

Research Article

An Improved BERT and Syntactic Dependency Representation Model for Sentiment Analysis

Wenfeng Liu,¹ Jing Yi ,² Zhanliang Hu,¹ and Yaling Gao¹

¹School of Computer, Heze University, Heze 274015, China

²School of Computer Science and Technology, Shandong Jianzhu University, Jinan 250101, China

Correspondence should be addressed to Jing Yi; yjzyj2006@163.com

Received 9 September 2021; Accepted 11 March 2022; Published 5 May 2022

Academic Editor: Amparo Alonso-Betanzos

Copyright © 2022 Wenfeng Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Text representation of social media is an important task for users' sentiment analysis. Utilizing the better representation, we can accurately acquire the real semantic information expressed by online users. However, existing works cannot achieve the best results. In this paper, we construct and implement a sentiment analysis model based on the improved BERT and syntactic dependency. Firstly, by studying the word embeddings of BERT, we have ameliorated the embeddings representation. Attention mechanism is added to the word embeddings, sentence embeddings, and position embeddings. Secondly, we have exploited the dependency syntax analysis of the text, and the dependency relationship of different syntactic components will be obtained. For different syntactic components, the hierarchical attention mechanism is used to construct the phrase embeddings or block embeddings. Finally, we splice the syntactic blocks for sentiment analysis. Extensive experiments show that the proposed model has a stronger ability than the baselines on two standard data sets.

1. Introduction

In recent years, with the popularization of social media such as WeChat, Face Book, Twitter, and Fetion, these media are changing people's lifestyles and habits. How to represent the text and understand their semantic information accurately is an important task. However, existing works cannot achieve the best results. In general, the composition of a text can be subdivided into paragraph-level, sentence-level, and word-level. The words are basic components, and the representation of text can be divided into a series of word combinations. Therefore, researching on the word-level representation is extremely important compared with the other two.

With the innovation of hardware technology, we can do a large number of calculations or parameter learning. However, how to integrate more semantic information on text representation is an important and difficult task for natural language processing. Harris has put forward an important idea on text representation as early as the 1950s, which is the famous distributed hypothesis: words with similar contexts have similar semantics. Firth elaborated

Harris' thoughts a few years later. A more direct expression is that the semantic information about a word is mainly determined by its context. In the last ten years, the computing capability has been greatly improved, especially the wide application of GPUs and TPUs, which have made the analysis, calculation, and processing of big data easier.

The contributions of our paper are as follows:

- (1) We have improved BERT (iBERT) to obtain a better representation. Respectively, the Token Embeddings (TEs), Segment Embeddings (SEs), and Position Embeddings (PEs) have different attention weights.
- (2) We have constructed the syntax tree based on syntactic dependency of block embeddings.
- (3) Combining with the attention mechanism, we have constructed the embeddings representation of the text.

2. Related Works

The expression of any language can be divided into several levels, such as paragraph-level, sentence-level,

and word-level. The basic unit of meaningful representation is word-level. There are two methods for the vectorized representation about words, one is the One-Hot model and the other is the Distributed Representation model.

The idea of One-Hot representation is very simple. The dimension of word embeddings is measured by the number of words appeared; that is, the dimension of each word is equal to the total number of words. Only the position where the word has appeared is represented by 1, and the remaining positions are represented by 0. For instance, the word embeddings of “computer” and “PC” are [0, 0, 1, 0, 0, 0, 0] and [0, 0, 0, 0, 1, 0], respectively. As we all know, the two words have the same meaning. Nonetheless, the similarity between them is zero. Therefore, One-Hot representation cannot express the similarity of words. If the amount of data is increased, it is prone to dimension disasters. Therefore, many applications have adopted the Distributed Representation model.

2.1. Distributed Representation. To acquire the semantic information about words and alleviate a series of problems in depth, there are two classic models, Word2Vec and BERT. In 2003, Bengio et al. [1] proposed the NNLM model, which obtained the word embeddings when training and constructing a language model. On this basis, Mikolov et al. [2] proposed the Word2Vec which contained two models (Continuous Bag-of-Words and Skip-gram) in 2013. The CBOW model used the context to predict the current word, while the SkipGram model used the current word to predict the context.

In 2018, Devlin et al. [3] proposed the BERT model (Bidirectional Encoder Representations from Transformers), which is another substantial achievement after Word2Vec. And it has achieved the optimal results on 11 tasks in natural language processing. This achievement also proved the importance of the two-way and pretraining model for text representation. Many related models have appeared one after another, such as SpanBert [4], RoBERTa, and XLNet [5]. To further improve the text language processing effect, a convolutional neural network model, Hybrid convolutional neural network (CNN), and Long Short-Term Memory (LSTM) based on the fusion of text features and language knowledge were proposed [6]. Chen et al. [7] proposed a new representation learning method combined with variational autoencoder (VAE) and density-based spatial clustering of applications with noise (DBSCAN).

2.2. Coarse-Grained Semantic Representation. Combining textual semantics, we can construct larger granularity of text representation, such as grammatical blocks, sentence-level, and document-level. The Paragraph Embeddings [8] and the Skip-Thoughts were the influential models. Paragraph Embeddings consisted of two submodels. One was to evaluate the central word-by-topic embeddings and context information. The other used paragraph or sentence level evaluated the probability of words. However, Skip-Thoughts had an integrated encoder and decoder which modeled the context-related topics of physically adjacent sentences.

