Expert Systems With Applications 200 (2022) 117084

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A joint FrameNet and element focusing Sentence-BERT method of sentence similarity computation

Tiexin Wang [a,b,c,*], Hui Shi [a], Wenjing Liu [a], Xinhua Yan [a,c]

[a] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, Jiangsu, China
[b] Key Laboratory of Safety-Critical Software, Nanjing University of Aeronautics and Astronautics, Ministry of Industry and Information Technology, Nanjing 210016, Jiangsu, China
[c] Nanjing DENET System Technology Co.LTD, Nanjing 211100, Jiangsu, China

ARTICLE INFO

Keywords:
Sentence similarity computation
FrameNet
Frame semantics
BERT
Deep learning

ABSTRACT

As one of the fundamental research areas of natural language processing, sentence similarity computation attracts researchers' attention. Considering two single independent sentences, it is difficult to measure the similarity between them without sufficient context information. To solve this issue, we propose a joint FrameNet and element focusing Sentence-BERT method of sentence similarity computation (FEFS3C). Considering the actual meaning of sentences, we adopt the frame semantics theory and adapt FrameNet in FEFS3C. Moreover, focusing on critical information conveyed in sentences, FEFS3C takes the superiority of deep learning technologies and proposes a new sentence representation model element focusing Sentence-BERT (EF-SBERT) which improves traditional sentence representations. Two primary considerations of sentences in FEFS3C "sentence meaning" and "critical sentence information" aim to better utilize the influence of sentences context. To evaluate the performance of FEFS3C, we carried out experiments on the standard test set "STS-B". Results show that FEFS3C has obtained better Spearman correlation compared with traditional methods.

1. Introduction

As a fundamental task in natural language processing (NLP), sentence similarity computation has been used in many language processing applications such as information retrieval (Huang & Hu, 2009), question answering (Yih, Chang, Meek, & Pastusiak, 2013), paragraph identification (Wan, Dras, Dale, & Paris, 2006) and text mining (Atkinson-Abutridy, Mellish, & Aitken, 2004), etc. For instance, sentence similarity computation is used as a text mining technique to discover potential patterns in textual databases (Atkinson-Abutridy et al., 2004). Automatic and efficient sentence similarity computation methods can effectively replace onerous and error-prone manual judgment processes.

Current mature sentence similarity computation methods can be classified into three kinds (Chandrasekaran & Mago, 2021), namely knowledge-based methods, corpus-based methods and deep neural network-based methods. However, all three kinds of methods above have their own limitations. Knowledge-based methods take the actual meaning of sentences into consideration. However, these methods only focus on keywords of sentences without considering whole sentences; meanwhile, they show poor performance when dealing with sentences taken from different domains or written with diverse languages. On the contrary, Corpus-based methods are suitable for more situations including across languages. However, these methods do not take the actual meaning of sentences into consideration. Focusing on computation results, deep neural network-based methods show obvious advantages over the former two kinds of methods. However, similar to other AI techniques-based methods, they require high computational resources (Xue, Wang, Liang, & Slowik, 2021; O'Neill, Xue, & Zhang, 2021) and lack of interpretability. Therefore, concerning limitations of the above methods, three main challenges can be summarized as follows.

- Concerning sentences context, how to consider one sentence as a whole while taking the actual meaning of sentences into consideration?
- How to make sentence similarity computation results interpretable?
- How to exploit advantages of each method and build a hybrid method to provide better results of sentence similarity computation?

To handle these challenges, we propose a joint FrameNet and

* Corresponding author at: College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, Jiangsu, China.
E-mail addresses: tiexin.wang@nuaa.edu.cn (T. Wang), yolosh2020@nuaa.edu.cn (H. Shi), liuwenjing@nuaa.edu.cn (W. Liu).

https://doi.org/10.1016/j.eswa.2022.117084
Received 5 November 2021; Received in revised form 8 February 2022; Accepted 28 March 2022
Available online 4 April 2022

element focusing Sentence-BERT (EF-SBERT) method of sentence similarity computation (FEFS3C). On one hand, we use FrameNet (Baker, Fillmore, & Lowe, 1998), a large lexical database of English that comes with sentences annotated with semantic frames, to mark and analyze sentences. On the other hand, we exploit the EF-SBERT model to improve traditional sentence representations, which uses advanced deep learning technologies. In short, FEFS3C exploits both the versatility of corpus-based methods and the superiority of deep neural network-based methods.

