Fine-grained Sentiment Analysis in Chinese Based on LSTM

58119125 JiangZhuoyang
58119132 LiKaixin

Abstract

We explore a fine-grained dichotomous sentiment orientation analysis based on LSTM in Chinese context. The basic process of emotion analysis engineering is clarified from beginning to end, and the methods and related models involved in each process are studied in depth theoretically, and the theoretical framework is summarized to form the analysis. On the engineering task, we used the collected corpus for text preprocessing and word vector generation, and then used the Tensorflow deep learning framework to build the basic and improved LSTM classification neural network. By practicing the model on the training set, the model obtained has achieved good evaluation result of sentiment analysis on the test set, and has good analysis ability on the real-time input text. On the research task, we traced the model ideas from RNN to LSTM, explored the improvement ideas of Attention and BiLSTM, made a comparative analysis of the model before and after improvement, summarized the advantages and disadvantages of the model, and proposed the follow-up research direction.

1 INTRODUCTION

Our task is a fine-grained dichotomous affective tendency analysis based on LSTM.

1.1 Background

When choosing tasks, we start with life. In the form of task-driven, our research theme and engineering objectives are presented. In recent years, we have witnessed many influences of the network public opinion environment on the society, and we have to admit that the rapid development of social networks has brought huge moral problems. For the study of sociology and social psychology, natural language, as a medium of communication, appears in the environment of network public opinion in a short and fine-grained pattern. These short subjective texts have great research significance. Among them, the measurement of the so-called ”subjectivity” can, to a certain extent, cognitively analyze public opinions in this society. A specific example of this ”subjective” analysis is the emotional tendency analysis of fine-grained public opinion texts.

1.2 Motivation

In addition to being driven by realistic tasks, there are more important reasons why we choose fine-grained sentiment analysis as our exploration target:

(1) Include a variety of natural language processing technologies.

As an engineering research task based on natural language processing, the complete process of sentiment analysis includes various important technologies of natural language processing, so we can establish a basic cognition of the model in each step by getting familiar with the engineering process.

(2) The model based on RNN does not require too much linguistic knowledge.

The essence of sentiment analysis is text classification. When selecting the model for text classification, we choose the classification model based on recurrent neural network. The framework of this model is very consistent with the characteristics of Chinese natural language, so we do not need to spend time learning the linguistic knowledge required in artificial feature engineering. In this way, I can focus on the study and practice of natural language processing.

(3) It is helpful to understand the cutting-edge methods of natural language processing.

Using recurrent neural network as the main model of sentiment analysis engineering can drive us to continue to understand the cutting-edge technology in the field of natural language processing, which is no longer an ordinary machine learning task, but a real task in the field of natural language processing. [1]

(4) It is helpful to understand the improvement ideas of the model.

The recurrent neural network model has undergone various improvements in the field of natural language processing. Starting from the most basic LSTM, we can not only establish cognition of the basic RNN emotion analysis model, but also preliminarily understand some ideas for improving the model, which will be of great help to our future scientific research.

(5) It can produce the actual application results for analysis.

As a practical task, sentiment analysis allows us to start from a practical task to obtain the actual application effect of the model, so as to be familiar with how to collect data, process data, build models, improve models and get results in the application of natural language processing step by step.

1.3 Task description

1.3.1 Engineering tasks

(1) Construct a complete project from fine-grained text corpus to emotion orientation evaluation value according to the process.

(2) Practice the methods involved in each step of the process and build corresponding models.

(3) Finally, a practical fine - grained binary classifier for sentiment analysis is obtained.

1.3.2 Research tasks

(1) The idea of RNN model for sentiment analysis task traceability: LSTM is the text classification model.

(2) Discuss the improvement idea of the model: BiLSTM and Attention are introduced to improve the LSTM model.

(3) Qualitative analysis and quantitative analysis were conducted by comparing the actual effect of the LSTM-based emotion analysis model and the improved emotion analysis model.

2 PROCESS WITH METHOD

We have sorted out the basic process of sentiment analysis engineering from the beginning to the end, studied the methods and related models involved in each process in depth, and understood their application methods. We will use our own understanding to describe these processes and related methods in the process.