Furthermore, to achieve accurate semantic information in multiple documents, Lin et al. [9] proposed a semantic search model for knowledge documents. Yan and Gao [10] studied the coupling of internal topics and topological structure, and they modeled large-grained semantics. Wu et al. [11] proposed a multigranularity and cross-text semantic matching method by a deep neural network, which had obtained better results in the text matching field.

In recent years, due to the wide application of deep learning in text processing, the combination of multiple models (such as RNN, CNN, LSTM, GRU, Transformer, and BERT) is very widely used. Sun et al. [12] proposed a secure indoor crowdsourced localization system, BERT-ADLOC, which was based on BLE fingerprints. The system consisted of two main parts: adversarial sample discriminator BERT-AD and indoor localization model BERT-LOC. Jiang and He [13] had presented an attention mechanism that differentiated the focus on the output of ResNet and the long short-term memory for the features of the sequences. Alahmadi et al. [14] proposed a smartphone-based periocular recognition which used a deep convolutional neural network and collaborative representation. Cross-modal convolution could enable the use of efficient CNN-style layers for multimodal sequential models.

In addition, other models which have obtained excellent performance in image fields have been gradually migrated and applied to some subtasks of text processing. The convolutional models, which have combined words and phrases, have achieved better results in classification and sentiment analysis [15].

3. The Text Representation Model of Online Social Media

Any language has its corresponding language features and grammatical rules which are the key requirements for the meaningful expression. Therefore, making use of the word embeddings and grammatical structure, we can construct a better semantic representation of the text. Firstly, given a text in social media, it is necessary to preprocess the text (such as word segmentation and part-of-speech). Secondly, we have proposed the improved BERT model to obtain better word embeddings and utilized the direct dependencies of words to build a dependency tree. Finally, the improved BERT model (iBERT) and dependency trees are used to construct the semantic representation of the text. The framework is shown in Figure 1.

3.1. Word Embeddings Based on the iBERT. BERT obtains the input embeddings by summing multiple embeddings. These embeddings include the Token Embeddings (TEs), Segment Embeddings (SEs), and Position Embeddings (PEs). We have improved the BERT. The final inputs are represented by attention summation of the three embeddings, as shown in the following equation:

$$\text{Input}_{\text{EmbeddingsBERT}} = \alpha * E_{\text{TE}} + \beta * E_{\text{SE}} + \gamma * E_{\text{PE}}, \quad (1)$$

where α , β , and γ are the attention weights of TE, SE, and PE.

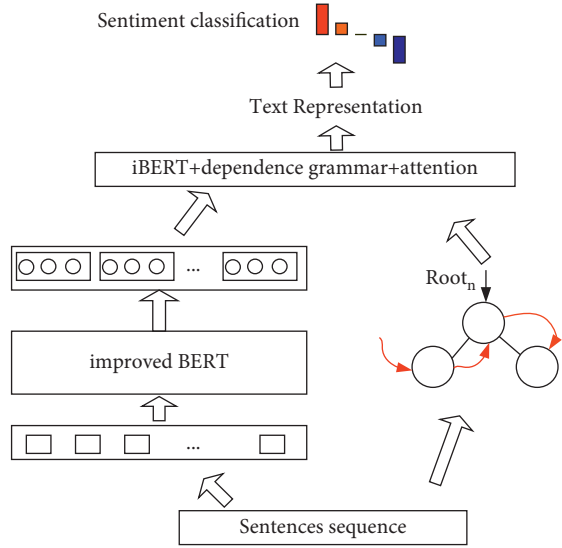


FIGURE 1: The proposed model.

As shown in Figure 2, $E_{TE} \in \mathbb{R}^{N \times d_{\text{model}}}$, $E_{PE} \in \mathbb{R}^{N \times d_{\text{model}}}$, N is the length of the input sequence, and d_{model} is the dimension of the word embeddings.

Position Embeddings $E_0, E_1, E_2, E_3, \dots, E_n$ are obtained by equations (2) and (3). To facilitate comparison with the standard BERT, our paper adopts the same formulas as the official.

$$E_{PE(\text{pos}, 2i)} = \sin\left(\frac{\text{pos}}{(10000^{2i/d_{\text{model}}})}\right), \quad (2)$$

$$E_{PE(\text{pos}, 2i)} = \cos\left(\frac{\text{pos}}{(10000^{2i/d_{\text{model}}})}\right), \quad (3)$$

where pos denotes the position number of the word in the input sequence. The word in the even position is calculated by equation (2) (in the odds by equation (3)).

The overall framework of the BERT model utilizes the officially released structure. Transformer that belongs to the encoder-decoder architecture uses a two-way and self-attention mechanism. The main operations of the encoder in Transformer module are the following equations:

$$e_{\text{mid}} = \text{ayer Norm}(e_{\text{in}}) + \text{Multi Head Attention}(e_{\text{in}}), \quad (4)$$

$$e_{\text{out}} = \text{Layer Norm}(e_{\text{mid}} + \text{FFN}(e_{\text{mid}})), \quad (5)$$

where $e_{\text{in}} \in \mathbb{R}^{N \times d}$ denotes the input of the encoder and $e_{\text{out}} \in \mathbb{R}^{N \times d}$ denotes the output of the encoder. Multi Head Attention (\cdot) represents a multiheaded attention mechanism. FFN (\cdot) is a feedforward neural network. Layer Norm (\cdot) represents layer normalization.