Three contributions of this paper are as follows.

- Considering contextual issues, FEFS3C is based on frame semantics and considers one sentence as a whole while taking the actual meaning of sentences into consideration.
- Focusing on information of sentences, we define a new sentence representation model that is built on advanced deep learning technologies. This model has better interpretability than other deep learning-based methods.
- We propose an integrated sentence similarity computation method, which combines the frame semantics theory and the sentence representation model idea.

The paper is structured as follows. Section 2 details preliminaries of FEFS3C. Section 3 discusses FEFS3C in detail. In section 4, we present experiments and analyze testing results. Section 5 gives the related work, and we conclude the paper in Section 6.

2. Preliminaries

2.1. FrameNet and frame semantics extension

FrameNet (Baker et al., 1998), a semantic database of English based on Frame Semantics theory (Ruppenhofer, Ellsworth, Schwarzer-Petruck, Johnson, & Scheffczyk, 2006), contains nearly 202,230 manually annotated sentences linked to more than 1,220 semantic frames. FrameNet defines three major components: frames, frame elements (FEs) and lexical units (LUs) (Zhang, Sun, & Wang, 2018) to restrict specific meaning of a word. Furthermore, frame elements are divided into two kinds core and non-core. Core frame elements uniquely define a frame (Petruck & Ellsworth, 2018). In contrast, non-core frame elements are relevant to common semantic components such as time and place.

Besides, FrameNet has linked semantic frames together by a system of frame relations, which is an asymmetric directional relation system. Eight kinds of frame relations are defined, which are inheritance (Inherits_from & Is_Inherited_by), perspective_on (Perspective_on & Is_Perspective_in), using (uses & Is Used by), subframe (Subframe_of & Has_Subframe_s), precedes (Precedes & Is_Proceded_by), inchoative_of, causative_of and see_also. These frame relations indicate how frames relate to each other in its hierarchy of frames. In our previous work (Liu, Wang, Yang, & Cao, 2020), to better use FrameNet for sentence similarity computation, we developed a visual tool "**Graphical Interpretation for FrameNet: GIFN**". GIFN considers all frames, frame elements and lexical units as nodes and regards frame relations as edges. Fig. 1 illustrates frame relations between frames in GIFN.

In Fig. 1, nodes (circles) represent frames and edges with arrows represent frame relations between two associated frames. The minimum number of connected edges between two associated frames in GIFN is considered as the shortest path between them.

In order to quantitatively analyze frame relations, we have assigned values to different frame relations and set a judgment threshold to "0.505" through a pilot study in our previous work (Wang, Liu, Yang, & Cao, 2021). The specific assignment values of frame relations and examples are shown in Table 1. These assignment values to frame relations are regarded as path weights. We use shortest paths and path weights to mark and analyze sentences. We detail the computation process in section 3.2.

The idea of assigning values to frame relations is inspired by the literature (Wang, Truptil, & Benaben, 2017). A larger corresponding value means a closer semantic relation between two frames. The process of assigning values to frame relations and setting the judgment threshold "0.505" is as follows. First, we invited five related scholars to

Table 1
Assignment values (Wang et al., 2021) to semantic relations in FrameNet and Examples.

frame relations	Initial assigned values	Final assigned values	Examples
Inherits from	0.6	0.55	Absorb_heat → Becoming
Is Inherited by	0.6	0.55	Becoming → Absorb_heat
Perspective on	0.5	0.45	Growing_food → Agriculture
Is Perspectivized in	0.5	0.45	Agriculture → Growing_food
Uses	0.3	0.3	Locale → Locative_relation
Is Used by	0.3	0.3	Locative_relation → Locale
Subframe of	0.4	0.35	Crime_process → Crime_scenario
Has Subframe(s)	0.4	0.35	Crime_scenario → Crime_process
Precedes	0.2	0.2	Activity_ongoing → Activity_stop
Is Preceded by	0.2	0.2	Activity_stop → Activity_ongoing
Is Inchoative of	0.3	0.3	Come_together → Aggregate
Is Causative of	0.3	0.3	Apply_heat → Absorb_heat
See also	0.4	0.4	Choosing → Deciding

