applied sciences

MDPI

Article

CWSXLNet: A Sentiment Analysis Model Based on Chinese Word Segmentation Information Enhancement

Shiqian Guo [1], Yansun Huang [2], Baohua Huang [1,*], Linda Yang [1] and Cong Zhou [1]

[1] School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China
[2] Auditing Bureau of Xixiangtang, Nanning 530001, China
[*] Correspondence: bhhuang66@gxu.edu.cn; Tel.: +86-152-9654-4306

Abstract: This paper proposed a method for improving the XLNet model to address the shortcomings of segmentation algorithm for processing Chinese language, such as long sub-word lengths, long word lists and incomplete word list coverage. To address these issues, we proposed the CWSXLNet (Chinese Word Segmentation XLNet) model based on Chinese word segmentation information enhancement. The model first pre-processed Chinese pretrained text by Chinese word segmentation tool, and proposed a Chinese word segmentation attention mask mechanism by combining PLM (Permuted Language Model) and two-stream self-attention mechanism of XLNet. While performing natural language processing at word granularity, it can reduce the degree of masking between masked and non-masked words for two words belonging to the same word. For the Chinese sentiment analysis task, proposed the CWSXLNet-BiGRU-Attention model, which introduces bi-directional GRU as well as self-attention mechanism in the downstream task. Experiments show that CWSXLNet has achieved 89.91% precision, 91.53% recall rate and 90.71% F1-score, and CWSXLNet-BiGRU-Attention has achieved 92.61% precision, 93.19% recall rate and 92.90% F1-score on ChnSentiCorp dataset, which indicates that CWSXLNet has better performance than other models in Chinese sentiment analysis.

Keywords: sentiment analysis; Chinese word segmentation; XLNet; attention mask; machine learning; natural language processing

check for updates

Citation: Guo, S.; Huang, Y.; Huang, B.; Yang, L.; Zhou, C. CWSXLNet: A Sentiment Analysis Model Based on Chinese Word Segmentation Information Enhancement. *Appl. Sci.* **2023**, *13*, 4056. https://doi.org/10.3390/app13064056

Academic Editor: João M. F. Rodrigues

Received: 26 February 2023
Revised: 19 March 2023
Accepted: 21 March 2023
Published: 22 March 2023

1. Introduction

In the era of big data, the amount of information on the Internet has increased dramatically, including a large number of reviews posted by users on the Web. Most of these reviews express users' opinions and evaluations of products and services, which contain a lot of potential value [1]. The purpose of sentiment analysis techniques is to uncover the emotions and attitudes expressed in them, but due to the complex nature of comment data, which is diverse, colloquial and abbreviated, it is particularly important to use computational techniques to achieve automatic, in-depth and accurate analysis and processing [2].

The development of sentiment analysis has had a significant impact on the field of natural language processing. Natural language processing techniques and text analysis methods are used to mine text and extract sentiment polarity from it [3]. Sentiment analysis has a wide range of applications, such as reputation management, market research, customer service, brand monitoring, and so on.

Based on the above, sentiment analysis is important in various fields such as business, politics and society to help people better understand and respond to different emotions and attitudes in society. Sentiment analysis has developed through three main stages: sentiment lexicons, machine learning and deep learning [4].

Sentiment lexicon-based approaches: The earliest approaches to sentiment analysis were mainly based on sentiment lexicons, which typically contained a large number of words, each of which was tagged with a sentiment polarity such as positive, negative or neutral. The main use of sentiment lexicons in sentiment analysis is to automatically identify and classify the sentiment polarity of texts. The creation of a sentiment lexicon typically involves two aspects: word selection and sentiment annotation. For word selection, a large number of words are usually collected from different sources (e.g., network texts, human written texts, annotated datasets, etc.). The annotators have to annotate each vocabulary with a positive, negative or neutral sentiment polarity according to predefined sentiment classification criteria [5].

Several sentiment lexicons have been developed and are widely used in natural language processing. In the early years of research, Sebastiani Fabrizio et al. in [6–8] proposed the SentiWordNet sentiment lexicon, a WordNet-based sentiment lexicon that associates each word with a set of sentiment strengths, including positive sentiment, negative sentiment, and neutral sentiment. After a few years, Wu Xing et al. in [9], inspired by social cognitive theories, combined basic sentiment value lexicon and social evidence lexicon to improve the traditional polarity lexicon. In 2016, Wang Shih-Ming et al. in [10] presented the ANTU (Augmented NTU) Sentiment Dictionary, which was constructed by collecting sentiment statistics for words from several sentiment annotation exercises. A total of 26,021 Chinese words were collected in ANTUSD. In 2020, Yang Li et al. in [11] proposes a new sentiment analysis model SLCABG based on a sentiment lexicon that combines a convolutional neural network (CNN) with a bi-directional gated recurrent unit (BiGRU) based on an attention mechanism. The sentiment lexicon was used to enhance the sentiment features in the comments. CNN and BiGRU networks are used to extract the main sentiment and contextual features from the comments and weight them using the attention mechanism.

The advantage of sentiment dictionary-based methods is that they are simple and fast, but they require manual construction and updating of sentiment dictionaries, and they do not work well for some special texts (e.g., texts with complex semantics such as irony and metaphor).

Machine learning based approaches: With the development of machine learning techniques, models such as RNN, LSTM, CRF and GRU are gradually being proposed by researchers and people are using machine learning algorithms for text sentiment analysis.

LSTM (Long Short-Term Memory) [12] is a recurrent neural network (RNN) model commonly used to process sequential data. LSTM can effectively solve the long-term dependency problem in RNN by introducing a special memory unit. The BiLSTM model can be thought of as processing the input sequence from left to right for one LSTM model and from right to left for another LSTM model, and finally merging their outputs. The advantage of this is that not only the previous information but also the subsequent information can be considered when processing the input at the current time step. Xiao Zheng et al. in [13] used a bidirectional LSTM (BiLSTM) model for sentiment analysis. The experimental results show that BiLSTM outperforms CRF and LSTM for Chinese sentiment analysis. Similarly, Gan Chenquan et al. in [14] proposed a scalable multi-channel extended CNN-BiLSTM model with attention mechanism for Chinese text sentiment analysis, in which the convolutional model CNN based on bridging the BiLSTM model and achieved better result on several public Chinese sentiment analysis datasets.

In addition to LSTM, some researchers select GRU (Gated Recurrent Unit) to handle sentiment analysis tasks. As a variant of LSTM, GRU has fewer parameters than LSTM, requires less training data and has a faster training speed. Miao YaLin et al. in [15] proposed the adoption of the application of CNN-BiGRU model in Chinese short text sentiment analysis, which introduced the BiGRU model based on CNN. Zhang Binlong et al. in [16] proposed Transformer-Encoder-GRU (T-E-GRU) to solve the problem of transformer being naturally insufficient compared to the recurrent model in capturing the sequence features in the text through positional encoding. Both have achieved good experimental results in the field of Chinese sentiment analysis.

To leverage the affective dependencies of the sentence, in 2020, Liang Bin et al. in [17] proposed a graph convolutional network based on SenticNet [18] according to the specific aspect, called Sentic GCN, and explored a novel solution to construct the graph neural networks via integrating the affective knowledge from SenticNet to enhance the dependency graphs of sentences. Experimental results illustrate that SenticNet can beat state-of-the-art methods. In the same year, Jain Deepak Kumar et al. in [19] proposed BBSO-FCM model for sentiment analysis, used Binary Brain Storm Optimization (BBSO) algorithm for the Feature Selection process and thereby achieved improved classification performance, and Fuzzy Cognitive Maps (FCMs) were used as a classifier to classify the incidence of positive or negative sentiments. Experimental values highlight the improved performance of BBSO-FCM model in terms of different measures. In 2021, Sitaula Chiranjibi et al. in [20] proposed three different feature extraction methods and three different CNNs (Convolutional Neural Networks) to implement the features using a low resource dataset called NepCOV19Tweets, which contains COVID-19-related tweets in Nepali language. By using ensemble CNN, they ensemble the three CNNs models. Experimental results show that proposed feature extraction methods possess the discriminating characteristics for the sentiment classification, and the proposed CNN models impart robust and stable performance on the proposed features.