In the Transformer module of iBERT, the main operations of the decoder are as follows:

$$\begin{aligned} d_{\text{mid1}} &= \text{LayerNorm}(d_{\text{in}} + \text{MaskedMultiHeadAttention}(d_{\text{in}})), \\ d_{\text{mid2}} &= \text{LayerNorm}(d_{\text{mid1}} + \text{MaskedM_H_Attention}(d_{\text{mid1}}, e_{\text{out}})), \\ d_{\text{out}} &= \text{LayerNorm}(d_{\text{mid2}} + \text{FFN}(d_{\text{mid2}})), \end{aligned} \quad (6)$$

where $d_{\text{in}} \in \mathbb{R}^{M \times d}$ denotes the input of the decoder. $d_{\text{out}} \in \mathbb{R}^{M \times d}$ denotes the output of the decoder. Multi Head Attention (\cdot), Layer Norm (\cdot), and FFN (\cdot) represent the same functions as those of the encoder. Masked Multi Head Attention (\cdot) is a masked and multihead attention mechanism.

3.2. Syntax Tree Construction Based on Syntactic Dependency.

The syntactic tree of a sentence is an interdependence graph of its words which determine their importance by the distance from the central word. Andor et al. [16] proposed a transformation-based dependency syntax analysis method. And they developed the SyntaxNet (<http://github.com/tensorflow/models/tree/master/syntaxnet>) system, which was the most popular construction method of the syntax tree. Through researching this system and making corresponding improvements, we have adopted a generation scheme for the syntax tree based on the arc transformation. This method uses a stack (STACK), buffer (BUFFER), and set (ARC_SET) [17]. $s_1 s_2 \dots s_j \dots s_n$ is a given text; s_j is the j th word. The execution is that the STACK only has the root node at the beginning. The ARC_SET is an empty set, while

the BUFFER saves the word sequence of input. There are three operations, LEFT_ARC, RIGHT_ARC, and SHIFT (see Algorithm 1). The LEFT_ARC operation is that the current word in the buffer will be added a left arc to the word on top of stack, the RIGHT_ARC operation will add a right arc as LEFT_ARC does, and the SHIFT operation will transfer the current word into stack. Until all words in the buffer are all processed, the state of STACK is consistent with the initial, and the construction of the syntax tree has been completed. Figure 3 is the dependency tree constructed by this method of the sentence “a woman washed the dishes.”

3.3. Text Representation Based on iBERT and Syntactic Dependence.

The dependency tree is constructed by Algorithm 1. According to the different attention weights in different syntactic positions, we combine and splice them into the corresponding text semantic representation. For a sentence $S = [s_1, s_2, \dots, s_p, \dots, s_n]$, s_i is the i -th word of sentence S . Embeddings (s_i) represents the word embeddings obtained by iBERT, and Attentions (s_i) is the weight of s_i obtained by grammatical analysis. It satisfies the normalization, as shown in the following equation:

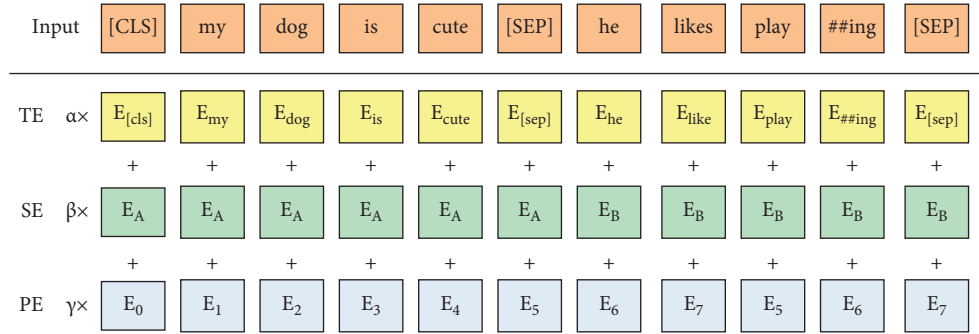


FIGURE 2: The improved word embeddings model.

```

//initial configuration
STACK stack = [root]
BUFFER buffer_words = [w_1, . . . , w_n]
POS pos = [w_1.pos, . . . , w_n.pos] // POS = {NN, JJ, VBZ, . . .}
ARCS arc_set = {empty} // ARCS = {dobj, amod, nsubj, . . .}
while (the buffer is not empty and the stack contains more than one node)
//The following operations are performed according to the words and their POS of the top stack and the buffer
    if (the shift will be used in the transition)
        then { run op_shift operation }
    if (the left arc will be used in the transition)
        then { run op_left_arc operation;
              add a left arc about the two words to the arc_set.}
    if (the right arc will be used in the transition)
        then { run op_right_arc operation;
              add a right arc about the two words to the arc_set.}
}
} //the end.

```

ALGORITHM 1: ARC_model ().