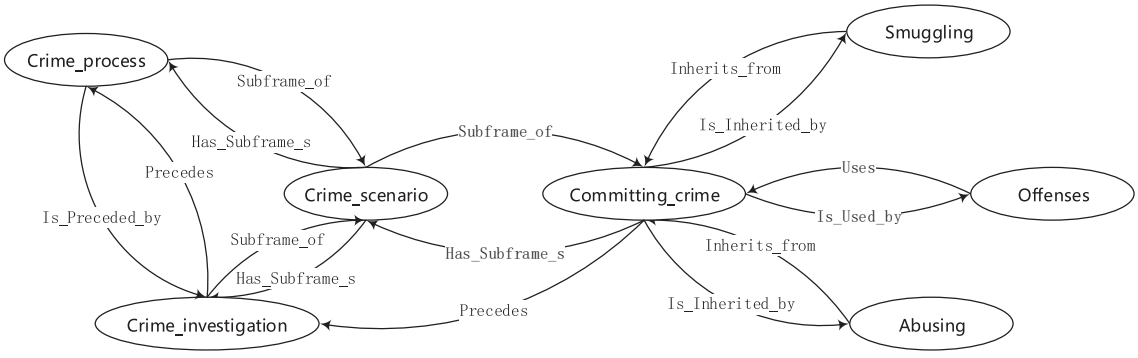

Fig. 1. An illustration of frame relations between frames in GIFN.

independently assign values to frame relations. Then, through a round-table discussion, all scholars jointly drafted a set of assignments, as shown in the second column of Table 1. Next, we randomly selected 60 pairs of sentences (of which 45 pairs are similar, the other 15 pairs are not similar) from "Microsoft Research Paraphrase Corpus: MSRP" as a validation set to verify the reliability of the initial assignments. We further adjusted frame relations assignments and the judgment threshold based on verification results. The final assigned values of frame relations are shown in the third column of Table 1.

To achieve frame semantic parsing, an open-source semantic parser called SEMAFOR (Kshirsagar, Thomson, Schneider, Carbonell, Smith, & Dyer, 2015) is employed. SEMAFOR works as follows. We employ SEMAFOR to address the English sentence "Kate drove home in a stupor.", and Table 2 lists its semantic parsing results. The first row lists all words in the sentence, the second row presents all frames that the words evoke and other rows present FEs of the frames. This sentence contains two frames, *Bringing* and *Foreign_or_domestic_country*. Take frame *Bringing* as an example, it is evoked by LU *drove* and contains two FEs, i.e., *Agent* and *Theme*.

SEMAFOR can automatically process English sentences according to the form of semantic analysis in FrameNet. However, we find that SEMAFOR cannot cover all keywords in sentences, which is because of both the incomplete coverage of FrameNet and the unsatisfying performance of SEMAFOR. To overcome this shortcoming, we use "the named entity recognizer: NER" to expand the coverage of FrameNet and SEMAFOR. In our previous work (Wang et al., 2021), we proposed a method referred to as "Extend Frame Semantics based Sentences Similarity Computing (EFS3C)" by using SEMAFOR to achieve frame semantic parsing and NER to achieve frame semantics extension. As an extension to that work, we improve our computation method in this paper.

2.2. Sentence-BERT model

In the area of deep learning, natural language representations are commonly employed as features for machine learning tasks or pre-training. Devlin, Chang, Lee, and Toutanova (2018) proposed a new pre-trained language representation model BERT, which has set an excellent performance on various sentence-pair similarity comparison tasks. BERT consists of several transformer encoder layers that enable models using it to extract deep language features on both token-level and sentence-level (Huang, Lee, Ma, Chen, Yu, & Chen, 2019). Despite performing well on many NLP tasks, BERT-induced sentence embeddings perform badly when retrieving semantically comparable sentences. This makes it difficult to apply BERT sentence embeddings to real-world contexts.

In order to improve the performance of the BERT model in computing sentence similarity, a coupling architecture based on two artificial neural networks, called the siamese network architecture, is used in BERT (Reimers & Gurevych, 2019). Fig. 2 is a simplification diagram of the siamese network architecture (Chopra, Hadsell, & LeCun, 2005).

The siamese network in Fig. 2 is made up of two sub neural networks. Each sub neural network takes in data, maps it to a high-dimensional feature space, and then outputs the resulting representation. By computing the distance between the two representations, such as cosine distance, researchers can compare the similarity of two input sentences. Based on the siamese network architecture, Sentence-BERT (SBERT) is developed.