However, machine learning-based methods require large amounts of annotated data to train the classifier, and require manual selection of features and algorithms. There is a degree of subjectivity in feature extraction and algorithm selection, with good or bad feature extraction directly affecting classification results [21] and not easily generalized to a new corpus.

Deep learning-based approaches: In recent years, the rise of deep learning techniques has brought new breakthroughs in text sentiment analysis. In particular, the use of pre-trained language models (e.g., BERT, RoBERTa, XLNet, etc.) for fine-tuning to solve sentiment analysis problems has yielded very good results. This approach does not require manual feature construction and can handle complex semantic relationships, and therefore has very promising applications in the field of text sentiment analysis.

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model based on the transformer structure proposed by Google in 2018, and is currently one of the most representative and influential models in the field of natural language processing [22]. Due to the excellent performance of the BERT model, various variants derived from it are also widely used in the field of natural language processing, such as RoBERTa [23], ALBERT [24], ELECTRA [25], etc. The emergence of the BERT model has greatly promoted the development of the field of natural language processing and has achieved leading scores in several benchmark tests, becoming an important milestone in the field of natural language processing. Li Mingzheng et al. in [26] proposed a novel sentiment analysis model for Chinese stock reviews based on BERT. This model relies on a pre-trained model to improve the classification accuracy. The model uses a BERT pre-training language model to perform sentence-level representation of stock reviews, and then feeds the obtained feature vector into the classifier layer for classification. In the experiments, we demonstrate that our method has higher precision, recall and F1 than TextCNN, TextRNN, Att-BLSTM and TextCRNN. Our model can achieve the best results, which indicates its effectiveness in Chinese stock review sentiment analysis. Meanwhile, our model has strong generalization ability and can perform sentiment analysis in many fields.

In 2019, Google proposed XLNet [27], which uses the Permuted Language Model (PLM) with a two-stream self-attention mechanism to outperform the BERT model in 20 natural language processing tasks, achieving the best results in 18 tasks. Currently, XLNet is widely used in natural language processing, covering tasks such as classification and named entity recognition [28,29].

As part of text classification, sentiment analysis was also an important application of XLNet. Gong Xin-Rong et al. in [30] proposed a Broad Autoregressive Language Model (BroXLNet) to automatically process the sentiment analysis task. BroXLNet integrates the advantage of generalized autoregressive language modeling and broad learning system, which has the ability of extracting deep contextual features and randomly searching highlevel contextual representation in broad spaces. BroXLNet achieved the best result of 94.0% in sentiment analysis task of binary Stanford Sentiment Treebank.

XLNet was trained on different languages. Alduailej Alhanouf et al. in [31] proposed AraXLNet model, which pre-trained XLNet model in Arabic language for sentiment analysis. For Chinese language, Cui Yiming et al. in [32] published an unofficial XLNet Chinese pre-training model, which was trained from the Chinese Wikipedia corpus, but its word segmentation model still suffers from the defects of excessively long word segmentation length, infrequent use of word segmentation, and incomplete coverage of the word list.

To address the above problems, this paper proposes the CWSXLNet (Chinese Word Segmentation XLNet) model, which improves the XLNet model. First, the original corpus is segmented in the text pre-processing stage and the corresponding segmentation codes are generated; in the pre-training stage, the corresponding segmentation mask codes are generated according to the random sequence of PLM. Combined with the two-stream self-attention mechanism and the attention mask with the segmentation mask codes, thus realising the improvement of Chinese sub-word location information while using the single Chinese character as the granularity. It is designed to solve the problem of the XLNet model for Chinese language processing in terms of character-to-word granularity.

For the Chinese sentiment analysis tasks, this paper combines the above researches and uses the BiGRU model in the downstream task, which can further extract the feature information of the context. In addition, the CWSXLNet-BiGRU-Attention model is proposed by introducing the self-attention mechanism in the model to increase its attention to sentiment-weighted words. It can further capture the sentiment keywords in the text and achieve better results in Chinese sentiment analysis tasks.

2. Related Works

2.1. XLNet

XLNet is an autoregressive language model proposed by Google in 2019, inherited from Transformer-XL, using the PLM, two-stream self-attention mechanism and the segment-level recurrence with relative position encoding in Transformer-XL, so that XLNet has better long text reading ability.

2.1.1. PLM

Traditional autoregressive language models are trained in a unidirectional direction; they can only predict based on the antecedent information of the predicted words, or from backwards to forwards through the posterior text of the predicted words. PLM is a model that generates several sequences of text in random order and masks the words at the end of the sequence as predictors, assuming that the semantic information of all the preceding words is available at the end of the sequence. PLM assumes that the end words of the sequence can contain the semantic information of all the preceding words, and so makes the prediction and finally completes the training of the model. This approach solves the problem of the possible relationship between multiple masks being ignored in the BERT model, and of small differences occurring in the pre-training and fine-tuning phases due to the use of masks.

2.1.2. Dual Stream Self-Attention Mechanism

To solve the problem of confusing the location information caused by rearranging the input information in PLM, XLNet introduces a two-stream self-attentive mechanism that divides attention into a content stream and a query stream, where the content stream can see the current predicted word and retains the semantic information of the current word; the query stream cannot see the current word and retains the location information of the current word. When making predictions, the content stream can see the current word itself, whereas the query stream cannot see the content of the current word. This is how prediction of disordered text is achieved.

2.2. Pre-Training Process of XLNet Model

The pre-training process of XLNet mainly consists of two stages: text pre-processing stage and pre-training stage, and the detailed flow chart is shown in Figure 1.

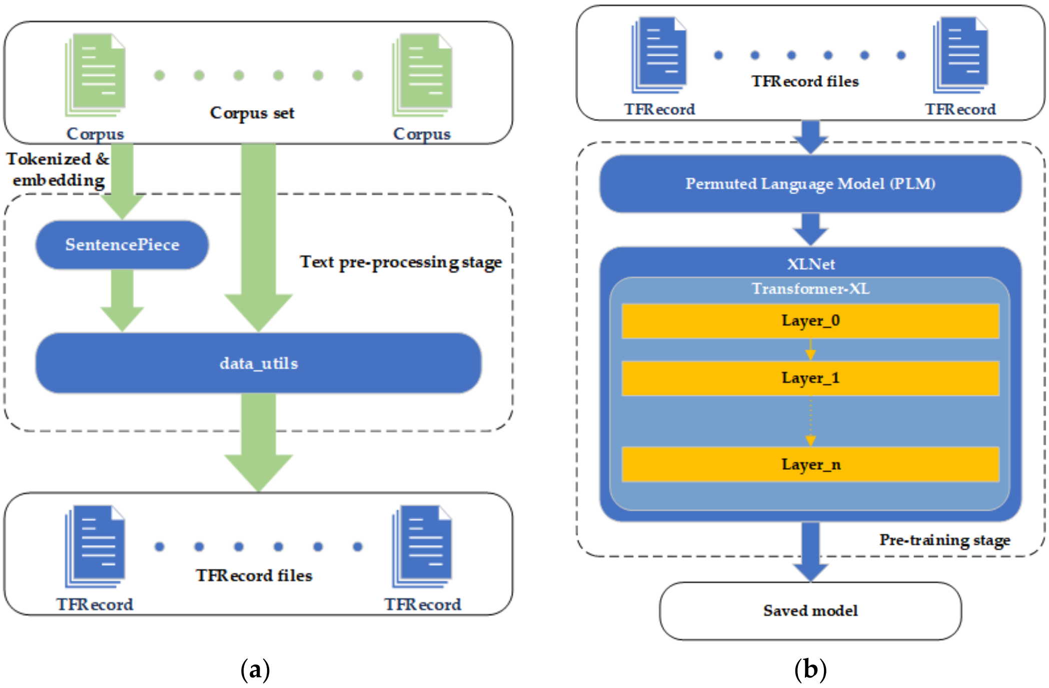


Figure 1. XLNet training process. (**a**) The text pre-processing process; (**b**) The pre-training process.