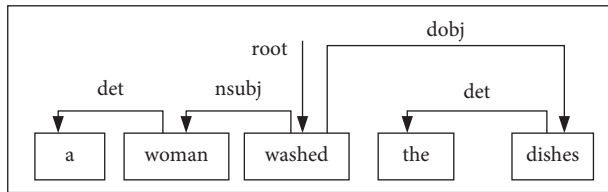


FIGURE 3: An example of a syntactic tree.

$$\sum_{i=1}^n \text{Attentions}(s_i) = 1. \quad (7)$$

Combining the attention mechanism and word embeddings, we construct sentence embeddings, as shown in equation (8). $\text{Attentioned_Embeddings}_{s_i}$ is abbreviated as $A_E_{s_i}$.

$$A_E_{s_i} = \text{Embeddings}(s_i) * \text{Attentions}(s_i). \quad (8)$$

According to the constructed syntactic tree, which contains the dependency relationship between words, we can construct the phrase embeddings of the syntax blocks (equations (9) and (10)). The phrase embeddings are attention_weighted of their words, as shown in Figure 4. Where

$\text{Represent}_{\text{sentence}}$ denotes the semantic embeddings and $\text{phrase}_{\text{vector}_r}$ represents the syntactic elements in the sentence which are mainly involving the subject, predicate, object, and other syntactic elements.

$$\text{phrase}_{\text{vector}_m} = \frac{1}{k} * \sum_j A_E_j. \quad (9)$$

$$\text{Represent}_{\text{sentence}} = [\text{phrase}_{\text{vector}_1}, \dots, \text{phrase}_{\text{vector}_r}, \dots, \text{phrase}_{\text{vector}_p}], \quad (10)$$

3.4. The Sentiment Analysis Model Based on iBERT and Syntactic Dependency. To verify the effectiveness of the proposed model, we construct a text sentiment model in this section, as shown in Figure 5.

We denote the text embeddings as $\text{Text}_1, \text{Text}_2, \dots, \text{Text}_n$. The stage from the vectorized representation to the sentiment categories is a fully connected network. Parameter weight W is obtained after training, and this matrix is locked (or fixed) during the test. Sentiment categories $C = \{c_1, c_2, \dots, c_k\}$. k is the total number of categories in the sentiment

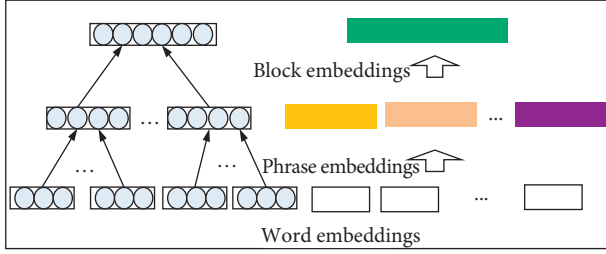


FIGURE 4: The block embeddings of Syntactic tree.

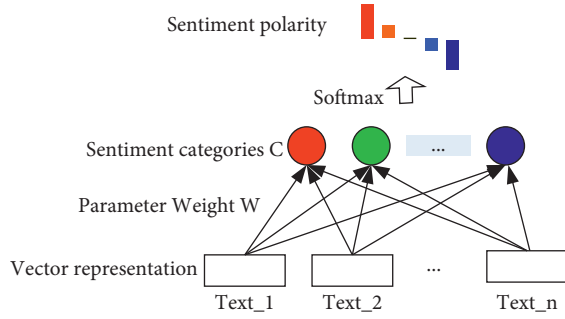


FIGURE 5: The sentiment analysis model.

classification, and the probability P_c that belongs to a certain category is obtained by the following formula:

$$P_c = W \cdot TE \quad (11)$$

$$= \{P_{c_1}, P_{c_2}, \dots, P_{c_k}\}.$$

We use the softmax function to normalize and obtain the category with the highest probability, as shown in the following equation:

$$P_{c_{\max}} = \text{softmax}\{P_{c_1}, P_{c_2}, \dots, P_{c_k}\}. \quad (12)$$

The cross-entropy Loss_{*i*} is used to train the model, and the formula is shown in the following equation:

$$\text{Loss}_i = -\frac{1}{k} \sum_m y_{im} \log(P_{c_{im}}), \quad (13)$$

where y_{im} represents the probability that the i -th sample belongs to the m -th class ($m \in C$). If it belongs to the m -class, y_{im} is 1; otherwise, it is zero. $P_{c_{im}}$ represents the prediction probability of the i -th sample belonging to the m -th category. To ensure obtaining a more robust model, we have utilized a dropout strategy. The dropout is used in a fully connected network with the vectorized representation TE to sentiment category C, and the value of dropout is set to 0.5.

4. Experiments and Results

4.1. Data Set and Evaluations

4.1.1. Data Set. The first data set is task 4 in SemEval 2014, which contains two subdata sets, one is the Laptop and the other is the Restaurant. Their format is described by XML. In the Laptop, the number of sentences in training is 3045 and in the test is 800. In the Restaurant, the number of sentences

in training is 3041, and the number of sentences in the test is 800.

Another data set is Subtask A [18] in SemEval 2017, which is mainly used for SDQC support and rumor classification. The classification of the training or testing is shown in Table 1.

S, D, Q, and C, respectively, represent the four categories, which are the support category (Support), the objection category (Deny), the doubt category (Query), and the irrelevant comment category (Comment). Category S denotes supporting related content. Category D represents the opposing related content. Category Q owns questions about related content, and category C expresses comments that have nothing to do with related content or themes.

