

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Sentiment Analysis with Comparison Enhanced Deep Neural Network

Yuan Lin¹, Jiaping Li², Liang Yang, Kan Xu², Hongfei Lin²

¹ Faculty of Humanities and Social Sciences, Dalian University of Technology, Dalian, China

² Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, China

Corresponding author: Liang Yang (e-mail: liang@dlut.edu.cn).

This work is partially supported by grant from the Natural Science Foundation of China (No. 61976036, 61702080, 61632011), the National Key Research and Development Program of China (No.2018YFC0832101), Postdoctoral Science Foundation of China (2018M631788), the Fundamental Research Funds for the Central Universities (No. DUT19RC(4)016).

ABSTRACT Sentiment analysis is a significant task in Natural Language Processing. It refers to classification based on the emotional tendency in text by extracting text features. The existing results show that models based on RNN and CNN have good performance. In order to improve the performance of text sentiment analysis, we reformulate the classification task as a comparing problem, and propose Comparison Enhanced Bi-LSTM with Multi-Head Attention (CE-B-MHA). In fact, it is efficient to classify by comparison mechanism instead of doing complex calculation. In this model, bidirectional LSTM is used for initial feature extraction, and valuable information is extracted from different dimensions and representation subspaces by Multi-Head Attention. The comparison mechanism aims to score the feature vectors by comparing with the labeled vectors. The experimental results show that CE-B-MHA has better performance than many existing models on three sentiment analysis datasets.

INDEX TERMS Sentiment analysis, Machine learning, Neural networks

I. INTRODUCTION

Today, countless text messages are produced in the internet every day. On social media, people express their opinions in text form. On pages with film reviews, they are also presented in text form. These text information with large quantity and rich content can become an important data resource.

The short texts published on various platforms contain strong emotional tendencies and reflect the diverse views held by users. Sentiment analysis of these massive text messages is of great value to various industries. For example, by analyzing citizens' different attitudes towards the same news event, the government can understand the public's opinions on social events and related policies. By analyzing the user's attitude towards a product function, the manufacturer can improve the product easily. The potential value of sentiment analysis attracts extensive attention from researchers in different fields, such as data mining and natural language processing. Sentiment analysis for user-generated text has become a research hotspot in relevant fields.

Sentiment analysis [1], also known as opinion mining [2], is a research field to analyze people's subjective feelings such as emotions, evaluations, opinions and attitudes towards products, services, organizations, individuals, events,

subjects and their attributes. Text sentiment analysis task is one of the most important tasks in the field of natural language processing. It is found that the neural network model which is based on Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) can effectively extract the context relations of sentences. The classification accuracy of the model can be improved by introducing Attention Mechanism to filter irrelevant information.

Attention mechanism was first proposed in the field of computer vision [3], and gradually entered the field of natural language processing [4]. The purpose of this mechanism is to allow the model to focus on important objects, which are mathematically represented as weighted sums. In "Attention is all you need" [5] published by Google machine translation team in 2017, the Multi-Head Attention mechanism used in this paper was proposed.

Sequential information is particularly important for natural language processing tasks. It represents the logic and structure of the text. However, Multi-Head Attention mechanism can obtain global structure, but it is hard to effectively capture sequence information. In this paper, Bi-LSTM is introduced to capture sequence information to get the order of the text and improve the model.

LSTM is a special type of recurrent neural network [6], has achieved great success in solving many problems. LSTM network introduces a self-cycling mechanism that makes it easier to learn long-term dependent information than a simple cycling structure. The bidirectional LSTM (Bi-LSTM) model inputs the forward and backward directions of the sequence into the LSTM network, which can better capture the contextual order information of the sequence [7].

Existing sentiment analysis models based on deep neural network have little to do with psychology, and have to use too much parameters. On the basis of deep neural network, we introduce comparison mechanism, a psychological concept, to enhance the learning ability of the model, which uses the comparison of text itself instead of using too much parameters. Comparison is the most intuitive and effective way of classification in people's daily life. People always learn new things by comparing. In this paper, we reformulate the sentiment analysis task as a comparing problem. Sentiment analysis task aims to train a model that maps the item feature to a score. We get the score by comparing the items with labeled samples, instead of fitting the hard-to-learn patterns.