2.2.1. Text Pre-Processing

XLNet model uses SentencePiece [33], a tokenizer provided by Google. The SentencePiece model is trained on the original text. It uses the BPE algorithm [34] to separate words and statistics from the original text, obtaining the word separation strategy by continuously merging the more frequent sub-words.

The trained SentencePiece is used to split the original text into vectors and complete the word embedding. After the transformation, merging, splitting and disrupting steps, each piece of training data is transformed into a Feature containing input, tgt, is_masked, seg_id and label and so on. The structure of input is shown in Figure 2 (the maximum length of the sentence is assumed to be 128).

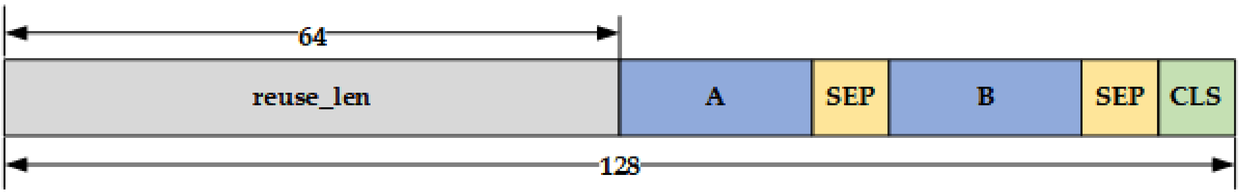


Figure 2. Structure of "input".

The first half of reuse_len is the reuse of the last 64 tokens of the previous input data, and the second half is the structure "A + <SEP> + B + <SEP> + <CLS>", where A and B are two vectors with a <SEP> control character at the end of A and B, respectively, as a statement separator A and B are two text vectors, and a <CLS> control character at the end of the entire data as the end of the entire input data.

Vector A and vector B have a 50% probability of being consecutive contexts and a further 50% probability of being randomly chosen vector B. The lengths of A and B are not fixed, but the length of "A + <SEP> + B + <SEP> + <CLS>" is equal to the maximum length of the model sentence.

Together with the input, tgt, is_masked, seg_id and label are generated, which together form a Feature, and Features together form TFRecord files.

Where tgt is the target vector of input, with a length of 128 tokens, the first 126 tokens are the next token corresponding to input, which is equivalent to shifting the whole of input to the left by one token length, and the last two tokens are <CLS>.

Is_masked indicates which of the 128 tokens in input are masked, assigning 0 to the unmasked and 1 to the masked.

Seg_id is used to distinguish vector A from vector B, where reuse + A + <SEP> is assigned 0, B + <SEP> is assigned 1 and <CLS> is assigned 2.

Label is used to distinguish whether vector A is continuous with vector B.

2.2.2. Pre-Training

In the pre-training phase, the XLNet model reads the TFRecords data generated in the pre-processing phase, performs random reordering on each data, and finally generates the perm_mask matrix corresponding to each data, as shown in Figure 3. Where perm_mask[i][j] = 1 (hollow circle) indicates that the ith token cannot detect the jth token after reordering; conversely, perm_mask[i][j] = 0 (solid circle) indicates that the ith token can detect the jth token.

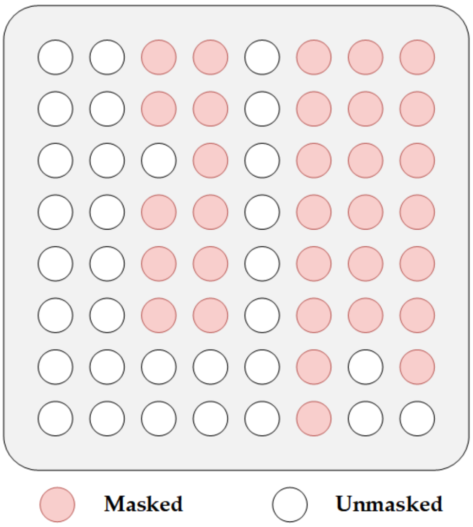


Figure 3. An example of perm_mask matrix.

In the subsequent pre-training process, the perm_mask matrix is transformed into attn_mask matrix after splitting, splicing and deformation operations, and is used in the calculation of attn_score with the formula as in Equation (1).

$$\text{attn\_score} = \text{attn\_score} - 10^{30} \times \text{attn\_mask}. \tag{1}$$

If the element in attn_mask[i][j] is 0, it means that i can notice j, and attn_score[i][j] remains unchanged at that time.

If the element in attn_mask[i][j] is 1, it means that i cannot notice j, and attn_score[i][j] becomes a large negative number, resulting in the probability of the next softmax operation being close to 0 to achieve the effect of the mask.

2.3. BiGRU

The Gated Recurrent Unit (GRU) is a type of recurrent neural network. Compared to LSTM, GRU streamlines the number of gates from three gates (forgetting gate, input gate and output gate) to two gates (reset gate and update gate), and merges cell state and hidden state. In addition, GRU has fewer parameters than LSTM, requires less training data and has a faster training speed.

The GRU architecture is shown in Figure 4.

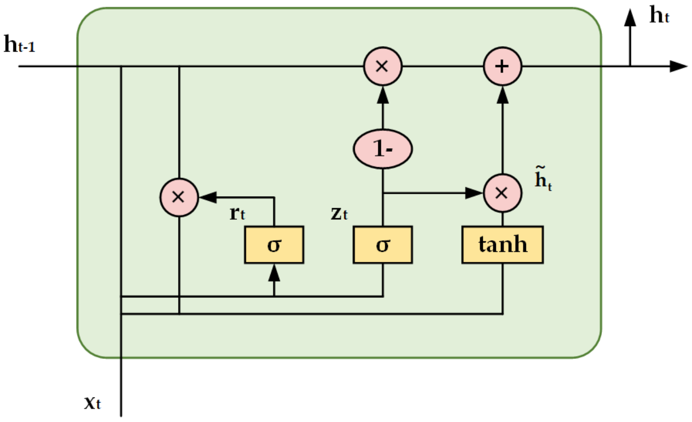


Figure 4. Structure of GRU unit.

The two doors of the GRU are calculated as follows:

Reset gate r_t:

$$r_t = \sigma(W^r x_t + U^r h_{t-1} + b_r). \tag{2}$$

Update gate z_t:

$$z_t = \sigma(W^z x_t + U^z h_{t-1} + b_z). \tag{3}$$

The formula for the calculation of the candidate hidden layers is

$$\widetilde{h_t} = tanh(Wx_t + U(r_t \odot h_{t-1}) + b). \tag{4}$$

Finally, the hidden layer information at time t is calculated h_t

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h_t}. \tag{5}$$

In the above equation, σ is the sigmoid activation function and $\odot$ is the Hadamard product operation of the matrix. The Hadamard product of matrix A and matrix B is denoted as A⊙B. For matrix A = [a_{ij}] and matrix B = [b_{ij}], the elements of matrix A⊙B are defined as the product of the corresponding elements of the two matrices, i.e., $(A\odot B)_{ij} = a_{ij}b_{ij}$.

To capture the contextual information of the text, BiGRU, a bidirectional GRU network, can be used, which consists of a forward GRU and a reverse GRU, as shown in Figure 5. The forward GRU provides the above information of the text and the reverse GRU provides the below information of the text to capture the contextual information.

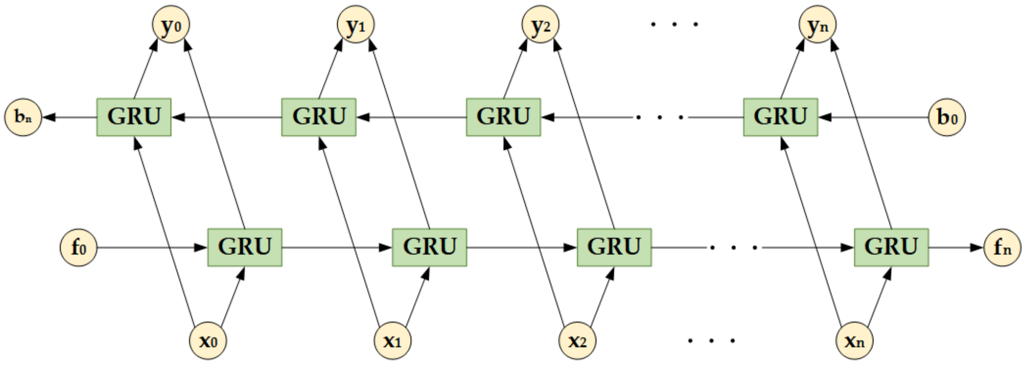


Figure 5. Network structure of BiGRU.

