracy.

Author Contributions: Software, M.K. and B.J.; Validation, M.K. and G.H.; Writing—original draft, B.J., M.K. and G.H.; Writing—review & editing, B.J., S.-u.K. and J.W.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea Grant funded by the Korea Government under Grant NRF-2019R1F1A1058058.

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

References

1. Zhang, L.; Wang, S.; Liu, B. Deep learning for sentiment analysis: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1253. [CrossRef]
2. Pouyanfar, S.; Sadiq, S.; Yan, Y.; Tian, H.; Tao, Y.; Reyes, M.P.; Shyu, M.-L.; Chen, S.-C.; Iyengar, S.S. A survey on deep learning: Algorithms, techniques, and applications. *ACM Comput. Surv. (CSUR)* **2018**, *51*, 1–36. [CrossRef]
3. Ikonomakis, M.; Kotsiantis, S.; Tampakas, V. Text Classification Using Machine Learning Techniques. *WSEAS Trans. Comput.* **2005**, *4*, 966–974.
4. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent convolutional neural networks for text classification. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
5. Zhang, Y.; Zheng, J.; Jiang, Y.; Huang, G.; Chen, R. A Text Sentiment Classification Modeling Method Based on Coordinated CNN-LSTM-Attention Model. *Chin. J. Electron.* **2019**, *28*, 120–126. [CrossRef]
6. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602–610. [CrossRef] [PubMed]
7. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
8. Liu, G.; Guo, J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing* **2019**, *337*, 325–338. [CrossRef]
9. Liu, H. Sentiment analysis of citations using word2vec. *arXiv* **2017**, arXiv:1704.00177.
10. Zhang, D.; Xu, H.; Su, Z.; Xu, Y. Chinese comments sentiment classification based on word2vec and SVMperf. *Expert Syst. Appl.* **2015**, *42*, 1857–1863. [CrossRef]
11. Peng, H.; Song, Y.; Roth, D. Event Detection and Co-reference with Minimal Supervision. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (ACL), Austin, TX, USA, 1–5 November 2016; pp. 392–402.
12. 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—SIGIR '15, Association for Computing Machinery (ACM), Santiago, Chile, 9–13 August 2015; pp. 959–962.

13. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A convolutional neural network for modelling sentences. *arXiv* **2014**, arXiv:1404.2188.
14. Yin, W.; Kann, K.; Yu, M.; Schütze, H. Comparative study of CNN and RNN for natural language processing. *arXiv* **2017**, arXiv:1702.01923.
15. Liang, D.; Zhang, Y. AC-BLSTM: Asymmetric convolutional bidirectional LSTM networks for text classification. *arXiv* **2016**, arXiv:1611.01884.
16. Zhou, P.; Qi, Z.; Zheng, S.; Xu, J.; Bao, H.; Xu, B. Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. *arXiv* **2016**, arXiv:1611.06639.
17. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991.
18. Wang, S.; Huang, M.; Deng, Z. Densely Connected CNN with Multi-scale Feature Attention for Text Classification. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence {IJCAI-18}, International Joint Conferences on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 4468–4474.
19. Du, C.; Huang, L. Text Classification Research with Attention-based Recurrent Neural Networks. *Int. J. Comput. Commun. Control.* **2018**, *13*, 50–61. [CrossRef]
20. Gao, S.; Ramanathan, A.; Tourassi, G. Hierarchical Convolutional Attention Networks for Text Classification. In Proceedings of the Third Workshop on Representation Learning for NLP, Association for Computational Linguistics (ACL), Melbourne, Australia, 20 July 2018; pp. 11–23.
21. Melamud, O.; Goldberger, J.; Dagan, I.; Riezler, S.; Goldberg, Y. context2vec: Learning Generic Context Embedding with Bidirectional LSTM. In Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, Association for Computational Linguistics (ACL), Berlin, Germany, 11–12 August 2016; pp. 51–61.
22. Ceraj, T.; Kliman, I.; Kutnjak, M. *Redefining Cancer Treatment: Comparison of Word2vec Embeddings Using Deep BiLSTM Classification Model*; Text Analysis and Retrieval 2019 Course Project Reports; Faculty of Electrical Engineering and Computing, University of Zagreb: Zagreb, Croatia, July 2019.
23. Xiao, L.; Wang, G.; Zuo, Y. Research on Patent Text Classification Based on Word2Vec and LSTM. In Proceedings of the 2018 11th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 8–9 December 2018; Volume 1, pp. 71–74. [CrossRef]
24. Rehman, A.U.; Malik, A.K.; Raza, B.; Ali, W. A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis. *Multimed. Tools Appl.* **2019**, *78*, 26597–26613. [CrossRef]
25. Luan, Y.; Lin, S. Research on Text Classification Based on CNN and LSTM. In Proceedings of the 2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Institute of Electrical and Electronics Engineers (IEEE), Dalian, China, 29–31 March 2019; pp. 352–355.
26. Wang, J.; Yu, L.-C.; Lai, K.R.; Zhang, X. Tree-Structured Regional CNN-LSTM Model for Dimensional Sentiment Analysis. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *28*, 581–591. [CrossRef]
27. 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. [CrossRef]
28. Salur, M.U.; Aydin, I. A Novel Hybrid Deep Learning Model for Sentiment Classification. *IEEE Access* **2020**, *8*, 58080–58093. [CrossRef]
29. 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), Institute of Electrical and Electronics Engineers (IEEE), Chongqing, China, 12–14 October 2018; pp. 1675–1680.
30. Dong, Y.; Liu, P.; Zhu, Z.; Wang, Q.; Zhang, Q. A Fusion Model-Based Label Embedding and Self-Interaction Attention for Text Classification. *IEEE Access* **2020**, *8*, 30548–30559. [CrossRef]
31. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
32. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T.; Lapata, M.; Blunsom, P.; Koller, A. Bag of Tricks for Efficient Text Classification. *arXiv* **2016**, arXiv:1607.01759.
33. Jasmir, J.; Nurmaini, S.; Malik, R.F.; Abidin, D.Z. Text Classification of Cancer Clinical Trials Documents Using Deep Neural Network and Fine Grained Document Clustering. In Proceedings of the Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019), Palembang, Indonesia, 16 November 2019; Atlantis Press: Paris, France, 2020; pp. 396–404.