In this paper, Comparison Enhanced Bi-LSTM with Multi-Head Attention (CE-B-MHA) is proposed to improve the performance of sentiment analysis. CE-B-MHA combines the ability of Multi-Head Attention to obtain global information with the ability of Bi-LSTM to obtain local sequence information, and enhances it with a comparison mechanism.

The main contributions of our work can be summarized as follows:

- We combine Bi-LSTM with Multi-Head Attention, and propose a new model with good performance. We use Bi-LSTM to capture the context relationship, and use Multi-Head Attention to capture long distance text features.
- We propose a text sentiment analysis model, namely CE-B-MHA. On the basis of constructing deep neural network with Bi-LSTM and Multi-Head Attention, we use comparison mechanism to enhance the model and improve its performance.
- We use public datasets to verify the efficiency of our model, and compare it with the current methods. Experimental results show that CE-B-MHA can improve the performance compared with several baselines.

The rest of our paper is structured as follows: Section II discusses related works, Section III gives details of our approach, Section IV gives our experimental results, and Section V summarizes this work.

II. Related Works

A. Sentiment Analysis Task

The sentiment analysis task can be divided into binary classification task and multi-class classification task according to the different classification objectives. In many cases, the researchers divided the emotional polarity of the text into positive and negative categories, commonly known as binary classification task. Generally, positive text indicates positive emotional tendency, while negative text indicates negative emotional tendency. This simple classification method can be applied in many real situations, such as analyzing whether users are favorable or unfavorable to a certain commodity and literary works, and public opinion towards a certain social event. In the research process, binary classification is also a common task to evaluate the classification ability of models.

In addition, multi-class classification task is also an important direction. Multi-class classification task can be divided into emotional level classification and fine-grained emotion classification. Emotional level classification refers to divide text emotional tendency from negative to positive into several levels. The classification of emotions into 1 to 5 can be called a five-classification problem. The fine-grained emotion is classified according to the categories of emotion. There is no common criterion for this classification, and it is generally self-determined based on research questions, such as emotions that can be divided into joy, anger, sadness, surprise, disgust, fear and neutrality.

B. Sentiment Analysis Methods

The methods used in text sentiment analysis can be divided into two categories: emotion dictionary method and machine learning method. Constructing an emotion dictionary and using it as a tool is the traditional method to judge the emotional polarity of texts [8]. Most emotional dictionaries need to be constructed manually. The principle is to summarize the words with emotional tendency to form a dictionary. Emotional words have strong indicating ability, which is one of the important signals that the text contains emotional tendency. When the text is entered, it is matched with the contents of the dictionary, looking for emotional words in the text to determine the emotional polarity of the text. However, there are some limitations in the emotion dictionary method. It covers insufficient forms of emotional expression and cannot timely cover the emerging forms of expression, which makes the accuracy of textual emotional judgment relatively low.