2.4. Self-Attention

Self-attention is able to notice the interconnectedness of words within the input utterance, giving the model a stronger ability to grasp emotionally weighted words. Its matrix form is computed as follows, first computing the three matrices Q, K and V:

$$Q = W^q \cdot I, \tag{6}$$

$$K = W^k \cdot I, \tag{7}$$

$$V = W^v \cdot I. \tag{8}$$

W^q, W^k, W^v are trainable parameter matrices and is a matrix composed of word vectors.

After calculating Q, K and V, we obtain A:

$$A = K^T \cdot Q. \tag{9}$$

A' is obtained by softmax normalization of A:

$$A' = softmax\left(\frac{A}{\sqrt{d_k}}\right). \tag{10}$$

The final output is calculated as O:

$$O = V \cdot A'. \tag{11}$$

Combining each of the above steps obtains the formula for self-attention:

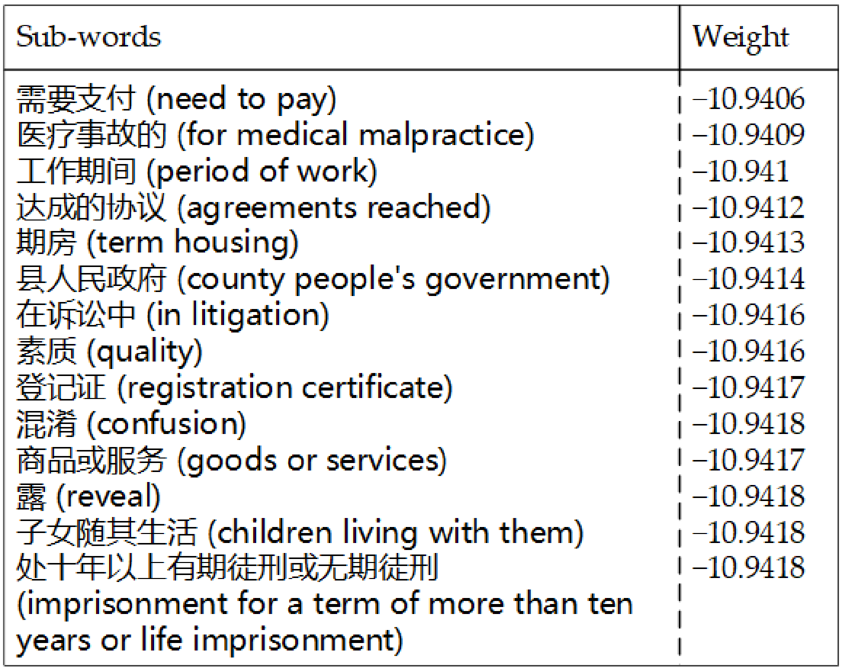
$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V. \tag{12}$$

3. CWSXLNet

3.1. Deficiencies of XLNet Model in Handling Chinese

When the XLNet model is used to process Chinese, because it uses SentencePiece for text segmentation, and SentencePiece tends to split longer sub-words when using the BPE method for word segmentation, these longer words, when separated from the original corpus, tend to be used less frequently in other text, causing the waste of the limited positional space of the word table. Furthermore, because SentencePiece tends to segment longer words rather than individual Chinese character, the required word list is unusually large. With a limited word list length of 32,000, there are still many commonly used Chinese words not included in the word list, which can only be represented by <UNK> during word embedding, affecting the model's understanding of semantics and sentiment analysis.

Figure 6 shows the sub-words of the word list in [31], which proposed Chinese pre-trained XLNet model. The numbers in Figure 6 indicate the weights. It can be seen that SentencePiece prefers to split out longer sub-words, which are less frequently used in the detached pre-trained corpus and waste space in the word list.



Sub-words	Weight
需要支付 (need to pay)	−10.9406
医疗事故的 (for medical malpractice)	−10.9409
工作期间 (period of work)	−10.941
达成的协议 (agreements reached)	−10.9412
期房 (term housing)	−10.9413
县人民政府 (county people's government)	−10.9414
在诉讼中 (in litigation)	−10.9416
素质 (quality)	−10.9416
登记证 (registration certificate)	−10.9417
混淆 (confusion)	−10.9418
商品或服务 (goods or services)	−10.9417
露 (reveal)	−10.9418
子女随其生活 (children living with them)	−10.9418
处十年以上有期徒刑或无期徒刑 (imprisonment for a term of more than ten years or life imprisonment)	−10.9418

Figure 6. Word lists and weights in Chinese XLNet.

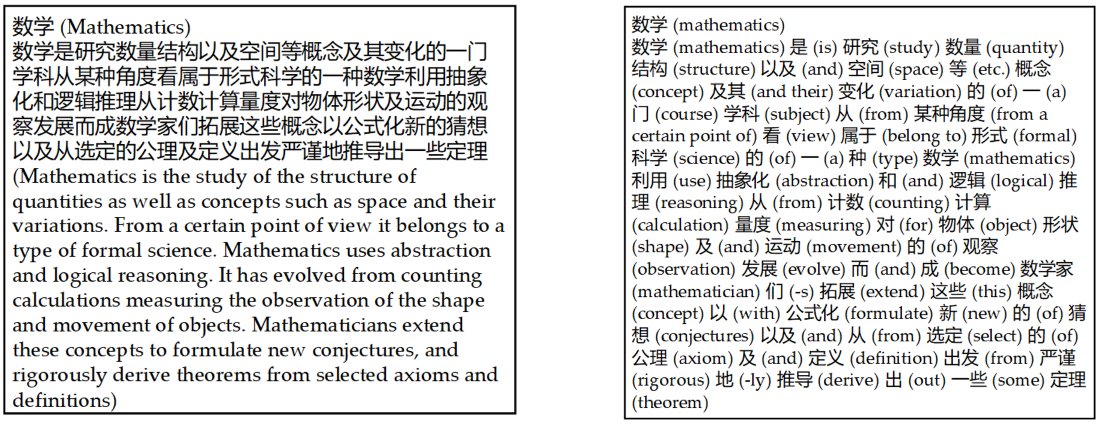
The basic unit of the Chinese language is the Chinese character. Several Chinese characters make up words. Several Chinese words form sentences. Unlike English, Chinese words are not separated by spaces, and there are approximately 55,000 commonly used Chinese words. For comparison, the length of XLNet's word list is 32,000. To avoid problems caused by long word lists, we reduce the granularity to the Chinese character instead of the Chinese word. However, this approach creates another problem: the relationship between characters from the same word is lost. To avoid this, we discovered a way to improve the connection between characters from the same word.

First, we want to use a tokenized tool to separate Chinese words from sentences, and use spaces to separate Chinese words like in English. Secondly, we reduce the granularity to the Chinese character. Finally, we need to improve the relationship between characters that form the same word, and make sure that the improvement fits perfectly to the XLNet model. This is a brief introduction to our CWSXLNet model; we explain the model in more detail below.

3.2. CWSXLNet Model

In this paper, we propose the CWSXLNet model, which aims to solve the natural non-adaptation problem of the SentencePiece model used in the XLNet model for Chinese. Improvements are made in the data pre-processing phase and the pre-training phase of XLNet.

In the data pre-processing stage, the model used the LTP [35] as a word separation tool to separate the original corpus with spaces between words. The following Figure 7 shows an example of the text after word separation using the LTP word separation tool.



