

## *Retraction*

# **Retracted: BRFP: An Efficient and Universal Sentence Embedding Learning Model Method Supporting Fused Syntax Combined with Graph Embedding Representation Algorithm**

### **Journal of Sensors**

Received 19 December 2023; Accepted 19 December 2023; Published 20 December 2023

Copyright © 2023 Journal of Sensors. 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.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

### **References**

- [1] Z. Li, W. Wu, and C. Shen, "BRFP: An Efficient and Universal Sentence Embedding Learning Model Method Supporting Fused Syntax Combined with Graph Embedding Representation Algorithm," *Journal of Sensors*, vol. 2022, Article ID 7471408, 14 pages, 2022.

## Research Article

# BRFP: An Efficient and Universal Sentence Embedding Learning Model Method Supporting Fused Syntax Combined with Graph Embedding Representation Algorithm

Zhifeng Li <sup>1</sup>, Wenwei Wu <sup>2,3</sup> and Chunlei Shen<sup>1</sup>

<sup>1</sup>The Information School, Guangdong University of Finance and Economic, Guangzhou 510320, China

<sup>2</sup>The Department of Computer Science, Hong Kong Baptist University, Hong Kong 999077, China

<sup>3</sup>Guangdong Rural Credit Union, Guangzhou 510627, China

Correspondence should be addressed to Zhifeng Li; [lizhifeng@gdufe.edu.cn](mailto:lizhifeng@gdufe.edu.cn) and Wenwei Wu; [howquax@life.hkbu.edu.hk](mailto:howquax@life.hkbu.edu.hk)

Received 21 June 2022; Revised 12 July 2022; Accepted 28 July 2022; Published 17 August 2022

Academic Editor: Yuan Li

Copyright © 2022 Zhifeng Li 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.

Due to the rapidly growing volume of data on the Internet, the methods of efficiently and accurately processing massive text information have been the focus of research. In natural language processing theory, sentence embedding representation is an important method. This paper proposes a new sentence embedding learning model called BRFP (Factorization Process with Bidirectional Restraints) that fuses syntactic information, uses matrix decomposition to learn syntactic information, and fuses and calculates with word vectors to obtain the embedded representation of sentences. In the experimental chapter, text similarity experiments are conducted to verify the rationality and effectiveness of the model and analyzed experimental results on Chinese and English texts with the current mainstream learning methods, and potential improvement directions are summarized. The experimental results on Chinese and English datasets, including STS, AFQMC, and LCQMC, show that the model proposed in this paper outperforms the CNN method in terms of accuracy and F1 value by 7.6% and 4.8. The comparison experiment with the word vector weighted model shows that when the sentence length is longer, or the corresponding syntactic structure is complex, the model's advantages in this paper are more prominent than TF-IDF and SIF methods. Compared with the TF-IDF method, the effect improved by 14.4%. Compared with the SIF method, it has a maximum advantage of 7.9%, and the overall improvement in each comparative experimental task is between 4 and 6 percentage points. In the neural network model comparison experiment, the model in this paper compared the CNN, RNN, LSTM, ST, QT, and InferSent models, and the effect significantly improved on the 14<sup>'</sup>OnWN, 14<sup>'</sup>Tweet-news, and 15<sup>'</sup>Ans.-forum datasets. For example, in the 14<sup>'</sup>OnWN dataset, the BRFP method has a 10.9% improvement over the ST method. The 14<sup>'</sup>Tweet-news dataset has a 22.9% advantage over the LSTM method, and the 15<sup>'</sup>Ans.-forum dataset has a 24.07% improvement over the RNN method. The article also demonstrates the generality of the model, proving that the model proposed in this paper is also a universal learning framework.

## 1. Introduction

In the Internet era, the amount of information shows a geometric progression, and the massive amount of text data hides inestimable social and economic value. Efficient and reasonable text information processing has become a hot and challenging point in current research [1]. The emergence of the word embedding method [2] provides an excellent idea, and researchers have focused their attention on the vector representation of natural language, usually based on

semantic similarity to model the words and sentences into continuous vectors. With the maturity and widespread application of word embedding methods, using vectors to represent [3–5] those higher language levels is also feasible.

Natural language is a product of the evolution and development of human society. As a carrier of information and a tool for communication and thinking, it has extensively promoted the development of human history. The difficulty of natural language processing lies in the flexibility of natural language semantic connotation and the ambiguity

of syntactic rules. Because of these characteristics, natural language modeling is complicated. In addition, the accuracy of the evaluation criteria for many downstream tasks in natural language processing also needs to be studied. Unlike the field of computer vision, for example, in the image classification task, even if the image labels are manually labeled, the labels' correctness can be significantly guaranteed due to the image's intuition. However, the high abstraction of natural language determines. Uncertainty in labeling training samples in supervised learning adds random and chance factors to the model's training process. In the final analysis, natural language, as a symbolic system, needs to be converted into a numerical model to realize computer processing and analysis.

The emergence of word embedding methods provides an excellent idea, and researchers have focused their attention on the vectorized representation of natural language, usually based on semantic similarity for modeling to map language into continuous vectors. In natural language, the words, sentences, paragraphs, and chapters constitute the four levels of division of natural language, among which words are the most basic language units. In addition, a sentence's general semantics must contain its constituent words' semantics. However, it is not a simple superposition of semantics but a fusion of lexical semantics under the constraints of specific syntactic rules.

The current main sentence embedding learning methods include bag-of-words model-based and neural network methods. Most of these methods start from the perspective of semantic similarity, directly regard the sentence as a whole without considering its internal composition, and learn the embedded representation of sentences by simply predicting various semantic connections between sentences. Therefore, these methods generally have the problem of a low degree of syntactic information fusion. The lack of syntactic information will lead to the overall semantic analysis and understanding deviation, ultimately affecting the semantic accuracy of sentence embedding.

This paper proposes a sentence embedding learning method based on the above background that fuses syntactic information. With the help of the syntactic analysis [6–8] method, the sentence is modeled as an attribute network containing node information [9], and matrix decomposition is used to learn the vector representation of syntactic information and fuse it with word vectors [10]. Embedding representation of sentences is dedicated to solving the deficiency of insufficient syntactic information fusion in the current research on sentence embedding [11]. The sentence embedding representation studied in this paper belongs to the fundamentally public nature problem in natural language processing. Sentence embedding can be understood as an extension of word embedding [12], which maps long text fragments into digital vectors, which points out potential fields for natural language processing research. At the same time, it is also the core link for solving many practical application problems such as machine translation, automatic question answering, sentiment analysis, and personalized recommendation. Therefore, this research topic has far-reaching theoretical significance and application value.

## 2. Related Research

Embedding representation of sentences is a fundamental theory in natural language processing and has become an indispensable part of natural language processing theory. In the early days, unsupervised representation learning has always occupied the mainstream position but supervised learning, and multitask learning is the current mainstream trends [13]. In addition, the emergence of pretrained language models has profoundly changed the landscape of natural language processing. So far, scholars from various countries have proposed many embedding representation learning methods. These methods mainly fall into the bag-of-words model-based and neural network methods.

*2.1. The Methods Based on Bag-of-Words Model.* The bag-of-words model was used initially in text classification tasks to represent documents as feature vectors by position index [14]. Methods based on the bag-of-words model usually ignore word order, grammar, and syntax in the text and only regard the text as a collection of words or a collection of word vectors.

Deerwester et al. believed that text generally contains several topics [15], and the similarity of text semantics can be approximated as the similarity of topics. A dimensionality-reduced latent semantic space was constructed and applied to text classification tasks by performing singular value decomposition on the vocabulary-text matrix. In 2017, Arora et al. proposed a smooth inverse frequency weighting model [16]. First, they presented the concept of public discourse vector, arguing that the generation of the corpus has obtained from a random walk of discourse vector, and introduced two smoothing factors to suppress those useless highs in the contribution of frequent words and ensure that information outside the context of words can also appear in sentences.