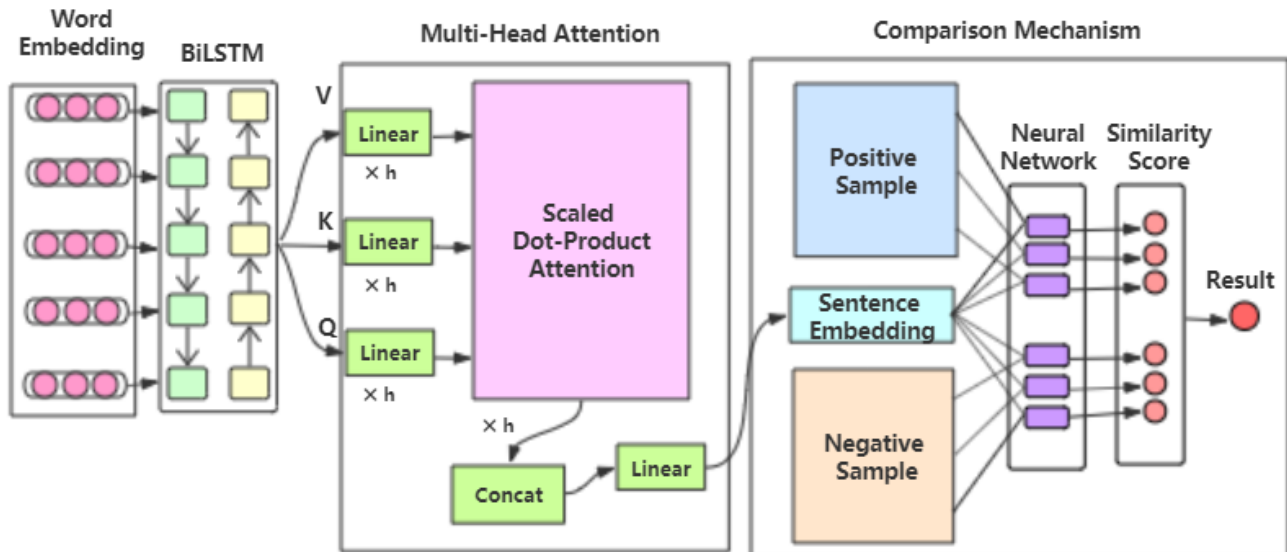


FIGURE 1. Comparison Enhanced Bi-LSTM with Multi-Head Attention (CE-B-MHA).

Nowadays, machine learning is a common method for researchers to analyze text emotion. The computer processes the text, extracts the text features and outputs the sentiment analysis. Machine learning methods have obvious advantages over emotion dictionary method which rely on manual work heavily. The machine learning methods of text sentiment analysis are mainly divided into supervised sentiment analysis and unsupervised sentiment analysis. The method proposed in this paper is a supervised sentiment analysis method, which is briefly introduced in the following part.

The basic principle of supervised sentiment analysis is to use the labeled text training model and use the trained model to conduct sentiment analysis on the unlabeled text. In addition to traditional machine learning methods, such as support vector machine, there are also deep learning methods such as CNN and RNN. Pang et al. [9] applied three representative classifiers (support vector machine, Naive Bayes and maximum entropy) to conduct experimental research on text sentiment analysis task, which has a high accuracy rate. Kim [10] proposed the classification of text CNN, becoming one of the important baselines of sentiment analysis task. Brueckner and Schuller [11] used Bi-LSTM in sentiment analysis task, which contributes to the solution of obtaining both historical information and future information by using the bidirectional propagation mechanism. Tang et al. [12] used two different RNNs to conduct sentiment analysis in combination with texts and themes. Wang et al. [13] proposed an Attention-based Long Short-Term Memory Network for aspect-level sentiment classification. The attention mechanism can concentrate on different parts of a sentence when different aspects are taken as input. Baziotis et al. [7] believed that deep LSTM with attention(D-LSTM) can improve the performance of the model. Shen et al. [14] proposed a novel LSTM, called ON-LSTM, to deal with natural language processing problems. The neurons in the

LSTM are specifically ordered to express richer information. Du et al. [15] proposed a new network architecture, called CRAN, which combines a recurrent neural network with convolution-based attention model and further stacks an attention-based neural model to build a hierarchical sentiment classification model. In recent years, more and more new methods have emerged in this field. Many researchers have realized the advantages of machine learning methods and applied them to the task of text sentiment analysis to improve the classification accuracy.

III. Model Description

The CE-B-MHA first generates word vectors based on text, and enters them into the Bi-LSTM network. Bi-LSTM can capture the context relationship of encoded word sequences initially. At the same time, the Multi-Head Attention mechanism in the model can capture long distance text features effectively. Finally, comparison mechanism scores the items by comparing with samples, instead of fitting the hard-to-learn patterns.

A. Word Embedding

Before text enter into the network, it needs to be converted into word vectors for computer processing. Therefore, Embedding Layer is added to encode the text. Word2Vec is a commonly used tool for training word vectors [16], which can convert a word into vector form quickly and effectively according to a given corpus. In this paper, Word2Vec is used in advance to process text and generate the required word vectors. After loading the text, the sentence is divided into words, and the stop words are removed. In the embedding layer, word vectors are read and input into the model as initialization values.

B. Bi-LSTM

We introduces LSTM network to capture contextual order information of sequences. LSTM [6] is a common method in processing sequence data. It introduces a self-cycling mechanism that makes it easier to learn long-term dependent information than a simple cycling structure.