数学 (Mathematics)
数学是研究数量结构以及空间等概念及其变化的一门学科从某种角度看属于形式科学的一种数学利用抽象化和逻辑推理从计数计算量度对物体形状及运动的观察发展而成数学家们拓展这些概念以公式化新的猜想以及从选定的公理及定义出发严谨地推导出一些定理
(Mathematics is the study of the structure of quantities as well as concepts such as space and their variations. From a certain point of view it belongs to a type of formal science. Mathematics uses abstraction and logical reasoning. It has evolved from counting calculations measuring the observation of the shape and movement of objects. Mathematicians extend these concepts to formulate new conjectures, and rigorously derive theorems from selected axioms and definitions)

(**a**)

数学 (mathematics)
数学 (mathematics) 是 (is) 研究 (study) 数量 (quantity) 结构 (structure) 以及 (and) 空间 (space) 等 (etc.) 概念 (concept) 及其 (and their) 变化 (variation) 的 (of) 一 (a) 门 (course) 学科 (subject) 从 (from) 某种角度 (from a certain point of) 看 (view) 属于 (belong to) 形式 (formal) 科学 (science) 的 (of) 一 (a) 种 (type) 数学 (mathematics) 利用 (use) 抽象化 (abstraction) 和 (and) 逻辑 (logical) 推理 (reasoning) 从 (from) 计数 (counting) 计算 (calculation) 量度 (measuring) 对 (for) 物体 (object) 形状 (shape) 及 (and) 运动 (movement) 的 (of) 观察 (observation) 发展 (evolve) 而 (and) 成 (become) 数学家 (mathematician) 们 (-s) 拓展 (extend) 这些 (this) 概念 (concept) 以 (with) 公式化 (formulate) 新 (new) 的 (of) 猜想 (conjectures) 以及 (and) 从 (from) 选定 (select) 的 (of) 公理 (axiom) 及 (and) 定义 (definition) 出发 (from) 严谨 (rigorous) 地 (-ly) 推导 (derive) 出 (out) 一些 (some) 定理 (theorem)

(**b**)

Figure 7. Original text after using LTP. (**a**) The original text; (**b**) After using LTP.

In order to reduce the training granularity to words while retaining the word separation information in the original text, CWSXLNet trains the SentencePiece model at the granularity of a single Chinese character. First, a total of 14,516 Chinese characters in the dictionary book are crawled using a crawler tool. These are fed into the SentencePiece model as training text for character-based partitioning training. Finally, the SentencePiece model is trained to segment Chinese texts as single characters.

The original corpus is fed into the SentencePiece model, which is trained as described above. In the subsequent training phase, the proposed model also uses the character "◎" as a word separation marker and refers to the character "◎" as a <TOK> marker, which is used to detect the boundary between words.

When the original text is processed to generate Features, the <TOK> token is used as a word delimiter to determine the position of each word. The tok_id vector of each data is generated to determine which word of the data the current character belongs to, where the tok_id of the <TOK> token is 0 and the tok_id of the control characters <SEP> and <CLS> is −1; the is_TOK vector of the data is generated at the same time, where the position of the TOK token is true and the rest position is false, which is used to determine whether the current token is a TOK token or not. The above two pieces of data are stored with input, tgt, label, seg_id and is_masked in the TFRecord files.

Figure 8 shows the process of generating the corresponding tok_id of a text with is_TOK. For illustration purposes, the lengths of reuse_len and the vector B are set to 0. The original text is " 今天天气真不错 (It's really nice weather today)". The LTP separated the text to " 今天 (today)", " 天气 (weather)", " 真 (really)", " 不错 (nice)".

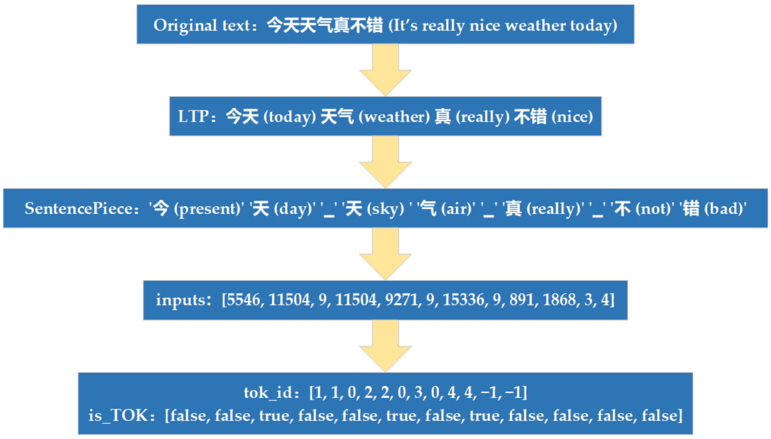

Original text：今天天气真不错 (It's really nice weather today)
LTP：今天 (today) 天气 (weather) 真 (really) 不错 (nice)
SentencePiece：'今 (present)' '天 (day)' '_' '天 (sky) ' '气 (air)' '_' '真 (really)' '_' '不 (not)' '错 (bad)'
inputs：[5546, 11504, 9, 11504, 9271, 9, 15336, 9, 891, 1868, 3, 4]
tok_id：[1, 1, 0, 2, 2, 0, 3, 0, 4, 4, −1, −1]
is_TOK：[false, false, true, false, false, true, false, true, false, false, false, false]

Figure 8. Generation of vector tok_id and is_TOK.

Inputs is the result of vectorizing the text by adding the <SEP> and <CLS> flags to the end of the original text, where 9 is <TOK>, 3 is <SEP> and 4 is <CLS>.

In the pre-training phase, the tok_mask matrix is computed using the same random sequence, while the random sequence is used to generate the perm_mask. Following the example in Figure 8, at this point, assuming the random sequence index = [4, 6, 7, 10, 3, 11, 0, 1, 8, 9, 2, 5], then the tok_index = [2, 3, 0, −1, 2, −1, 1, 1, 4, 4, 0, 0], which is how reordered version of tok_id by index is obtained.

Matrix A is obtained by transposing tok_index and broadcasting the columns, and matrix B is obtained by broadcasting the rows of tok_id. Elements in tok_mask matrix are set to 1 if Matrix A and B's corresponding elements are equal and 0 if they are not. The calculation flowchart is shown in Figure 9.

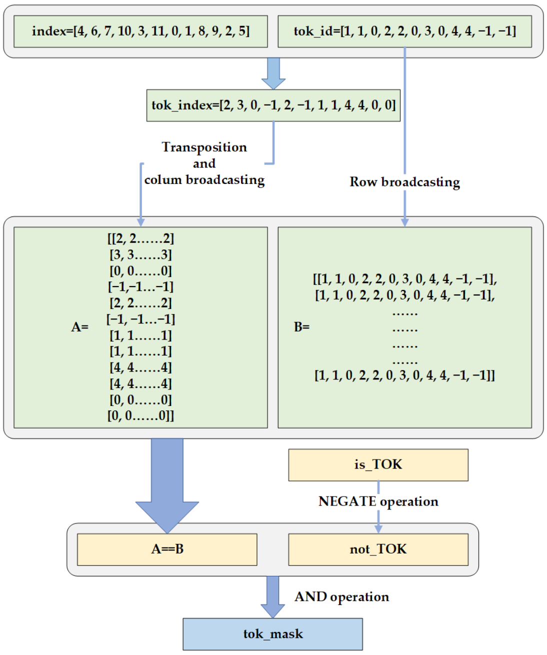


Figure 9. Generation of the tok_mask matrix.

The structure of the tok_mask matrix is shown in Figure 10. tok_mask[i][j] = 1 (diagonal circle) means that the ith token belongs to the same Chinese word as the jth token after disordering, and conversely tok_mask[i][j] = 0 (hollow circle) means that the ith token does not belong to the same Chinese word as the jth token.