34. Schmaltz, A.; Beam, A. Exemplar Auditing for Multi-Label Biomedical Text Classification. *arXiv* **2020**, arXiv:2004.03093.
35. Wang, Y.-B.; You, Z.-H.; Yang, S.; Li, X.; Jiang, T.; Zhou, X. A High Efficient Biological Language Model for Predicting Protein–Protein Interactions. *Cells* **2019**, *8*, 122. [CrossRef] [PubMed]
36. Bergman, P.; Berman, S.J. *Represent Yourself in Court: How to Prepare & Try a Winning Case*; Nolo: Berkley, CA, USA, 2016.
37. Li, P.; Zhao, F.; Li, Y.; Zhu, Z. Law text classification using semi-supervised convolutional neural networks. In Proceedings of the 2018 Chinese Control and Decision Conference (CCDC), Institute of Electrical and Electronics Engineers (IEEE), Shenyang, China, 9–11 June 2018; pp. 309–313.
38. Zhang, J.; Kowsari, K.; Harrison, J.H.; Lobo, J.M.; Barnes, L.E. Patient2Vec: A Personalized Interpretable Deep Representation of the Longitudinal Electronic Health Record. *IEEE Access* **2018**, *6*, 65333–65346. [CrossRef]
39. Srivastava, S.K.; Singh, S.K.; Suri, J.S. A healthcare text classification system and its performance evaluation: A source of better intelligence by characterizing healthcare text. In *Cognitive Informatics, Computer Modelling, and Cognitive Science*; Elsevier BV: Amsterdam, The Netherlands, 2020; pp. 319–369.
40. Seguí, F.L.; Aguilar, R.A.E.; De Maeztu, G.; García-Altés, A.; Garcia-Cuyàs, F.; Walsh, S.; Castro, M.S.; Vidal-Alaball, J. Teleconsultations between Patients and Healthcare Professionals in Primary Care in Catalonia: The Evaluation of Text Classification Algorithms Using Supervised Machine Learning. *Int. J. Environ. Res. Public Health* **2020**, *17*, 1093. [CrossRef] [PubMed]
41. Kang, M.; Ahn, J.; Lee, K. Opinion mining using ensemble text hidden Markov models for text classification. *Expert Syst. Appl.* **2018**, *94*, 218–227. [CrossRef]
42. Loureiro, S.M.; Guerreiro, J.; Eloy, S.; Langaro, D.; Panchapakesan, P. Understanding the use of Virtual Reality in Marketing: A text mining-based review. *J. Bus. Res.* **2019**, *100*, 514–530. [CrossRef]