*2.2. The Methods Based on Neural Network Model.* Kiros et al. proposed the Skip-Thought model [17], which adopts the most commonly used encoder-decoder architecture in machine translation and uses the midsentence to predict the sentence of the context to learn the sentence. Experiments show that the sentence vectors generated by Skip-Thought are not all optimal but generally perform well in multiple tasks. This result proves that the model has good generality. However, Skip-Thought predicts the context without any syntactic, semantic, and contextual information after training is finished and discards the encoder, resulting in a lack of model training efficiency. In 2018, Logeswaran et al. proposed Quick-Thought [18], which improved the decoder structure of Skip-Thought and replaced the original prediction behavior with a classification task, which improved the efficiency of the sentence encoding process and downstream jobs.

Collobert and Weston proposed the C&W model [19] to design models and objective functions directly from the distributed hypothesis, with the ultimate goal of learning word vectors. Mikolov et al. proposed word2vec in 2013. The method includes the CBOW model and the Skip-gram model [20], which are calculated based on local sliding windows and entirely use local contextual features. Pennington

et al. proposed the Glove model [21], which fuses global statistical elements and local contextual features. In 2016, Facebook released the FastText model, similar to the CBOV model of word2vec, including the input, hidden, and output layers. FastText can be used for embedding learning in different languages and is an excellent general term embedding model. In 2018, Paul G. Allen published the ELMo [22] model. Before it appeared, almost all word embedding learning models could not solve the problem of polysemy because each word has only a unique vector. The ELMo model is no longer a direct correspondence between words and vectors but infers the word vector corresponding to each word according to an input sentence or paragraph. Even in the context of polysemy, the model can combine contexts for polysemy understanding.

**2.3. Syntax Analysis Method.** Different from artificial language, there are many ambiguities in natural language, such as part-of-speech ambiguity, structural ambiguity, and referential ambiguity, and the process of syntactic analysis can eliminate the ambiguity problem [23] existing in natural language processing, which is an important method to solve natural language understanding [24]. Syntactic analysis is a key challenge [25] in natural language processing. Its purpose is to analyze the grammatical functions of words in sentences. The syntactic analysis includes syntactic structure analysis and dependency analysis.

Syntactic structure analysis [25] is aimed at obtaining the syntactic structure or temporary sentence structure of the whole sentence and organizing the structural relationship into a tree structure, called the syntactic structure tree, in which each node is composed of part-of-speech tags, phrase tags, or clause structure tags to refer to syntactic elements at different levels.

**2.4. Graph Embedding Representation Algorithm.** The core idea of graph embedding [26] is to find a mapping function to convert each node or the entire graph in the graph structure into a low-dimensional vector representation. This concept was originally an extension of the word embedding technology in natural language processing [27]. In the research field of graph embedding, many excellent algorithms, such as DeepWalk [28], node2vec [29], LINE [30], and GarphGAN [31], have been proposed, but the most influential is DeepWalk, proposed by Perozzi et al. [28] in 2014. The main idea is to perform random walks on the graph structure to generate node sequences, treat the node sequences as text sequences in natural language, input them into the word2vec model, and train them to obtain the vector representation of graph nodes. In 2015, Yang et al. proved that the DeepWalk algorithm based on the random walk, which in Figure 1 shows its algorithm matrix, is equivalent to the matrix decomposition of matrix  $M$  [32], in the form of  $M = W^T H$ .

### 3. Methodology

This article will innovate and propose new algorithm models based on syntax analysis methods [33] and graph embed-

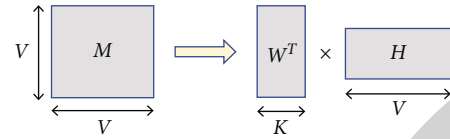


FIGURE 1: DeepWalk algorithm equivalent matrix.

ding algorithms [34]. We model syntactic and semantic information as attribute networks containing vertex information and apply algorithmic ideas in the field of graph embedding [35] to sentence embedding [26] representation learning.

**3.1. The Principle of the BRFP Model.** In the study of sentence embedding, syntactic information is an important aspect [36] because, from a grammatical point of view, the overall semantics of a sentence not only depends on the independent semantics of each word in the sentence [37] but also generates new semantics, even opposite semantics under the action of syntax.

In addition, the high abstraction of natural language determines the abstraction of syntactic structure, so a quantifiable modeling method is needed to describe syntactic information.

In this paper, the BRFP model introduced the Stanford parser [38] to parse the sentence to generate a syntactic structure tree. Because the tree can be regarded as a graph without rings in the data structure, BRFP constructs an undirected graph structure containing node information, shown in Figure 2, based on the syntactic structure tree and defines each component label in the tree as a graph vertex.

In the graph, triple represents the node information of the vertex and the semantic vector. There are two types of vertices in the graph. The first type is terminal vertex, such as vertex 4, 5, 8, 10, 12, 13, and 14. Each vertex corresponds to an entity word in the sentence, and the semantic vector of such vertex is the trained word vector. The rest of the vertices are nonterminal vertex and only correspond to syntactic components rather than an entity word. The BRFP model uses the mean of the semantic vectors of all subvertex vertices as the semantic vector of the vertices. The calculation rule is as Equation (1), which represents the syntax vector, which is also the part that requires model training. Furthermore, it represents various syntactic labels in the syntactic parse tree:

$$V_i = \begin{cases} \frac{1}{|C|} \sum_{v_j \in C} \text{vec}_j, & \text{otherwise,} \\ \text{vec}_i, & v_i \text{ is terminal,} \end{cases} \quad C \text{ is child set of } v_i. \quad (1)$$

The abovementioned undirected graph model contains the semantic information represented by word vectors and the syntactic information represented by the graph topology and vertex syntax labels. The embedding representation of the sentence is transformed into the embedding





FIGURE 2: Undirected graph with node information.

representation of the learning graph, which is precisely the embedding of the attribute network containing the vertex information.

BRFP decomposes a matrix  $M$  into the product form of four matrices  $E$ ,  $W$ ,  $H$ , and  $R$ , where  $W$  and  $H$  are parameter matrices.  $R$  represents the semantic matrix, which is formed by splicing the semantic vectors of each vertex in the graph structure horizontally.  $E$  represents the syntactic mutual information matrix, and the element  $E_{ij}$  is defined as follows:

$$E_{ij}^n = \frac{|\Omega_i^n \cap \Omega_j^n|}{|\Omega_i^n| + |\Omega_j^n|}. \quad (2)$$

$\Omega_i^n$  is a multilabel set of order  $n$  corresponding to a vertex, representing the set of syntactic labels of all neighbor vertex within  $n$  steps away from a vertex. In the following example, the parameter  $n$  is set to 2, and  $E_{56}$  is calculated: the vertex within 2 steps from vertex 5 is 1, 3, and 6, and the vertex within 2 steps from vertex 6 is 1, 3, 5, 7, 8, and 9; the specific calculation process is as follows:

$$\left. \begin{aligned} \Omega_5^k &= \{S, VP, S\} \\ \Omega_6^k &= \{S, VP, VBD, VP, TO, VP\} \end{aligned} \right\} \Rightarrow E_{56} = \frac{|\{S, VP\}|}{|\Omega_5^k| + |\Omega_6^k|} = \frac{2}{3+6} = \frac{2}{9}. \quad (3)$$

The matrix  $E$  reflects the syntactic similarity between vertices. The BRFP model uses matrix  $E$  and matrix  $R$  to add syntactic and semantic information in the decomposition process.  $E$  and  $R$  being fixed can also be regarded as two constraints on the matrix decomposition process, corresponding to the bidirectional restraints in the BRFP model. For matrix decomposition, this paper adopts the following loss function:

$$\min_{W,H} \|M - E^T W^T H R\|_F^2 + \frac{\lambda}{2} (\|W\|_F^2 + \|H\|_F^2). \quad (4)$$

Further, the second term is the regularization part, and  $\lambda$  is the introduced harmonic factor. During the optimization process,  $E$  and  $R$  are fixed, and it is only necessary to alternately fix and update  $W$  and  $H$  until the model converges:

$$\begin{aligned} \frac{\partial}{\partial W} \left( \|M - S^T W^T H T\|_F^2 + \frac{\lambda}{2} (\|W\|_F^2 + \|H\|_F^2) \right) \\ = \lambda W - 2S_\alpha (M^T - S_\alpha^T W E) E^T, S_\alpha = H R, \end{aligned} \quad (5)$$

$$\begin{aligned} \frac{\partial}{\partial H} \left( \|M - S^T W^T H T\|_F^2 + \frac{\lambda}{2} (\|W\|_F^2 + \|H\|_F^2) \right) \\ = \lambda H - 2S_\beta (M - S_\beta^T H R) R^T, S_\beta = W E. \end{aligned} \quad (6)$$

Finally, by splicing the matrix  $E * W$  with the matrix  $H * R$ , the  $k$ -dimensional network representation vector with attribute information can be obtained. It is assigned to the syntactic vectors  $E_i$  of  $V$  vertices in turn. These representation vectors not only contain the topological structure information of the syntactic parse tree but also contain the syntactic label information of the nodes and the semantic information of the word vector.