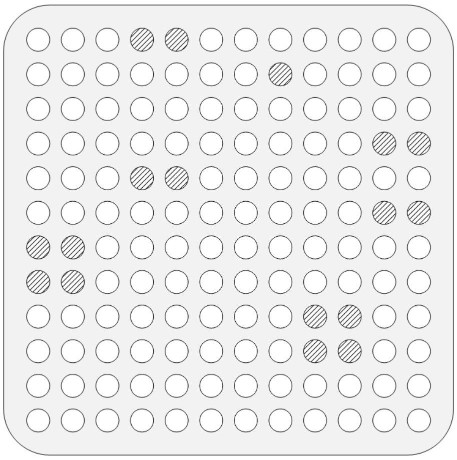


Figure 10. The tok_mask matrix.

After generating tok_mask, the following operations are performed on perm_mask:

$$perm\_mask = perm\_mask - tok\_mask \times \alpha, \tag{13}$$

$$perm\_mask = clip(perm\_mask, \alpha - 1, 1), \tag{14}$$

where α is the masking argument of tok_mask in the range (0,1), which is used to control the degree of "masking" of the mask, and clip is the clipping operation, which sets the range of each element in the perm_mask matrix from the original $(-\alpha, 1)$ to $(\alpha - 1, 1)$.

After the above operation, the degree of masking of some originally masked words in perm_mask is "reduced", while the degree of unmasking of some non-masked words is "enhanced" (as shown in Figure 11).

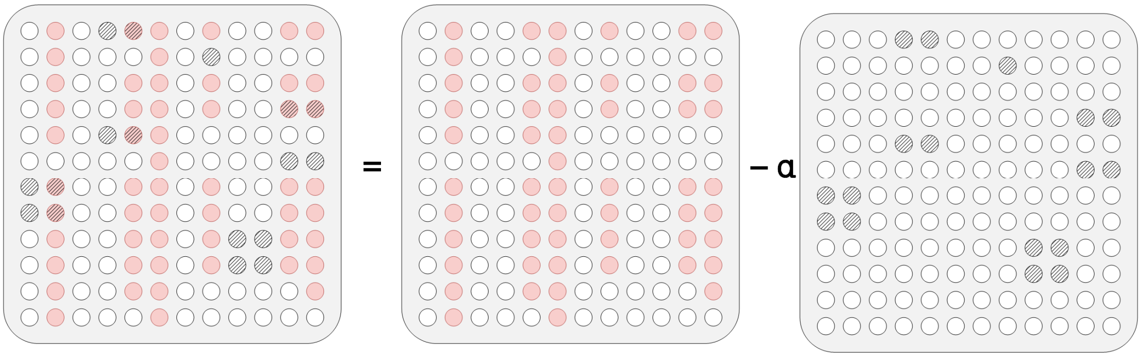


Figure 11. Masking of perm_mask.

This is reflected in Equation (1), which calculates the attn_score.

At this point, the element values in the attn_mask matrix can be one of the four cases 1, 0, $1 - \alpha$, $\alpha - 1$, as shown in Figure 12.

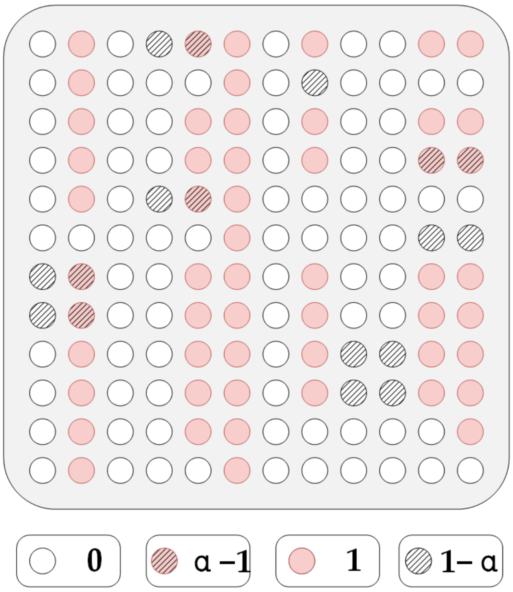


Figure 12. The attn_mask matrix, where 1 (solid circle) and 0 (hollow circle) represent masked and non-masked cases.

As we see in Figure 8, the LTP shows that Chinese word " 天气 (weather)" is separated by a space. It indicates that Chinese character " 天 (sky)" and " 气 (air)" belong to Chinese word " 天气 (weather)", and we hope to enhance the relationship between " 天 (sky)" to " 气 (air)", and, conversely, " 气 (air)" and " 天 (sky)". From the perm_mask matrix in the middle of Figure 10, we know that “ 天 (sky)” is masked; to enhance the relationship, we use the tok_mask matrix to reduce the degree of mask from “ 气 (air)” to “ 天 (sky)” and increase attention from “ 天 (sky)” to “ 气 (air)”.

If the element in attn_mask[i][j] is 1-α (solid diagonal circle), this means that originally i could not notice j, but since i and j belong to the same Chinese word, the attentional masking of j by i is “reduced” and the probability of attn_score[i][j] is reduced compared to the original one, which increases the probability of the subsequent softmax prediction. For example, the element in row 1 and column 5 in Figure 12 means that the first element “ 气 (air)” would not be able to notice the unordered fifth element “ 天 (sky)” after disordering, because “ 天 (sky)” is masked for “ 气 (air)”. However, since “ 天 (sky)” and “ 气 (air)” belong to the same Chinese word “ 天气 (weather)”, the extent of mask from “ 气 (air)” to “ 天 (sky)” is “reduced”. Since “ 气 (air)” is not masked for “ 天 (sky)”, the attention is increased from “ 气 (air)” to “ 天 (sky)”.

As we see in Figure 8, the LTP shows that Chinese word “ 不错 (nice)” is separated by space. It indicates that Chinese character “ 不 (not)” and “ 错 (bad)” belong to Chinese word “ 不错 (nice)”, and we hope to enhance the relationship between “ 不 (not)” and “ 错 (bad)”, and, conversely, “ 错 (bad)” and “ 不 (not)”. From the perm_mask matrix in the middle of Figure 10, we know that both “ 不 (not)” and “ 错 (bad)” are unmasked; to enhance the relationship, we use the tok_mask matrix to increase attention from “ 不 (not)” to “ 错 (bad)” and “ 错 (bad)” to “ 不 (not)”.

If attn_mask[i][j] is $\alpha-1$ (hollow diagonal circle), this means that originally i can notice j, but because i and j belong to the same phrase, i’s attention to j is “increased” compared to the original, attn_score[i][j] is increased, which also increases the probability of softmax prediction. For example, the elements in row 9 and column 10 in Figure 12 indicate that the ninth element “ 不 (not)” after disordering is able to detect the tenth element “ 错 (bad)”, which is not disordered. Since “ 不 (not)” and “ 错 (bad)” belong to the same Chinese word, we increase the attention from “ 不 (not)” to “ 错 (bad)” and “ 错 (bad)” to “ 不 (not)”.

In summary, we have reduced the granularity of Chinese natural language processing from Chinese sub-word to single Chinese character. To solve the problem of information loss by splitting Chinese words into characters, we proposed a method to enhance the relationship between characters from the same word. The embodiment of this approach in the XLNet model is the tok_mask matrix. To further enhance the sentiment analysis capability, we combined BIGRU and self-attention to form CWSXLNet. In Section 4, the experimental result shows that it definitely improves the ability of Chinese sentiment analysis.

4. Experiment

To demonstrate the effectiveness of the CWSXLNet model and the CWSXLNet-BiGRU-Attention structure proposed in this paper, experiments were conducted on two public Chinese sentiment analysis datasets.

4.1. Experimental Environment

The experimental code is developed based on the TensorFlow framework, and the cloud GPU server is used as the experimental runtime environment, and the experimental environment specifics are shown in Table 1.

Table 1. Experimental environment.

Name	Parameters
Operating System	Ubuntu 18.04
Memory	32 G
GPU	Tesla T4
GPU Memory	16 G

4.2. Parameters

The pre-training corpus was selected from the Chinese Wikipedia corpus, with a total size of about 2.5 G, and the tokenizer was selected from the LTP/base model. The details of the parameters in text pre-processing are shown in Table 2.

Table 2. Parameters in text pre-processing.