4.1.2. Evaluation. We have used the accuracy (AC) for evaluation of the experiments as shown in the following equation:

$$AC = \frac{TP + TN}{TP + TN + FP + FN} \quad (14)$$

TP (True Positive) indicates that the predicted (positive) is consistent with the actual (positive). FP (False Positive) denotes that the predicted (positive) is inconsistent with the actual (negative). TN (True Negative) represents that the predicted (negative) is consistent with the actual (negative). FN (False Negative) indicates that the predicted (negative) is inconsistent with the actual (positive).

4.2. Parameter Settings. In the learning stage of word embeddings, the number of layers used is 12 (num_hid den_layers = 12). The number of neurons in the hidden layer of the neural network is 768 (hidden_size = 768), and the length of the input text is uniformly set to 512 characters (num_hidden_layers = 512), the dropout is set to 0.1 (attention_probs_dropout_prob = 0.1), the activation function that used is gelu function (hidden_act = "gelu"), and the number of parameters is about 110 M.

During the construction of the syntactic tree, we use the default parameters in the SyntaxNet system, the small batch size of the syntax analyzer is 32 (parser_batch_size = 32), the learning rate is 0.08 (learning_rate = 0.08), and the momentum is 0.85 (momentum = 0.85)).

4.3. Baselines. The comparison models are as follows:

- (1) TLSTM [19] divides words into two subsequences, one subsequence is from left to right and the other is from right to left, so two different embeddings will be obtained. The two embeddings coalesce into the final embeddings.
- (2) Att-LSTM [20] has utilized the attention mechanism in which the words have different attention weights. The text representation is constructed by the weighted words.
- (3) CABSAs [21] combines the cyclic neural network, RNN, attention mechanism, and the memory

TABLE 1: The number of training and testing on SemEval 2017.

	S	D	Q	C
Training	910	344	358	2907
Testing	94	71	106	778

network to acquire the representation through different directions in the sentence.

- (4) AGCN [22] uses two gated-based convolutional neural networks. They can obtain different representations, and the gated mechanism can learn the relational information of words.
- (5) BERT [3] utilizes the transformer as a submodule and obtains word embeddings by a two-way mechanism.
- (6) GCNDA [23] obtains the weight of words by combining the graphed attention mechanism, and it has two attentions, global and local.

Since there are fewer available comparison models in Subtask A, this paper uses eight models in the system which are released for comparison experiments.

4.4. Experiment Results

4.4.1. Parameters α , β , and γ . α , β , and γ , which, respectively, denote the parameters of word embeddings, sentence embeddings, and position embeddings, take the same value in the iBERT model. For better verifying the effects, we have fixed one parameter and adjust the other two. Task_1 in BERT is used for measurement between the new embeddings and the standard word embeddings. During the experiments, the parameters are normalized by $\alpha + \beta + \gamma = 1$.

For example, α is {0, 0.2, 0.4, 0.6, 0.8, 1}. β is fixed to 1, and γ is {1, 0.8, 0.6, 0.4, 0.2, 0}, respectively. After normalization, the values of parameter α are {0, 0.1, 0.2, 0.3, 0.4, 0.5}, β is 0.5, and γ is {0.5, 0.4, 0.3, 0.2, 0.1, 0}. The settings of the three parameters are shown in Table 2.

As shown in Figures 6(a)–6(c), the parameters alpha, beta, and gamma refer to α , β , and γ , respectively. After in-depth analysis of the composition of embeddings, the weight of the word is relatively high, followed by the sentence embeddings and the position embeddings. With fixed position embeddings, the final effect is gradually improved, and the main reason is that part of the word information is contained in the sentence embeddings. The composition of the sentence embeddings can be regarded as embeddings with larger granularity. And all words in the same sentence are used with the same sentence embeddings. To a certain extent, it weakens the representation of word embeddings in the same sentence. However, from another perspective, the word embeddings added to the sentence have a degree of distinction between sentences. Therefore, the sentence embeddings are meaningful in the sentence representation. Simultaneously, the calculation of the position embeddings is obtained by equations (2) and (3), which is the empirical formula of BERT team. The main reason is that the different position has different weights for the composed embeddings.

TABLE 2: Experiments of parameters α , β , and γ .

α	Group_1			Group_2			Group_3		
	β	γ	α	β	γ	α	β	γ	
0.5	0	0.5	0	0.5	0.5	0	0.5	0.5	
0.5	0.1	0.4	0.1	0.5	0.4	0.1	0.4	0.5	
0.5	0.2	0.3	0.2	0.5	0.3	0.2	0.3	0.5	
0.5	0.3	0.2	0.3	0.5	0.2	0.3	0.2	0.5	
0.5	0.4	0.1	0.4	0.5	0.1	0.4	0.1	0.5	
0.5	0.5	0	0.5	0.5	0	0.5	0	0.5	

After a number of experimental analyses, when α , β , and γ are, respectively, 0.65, 0.20, and 0.15, better word embeddings can be obtained.

4.4.2. Experimental Results on SemEval 2014. From the results in Table 3 and Figure 7, we can get the following conclusions. The TLSTM model has the lowest accuracy among all baselines. The main reason is that this model only considers part of the content and ignores the representation of deep features. The Att-LSTM model can capture the deep features through the long short-term memory network. Simultaneously, it combines the attention mechanism to obtain the relationship of words in different locations. Hence, this model is more accurate than the TLSTM model. The CABSA model uses a memory network, and the effect of memory-based network is better than seq2seq-related models. The CABSA model can memorize the preceding or subsequent text feature through the memory network. So, this method has achieved a certain improvement to the previous two models.