Many natural language processing models process sentences as time series, such as recurrent neural networks [29, 39]. From the perspective of syntactic parsing, sentences are organized into a tree-like structure with hierarchical relationships. The structure is expanded spatially, which is more in line with the process in which words combine through syntactic rules to form complex structures in grammar.

For example, the computational linguist Schubert believes that a phrase is a language unit that has an aggregate relationship with other words and phrases. A syntactic relationship exists between the words within a phrase, forming a language combination [30, 40]. Therefore, according to the parse tree's recursive structure and the subsequent tree traversal, the BRFP model fuses the semantic vectors of each node in the parse tree. The basis for the fusion is the syntactic vector trained by the model, and finally, the embedded representation of the entire sentence is obtained from the root node. The node recursive fusion process can also be

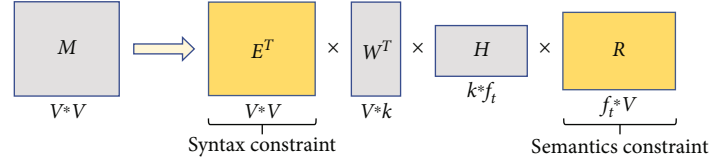
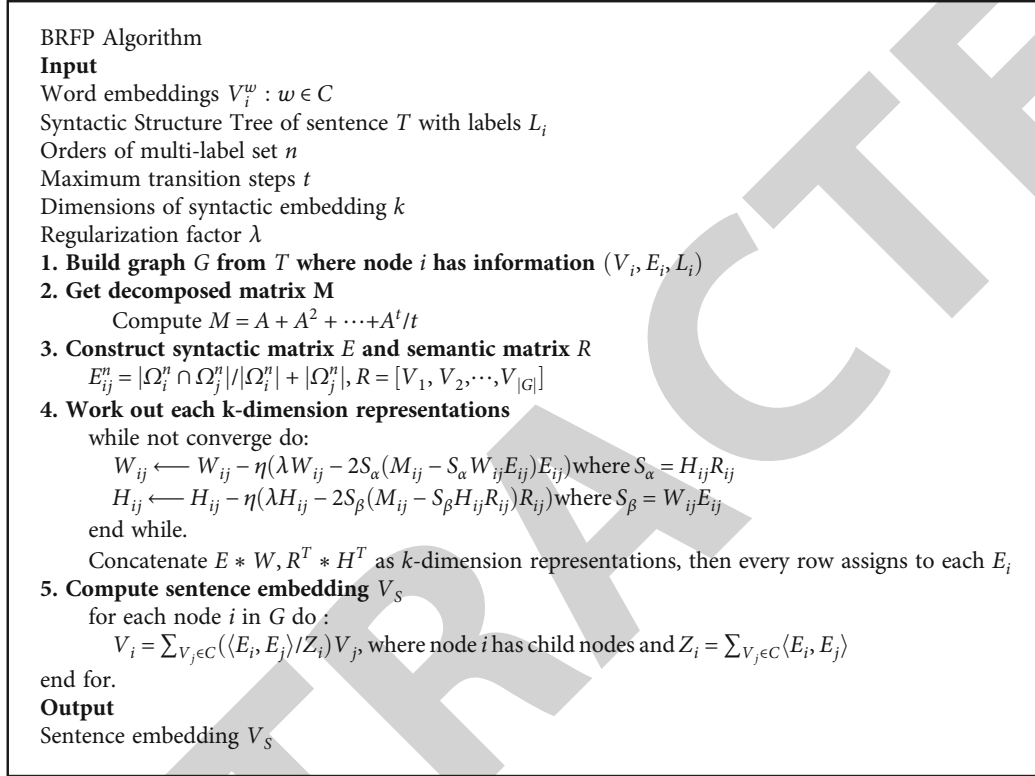


FIGURE 3: BRFP model decomposition process.



ALGORITHM 1: The model calculation steps.

regarded as a semantic fusion process based on syntactic information. The fusion calculation rules refer to

$$V_i = \sum_{V_j \in C} \frac{\langle E_i, E_j \rangle}{Z_i} V_j, \quad \text{where } Z_i = \sum_j \langle E_i, E_j \rangle, \quad (7)$$

where  $C$  is the set of child nodes of node  $V_i$ ,  $Z_i$  represents the normalization factor. The larger the inner product value of  $E_j$  and  $E_i$  is, the closer the syntactic component represented by node  $V_j$  is to the syntactic component represented by node  $V_i$ , which means that the semantic component corresponding to  $V_j$  occupies a greater proportion in the upper-level syntactic structure.

**3.2. The Calculation Process.** The author improved the DeepWalk method proposed by Perozzi et al. [28] and proposed a BRFP model, which constructs an undirected graph network [35] with node information based on the syntactic structure tree and word vectors, uses matrix decomposition to learn the syntactic information in the graph structure, and fuses

the word vectors to generate the embedded representation of the sentence.

Figure 3 shows the decomposition process of the model, and the calculation steps are shown in Algorithm 1.

## 4. Classification Experiments and Results

**4.1. Experiment Setup.** In this experiment, the Stanford parser [38] is used to analyze the Chinese and English corpus syntactically. The diversified language model provided by Stanford can support the syntactic analysis of English, Chinese, Arabic, and other languages. The English model of the Stanford analyzer has a total of 67 component labels, including 36 part-of-speech labels, 22 phrase labels, and nine clause structure labels. The standard component labels in the Chinese model of the Stanford analyzer mainly include 33 part-of-speech labels and 17 phrase labels.

In the English data experiment in this paper, due to the Pearson correlation coefficient being used to respect the final evaluation index in the datasets, in the calculation step, firstly, the similarity value of all sentence pairs is calculated

TABLE 1: The STS datasets used in the experiment.

STS-12	STS-13	STS-14	STS-15	STS-16
MSRpar (1500)	HDL (750)	HDL (750)	HDL (750)	HDL (249)
MSRvid (1500)	FNWN (189)	OnWN (750)	Images (750)	Plagiarism (230)
OnWN (750)	OnWN (561)	Deft-forum (450)	Ans.-student (750)	Post-editing (244)
SMTnews (750)	SMT (750)	Deft-news (300)	Ans.-forum (375)	Ans.-Ans. (254)
SMTeuroparl (750)		Images (750)	Belief (375)	Quest.-Quest. (209)
		Tweet-news (750)		

TABLE 2: English text experimental results.

Model	STS - 12	STS - 13	STS - 14	STS - 15	STS - 16
TF - IDF	55.07	52.14	57.39	60.85	62.39
SIF	57.52	61.83	68.50	<b>73.73</b>	71.50
CNN	56.10	54.22	59.55	52.75	60.35
RNN	48.21	59.36	58.60	57.29	60.31
LSTM	53.56	58.49	62.56	66.53	67.30
ST	50.84	42.87	61.49	62.03	55.73
QT	58.57	57.70	66.52	68.45	69.05
InferSent	<b>59.28</b>	60.67	67.73	72.50	<b>72.41</b>
BRFP	59.08	<b>65.32</b>	<b>71.80</b>	73.43	72.35

by cosine similarity. Then, the Pearson correlation coefficient [41] is calculated with the standard similarity score in the datasets as the final evaluation index.

The Pearson correlation coefficient is a statistical concept used to measure the linear correlation between two random variables with a value between -1 and 1. Here,  $X_i$  is the text similarity prediction value,  $Y_i$  is the given standard similarity value, and the calculation equation is as follows:

$$\text{Pearson} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}. \quad (8)$$

The Chinese data experiments are typical binary classification problems [42], so this paper uses the accuracy and F1 value as the evaluation indicators. First, the cosine similarity is used to calculate the similarity value of all sentence pairs, and then, an optimal threshold is determined by the model. In addition, this experiment uses the Chinese and English word vectors trained by word2vec, and the default length is 100 dimensions.