Name	Parameters	Description
batch_size	8	Batch size.
seq_len	512	Sequence length.
reuse_len	256	Number of tokens that can be reused as memory.
bi_data	True	Whether to create bidirectional data.
mask_alpha	6	The number of tokens to form a group.
mask_beta	1	The number of tokens to mask within each group.
num_predict	85	The number of tokens to predict.

The details of the parameters in pre-training are shown in Table 3.

Table 3. Parameters in pre-training.

Name	Parameters	Description
n_layer	6	Number of layers.
d_model	768	Dimension of the model.
d_embed	768	Dimension of the embedding.
n_head	12	Number of attention heads.
d_head	64	Dimension of each attention head.
d_inner	3072	Dimension of inner hidden size in position-wise feed-forward.
weight_decay	0.01	Weight decay rate.
adam_epsilon	1×10^{-6}	Adam epsilon.
learning_rate	2.5×10^{-5}	Maximum learning rate.
dropout	0.1	Dropout rate.
dropatt	0.1	Attention dropout rate.
train_steps	100k	Total number of training steps.
ff_activation	gelu	Activation type used in position-wise feed-forward.
mask_arg	1×10^{-34}	Mask argument α in Equation (13).

The details of the parameters in fine tune are shown in Table 4.

Table 4. Parameters in fine tune.

Name	Parameters	Description
Learning rate	2×10^{-5}	Maximum learning rate.
GRU_units	128	Number of GRU units.
GRU_dropout	0.5	GRU dropout rate.
dropout	0.3	Dropout rate.

4.3. Data Sets

4.3.1. ChnSentiCorp Dataset

The dataset is the hotel accommodation reviews collected by Songbo Tan, which is the dichotomous sentiment analysis dataset.

4.3.2. Weibo_senti_100k Dataset

The dichotomous dataset obtained from Sina Weibo comment data has 119,988 pieces of data, of which there are 59,993 positive samples and 59,995 negative samples.

4.4. Evaluation Indicators

The evaluation index consists of the precision rate P, the recall rate R and the F1-score, which are calculated as follows:

$$P = \frac{TP}{TP + FP} \tag{15}$$

$$R = \frac{TP}{TP + FN} \tag{16}$$

$$F_1 = \frac{2RP}{R + P} \tag{17}$$

where TP is the number of positive samples considered positive by the model, FP is the number of negative samples considered positive by the model, and FN is the number of positive samples considered negative by the model.

5. Results and Discussion

The experimental results of the ChnSentiCorp dataset are shown in the following Table 5 and Figure 13, and the LTP process was performed on the training corpus to maintain consistency between pre-training and fine-tuning.

Table 5. ChnSentiCorp dataset experimental results.

Model	P (%)	R (%)	F1-Score (%)
LSTM	84.25	84.01	84.12
BiLSTM	86.94	86.50	86.72
BERT	89.92	89.90	89.91
XLNet	88.83	87.92	88.37
CWSXLNet	89.91	91.53	90.71
CWSXLNet-BiGRU-Attention	92.61	93.19	92.90

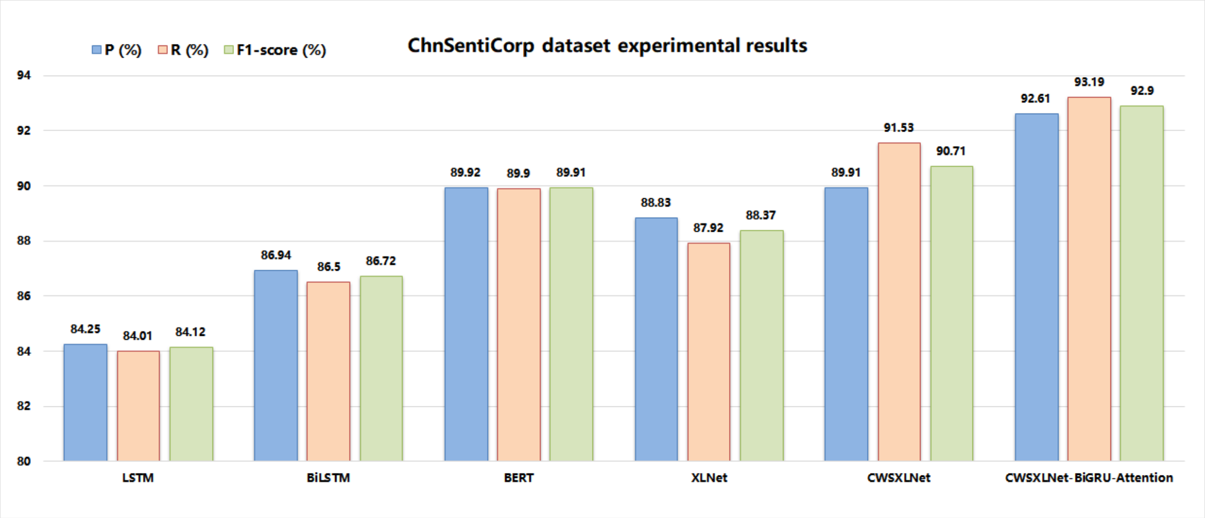


Figure 13. ChnSentiCorp dataset experimental results.

The experimental results of the Weibo_senti_100k dataset are shown in Table 6 and Figure 14 below, and the LTP process was performed on the training corpus to maintain consistency between pre-training and fine-tuning.

Table 6. Weibo_senti_100k dataset experimental results.

Model	P (%)	R (%)	F1-Score (%)
LSTM	90.27	90.58	90.42
BiLSTM	93.28	93.30	93.29
BERT	94.24	94.23	94.24
XLNet	94.28	94.15	94.21
CWSXLNet	95.02	94.83	95.01
CWSXLNet-BiGRU-Attention	95.67	95.48	95.57

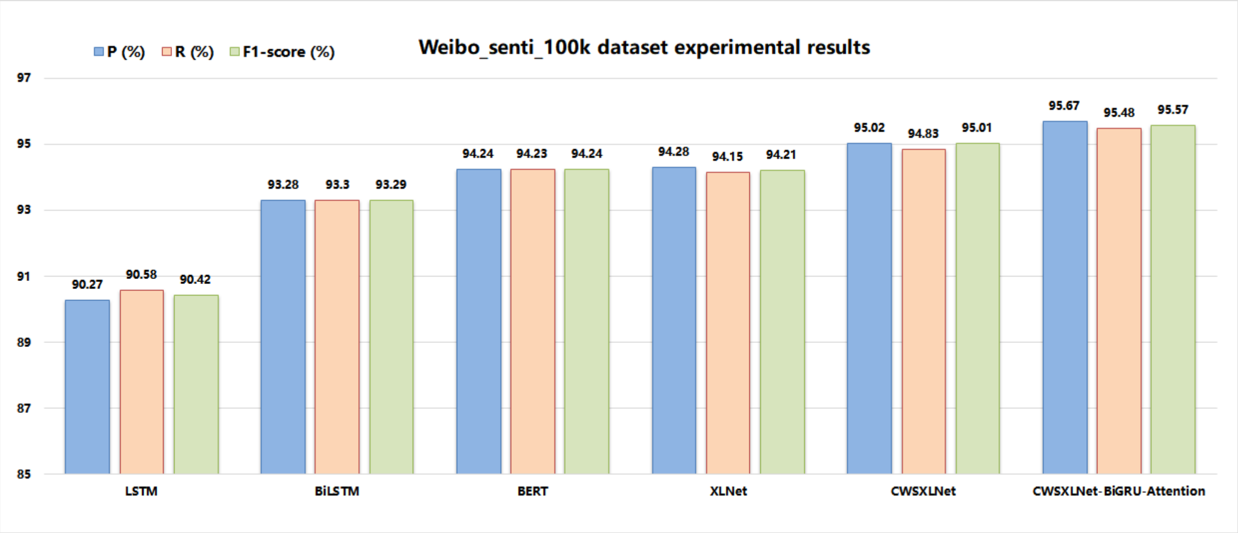


Figure 14. Weibo_senti_100k dataset experimental results.