2.1 Preprocessing

The processing object of fine-grained sentiment analysis task is sentence text, so it is necessary to collect a large amount of sentence information first, conduct basic data cleaning on these original sentences to build a corpus, and at the same time conduct artificial emotion orientation judgment on the sentences to build basic sentence training set and test set. After that, on the basis of the training set and test set, word segmentation is further preprocessed to convert the statements of the training set and test set into word array, which is convenient for the subsequent digital description.

2.1.1 Corpus building

In the specific task, we need to batch crawl the statement information in the sentiment analysis environment and perform data cleaning on the original text, such as processing stop word in the statement text.

After that, one to one artificial emotional orientation judgment is carried out to determine the emotional label of the item, so as to build the basic sentence training set and test set.

2.1.2 Jieba

The most important step in the following preprocessing is to segment the sentence elements in the training set and test set into the word array elements, laying the foundation for the following word vector expression.

Since the processing object is Chinese text, we choose to use jieba tool for word segmentation. We have roughly understood the principle of jieba word segmentation. Summarized as follows:

- Realize efficient word graph scanning based on prefix dictionary, and generate a directed acyclic graph (DAG). Generate a trie prefix tree with word frequency.

- Dynamic programming to find the maximum probability path and find the maximum segmentation combination based on word frequency.

- For unknown words, HMM model based on Chinese word formation ability is adopted, and Viterbi algorithm is used.

- Inputing a sentence to be segmented, the HMM with four states(BEMS) need to find the best BEMS sequence, with viterbi algorithm whose parameter matrix had been trained previously.

2.2 Embedding

After that, we can make mathematical representation of the result of word segmentation. After word segmentation, the text is transformed into an array of words, and we can further mathematically describe the lexical elements in these arrays by means of word vector.

2.2.1 Word Vector Representation

(1) Motivation of Word Vector

When a computer processes text, the model cannot understand the text directly. So the first step in NLP is to convert words into numbers that model can understand.

In machine learning, deep learning, text must be described as numbers. The most common method is to convert text into dictionary order. The main problem is that the lexicographical order of text only represents the lexicographical order of each text and has no concrete meaning in practice.

To solve the lexicographical problems, we can use one-hot encoding the word into a vector which has only one dimension is 1 and all the others are 0. One-hot can correspond to only one word, and its elements of 0 and 1 are suitable for probabilistic information.

However, One-hot tends to be too long and has no representation of semantics. In view of the above problems, we propose a word vector model.

(2) Concept of Word Vector

Word vector, also known as a group of language modeling and feature learning techniques in Word embedded Natural language processing (NLP) [7], in which words or phrases from vocabulary are mapped to vectors of real numbers. Conceptually, it involves the mathematical embedding of Spaces from one dimension per word to continuous vector Spaces with lower dimensions.

Embedding can be abstractively understood as: it is designed to describe the features of different dimensions of one-hot in order to make one-hot have stronger semantic features. It's just that the dimensions and characteristics are learned in some way.

The word vector has spatial meaning and is not a simple mapping. Vector representation of words is also called word embedding (mapping words into space to make a spatial representation).

2.2.2 Word2Vec and Embedding

Word2vec is one of the methods of Word Embedding.

We need to ask the question: how do we use contextual information to represent the meaning of a word?

Intuitively speaking, Word2vec is further in the language algorithm expression, do a mapping task: language $\rightarrow$ Function $\rightarrow$ Target tasks. The reasons for introducing target tasks is for machines to explore the semantic features of words [8]. Because semantic characteristics are derived from the context, a task is introduced that is relevant to the following processing. When training this task, we use some parameters to establish the corresponding relation between the input word and its context. These parameters can be used to indicate the embedding of this word.

There are two ways to do this following with the related target tasks:

- skip-gram: Given the current word in the sentence, predict the surrounding words.

- CBOW: Given the surrounding words in the sentence, predict the current word.

We can train the parameters as our embedding methods through completing the task.

2.3 Training Set and Test Set

After that, we can further describe the word elements in the sentence array with the word vector, and construct the training set and test set of the sentence emotion type described by the word vector, laying a foundation for the training of the classification model.

2.4 Classification Model

Next, we train the classification model. First, we understand the basic ideas and infrastructure of the classification model.

2.4.1 The RNN thought

(1) Motivation of RNN

1) Disadvantages of traditional machine learning methods:

Text sentiment orientation analysis is similar to traditional theme-based text classification, but sentiment classification is not based on the content itself, but based on the emotion and attitude of the text. Based on this, traditional machine learning needs to first extract the emotional features held by the text, such as emotional words, part of speech, syntactic structure, negative expression template, connection, semantic topic, etc.