Because the AGCN model has used two gated convolutional networks, the relationship of words can be obtained to a certain extent, but the syntactic structure cannot be captured commendably. Since BERT is an excellent and pretraining model in recent years, expression ability of word embeddings can be optimized, but the word embeddings constructed by the addition of embeddings can weaken the characteristic information such as syntactic structure. The GCNDA model is the best model among the baselines. The main reason is that this model has used a graphed convolution and combined with an attention mechanism, so that this model can obtain part of the structured information.

Our proposed model is BDPT, which combines the improved BERT and the syntactic structure. It has used the attention mechanism and combined syntactic blocks to construct a combinative text representation. Therefore, our method can obtain a deep-level representation of semantic information, and it has achieved higher precision in the classification task of sentiment analysis. Compared with the best model in baselines, our model has improved by 2.1% on the Restaurant and 1.9% on the Laptop.

Specifically, the time (seconds per ten sentences) consumed by BDPT is also the lowest (in Figure 8). Through in-depth analysis, the time complexity of TLSTM is $O(n * m + n * n + n)$, n denotes Hidden_size, and m represents input_size. Att-LSTM adds a weight matrix, and the time complexity is $O(n * m + n * n + n + a * a)$, a denotes the

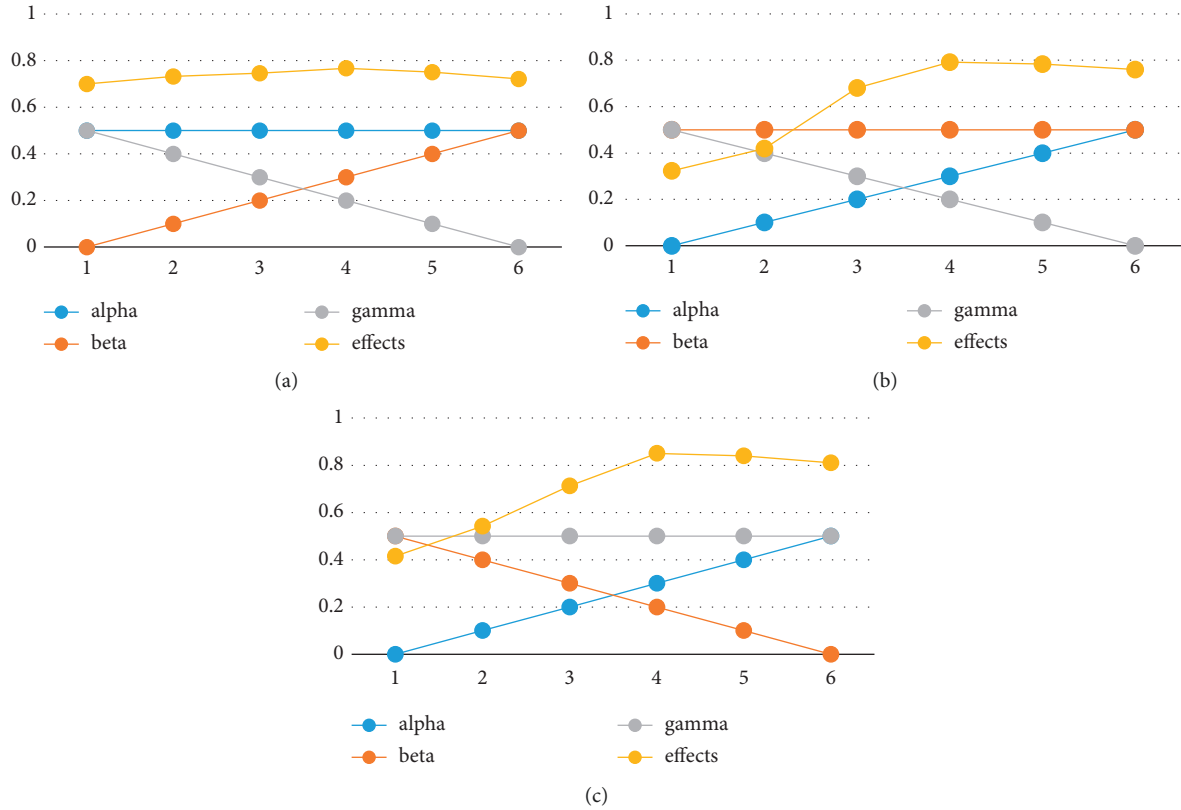


FIGURE 6: Experimentson α , β , and γ : (a) fix α ; (b) fix β ; (c) fix γ .

TABLE 3: The accuracy on the Restaurant and Laptop.

Model	Restaurant	Laptop
TLSTM	0.752	0.668
Att-LSTM	0.761	0.672
CABSA	0.771	0.687
AGCN	0.772	0.688
BERT	0.782	0.694
GCNDA	0.789	0.702
BDPT	0.810	0.721

attention weight matrix. AGCN is equivalent to having two gates, and it is close to double times of TLSTM. GCNDA adopts a four-layer network structure and involves the relationship matrix between words, so the time consumption is close to CABSA. Among all the comparison models, BERT has the lowest time complexity. The main reason is that its pretraining time is not taken into account.