LSTM is a special type of recurrent neural network, which has internal structure named "LSTM cell". In the model, the "gate" structure is used to realize the selective passing of information. The gate consists of a sigmoid layer with weights in the [0,1] and a multiplication operation to remove or add information to the cell state. Three "gate" structures were set in each LSTM cell: Forget Gate, Input Gate and Output Gate, performing different functions to control cell state.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (1)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (3)$$

$$\hat{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (4)$$

$$c_t = f_t * c_{t-1} + i_t * \hat{c}_t \quad (5)$$

$$h_t = o_t * \tanh(c_t) \quad (6)$$

In LSTM cell, x_t represents the input of the current cell, h_{t-1} represents the output of the previous cell, h_t represents the output of the current cell, c_{t-1} and c_t represent the previous cell state and the current cell state. f_t , i_t and o_t are the outputs of three gates.

The model uses the Forget Gate to determine what information to discard from the cell state. The gate inputs h_{t-1} and x_t , and outputs a weight between 0 and 1 multiplied by the number in the cell state c_{t-1} : 1 for "completely retained" and 0 for "completely discarded". Use the input gate to determine how to add new information to the cell state. Candidate value vectors are generated using a tanh layer and multiplied with the results of the sigmoid layer to determine the values that should be added to the cell state. Add this value to the cell state and update the old cell state to c_t . The output gate is used to determine the output value based on the cell state c_t . The cell state c_t is processed by tanh and multiplied with the result of the sigmoid layer to get the cell output h_t . LSTM can output all the hidden vectors including h_1 to h_t .

LSTM cell is abstracted as a function with inputs of h_{t-1} and x_t , outputs of h_t . which can be expressed as follows:

$$h_t = LSTM(x_t, h_{t-1}) \quad (7)$$

The bidirectional LSTM (Bi-LSTM) can extract the features of sequences from the front and back directions, and can better capture the contextual order information of sequences. In this paper, Bi-LSTM is adopted to extract local order information of text, and the forward LSTM and backward LSTM are combined to form Bi-LSTM. The forward and backward LSTM respectively corresponding to the output \vec{h}_t and \bar{h}_t .

$$\vec{h}_t = LSTM(x_t, \vec{h}_{t-1}) \quad (8)$$

$$\bar{h}_t = LSTM(x_t, \bar{h}_{t-1}) \quad (9)$$

$$h_t = Concat(\vec{h}_t, \bar{h}_t) \quad (10)$$

The inputs of forward LSTM cell are \vec{h}_{t-1} and x_t , and the output is \vec{h}_t . Similarly, the inputs of backward LSTM cell are \bar{h}_{t-1} and x_t , and the output is \bar{h}_t . Will \vec{h}_t and \bar{h}_t splicing (*Concat*) together, generate a new vector. it contains all information of forward and backward. The final output of Bi-LSTM layer is a matrix made up of vector h_1 to h_t .

C. Multi-Head Attention

Multi-Head Attention (MHA) mechanism is used to fully capture long-distance features and obtain global information. The output vectors of Bi-LSTM layer, h_1 to h_t , are combined to form a matrix, which will become the three inputs of MHA, named Q (Query), K (Key), V (Value) ($Q=K=V$), as shown in FIGURE 1.

$$Q=K=V=[h_1, h_2, \dots, h_t] \quad (11)$$

$$SDPA(Q, K, V) = \text{soft max} \left(\frac{QK^T}{\sqrt{d_K}} \right) V \quad (12)$$

The Scaled Dot-Product Attention (SPQA) compute the dot product between Q and K , and going to divide it by a scale of $\sqrt{d_K}$. Purpose of $\sqrt{d_K}$ is to play a regulatory role, so that the inner product is not too large. Soft-max operations are used to normalize the result into a probability distribution, and then multiplied by the matrix V .

$$head_i = SDPA(QW_i^Q, KW_i^K, VW_i^V) \quad (13)$$

$$MHA(Q, K, V) = Concat(head_1, \dots, head_h)W \quad (14)$$

In Multi-Head Attention, a linear transformation is performed for Q , K and V with different parameters W_i^Q , W_i^K and W_i^V for h times, and input results into Scaled Dot-Product Attention (SDPA). Each input is the original input transformed by a linear transformation. Make each result $head_i$ learn features in different representation spaces. The results $head_1$ to $head_h$ are spliced (*Concat*) after h times of Scaled Dot-Product Attention, and they are combined into a large matrix. In this way, we get the information from h different representation spaces. The spliced matrix is transformed linearly by the parameter W , and the obtained value is the result of Multi-Head Attention (MHA).