From the experimental results, both the CWSXLNet model and the CWSXLNet-BiGRU-Attention model achieve better results in dealing with Chinese sentiment analysis tasks.

On ChnSentiCorp dataset, CWSXLNet achieved 89.91% precision, 91.53% recall rate and 90.71% F1-score, and CWSXLNet-BiGRU-Attention has achieved 92.61% precision, 93.19% recall rate and 92.90% F1-score. For comparison, Chinese pre-trained XLNet model proposed by [31] achieved 88.83% precision, 87.92% recall rate and 88.37% F1-score on the same dataset.

On Weibo_senti_100k dataset, CWSXLNet achieved 95.02% precision, 94.83% recall rate and 95.01% F1-score, and CWSXLNet-BiGRU-Attention has achieved 95.67% precision, 95.48% recall rate and 95.57% F1-score. For comparison, Chinese pre-trained XLNet model proposed by [31] achieved 94.28% precision, 94.15% recall rate and 94.21% F1-score.

The experimental results indicated that the Chinese word separation information can help the XLNet model to understand Chinese semantics, and the performance of CWSXLNet-BiGRU-Attention is better than that of CWSXLNet alone, indicating that the BiGRU network and the self-attention mechanism are more accurate and effective in controlling the sentiment keywords.

6. Conclusions

In this paper, we proposed a method to improve the XLNet model for Chinese language processing by addressing the importance of word separation in Chinese language processing and combining it with the SentencePiece tool used by the XLNet model. Experimental evidence shows that the CWSXLNet proposed in this paper outperforms XLNet on Chinese sentiment analysis tasks. Meanwhile, the CWSXLNet-BiGRU-Attention structure model proposed in this paper proceeds further and achieves better performance on the Chinese sentiment analysis task. However, the pre-training method of the XLNet model proposed in this paper still has some shortcomings; for example, there is no better treatment for English and numbers. In other words, the CWSXLNet model is language-dependent and only supports Chinese at present. In further studies, we will focus on these shortcomings and commit to constructing word lists in different languages.

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

Funding: This research was funded by National Natural Science Foundation of China grant number 61962005.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data underlying this article will be shared upon reasonable request to the corresponding author.

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

References

1. Wang, H.; Zhou, C.; Li, L. Design and Application of a Text Clustering Algorithm Based on Parallelized K-Means Clustering. *Rev. D'intelligence Artif.* **2019**, *33*, 453–460. [CrossRef]
2. Kiritchenko, S.; Zhu, X.; Mohammad, S.M. Sentiment analysis of short informal texts. *J. Artif. Intell. Res.* **2014**, *50*, 723–762. [CrossRef]
3. Yadollahi, A.; Shahraki, A.G.; Zaiane, O.R. Current state of text sentiment analysis from opinion to emotion mining. *ACM Comput. Surv. (CSUR)* **2017**, *50*, 1–33. [CrossRef]
4. Bansal, N.; Sharma, A.; Singh, R.K. An Evolving Hybrid Deep Learning Framework for Legal Document Classification. *Ingénierie Des Systèmes D'information* **2019**, *24*, 425–431. [CrossRef]
5. Khoo, C.S.; Johnkhan, S.B. Lexicon-based sentiment analysis: Comparative evaluation of six sentiment lexicons. *J. Inf. Sci.* **2018**, *44*, 491–511. [CrossRef]
6. Sebastiani, F.; Esuli, A. Sentiwordnet: A publicly available lexical resource for opinion mining. In Proceedings of the 5th International Conference on Language Resources and Evaluation, Genoa, Italy, 22–28 May 2006.
7. Esuli, A.; Sebastiani, F. SentiWordNet: A high-coverage lexical resource for opinion mining. *Evaluation* **2007**, *17*, 26.
8. Baccianella, S.; Esuli, A.; Sebastiani, F. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. *Lrec* **2010**, *10*, 2200–2204.
9. Wu, X.; Lü, H.; Zhuo, S. Sentiment analysis for Chinese text based on emotion degree lexicon and cognitive theories. *J. Shanghai Jiaotong Univ.* **2015**, *20*, 1–6. [CrossRef]
10. Wang, S.M.; Ku, L.W. ANTUSD: A large Chinese sentiment dictionary. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, 23 May 2016.
11. Yang, L.; Li, Y.; Wang, J.; Sherratt, R.S. Sentiment analysis for E-commerce product reviews in Chinese based on sentiment lexicon and deep learning. *IEEE Access* **2020**, *8*, 23522–23530. [CrossRef]
12. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans Neural Netw Learn Syst.* **2016**, *28*, 2222–2232. [CrossRef]
13. Xiao, Z.; Liang, P. Chinese sentiment analysis using bidirectional LSTM with word embedding. In Proceedings of the Cloud Computing and Security: Second International Conference, Nanjing, China, 29–31 July 2016.
14. Gan, C.; Feng, Q.; Zhang, Z. Scalable multi-channel dilated CNN–BiLSTM model with attention mechanism for Chinese textual sentiment analysis. *Future Gener. Comput. Syst.* **2021**, *118*, 297–309. [CrossRef]
15. Miao, Y.; Ji, Y.; Peng, E. Application of CNN-BiGRU Model in Chinese short text sentiment analysis. In Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence, Sanya, China, 20–22 December 2019.
16. Zhang, B.; Zhou, W. Transformer-Encoder-GRU (TE-GRU) for Chinese Sentiment Analysis on Chinese Comment Text. *Neural Process. Lett.* **2022**, 1–21. [CrossRef]
17. 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]
18. Cambria, E.; Liu, Q.; Decherchi, S.; Xing, F.; Kwok, K. SenticNet 7: A commonsense-based neurosymbolic AI framework for explainable sentiment analysis. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, Marseille, France, 21–23 June 2022.
19. 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]

20. Sitaula, C.; Basnet, A.; Mainali, A.; Shahi, T.B. Deep learning-based methods for sentiment analysis on Nepali COVID-19-related tweets. *Comput. Intell. Neurosci.* **2021**, *2021*, 2158184. [CrossRef] [PubMed]
21. Shang, C.; Li, M.; Feng, S.; Jiang, Q.; Fan, J. Feature selection via maximizing global information gain for text classification. *Knowl. Based Syst.* **2013**, *54*, 298–309. [CrossRef]
22. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
23. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lweis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
24. 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.
25. Clark, K.; Luong, M.T.; Le, Q.V.; Manning, C.D. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv* **2020**, arXiv:2003.10555.
26. Li, M.; Chen, L.; Zhao, J.; Li, Q. Sentiment analysis of Chinese stock reviews based on BERT model. *Appl. Intell.* **2021**, *51*, 5016–5024. [CrossRef]
27. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1–18.
28. Salma, T.D.; Saptawati, G.A.P.; Rusmawati, Y. Text Classification Using XLNet with Infomap Automatic Labeling Process. In Proceedings of the 2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA), Bandung, Indonesia,, 29–30 September 2021.
29. Yan, R.; Jiang, X.; Dang, D. Named entity recognition by using XLNet-BiLSTM-CRF. *Neural Process. Lett.* **2021**, *53*, 3339–3356. [CrossRef]
30. Gong, X.R.; Jin, J.X.; Zhang, T. Sentiment analysis using autoregressive language modeling and broad learning system. In Proceedings of the 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), San Diego, CA, USA, 18–21 November 2019.
31. Alduailej, A.; Alothaim, A. AraXLNet: Pre-trained language model for sentiment analysis of Arabic. *J. Big Data* **2022**, *9*, 72. [CrossRef]
32. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Wang, S.; Hu, G. Revisiting pre-trained models for Chinese natural language processing. *arXiv* **2020**, arXiv:2004.13922.
33. Kudo, T.; Richardson, J. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv* **2018**, arXiv:1808.06226.
34. Sennrich, R.; Haddow, B.; Birch, A. Neural machine translation of rare words with subword units. *arXiv* **2015**, arXiv:1508.07909.
35. Che, W.; Feng, Y.; Qin, L.; Liu, T. N-LTP: An open-source neural language technology platform for Chinese. *arXiv* **2020**, arXiv:2009.11616.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.