4.4.3. Results on SemEval 2017. Results on SemEval 2017 are shown in Table 4 and Figure 9. The DFKI_DKT model only uses sparse word embeddings as input, which has achieved the worst effect among all comparison models. The IITP model uses pairs of the original text and its response as the input. The IKM model uses the convolutional neural network to obtain the text representation, and it uses the softmax classifier to assign the probability that each category belongs to. IITP and NileTMRG are implemented by linear and polynomial kernel classifiers, respectively, while they are

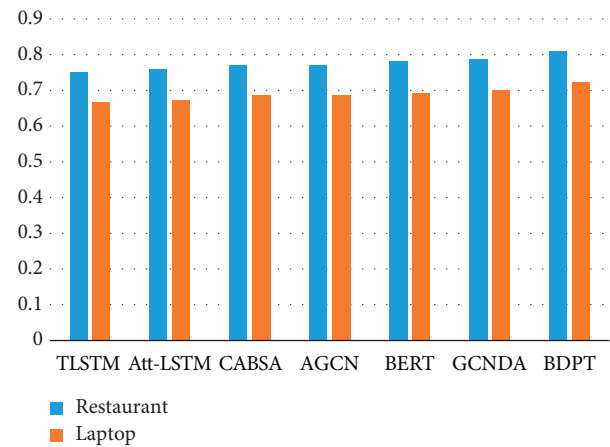


FIGURE 7: Model comparison diagram.

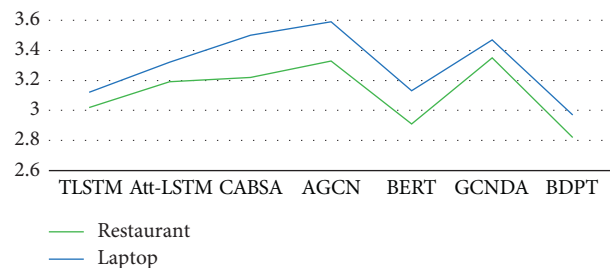


FIGURE 8: Elapsed time for different models.

TABLE 4: Results on SemEval 2017 (Subtask A).

Models	AC
DFKI_DKT	0.635
IITP	0.641
IKM	0.701
NileTMRG	0.709
MamaEdha	0.749
ECNU	0.778
Uwaterloo	0.78
Turing	0.784
BDPT	0.799

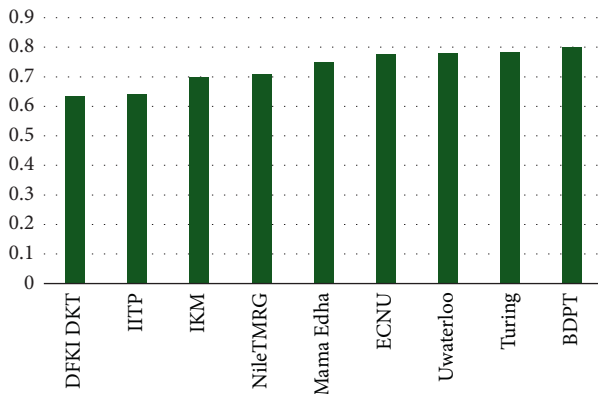


FIGURE 9: Results on SemEval 2017 (Subtask A).

less effective. The MamaEdha model has mixed and used a variety of neural networks as classifiers. The ECNU system has solved the problem of information imbalance by decomposing it into a two-step classification task. DFKI-DKT, MamaEdha, ECNU, and UWaterloo use integrated classifiers, and the results of the classification are obtained through a voting mechanism. The three models, DFKI-DKT, ECNU, and MamaEdha, use the mixture of deep learning, machine learning, and manual rules to assign different labels with different weights.

All these compared models have used carefully designed feature engineering. IITP, NileTMRG, ECNU, and UWaterloo have utilized keywords and key sentences, as well as features in the Tweet (such as metadata, tags, and keywords for specific events). IKM and MamaEdha have used fewer features and exploited the word embeddings obtained from the CCN network.

The Turing model uses the LSTM network to implement sequence-to-sequence classification. This model comprehensively considers the word embeddings, punctuation embeddings, and the similarity between words, and it has incorporated more feature information. Consequently, it has obtained the best result in baselines. Compared with all the baselines, our proposed method has incorporated more in-depth features (such as improved BERT and syntactic dependency trees). And it has achieved the better result (1.5% higher than the best baseline). Further, our model has a better representation than all of them because syntactic structure plays a very important role in the text representation too. At the same time, our model takes the least amount of processing time.

5. Conclusions and Future Work

How to represent the text better is an important task in data mining and data analysis. This paper combines the existing research results and conducts a further study. In addition, we have proposed a novel model which has combined the improved BERT and grammatical dependency structure. Incorporating the deep semantic features into text representation, we can obtain a better sentiment analysis model. First of all, we have constructed a better text representation by studying the grammatical structure and iBERT. Then, we construct the syntactic dependency graph of words. Finally, extensive experiments have been performed on SemEval 2014 and SemEval 2017. Our model has achieved the state-of-art. Experiments show that syntactic structure plays an essential role in the text representation. The next step is to combine more deep-level features (such as the syntactic structure combined graph convolutional neural networks) for researching text and image sentiment analysis.