*4.2. Analysis of Experimental Results in Chinese and English.* The English corpus uses the text semantic similarity task of the 2012-2016 International Workshop on Semantic Evaluation [43]. The competition publishes several datasets each year. The corpus is collected from the news, videos, forums, picture descriptions, Twitter, and many other fields, and the number of datasets and corpus is adjusted annually. For example, STS-12 provides five datasets, including MSRpar, MSRvid, OnWN, SMTnews, and SMTeuroparl, and STS-13

includes four datasets of HDL, FNWN, OnWN, and SMT, replacing some datasets based on STS-12. The number of samples also fluctuates to a certain extent. The STS datasets used in the experiment are shown in Table 1. The data in brackets is the number of samples in the dataset. The STS task is designed to measure the degree of semantic similarity. Each data sample consists of a sentence pair and a standard similarity score. The similarity score ranges from 0 to 5. The higher the score, the closer the semantics of the two sentences.

Due to the differences in the sample sources of the datasets included in the STS tasks in different years, the average Pearson coefficient of each dataset is used here as the final presentation index. As a widely used text semantic similarity dataset worldwide, the STS task can thoroughly verify the rationality and effectiveness of the BRFP model. For intuitive expression, all Pearson correlation coefficients in the experimental data in this paper are multiplied by 100 by default.

In Table 2, STS-12 to STS-16 represent the datasets of different years from 2012 to 2016 in the International Workshop on Semantic Evaluation.

The bold numbers in Table 2 represent the best performance on the specific dataset. The experimental results show that InferSent [44] achieves the best performance on STS-12 and STS-16 datasets and has good performance on other datasets, proving that InferSent is an excellent baseline model. Our method achieves the highest scores on the STS-13 and STS-14 tasks, which are 3.5 and 2.3 higher than the other state-of-the-art performances, respectively, even though our method fails to achieve the best results on all datasets. Nevertheless, they are also close compared to the

TABLE 3: The Chinese text experimental results on the AFQMC dataset.

Model	Accuracy (%)	F1 (%)
TF-IDF	63.5	62.4
CNN	69.3	70.7
LSTM	75.7	75.2
QT	68.9	67.4
BRFP	76.8	75.5

TABLE 4: The Chinese text experimental results on the LCQMC dataset.

Model	Accuracy (%)	F1 (%)
SIF	77.4	78.2
RNN	76.7	75.7
ST	68.3	68.9
InferSent	79.4	78.5
BRFP	79.0	78.4

best methods, with differences of 0.2, 0.3, and 0.06, respectively. It is difficult for any single model to apply to language representations in different fields, considering the complexity of natural language and the diversity of application scenarios, but the BRFP model has shown relatively good performance.

The Chinese dataset uses the Ant Financial Question Matching Corpus dataset (AFQMC) [45] and a Large-scale Chinese Question Matching Corpus dataset (LCQMC) [46], both of which are binary classification tasks to determine whether two sentences are semantically similar. For the AFQMC dataset, 100,000 pairs of labeled data are used in this experiment. For the LCQMC dataset, 238,766 training samples, 8,802 validation samples, and 12,500 test samples are used in the experiment.

According to the above experimental results in Tables 3 and 4 and Figure 4, the all-around performance of BRFP and LSTM methods [47] on the AFQMC dataset is significantly better than the other three models. Compared with LSTM, BRFP exceeds 1.1 and 0.3 in accuracy and F1 value, respectively, and the performance improvement is relatively apparent. Although BRFP failed to achieve the highest accuracy rate on the LCQMC dataset, it significantly surpassed neural network models such as RNN [48] and Quick-Thought (QT) [18], only 0.4 lower than the best-performing InferSent and the same as InferSent in F1 value.

The AFQMC and LCQMC datasets have a wide range of sample sources. The AFQMC dataset is derived from the interactive Q&A between customer service and users in the production environment and involves some proper nouns in finance and e-commerce, which can thoroughly test the model's ability to deal with unknown and low-frequency words.

In addition, the LCQMC dataset contains colloquial description text, which involves many different fields, which is a massive challenge to the model's generalization ability.

Therefore, the stable performance of different Chinese data further verifies the generality of the BRFP model.

*4.3. Compared with the Word Vector Weighted Model.* This subsection compares the BRFP model with the weighted model based on word vectors, including TF-IDF and SIF methods. This subsection uses the text semantic similarity of the 2012 and 2013 International Semantic Evaluation Competition, in which STS-12 includes five datasets and STS-13 includes four datasets.

Since different years may contain datasets with the same name, the following experiments are distinguished by the year plus the dataset name. For example, the OnWN datasets of the STS-12 task and STS-13 task are denoted as 12'OnWN and 13'OnWN. It is not used as the experimental dataset in this section, considering that the total number of samples in the 13'FNW dataset is small. After excluding 13'FNW, the experiment in this section actually contains 8 datasets, namely, 12'MSRpar, 12'MSRvid, 12'OnWN, 12'SMTnews, 12'SMTeuoparl, 13'HDL, 13'OnWN, and 13'SMT.

The experimental results in Table 5 show that the BRFP model outperforms the comparative baselines in five of the eight selected datasets and achieves the best performance, which is reasonably competitive. In addition, the BRFP model has achieved significant performance improvements on some datasets such as 12'MSRpar, 12'SMTeuoparl, 13'OnWN, and 13'SMT, and the overall improvement is between 4 and 6 percentage points.

Both TF-IDF and SIF measure the contribution of each word in sentence semantics based on word frequency but do not reflect the impact of the interaction between syntactic components on the overall semantics of the sentence. As the experimental result shows, the corresponding syntactic structure is more complex, and the effect of the BRFP model is improved more obviously.

*4.4. Compared with Neural Network Model.* This section compares the BRFP model with some neural network models and baseline models including CNN [49], RNN [50], LSTM [51], Skip-Thought (ST) [17], Quick-Thought QT [18], and InferSent [44]. Given the small number of samples in each dataset of the STS-16 task, this section uses 11 datasets provided by STS-14 and STS-15. The content of these datasets involves news, forums, images, and many other fields. The experiment uses different experiments on datasets in the field to fully verify the model's generality.

From the data in Table 6, it can be seen that the BRFP model's performance exceeds the baseline model's performance on more than half of the datasets, and the effect is significantly improved on the 14'OnWN, 14'Tweet-news, and 15'Ans.-forum datasets, which are definite improvements of 2.87, 3.52, and 2.4. However, A particular gap has been found in the BRFP model compared to the best comparison method in the two datasets of 14'Images and 15'Images. The possible reason is that these two datasets contain a large number of unique characters, such as “%” and “3/4.” The existence of special characters affects the accuracy of



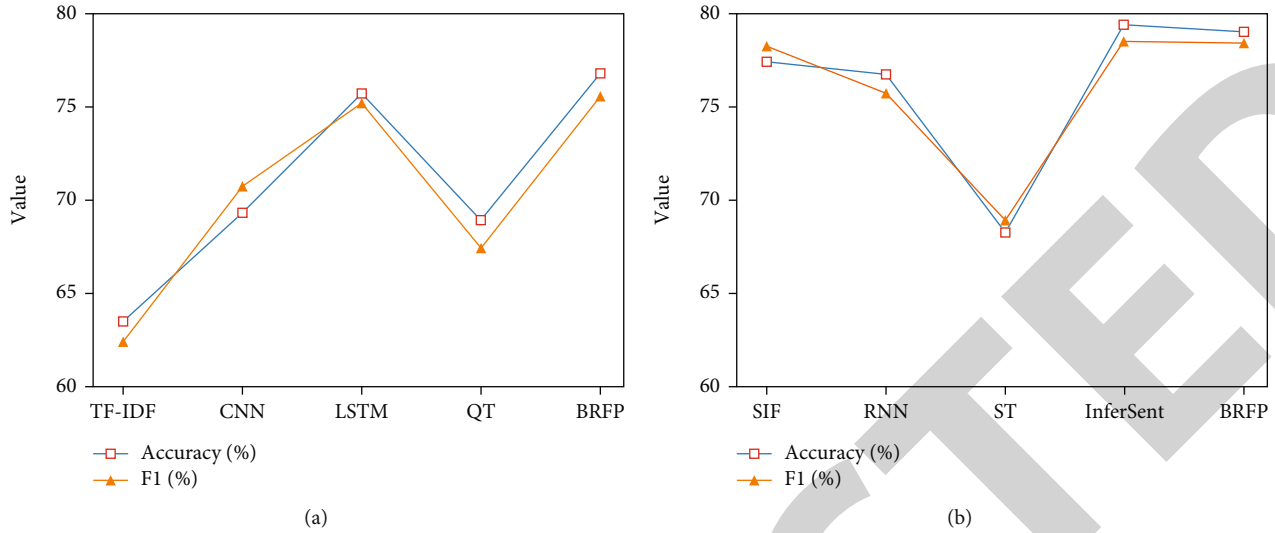


FIGURE 4: (a) Description of the result on AFQMC dataset; (b) description of the result on LCQMC dataset.