Similar to the idea of introducing neural network models in other fields, people hope to use neural network models to automatically mine high-level features to make up for the limitations of human feature cognition.

2) Disadvantages of traditional NN network:

In the neural network model, the traditional fully connected neural network can realize the function of feature mining in training. However, from the perspective of its workflow, the traditional fully connected neural network can only process each input in turn, and ignores the relationship between the input before and after.

In sentiment analysis, the input of classification model is text on the whole, and text is divided into word input neural network model on the part. It can be seen from the nature of Chinese natural language that there must be semantic connection between the two words. Abstractly, Chinese natural language text should be a sequence, not a Mosaic of independent elements. For input to the model, there is a relationship between the words entered before and the words entered after. So we want the model to be able to better process the sequence of information, and each time we process a new input, we need to take the previous input information into account.

3) Adaptability of RNN:

The essence of sequence takes time dimension into consideration. RNN is a neural network model that can connect the two input information in time dimension. This determines that the idea of RNN has a good adaptability to emotion analysis tasks.

(2) Idea of RNN

As shown in Figure 1, the value s[t] of the hidden layer of the recurrent neural network depends not only on the current input X [t], but also on the value S [t-1] of the last hidden layer. The weight matrix W is the weight of s[t-1] as the input of this time (t).

In this way, we are able to relate the previous and subsequent inputs in a temporal dimension.

We can expand RNN in the time dimension to further explore the influence of its time characteristics on the sequential input. As shown in the Figure 1, after the network receives input X_t at time t, the value of the hidden layer is S_t and the output value is O_t. The value of S_t depends not only on X_t but also on S_{t-1}. Thus we can use the following formula to describe the core idea of RNN [14].

$$\mathrm{O}_t = g(V \cdot S_t)$$
$$S_t = f(U \cdot X_t + W \cdot S_{t-1})$$

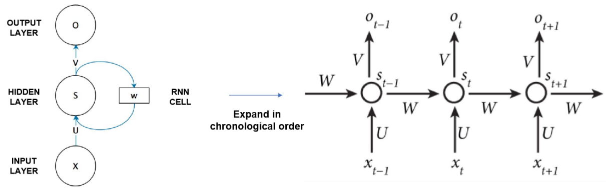


Figure 1: RNN Timeline expansion diagram

(3) Sentiment Analysis with RNN

The sentiment analysis task based on neural network requires feature mining and classification of text. Text, as a sequential data input model, needs to consider its contextual association information in the time dimension, so RNN can effectively memorize such association information.

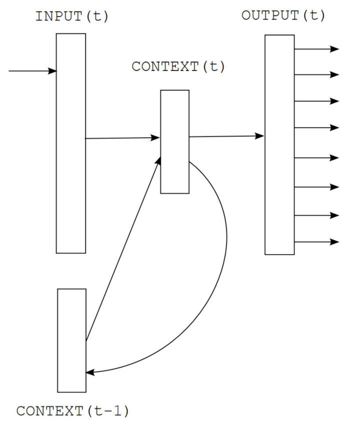


Figure 2: RNN structure in text information processing

One problem with the basic idea of RNN in natural language processing, however, is that not all information in text sequences is related in a ”close” way.

Suppose we try to predict ”I am a Chinese living in a beautiful countryside nbeside a vast lake where people here speaking Fluent” Mandarin is the last word. Current information suggests that the next word may be the name of a language, but if we need to figure out what language it is, we need the context of the Previously mentioned Chinese far from the current position. This means that the gap between the relevant information and the current predicted position must become quite large. Unfortunately, as this interval increases, the RNN loses the ability to learn to connect information so far away.

Essentially, this phenomenon is mainly reflected in the problem of gradient disappearance or gradient explosion in the process of training RNN.

For example, when RNN is trained by backward propagation of stochastic gradient descent method [9], partial derivatives of weight parameters need to be obtained by loss function. This may produce long-term dependence problems with time series. Because the absolute value of the activation function is less than 1, if weight is also less than 1 and the time series length is large, the multiplication in chain rule will turn the whole item into 0, which is the disappearance of the gradient of long-term dependence. Similarly, when the weight is very large, the partial derivative of the chain rule goes to infinity, which is the gradient explosion that depends on the long term.