SBERT consists of two BERT networks and each BERT receives a sentence as input. Moreover, SBERT performs a pooling operation on the output of BERT to produce a fixed-sized sentence embedding. The similarity score of two input sentences is computed by computing the cosine distance between two sentence embeddings "u" and "v". Fig. 3 is the SBERT architecture for computing similarity scores. We further improve inputs and outputs of BERT in SBERT. The detailed computation process, using the adapted version of SBERT architecture, is given in section 3.3.

3. Methodology

This section contains three subsections. First, we introduce the workflow of FEFS3C. Then, the second subsection details the FrameNet-based sentence similarity computation method. Finally, an adapted version of Sentence-BERT is given.

3.1. FEFS3C architecture overview

Ensemble methods have exhibited apparent advantages in many applications. They combine multiple models into one and usually obtain better predictive performance than the best of its components. Creating an ensemble method entails two steps: (1) creating a variety of models and (2) integrating their results (Seni & Elder, 2010). Zhang et al. (2018) used an ensemble method that combines a traditional model and a neural network model in their work, and achieved better results than the independent model in the Duplicate Question Identification (DQI) task. Inspired by this work, we apply the ensemble model to sentence similarity computation tasks.

We take a straightforward but effective ensemble method that combines the FrameNet-based method and the EF-SBERT by a hyper-parameter α (Fig. 4). The Equation is as follows.

$$P = \alpha \bullet P^F + (1-\alpha) \bullet P^{EF} \tag{1}$$

where $\alpha \in [0,1]$ is a hyper-parameter, P is the combined result, P^F is the score calculated by the FrameNet-based method, and P^{EF} is the score calculated by EF-SBERT.

In order to set the value of α to achieve better performance of FEFS3C, we carried out experiments on the standard set "STS-B" using Equation (1) with adjusting the value of α between 0 and 1.0. Fig. 5 shows the change of testing results with the alteration of α.

As shown in Fig. 5, when α is close to 0.2, FEFS3C performs the best, and FEFS3C gets a negative feedback as α becomes larger or smaller. It also indicates that EF-SBERT contributes more to FEFS3C. Hence, the hyper-parameter α is finally set to 0.2.

3.2. FrameNet-based sentence similarity computation

The architecture of FrameNet-based sentence similarity computation is shown in Fig. 6. Depending on computation elements, the architecture can be divided into three parts.

Part 1. Counting same frames between two sentences.

We extract frames that are evoked by words in sentences by employing open-source semantic parser SEMAFOR. Meanwhile, for the sake of expanding the coverage of FrameNet and SEMAFOR, we make use of NER to get supplement of frames that are missed by SEMAFOR.

Some definitions involved are described as follows.

Table 2
Frame Semantic Parsing of a sentence.

words in the sentence	Kate	drove	home	in	a	stupor	.
frame		**Bringing**	**Foreign_or_domestic_country**				
frame elements (FEs)		Agent	Current_country				
		Theme					