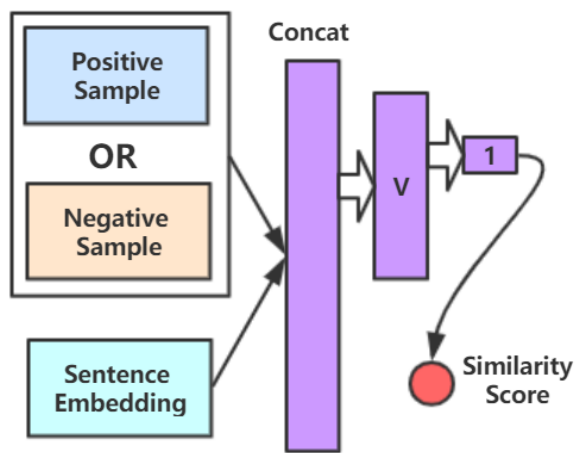


FIGURE 2. Details of Comparison Mechanism.

Bi-LSTM layer can effectively obtain the context order information of sequences; Multi-Head Attention mechanism can learn information from different dimensions and representation subspaces, and fully capture long-distance text features. They complement each other and can improve the emotional analysis ability of the model effectively.

D. Comparison Mechanism

We introduce comparison mechanism to enhance the learning ability of the model. Comparison mechanism scores the sentence embedding which generate from MHA by comparing with samples. Positive samples and negative samples are selected from the labeled training data. The samples are selected by random in this paper. The number of positive samples and negative samples should be equal to obtain better results. Corresponding sentence vectors of these samples are generated to become a part of the model.

A simple method is used to generate sentence vectors of positive and negative samples. We get word vectors of

samples from Embedding Layer, and sample vectors are obtained by averaging all the word vectors in the sentence.

We use neural network with hidden layer as a similarity function to get a similarity score for classification. The activation functions of neural networks are respectively “relu” and “sigmoid”. The input of the neural network is the connection (*Concat*) between the sentence embedding and a sample vector, and the output layer size is 1. Its hidden layer size V is the length of sentence vector.

$$Score = W_2(W_1Concat(sample, s) + b_1) + b_2 \quad (15)$$

In the formula, the neural network is represented by two linear transformations and bias, and the parameters are W_1 , b_1 , W_2 and b_2 . The s represents the sentence embedding, and $sample$ represents the sample vector.

Every sample can calculate similarity score with sentence embedding. The method to integrate them is to calculate weighted sum of the scores. The result of the Comparison Mechanism can be obtained in this way.

$$r = \sum_{i=1}^K (w_i Score_i) \quad (16)$$

In the formula, w_i represents the weight of each score. We use a layer of neural network to calculate the weights. K represents the size of samples selected.

IV. Experiments

A. Datasets

The experiment was performed on three datasets, which are Large Movie Review Dataset [17], Semeval2017-task4-A English [18] and Stanford Sentiment Treebank [19]. In the experiment, their training and test sets were directly used. 20% of the original training set was assigned as the verification set, and the rest was assigned as training set.

The Large Movie Review Dataset (IMDB) is a commonly used IMDB comment sentiment analysis dataset, which contains 25,000 positive and 25,000 negative samples. Semeval2017-task4-A English (Semeval2017) is a dataset provided by task4 of the Semeval2017 competition, containing more than 7,000 positive and 3,000 negative samples. Stanford Sentiment Treebank (SST) is a sentiment analysis dataset provided by Stanford, containing 5,000 positive and 4,500 negative samples. For the datasets with three sentiments, the neutral samples were removed.

B. Results

RNN, CNN, ON-LSTM[14], D-LSTM[7], and Bi-LSTM with attention were used as baselines. We did ablation experiments on Bi-LSTM, MHA, B-MHA, CE, and CE-B-MHA. Where B-MHA refers to Bi-LSTM with MHA, and CE refers to independent Comparison Mechanism.