Therefore, we need to improve the structure of RNN cells to enable them to perform special screening and memory for long-term dependent information.

2.4.2 The LSTM model

(1) Motivation of LSTM

As can be seen from the previous description of RNN, if we want to use the neural network model to carry out feature mining and text classification for sentiment classification tasks that take text sequences as processing objects, we need to introduce the idea of RNN to consider the contextual information in the time dimension. When the interval of associated information is too large, ordinary RNN cannot effectively remember the distant long-term dependent information. Therefore, we need to improve the structure of RNN cells and design a network model with stable weight effect for long-term memory [5].

In view of this, LSTM improves the basic RNN by cleverly designing the network structure, which not only extends the memory distance of context information, but also filters the contents of memory, avoiding the problems of gradient disappearance and gradient explosion in training caused by long-term dependence.

(2) Solution of LSTM

We studied a classic LSTM blog and deconstructed the LSTM process [4].

Firstly, the overall framework of LSTM is described. By comparing THE FRAMEWORK of RNN and LSTM, we can find that there are four processing units in THE RNN cell of LSTM, which are different from the single neural network layer, and they interact in a very special way.

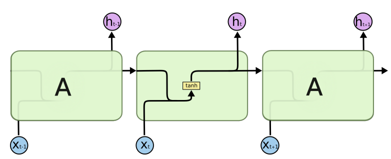


Figure 3: RNN

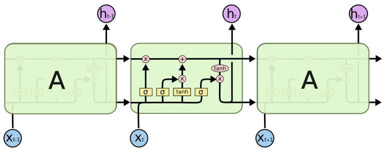


Figure 4: LSTM

We can describe the basic elements of these interactions:

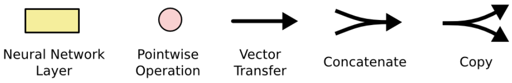


Figure 5: Interaction Mode description

After that, we must know that the core improvement of LSTM compared with RNN is the introduction of cell state, as shown in Figure 14. Cell state is the memory information transmitted across the top of each RNN cell structure. The cellular state is like a conveyor belt. It runs directly along the chain, with just a few linear interactions. It would be easy for the message to circulate and stay the same.

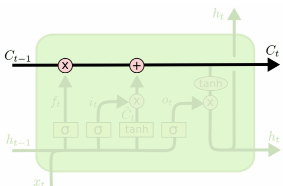
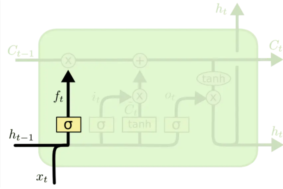


Figure 6: State of the cell

Among them, there is a very important processing unit, which is the so-called "gate", which enables LSTM to remove or add information to the cell state, solves the long-term screening and memory of information, and avoids direct long-term dependence.

Then, according to Colah's Blog, we followed the steps to understand LSTM:

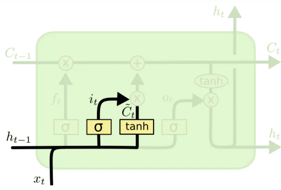
The first step is to decide what information we're going to throw away from the cellular state. This decision is made through a portal called the oblivion Gate. The gate reads h_{t-1} and x_t and outputs a value between 0 and 1 for each number in cell state C_{t-1}. 1 indicates completely reserved, 0 indicates completely discarded.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Figure 7: Decide to discard information

The second step is to determine what new information is stored in the cellular state. There are two parts to this. First, the $sigmoid$ layer, called the "input gate layer", determines what values we are going to update. Next, a $tanh$ layer creates a new vector of candidate values $\tilde{C}_t$ that are added to the municipality. Next, we'll talk about these two pieces of information to generate state updates.

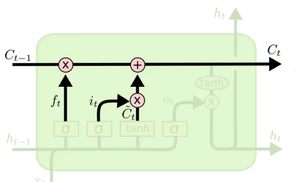


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Figure 8: Identify the updated information

The third step is to update the old cell state, C_{t-1} to C_t. The previous steps have determined what will be done, and we are now going to actually do it.