TABLE 5: The result of comparison with the word vector weighted model.

Dataset	TF-IDF	SIF	BRFP
12'MSRpar	43.53	41.87	47.50
12'MSRvid	69.34	79.07	78.93
12'OnWN	63.58	66.15	67.09
12'SMTnews	68.05	58.63	66.58
12'SMTeuroparl	43.07	53.25	57.47
13'HDL	47.56	51.34	50.09
13'OnWN	50.09	57.80	63.94
13'SMT	51.77	49.55	56.30

syntactic analysis and reduces the accuracy of syntactic information learned by the model.

According to statistics, the number of samples containing special characters in 14'Images and 15'Images accounts for 9% and 11% of the total samples, respectively. The existence of a large number of special characters weakens the ability of the BRFP model to capture effective syntax.

**4.5. Validity Experiment of Syntactic Information.** This subsection will illustrate the effectiveness of syntactic information in the model from another perspective to further illustrate competitiveness and interpretability. The BRFP model transforms sentence embedding representation learning into graph embedding representation learning with vertex information and adopts a triple to represent vertex information. Whether the syntactic vector learned by the model can reflect syntactic information is also an aspect worthy of further discussion.

In this paper, 3000 sentences are randomly selected from the STS task historical dataset and input to the BRFP model, and 3000 trained undirected graphs with complete vertex information can be obtained. Then, for all undirected graphs, the average similarity between the syntactic vectors corresponding to the vertices of different component labels

is calculated. Since the English syntactic analysis of the Stanford analyzer has a total of 67 component labels, there are many combinations between labels, and only the similarity relationship data of some component labels are displayed in Table 7.

According to the data in Table 7, it can be observed the similarity distribution between the syntax vectors.

For example, an adjective phrase's most common syntactic composition (ADVP) is "adjective+noun." From the data in Table 7, it can be seen that ADVP is related to the adjective (JJ), adjective comparative (JJR), adjective superlative (JJS), and noun singular (NN). The average similarity of the syntactic vectors corresponding to the component labels such as noun plural (NNS) is significantly higher than other labels. Verb phrases (VP) and noun labels (NN, NNS) and verb labels (VB, VBD, VBN, VBG, VBP, and VBZ) are much more similar than the rest of the component labels.

In addition, the syntactic similarity between prepositional phrases (PP), position conjunctions (IN), and proper nouns (NNP) is high, which also fits the syntactic structure of "preposition+place noun."

It can be seen that the syntactic vectors learned by BRFP reflect the regularity of collocation between different syntactic components, and the process of using syntactic vectors to weight and fuse word vectors is also a process of semantic fusion based on syntactic information.

**4.6. Hyperparameter Analysis.** The BRFP model has four essential parameters  $k$ ,  $t$ ,  $n$ , and  $\lambda$ , of which the maximum number of transition steps  $t$  and the multilabel set order  $n$  both reflect the utilization range of the information around the vertex, so in the practical application of the model,  $t$  and  $n$  usually take the same value to simplify model parameters.

Therefore, this subsection will focus on the impact of changes in parameters  $k$ ,  $n$ , and  $\lambda$  on model performance. First, the STS-15 and STS-16 datasets were selected to study the influence of the parameter  $k$ , and five groups of different

TABLE 6: The result of comparison with neural network model.

Dataset	CNN	RNN	LSTM	ST	QT	InferSent	BRFP
14 HDL	60.18	55.40	62.66	61.05	<b>70.46</b>	67.51	69.53
14 OnWN	63.50	61.35	58.36	55.47	61.25	59.35	<b>66.37</b>
14 Deft – forum	58.37	59.67	65.74	58.36	65.58	70.97	<b>71.02</b>
14 Deft – news	51.03	42.16	67.05	65.00	<b>67.60</b>	66.32	66.47
14 Images	67.25	67.39	61.50	61.92	71.09	<b>77.60</b>	74.69
14 Tweet – news	71.93	70.05	56.04	67.50	74.35	75.42	<b>78.94</b>
15 HDL	73.54	65.37	66.50	68.57	66.83	71.60	<b>73.95</b>
15 Images	64.30	53.02	60.37	58.93	61.07	<b>67.52</b>	64.18
15 Ans.–student	59.75	51.60	71.53	53.66	70.43	74.38	<b>75.67</b>
15 Ans.–forum	62.95	60.50	70.48	61.10	72.45	70.80	<b>74.85</b>
15 Belief	65.83	61.72	67.25	70.35	68.50	<b>72.95</b>	72.50

TABLE 7: The English ingredient label similarity results.

	ADJP	NP	PP	QP	VP
CD	0.17	0.73	0.65	0.92	0.20
DT	0.40	0.85	0.67	0.87	0.37
IN	0.39	0.14	0.98	0.13	0.51
JJ	0.96	0.80	0.48	0.17	0.46
JJR	0.74	0.69	0.30	0.14	0.35
JJS	0.83	0.73	0.35	0.07	0.38
NN	0.91	0.89	0.77	0.81	0.73
NNS	0.87	0.83	0.71	0.85	0.82
NNP	0.65	0.71	0.87	0.62	0.69
VB	0.62	0.34	0.45	0.34	0.93
VBD	0.57	0.38	0.33	0.23	0.84
VBN	0.51	0.27	0.20	0.30	0.73
VBG	0.57	0.10	0.16	0.17	0.90
VBP	0.44	0.14	0.24	0.36	0.81
VBZ	0.40	0.09	0.32	0.29	0.85

$n$  and  $\lambda$  values were randomly selected for each dataset. The experimental results are shown in Figure 5.

Under different combinations of  $n$  and  $\lambda$  values, most of the curves show the peak value of the Pearson coefficient in the interval of 80-100, and the model effect reaches the optimal state. Considering the overall data, 70-100 is still the most ideal value range for the syntactic vector dimension  $k$ , although a small number of curves show that  $k$  reaches a peak when 60 or 100 is taken, followed by a slight downward trend. In order to control the variables, the default  $k$  is 90 in the subsequent experiments in this subsection.

It can be seen from Figure 6(a) that the increase of  $n$  significantly improves the effect of the experiment. When  $n$  exceeds 8, most of the curves begin to show a downward trend; especially when  $n$  is greater than 10, the Pearson coefficient generally drops sharply.

Further, it can be inferred that the ideal value of  $n$  is between 4 and 7. The main reason is that the principle of multiple label sets is to use the component labels of the neighboring vertices of the vertices to approximate the syntactic similarity between vertices, which is similar to the  $K$  nearest neighbor algorithm.

When  $n$  is relatively small, the increase of  $n$  means that the multilabel set can cover more label information of neighbor vertices, and it will be more accurate in the judgment of syntactic similarity between two vertices.

However, when the value of  $n$  is large enough, because the syntactic structure tree of short sentences has relatively few nodes, high-order multilabel sets will produce convergence. In extreme cases, if  $n$  exceeds the total number of nodes in the syntactic structure tree, the multiple label sets corresponding to any two nodes are approximately the same, which will weaken the constraining effect of syntactic information in the subsequent matrix decomposition process.

In addition, as can be seen from Figure 6(b), after the harmonic factor  $\lambda$  reaches 0.6, the performance is basically in a relatively ideal state. Although the subsequent curve trend fluctuates, the overall stability remains stable.