We implemented the model by using the Keras, a popular deep learning tool based on Python. We use Adam[20] as

optimizer, and cross entropy as the loss. TABLE I, II and III respectively show the experimental results of the three metrics. The item with the highest score in each column is highlighted in bold.

TABLE I
THE ACC OF BASELINES AND ABLATION EXPERIMENTS

ACC	IMDB	Semeval2017	SST
RNN	0.721	0.727	0.657
CNN	0.881	0.745	0.713
ON-LSTM	0.876	0.754	0.714
D-LSTM	0.883	0.780	0.721
BiLSTM-Attention	0.880	0.749	0.745
BiLSTM	0.874	0.787	0.724
MHA	0.871	0.778	0.743
B-MHA	0.885	0.795	0.740
CE	0.881	0.801	0.729
CE-B-MHA	0.895	0.804	0.735

TABLE I shows the ACC of three datasets. For IMDB, the ACC of CE-B-MHA is the highest among all models, which is about 1.4% higher than baselines. After combining the Bi-LSTM and MHA, the effect of B-MHA is better than either one alone. After joining CE, the performance has been further improved. For Semeval2017, the ACC of CE-B-MHA is the highest among all models, which is about 3.1% higher than baselines. The ablation experiments have similar results as IMDB. For SST, the dataset is smaller and the sentences in the dataset are shorter. Overly complex models are prone to serious overfitting, and the simpler model gets better results.

TABLE II
THE AUC OF BASELINES AND ABLATION EXPERIMENTS

AUC	IMDB	Semeval2017	SST
RNN	0.806	0.802	0.730
CNN	0.935	0.881	0.832
ON-LSTM	0.964	0.899	0.840
D-LSTM	0.962	0.865	0.805
BiLSTM-Attention	0.962	0.878	0.726
BiLSTM	0.928	0.897	0.772
MHA	0.936	0.897	0.738
B-MHA	0.934	0.840	0.744
CE	0.935	0.890	0.844
CE-B-MHA	0.969	0.905	0.823

TABLE III
THE PRECISION OF BASELINES AND ABLATION EXPERIMENTS.

Precision	IMDB	Semeval2017	SST
RNN	0.708	0.593	0.663
CNN	0.899	0.657	0.720
ON-LSTM	0.915	0.632	0.673
D-LSTM	0.897	0.662	0.792
BiLSTM-Attention	0.896	0.621	0.764

BiLSTM	0.897	0.716	0.683
MHA	0.854	0.650	0.739
B-MHA	0.915	0.736	0.711
CE	0.877	0.699	0.767
CE-B-MHA	0.929	0.703	0.732

TABLE II and III respectively show the AUC and Precision of three datasets. For IMDB, the AUC and Precision of CE-B-MHA are the highest among all models. For Semeval2017, the AUC of CE-B-MHA is the highest among all models. Although CE-B-MHA can achieve good results comparing with the baselines, it is not optimal on all metrics. The enhancement effect of Comparison Mechanism on Precision is weak. This may be due to an imbalance in the number of positive and negative data.

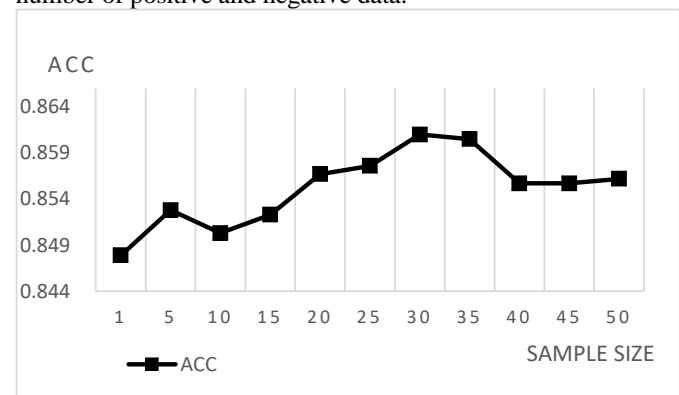


FIGURE 3. The Effect of Comparison Mechanism. We train CE-B-MHA on Semeval2017 with different sample sizes. This result is generated from the validation set.