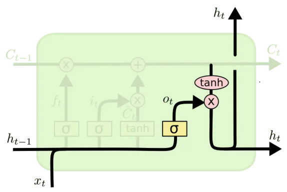
We multiply the old state by f_t, discarding the information we know we need to discard. Also add $i_t * \tilde{C}_t$. This is the new candidate value, varying according to how much we decide to update each state.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 9: Update cell state

Finally, we need to decide what value to output. This output will be based on our cell state, but also a filtered version. First, we run a $sigmoid$ layer to determine which part of the cell state will be output. Next, we process the cell state through $tanh$ (to get a value between -1 and 1) and multiply it by the output of the $sigmoid$ gate. Finally, we will only output what we are sure to output.



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 10: Output information

2.5 Classification Model Training

After understanding the basic principle and architecture of LSTM classification model, we can use the existing deep learning framework to build a model and train the classification model on the training set of sentence emotion category represented by word vector.

2.6 Model Application

Finally, we can use the trained model to conduct batch sentiment orientation analysis or real-time single sentence emotional orientation analysis for sentences in context. We will reproduce and analyze these two operations in practice.

3 EXPERIMENTATION

3.1 Data Acquisition and Calibration

We chose the open source microblogging corpus of https://github.com/dengxiuqi/weibo2018 as a data source, in the corpus, construction of the corpus climb took a considerable amount of weibo corpora, and has carried on the preliminary data cleaning, filter out the advertising/too short/long/ideographic unclear corpora, Regular expressions are used to standardize the translation of special text information such as micro-blog expressions and hyperlinks.

After that, the corpus builders labeled each fine-grained text corpus by dichotomy and constructed 10,000 training samples and 500 test samples. The ratio of positive and negative corpora in the training set was about 55:45, and that in the test set was about 7:3.

The data cleaning of corpus is more rigorous, and the annotation of emotion is also very careful. Although the corpus is small, it is a high-quality corpus in general. It provides a good help for our project reproduction.

3.2 Data Preprocessing

In the data preprocessing stage, we first remove hyperlinks, user names and texts filled in by non-users in the expected part of the sample by regular expression, and finally only the valid texts are retained as the sentence pattern corpus. Jieba word segmentation is performed on the expected part of the cleaned sample, and the expected part of the sample points is constructed into an array of words.

3.3 Word Vector Generation

After understanding the word vector construction method of Word2Vec, we choose to use fastText method to train word vector. The framework of fastText is similar to that of CBOW in Word2Vec, but the target task is different. The target task of fastText is related to text classification, specifically, the prediction of text labels. Based on this target task, word vector training can increase n-gram features and better adapt to the situation of small corpus.

In order to unify the length of the input sentence vectors, we set the upper limit of the number of sentence vectors, add zero vector to the sentences with insufficient length, and cut off the sentences with too long length.

3.4 LSTM Building and Training

Based on TensorFlow architecture, we construct two layers of LSTM and two layers of MLP, and build a basic LSTM classification model. After setting batch size, optimization stop value, hidden size and other hyperparameters, the model was trained with epoch for 1000 times.

4 MODEL IMPROVEMENT

In order to further deepen the knowledge of natural language processing in sentiment analysis engineering, we need to study some improvement methods in addition to the LSTM method of traceability. Here, we improve LSTM from BiLSTM and Attention.

4.1 Improvement Ideas

In order to further deepen the knowledge of NLP in emotion analysis engineering, we need to study some improvement methods in addition to the LSTM method of traceability. Here, we improve LSTM from BiLSTM and Attetion.

4.1.1 LSTM Improvement – BiLSTM

LSTM modeling is a problem, can't code from the back forward information, this is a big flaw, because in the classification of fine-grained, such as strong degree of good, low degree of good, neutral and weak degree of derogatory, strong degree of derogatory five classification task, degree of words, negative emotional words need to be paid attention to the interaction between, for example, "This restaurant is not as dirty as next door." "No" is a modification of "dirty" degree.

BiLSTM (bi-directional Long short-term Memory) is a combination of forward-lSTM and forward-LSTM to better capture bi-directional semantics. To put it simply, the forward LSTM part of BiLSTM takes sentence vectors in normal word order as input, and the backward LSTM part takes sentence vectors in reverse order as input. After hidden layers, vectors in two directions are obtained, and these two vectors are splicing into a vector for sentiment analysis.

4.1.2 LSTM Improvement – Attetion

BiLSTM + Attention is to add Attention layer to BiLSTM model. In BiLSTM, we will use the last output vector of timing sequence as the feature vector, and then conduct softmax classification [2]. Attention first calculates the weight of each time sequence, then weights and sums all vectors of time sequence as feature vectors, and then conducts softmax classification [15].