## 5. Model Feature of Efficient and Universal

The model proposed in this paper is an efficient and universal sentence embedding learning framework that can flexibly integrate different word embedding schemes and syntactic analysis techniques, considering word embedding and syntactic analysis as the framework's building blocks.

For the word embedding method, word2vec is a typical method, and C&W, Glove, FastText, and ELMo are also popular word embedding methods. In order to verify the usability of the model combination, this paper integrates five word embedding methods, word2vec, C&W, Glove, FastText, and ELMo, into the BRFP model and conducts experiments on the classification effect of the model in the STS2012-2015 dataset.

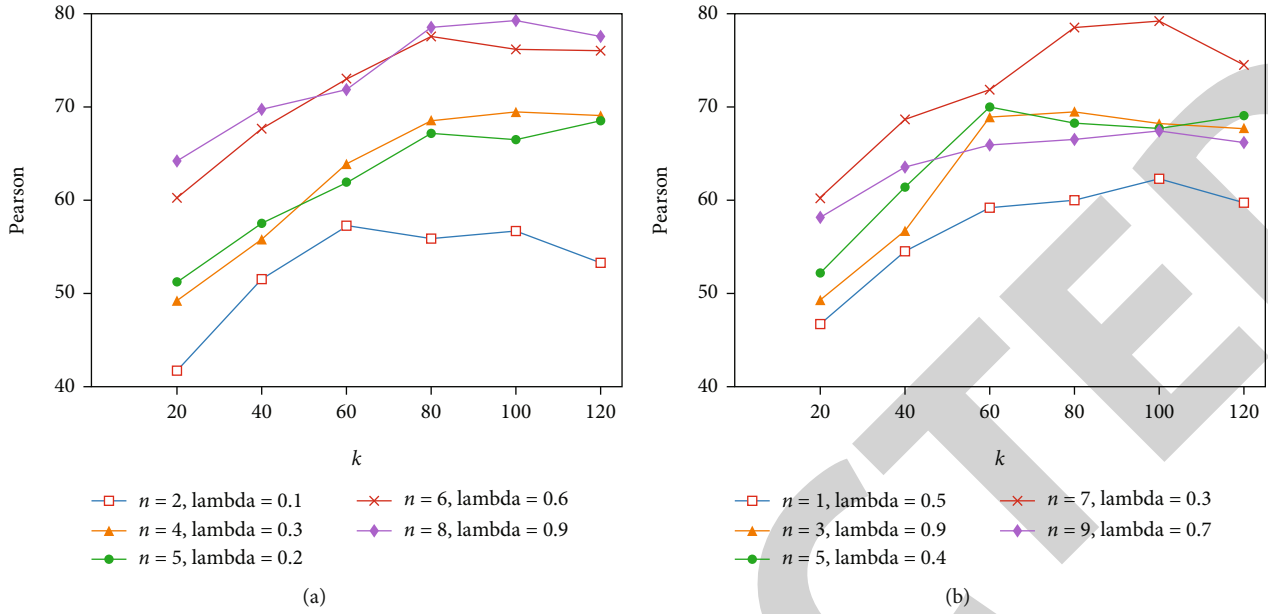


FIGURE 5: (a) Description of the result on STS-15; (b) description of the result on STS-16.

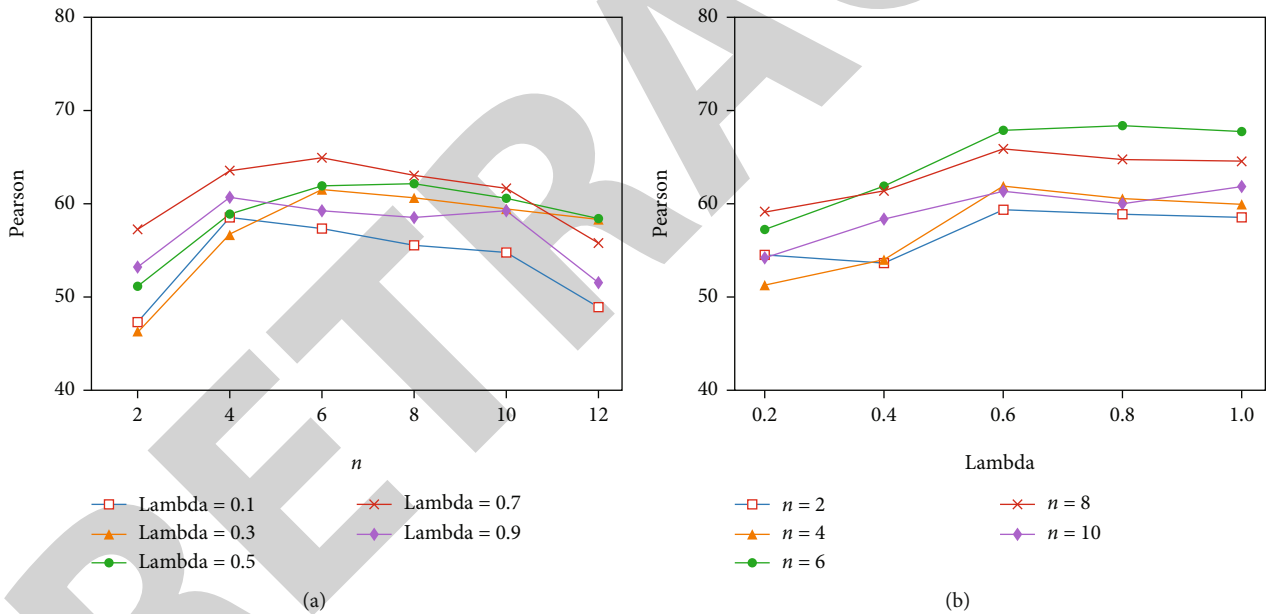


FIGURE 6: (a) Description of the result on parameter  $n$ ; (b) description of the result on parameter  $\lambda$ .

For syntactic analysis, this paper constructs BRFP model variants based on two different syntax tree structures, syntax structure tree (SST) and Dependency Syntactic Tree (DST), and conducts comparative experiments.

The model proposed in this paper has adopted the syntax tree and achieved good performance. Studying whether the dependency syntax tree can achieve good results as a model building block is also precious. In addition to the Stanford analyzer, the Language Technology Platform (LTP) [52] developed by the Harbin Institute of Technology can provide complete technical support for Chinese natural language processing.

As a complete set of Chinese natural language processing systems, including a series of language processing modules such as lexical, syntactic, and semantic, LTP has become one of the most influential Chinese language processing platforms at home and abroad.

As shown in Figure 7, this paper will use the syntactic structure analysis and syntactic dependency analysis of the Stanford analyzer and LTP to construct four variant methods, namely, Stanford+SST, Stanford+DST, LTP+SST, and LTP+DST, where SST and DST are the abbreviations for syntax structure tree and Dependency Syntax Tree.

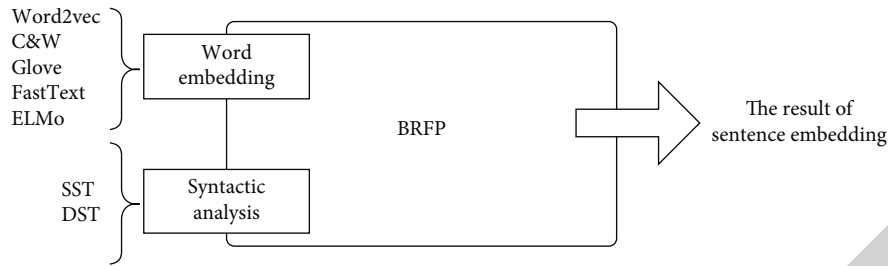


FIGURE 7: Schematic diagram of model combination.