Data Availability

The data and the authors' source code used to support the findings of this study will be available at <https://alt.qcri.org/semeval2017> (semeval2014) and <https://gitee.com/hzxylwf/model>.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the Shandong Province Social Science Popularization and Application Research Project under Grant 2020-SKZZ-51, in part by the Social Science Planning Office of Heze City under Grant 2020_zz_55, in part by the Heze University Doctoral Research and Development Fund under Grant XY20BS19, in part by the Shandong Province Educational Science Planning under Grant BYZN201910, in part by the Shandong Province Social Science Planning Project under Grant 20CDCJ01, in part by the NSFC-Xinjiang Joint Fund under Grant U1903127, and in part by the Natural Science Foundation of Shandong Province under Grant ZR2020MF052.

References

- [1] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, no. 6, pp. 1137–1155, 2003.
- [2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, <https://arxiv.org/abs/1301.3781>.
- [3] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for*

- Computational Linguistics*, pp. 4171–4186, NAACL-HLT, Minneapolis, MIN, USA, June 2019.
- [4] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, “Spanbert: improving pre-training by representing and predicting spans,” *Transactions of the Association for Computational Linguistics ACL*, vol. 8, pp. 64–77, 2020.
- [5] Z. Yang, Z. Dai, Y. Yang, and J. Carbonell, “Xlnet: generalized autoregressive pretraining for language understanding,” *NeurIPS*, vol. 32, pp. 1–11, 2019.
- [6] S. Zhang, “language processing model construction and simulation based on Hybrid CNN and LSTM,” *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 2578422, 11 pages, 2021.
- [7] F. Chen, T. Zhang, and R. Liu, “An Active Learning Method Based on Variational Autoencoder and DBSCAN Clustering,” *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 9952596, 11 pages, 2021.
- [8] Q. V. Le and T. Mikolov, “Distributed representations of sentences and documents,” 2014, <https://arxiv.org/abs/1405.4053>.
- [9] Z. Lin, Y. Zou, and J. Zhao, “Software text semantic search approach based on code structure knowledge,” *Journal of Software*, vol. 12, pp. 3714–3729, 2019.
- [10] R. Yan and G. Gao, “Pseudo topic analysis based on topic network,” *Journal of Chinese Information Processing*, vol. 32, pp. 100–108, 2018.
- [11] S. Wu, D. Peng, W. Yuan, C. Zhan, and L. Cong, “MGSC: a multi-granularity semantic cross model for matching short texts,” *Journal of Chinese Computer Systems*, vol. 40, pp. 1148–1152, 2019.
- [12] X. Sun, H. Ai, J. Tao, and T. Y. Hu, “Bert-adloc: a secure crowdsourced indoor localization system based on BLE fingerprints,” *Applied Soft Computing*, vol. 104, Article ID 107237, 2021.
- [13] D. Jiang and J. He, “Text semantic classification of long discourses based on neural networks with improved focal loss,” *Computational Intelligence and Neuroscience*, vol. 20219 pages, Article ID 8845362, 2021.
- [14] A. Alahmadi, M. Hussain, H. Aboalsamh, and A. Azmi, “ConvSRC: SmartPhone-based periocular recognition using deep convolutional neural network and sparsity augmented collaborative representation,” *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 3, pp. 3041–3057, 2020.
- [15] Y. Kim, “Convolutional neural networks for sentence classification,” 2014, <https://arxiv.org/abs/1408.5882>.
- [16] D. Andor, C. Alberti, D. Weiss et al., “Globally normalized transition-based neural networks,” 2016, <https://arxiv.org/abs/1603.06042>.
- [17] W. Liu, P. Liu, Y. Yang, and J. Z. Yi, “A embedding model for text classification,” *Expert Systems*, vol. 36, no. 6, pp. 1–8, 2019.
- [18] L. Derczynski, K. Bontcheva, M. Liakata, R. Procter, G. W. Sak Hoi, and A. Zubiaga, “SemEval-2017 Task 8: RumourEval: determining rumour veracity and support for rumours,” 2017, <https://arxiv.org/abs/1704.05972>.
- [19] D. Tang, B. Qin, X. Feng, and T. Liu, “Target-dependent sentiment classification with long short term memory,” 2015, <https://arxiv.org/abs/1512.01100v1>.
- [20] Y. Wang, M. Huang, and L. Zhao, “Attention-based LSTM for Aspect-Level Sentiment Classification,” in *Proceedings of the EMNLP 2016: Conference on Empirical Methods in Natural Language Processing*, pp. 606–615, EMNLP, Austin, Texas, USA, November 2016.
- [21] Q. Liu, H. Zhang, and Y. Zeng, “Content Attention Model for Aspect Based Sentiment Analysis,” in *Proceedings of the web conferences WWW*, pp. 1023–1032, Lyon, France, April 2018.
- [22] X. Wei and L. Tao, “Aspect based sentiment analysis with gated convolutional networks,” 2018, <https://arxiv.org/abs/1805.07043>.
- [23] J. Chen, H. Hou, Y. Ji, and J. Gao, “Graph convolutional networks with structural attention model for aspect based sentiment analysis,” in *Proceedings of the International Joint Conference on Neural Networks*, pp. 1–7, IJCNN, Budapest, Hungary, July 2019.