4.2 BiLSTM Building and Training

Similar to the LSTM model, we constructed two layers of LSTM and two layers of MLP based on TensorFlow architecture, and established the basic LSTM classification model. After setting batch size, optimization stop value, hidden size and other hyperparameters, the model was trained with epoch for 1000 times.

5 RESULT DISCUSSION

5.1 Quantitative Analysis on Test Set

Firstly,we do the quantitative analysis.

5.1.1 Training Two Times

We trained the two models twice and evaluated them twice. In the two evaluations, there was no difference in the quantitative relationship between the corresponding evaluation results of the two models, which proved that the model was relatively robust.

5.1.2 Test Results

The evaluation results on test sets of the two training turns are shown in Table 1 and Table 2:

Table 1: Quantitative results on test set 1

Criteria	Train 1	
	LSTM	BiLSTM
Accuracy	0.836	0.850
P-Precision	0.880	0.900
N-Precision	0.730	0.750
P-Recall	0.880	0.880
N-Recall	0.740	0.770

Table 2: Quantitative results on test set 2

Criteria	Train 2	
	LSTM	BiLSTM
Accuracy	0.840	0.864
P-Precision	0.890	0.910
N-Precision	0.730	0.770
P-Recall	0.880	0.890
N-Recall	0.760	0.810

1) Accuracy: the evaluation Accuracy of the correct and wrong levels of the general classification results.

2) Precision of positive emotion: Based on the prediction results, the proportion of samples predicted as positive emotion is really positive emotion samples.

3) Recall of positive emotions: Based on the actual situation, it is the proportion of samples with positive emotions that are predicted to be positive emotions.

4) Precision of negative emotion: Based on the predicted results, the proportion of samples predicted as negative emotion is really negative emotion.

5) Recall of negative emotion: Based on the actual situation, it is the proportion of samples with positive emotion that are predicted to be positive emotion.

5.1.3 Accuracy Analysis

For the emotional orientation dichotomy model based on LSTM and the emotional orientation dichotomy model based on BiLSTM+Attention, the batch test accuracy on the test set reached 80% in the two training and testing. From the perspective of the classification task itself, the two have met the performance requirements of the task model.

The classification accuracy effect of BiLSTM+Attention classification model is higher than that of LSTM classification model, indicating that BiLSTM+Attention does improve LSTM on the whole. However, in general, there is no obvious difference in classification accuracy between BiLSTM and LSTM, which is caused by the small corpus. If the training is conducted on a larger corpus, the performance gap between the two will gradually increase, and the advantages of BiLSTM+Attention will continue to emerge.

5.1.4 Prediction Analysis

In terms of the accuracy of predicted values, since there are more positive emotion samples than negative emotion samples in the training corpus, the accuracy of predicting positive emotion samples in both models is higher than that of predicting negative emotion samples, which can be seen from the Precision evaluation results of the two models. In the two training and testing, The Precision of positive emotion samples of the two models is higher than that of negative emotion samples.

Comparing the Precision of the two models, in the two training and testing, the Precision of positive and negative emotion samples of BiLSTM+Attention model is higher than that of the basic LSTM. It reflects that BiLSTM+Attention model has a stronger ability to predict accurately on the whole. In principle, this is related to the introduction of Attetion mechanism.

5.1.5 Recall Analysis

In terms of the comprehensiveness of the real value judgment, the model also has a higher comprehensiveness of the positive emotion sample judgment than the negative emotion sample judgment because there are more positive emotion samples than negative emotion samples in the training corpus. This can be seen from the Recall evaluation results of the two models. The recall of positive emotion samples of the two models was higher than that of negative emotion samples.

Comparing the recall of the two models, it can be seen that there is little difference between the two models in the comprehensiveness of identification of positive emotion samples. However, BiLSTM+Attention has a more comprehensive identification of the real situation of negative emotion samples than LSTM in the two training and tests. In principle, BiLSTM+Attention can select real negative emotion samples more easily, largely because its bidirectional sequence design partially solves the postposition of degree adverbs in Chinese natural language.

5.1.6 P-R curve Analysis

We draw P-R curves for Precision and Recall evaluation of the two models.