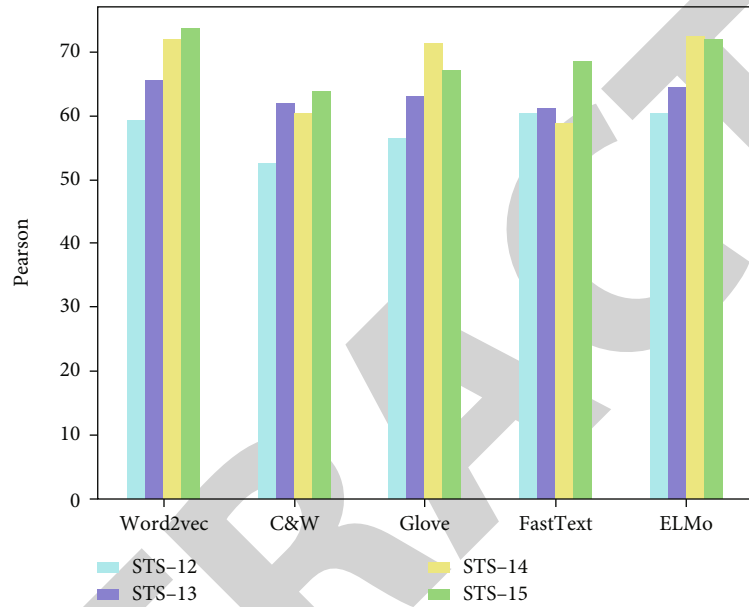


FIGURE 8: Different word embedding method fusion experimental result.

5.1. *The Experimental Results of BRFP and Word Embedding Fusion.* The dataset used in this experiment is the text semantic similarity task of the 2012-2015 International Workshop on Semantic Evaluation. In order to ensure the same experimental conditions, the experiments in this section only change the category of word vectors. All word vectors use the official pretraining model, and other variables such as syntactic analysis method, word vector dimension, and hyperparameter values are the same.

As shown in Figure 8, the five word embedding methods, word2vec, C&W, Glove, FastText, and ELMo, can all achieve good classification results after they are integrated into BRFP. This experiment proves efficient and universal to the BRFP framework. Different word embedding methods can be selected according to application scenarios to achieve the best results. It is also possible to use more powerful word embedding methods to improve the BRFP model’s performance in the future.

5.2. *The Experimental Results of BRFP and Syntactic Analysis Fusion.* In this experiment, the AFQMC and the LCQMC datasets are selected, both of which are classification tasks. The F1 value is used as the evaluation index to judge whether the two sentences’ semantics are similar. In order

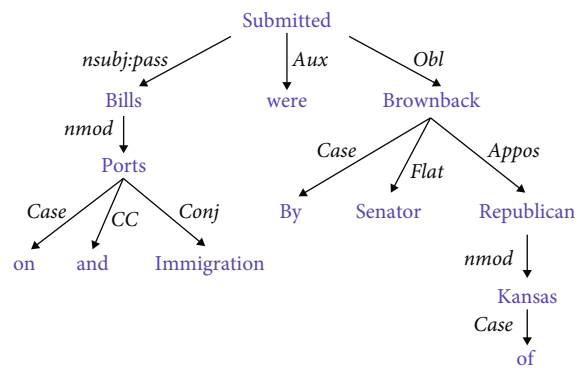


FIGURE 9: Dependency syntax tree.

to ensure the fairness of the comparison environment of variant methods, this section only adjusts the syntactic analysis technology, and other conditions remain the same.

Syntactic analysis is a critical method in the field of NLP, and in addition to structural analysis, syntactic analysis methods also include dependency analysis. Dependency analysis considers that the syntactic functions of words to other words are described by dependencies, emphasizing



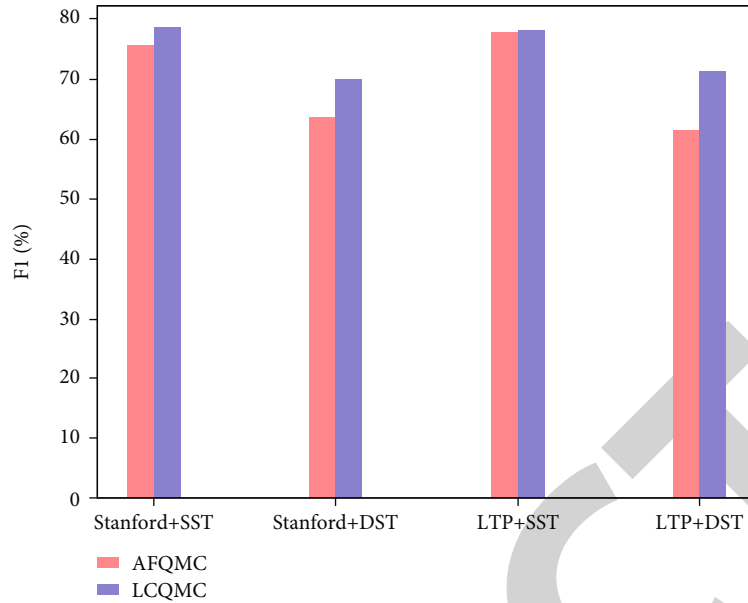


FIGURE 10: Different syntactic analysis method fusion experimental result.

the dependencies between local words and constraining them into a tree structure.

The arrow direction in Figure 9 points from the dominant word to the subordinate word, and the label on the edge is the relation type. Representing the dependencies of all words in a sentence in the form of directed edges results in a tree called a dependency syntax tree.

In modern dependency grammars, linguist Robinson [53] proposes four binding axioms for dependency syntax trees:

- (1) There is only one word or virtual root node that does not depend on other words
- (2) All words except the virtual root node must depend on other words
- (3) Each word cannot depend on multiple words
- (4) If A depends on B, then A and B can only depend on words between A and B

The above four axioms constrain the uniqueness, connectivity, acyclicity, and projectivity of the root node of the dependency syntax tree, respectively. Therefore, the dependency syntax tree and the syntax structure tree are structurally homogenous and can also be used as the input of the BRFP model.

As shown in Figure 10, the syntactic information expressed by the generated tree structure is different due to the different focuses of the two analyses on the two syntactic analysis platforms of Stanford analyzer and LTP, with the practical effect of using the syntactic structure tree significantly better than the Dependency Syntax Tree. Since the syntactic structure analysis is to decompose the sentence structure layer by layer starting from the global structure, the proposition of concern is the sentence generation pro-

cess. On the other hand, dependency analysis emphasizes the grammatical connection between local words, believes there is a master-slave relationship between words, and constrains this master-slave relationship into a tree structure. The experimental results also show that the syntactic structure tree is more suitable for the text-similarity task.

By comparing the Stanford+SST and LTP+SST methods horizontally, it is found that the F1 values of the two methods on the LCQMC dataset are the same, but the LTP performs better on the AFQMC dataset, and the F1 value increases by about 1.5 percentage points. The difference is the granularity of word segmentation, which is more evident in Chinese texts. Word segmentation is usually not required in English text preprocessing, or spaces are directly split. The Chinese word segmentation granularity of LTP is smaller than that of the Stanford analyzer. For example, LTP will treat 北京大学 (Peking University) as a whole noun and will not split it, while the word segmentation result of the Stanford analyzer is 北京(Peking) and 大学 (University). The original node in the tree will be split into two, thus changing the entire tree structure, which will split the original semantics of some words.

## 6. Discussion

Text semantic understanding has always been a critical problem in natural language processing. The current experimental results show that the method in this paper can surpass many baseline models on the text semantic similarity task, but there are still some areas for improvement.

First, text semantic similarity is a fundamental problem in natural language processing. Currently, the English STS corpus is rich in training data, but the Chinese STS corpus is relatively scarce, and most belong to classification tasks, which have no accurate similarity score for two sentences like the English STS corpus. In future work, the authors will

try to collect more Chinese datasets similar to English STS for experimental comparison.

Secondly, the existence of special characters in the dataset will reduce the accuracy of syntactic analysis and affect the model effect. The method of dealing with nodes containing special characters effectively and reasonably in the process of matrix decomposition without destroying the original syntactic structure is also worthy of further study.

## 7. Conclusions

This paper proposes a new sentence embedding learning model that integrates syntactic information called BRFP, aimed at the low degree of syntactic information fusion in the current research on sentence embedding representation, which model uses matrix decomposition to learn syntactic information and fuses it with word vectors to obtain the embedded representation of sentences.

The experiments in this paper prove that the sentence embedding representation learned by the BRFP model surpasses most of the baseline models in the semantic similarity task of Chinese and English texts, and the accuracy of each experiment has achieved significant advantages.