C. The Effect of Comparison Mechanism

We tested the effects of different sample sizes on the enhancement of Comparison Mechanism. The sample size means the number of positive and negative sample pairs. FIGURE 3 shows the results of CE-B-MHA on Semeval2017 with different sample sizes. With fewer samples, the enhancement of Comparison Mechanism is not obvious. When the sample size is higher than 35, the influence of sample size on the model performance decreases.

V. Conclusion

We propose a method to analyze text sentiment named Comparison Enhanced Bi-LSTM with Multi-Head Attention (CE-B-MHA). Experiments show that the sentiment analysis effect of CE-B-MHA is improved, compared with existing classification models.

CE-B-MHA has a complex internal structure. It combines the ability of Multi-Head Attention to obtain global information with the ability of Bi-LSTM to obtain sequence information. The Bi-LSTM network is used to obtain the internal relation between the front and back directions of sentences and obtain the local order information. In addition, Multi-Head Attention mechanism is used to fully capture the features of long distance and learn relevant information from different dimensions and representation subspaces. On the

other hand, CE-B-MHA introduces comparison mechanism to enhance the learning ability of the model. Comparison mechanism scores the items by comparing with samples

instead of fitting the hard-to-learn patterns, and achieves good result.

REFERENCES

- [1] Tetsuya Nasukawa, Jeonghee Yi. "Sentiment analysis: Capturing favorability using natural language processing." *Proceedings of the 2nd international conference on Knowledge capture*. ACM, 2003: 70-77.
- [2] Bing Liu. "Sentiment analysis and opinion mining." *Synthesis lectures on human language technologies* 5.1 (2012): 1-167.
- [3] Volodymyr Mnih, Nicolas Heess, and Alex Graves. "Recurrent models of visual attention." *Advances in neural information processing systems*. 2014: 2204-2212.
- [4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *arXiv preprint arXiv:1409.0473* (2014).
- [5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser. "Attention is all you need." *Advances in neural information processing systems*. Long Beach, USA, 2017:5998-6008.
- [6] Sepp Hochreiter, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.
- [7] Christos Baziotis, Nikos Pelekis, and Christos Doukeridis. "Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis." *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. 2017: 747-754.
- [8] Janyce Wiebe, Theresa Wilson, Claire Cardie. "Annotating expressions of opinions and emotions in language." *Language resources and evaluation* 39.2-3 (2005): 165-210.
- [9] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002: 79-86.
- [10] Yoon Kim. "Convolutional neural networks for sentence classification." *arXiv preprint arXiv:1408.5882* (2014).
- [11] Raymond Brueckner and Bjorn Schuller. "Social signal classification using deep BLSTM recurrent neural networks" in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 4823-4827.
- [12] Duyu Tang, Bing Qin, Xiaocheng Feng, Ting Liu. "Effective LSTMs for target-dependent sentiment classification." *arXiv preprint arXiv:1512.01100* (2015).
- [13] Yequan Wang, Minlie Huang, Li Zhao. "Attention-based LSTM for aspect-level sentiment classification." *Proceedings of the 2016 conference on empirical methods in natural language processing*. 2016: 606-615.
- [14] Yikang Shen, Shawn Tan, Alessandro Sordani, Aaron Courville. "Ordered neurons: Integrating tree structures into recurrent neural networks." *arXiv preprint arXiv:1810.09536* (2018).
- [15] Jiachen Du, Lin Gui, Yulan He, Ruifeng Xu, Xuan Wang. Convolution-Based Neural Attention with Applications to Sentiment Classification. *IEEE ACCESS*. Vol. 7, Issue 1, 22983-27992, 2019.
- [16] Tomas Mikolov, Kai Chen, Greg Corrado, Jeff Dean. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).
- [17] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. "Learning word vectors for sentiment analysis." *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*. Association for Computational Linguistics, 2011: 142-150.
- [18] Sara Rosenthal, Noura Farra, and Preslav Nakov. "SemEval-2017 task 4: Sentiment analysis in Twitter." *arXiv preprint arXiv:1912.00741* (2019).
- [19] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang. "Recursive deep models for semantic compositionality over a sentiment treebank." *Proceedings of the 2013 conference on empirical methods in natural language processing*. 2013: 1631-1642.
- [20] Diederik Kingma and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980* (2014).