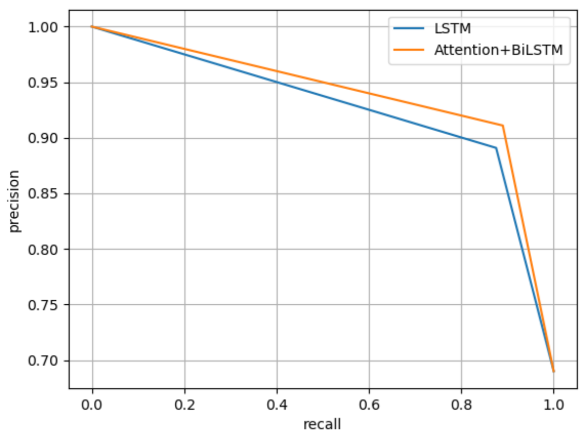


Figure 11: P-R curve

The P-R figure can intuitively show the comprehensive performance of Precision and Recall of binary classifiers. In comparison, if the P-R curve of one binary classifier is completely covered by the P-R curve of another binary classifier, it can be asserted that the latter has better performance than the former. It can be seen that the P-R curve of BiLSTM+Attention completely covers the P-R curve of LSTM, so our improvement is effective.

5.2 Qualitative Analysis in Real-time-test

We also conduct real-time emotion classification test for some special Chinese sentences.

LSTM	
Input Text	Predicition
"我永远喜欢东南大学"	0.9998994470
"南京的冬天美得不行"	0.9996442795
"某些建筑物丑得不行"	0.0021654963

Figure 12: LSTM

BiLSTM+Attention	
Input Text	Predicition
"我永远喜欢东南大学"	1.0000000000
"南京的冬天美得不行"	0.9992145300
"某些建筑物丑得不行"	0.0000218153

Figure 13: BiLSTM+Attention

As we can see, the improved model can classify positive and negative emotion samples more strongly

6 CONCLUSION

6.1 The Superiority of Model Basic Idea

In the classification of natural language text sequence input, LSTM's classification model has obvious advantages over ordinary Neural Network model. It takes into account the relevance information between inputs in the sequence time dimension.

Compared with the basic RNN, the LSTM classification model solves the long-term dependencies problem related to gradient disappearance or gradient explosion, and has a better memory efficiency for distant related information.

6.2 The Superiority of Improvement Idea

From the intuitive results, the Attention mechanism is introduced to introduce a contribution of input words in the context, and the structure of LSTM is improved to BiLSTM, which does improve the precision of classification to a certain extent, and improves the Recall of negative samples.

6.3 Disadvantages of Model

- Although BiLSTM is indeed much improved than the basic RNN, it still cannot transmit the information of the starting point of the sequence well for excessively long sequences.
- BiLSTM's sequence processing model cannot calculate the result of the next moment without calculating the result of the previous moment. Therefore, parallel computing is impossible and its processing efficiency is generally not high.

6.4 Future Work

LSTM does not have efficient parallel computing capability. To solve this problem, SRU [6] and Sliced RNN [13] are the two improvement ideas.

In addition to the sentiment orientation classification model based on RNN, Kim also introduced CNN [10] into NLP. There are also some excellent works on sentiment analysis based on CNN [12], and there are also many methods to improve model performance in this direction, which are worth studying.

In addition, with Transformer [11] and Bert [3] popular today, we can also improve the performance of the model from this aspect, and have a further understanding of the current research hotspot of natural language processing.

References

[1] Yonatan Belinkov and James Glass. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72, 2019.

[2] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine

translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, 2014. Association for Computational Linguistics.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[5] Bo-Jian Hou and Zhi-Hua Zhou. Learning with interpretable structure from gated rnn. *IEEE transactions on neural networks and learning systems*, 31(7):2267–2279, 2020.

[6] Tao Lei, Yu Zhang, Sida I. Wang, Hui Dai, and Yoav Artzi. Simple recurrent units for highly parallelizable recurrence. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4470–4481, Brussels, Belgium, 2018. Association for Computational Linguistics.

[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[8] Xin Rong. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*, 2014.

[9] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[10] Zhiqiang Toh and Jian Su. NLANGP at SemEval-2016 task 5: Improving aspect based sentiment analysis using neural network features. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 282–288, San Diego, California, 2016. Association for Computational Linguistics.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[12] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*, 2017.

[13] Zeping Yu and Gongshen Liu. Sliced recurrent neural networks. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2953–2964, 2018.

[14] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[15] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany, 2016. Association for Computational Linguistics.