## Data Availability

Publicly available datasets were analyzed in this study. This data can be found here: (1) STS (2012-2016) datasets: <https://paperswithcode.com/dataset/semantic-textual-similarity-2012-2016>, (2) AFQMC dataset: <https://tianchi.aliyun.com/dataset/dataDetail?dataId=106411>, and (3) LCQMC dataset: <http://icrc.hitsz.edu.cn/Article/show/171.html>.

## Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Acknowledgments

This research was funded by the Philosophy and Social Science Fund of Guangdong Province (GD15XJY01) and a major platform and project of Department of Education of Guangdong Province (Project Grant No: 2017KTSCX013).

## References

- [1] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, Mendu, Barnes, and Brown, "Text classification algorithms: a survey," *Information*, vol. 10, no. 4, p. 150, 2019.
- [2] B. Yoshua, D. Rejean, V. Pascal, and J. Christian, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, no. 6, pp. 1137–1155, 2003.
- [3] Y. Tay, L. A. Tuan, and S. C. Hui, "Compare, compress and propagate: enhancing neural architectures with alignment factorization for natural language inference," 2017, <https://arxiv.org/abs/1801.00102>.
- [4] K. Shuang, Y. Tan, Z. Cai, and Y. Sun, "Natural language modeling with syntactic structure dependency," *Information Sciences*, vol. 523, pp. 220–233, 2020.
- [5] B. Wang, A. Wang, F. Chen, Y. Wang, and C.-C. J. Kuo, "Evaluating word embedding models: methods and experimental results," *APSIPA transactions on signal and information processing*, vol. 8, no. 1, 2019.
- [6] S. Kurita, D. Kawahara, and S. Kurohashi, "Neural joint model for transition-based Chinese syntactic analysis," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, 2017.
- [7] O. G. Iroju and J. O. Olaleke, "A systematic review of natural language processing in healthcare," *International Journal of Information Technology and Computer Science*, vol. 8, pp. 44–50, 2015.
- [8] H. K. Janda, A. Pawar, S. du, and V. Mago, "Syntactic, semantic and sentiment analysis: the joint effect on automated essay evaluation," *Access*, vol. 7, pp. 108486–108503, 2019.
- [9] P. Chunaev, "Community detection in node-attributed social networks: a survey," *Computer Science Review*, vol. 37, article 100286, 2020.
- [10] S. Henry, C. Cuffy, and B. T. McInnes, "Vector representations of multi-word terms for semantic relatedness," *Journal of Biomedical Informatics*, vol. 77, pp. 111–119, 2018.
- [11] K. Kann, A. Warstadt, A. Williams, and S. R. Bowman, "Verb argument structure alternations in word and sentence embeddings," vol. 16, no. 7, pp. 106–120, 2018, <https://arxiv.org/abs/1811.10773>.
- [12] Z. Lin, M. Feng, C. N. Santos et al., "A structured self-attentive sentence embedding," 2017, <https://arxiv.org/abs/1703.03130>.
- [13] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Transactions on Knowledge and Data Engineering*, 2021.
- [14] J. Wang, P. Liu, M. F. She, S. Nahavandi, and A. Kouzani, "Bag-of-words representation for biomedical time series classification," *Biomedical Signal Processing & Control*, vol. 8, no. 6, pp. 634–644, 2013.
- [15] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.
- [16] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," in *International conference on learning representations*, Toulon, France, 2017.
- [17] R. Kiros, Y. Zhu, R. R. Salakhutdinov et al., "Skip-thought vectors," *Advances in neural information processing systems*, vol. 28, 2015.
- [18] L. Logeswaran and H. Lee, "An efficient framework for learning sentence representations," 2018, <https://arxiv.org/abs/1803.02893>.
- [19] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *Machine Learning, Proceedings of the Twenty-Fifth International Conference*, Helsinki, Finland, 2008.
- [20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, <https://arxiv.org/abs/1301.3781>.
- [21] J. Pennington, R. Socher, and M. C. Glove, "Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1535–1541, Association for Computational Linguistics, Doha, 2014.
- [22] E. Peters Matthew, N. Mark, I. Mohit, G. Matt, C. Christopher, and L. Kenton, "Deep contextualized word representations," 2018.

- [23] U. S. Shah and D. C. Jinwala, "Resolving ambiguities in natural language software requirements," *ACM SIGSOFT Software Engineering Notes*, vol. 40, no. 5, pp. 1–7, 2015.
- [24] B. Galitsky, "Machine learning of syntactic parse trees for search and classification of text," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 3, pp. 1072–1091, 2013.
- [25] A. Torfi, R. A. Shirvani, Y. Keneshloo, N. Tavaf, and E. A. Fox, "Natural language processing advancements by deep learning: a survey," 2020, <https://arxiv.org/abs/2003.01200>.
- [26] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: problems, techniques, and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1616–1637, 2018.
- [27] S. Yan, D. Xu, B. Zhang, and H. J. Zhang, "Graph embedding: a general framework for dimensionality reduction," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2005.
- [28] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, New York, USA, 2014.
- [29] A. Grover and J. Leskovec, "node2vec: scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, San Francisco, California, USA, 2016.
- [30] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: large-scale information network embedding," *Microsoft Research Asia*, vol. 17, no. 4, pp. 267–283, 2015.
- [31] H. Wang, J. Wang, J. Wang et al., "GraphGAN: graph representation learning with generative adversarial nets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 41, no. 5, pp. 239–249, 2017.
- [32] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Chang, "Network representation learning with rich text information," *Twenty-fourth international joint conference on artificial intelligence*, vol. 39, no. 3, pp. 52–61, 2015.
- [33] R. Köhler, *Quantitative Syntax Analysis*, De Gruyter Mouton, 2012.
- [34] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: a survey," *Knowledge-Based Systems*, vol. 151, pp. 78–94, 2018.
- [35] R. Mihalcea and D. Radev, *Graph-Based Natural Language Processing and Information Retrieval*, Cambridge university press, 2011.
- [36] A.-D. Vo, Q.-P. Nguyen, and C.-Y. Ock, "Semantic and syntactic analysis in learning representation based on a sentiment analysis model," *Applied Intelligence*, vol. 50, no. 3, pp. 663–680, 2020.
- [37] A. D. Friederici and J. Weissenborn, "Mapping sentence form onto meaning: the syntax-semantic interface," *Brain Research*, vol. 1146, pp. 50–58, 2007.
- [38] M.-C. De Marneffe and C. D. Manning, "The Stanford typed dependencies representation," in *Coling 2008: proceedings of the workshop on cross-framework and cross-domain parser evaluation*, Manchester, 2008.
- [39] M. Morchid, "Parsimonious memory unit for recurrent neural networks with application to natural language processing," *Neurocomputing*, vol. 314, pp. 48–64, 2018.
- [40] X. Jing, L. Jun-Hui, Z. Qiao-Ming, and L. Pei-Feng, "Chinese semantic role labeling based on phrase and dependency syntactic structure," *Computer Engineering*, vol. 37, no. 24, pp. 169–172, 2011.
- [41] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise Reduction in Speech Processing*, pp. 1–4, Springer, Berlin, Heidelberg, 2009.
- [42] L. Xu, H. Hu, X. Zhang et al., "CLUE: a Chinese language understanding evaluation benchmark," 2020, <https://arxiv.org/abs/2004.05986>.
- [43] "International Workshop on Semantic Evaluation," Available online: <https://semeval.github.io>.
- [44] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," 2017, <https://arxiv.org/abs/1705.02364>.
- [45] B. An, "Chinese paraphrase dataset and detection," in *2021 International Conference on Asian Language Processing (IALP)*, Singapore, Singapore, 2021.
- [46] X. Liu, Q. Chen, C. Deng et al., "Lcqm: a large-scale chinese question matching corpus," in *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA, 2018.
- [47] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: a search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2017.
- [48] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, article 132306, 2020.
- [49] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2489–2515, 2011.
- [50] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu, "Towards universal paraphrastic sentence embeddings," in *Proc of 4th International Conference on Learning Representations*, San Juan, 2016.
- [51] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [52] W. Che, Z. Li, and T. Liu, "LTP: a Chinese language technology platform," in *The 23rd International Conference on Computational Linguistics*, Beijing, 2010.
- [53] S. Kübler, R. McDonald, and J. Nivre, "Dependency parsing," *Synthesis Lectures on Human Language Technologies*, vol. 2, no. 1, pp. 1–127, 2009.