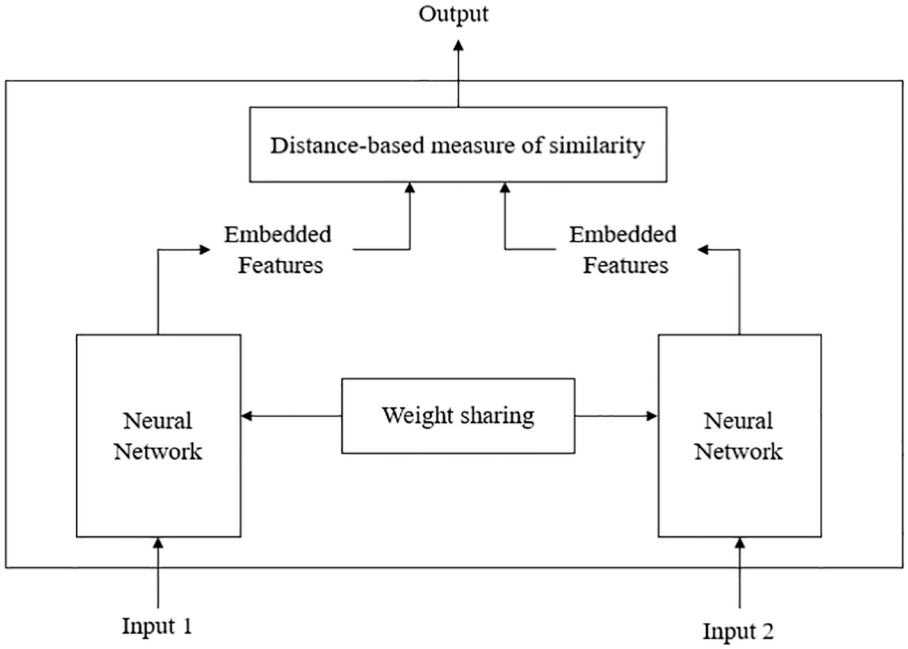

Fig. 2. The siamese network architecture (Chopra et al., 2005).

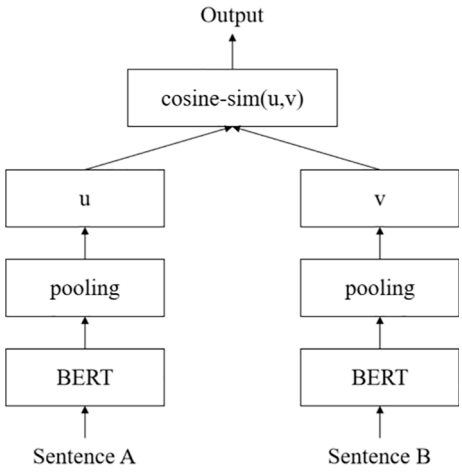

Fig. 3. SBERT architecture for computing similarity scores (Reimers et al., 2019).

Definition 1. *frameOne* is a set of frames evoked by Sentence One.

Definition 2. *frameTwo* is a set of frames evoked by Sentence Two.

Definition 3. *frameSame* = *frameOne* ∩ *frameTwo*={$frame_k$ | $frame_k$ ∈ *frameOne and* $frame_k$ ∈ *frameTwo*}, *i.e. frameSame* is the set of same frames between Sentence One and Two.

Definition 4. *frameOneR* = *frameOne* - *frameSame*, representing remaining frames in *frameOne*, where "-" represents the set difference.

Definition 5. *frameTwoR* = *frameTwo* - *frameSame*, representing remaining frames in *frameTwo*, where "-" represents the set difference.

Part 2. Computing shortest path scores.

Through the computation in Part 1, we get same frames between Sentence One and Two. Meanwhile we get the rest of frames after removing same frames from *frameOne* and *frameTwo*, i.e. *frameOneR* and *frameTwoR*. The main objective of this part is to calculate scores between the remaining frames in two sentences. The scores are determined by shortest paths that exist between *frameOneR* and *frameTwoR* as well as path weights.

We define two equations to compute the shortest path scores between frames of two sentences.

One definition involved in Equation (2) is described as follows.

Definition 6. *frameRel*={<$frame_i$, $frame_j$> | $frame_i$ ∈ *frameOneR*, $frame_j$ ∈ *frameTwoR* and there exists a shortest path between $frame_i$ and $frame_j$}, *i.e. frameRel* is the set of frame pairs that have shortest paths in GIFN between *frameOneR and frameTwoR*. |*frameRel*| is the number of frame pairs in frameRel.

Two Equations of computing shortest path scores between two sentences are as follows.

$$PathValue = \frac{\sum_{i=1}^{ShortestPath} pathWeight_i}{ShortestPath} \tag{2}$$

PathValue is the ratio of the cumulative sum of path weights of two frames in GIFN to *ShortestPath*. *pathWeight* corresponds to the assigned values of semantic relations in FrameNet. *ShortestPath* in Equation (2) is the minimum number of connected edges between two frames in GIFN.

$$PathScore = \sum_{i=1}^{|frameRel|} PathValue_i \tag{3}$$

PathScore is the cumulative sum of *PathValue*, i.e. the shortest path score between frames of two sentences.

Part 3. Computing sentence similarity of two sentences.

This part is the main extension of our previous work. We aim to take the importance of frames into consideration when mark and analyze sentences. From the perspective of fine-grained frame elements, we measure the importance of frames. The frame importance depends on the number of core frame elements covered by the frame. If the number of core frame elements covered by two frames is the same, the importance of two frames is considered to be the same. Table 3 shows the frame importance heat map of sentences from two different data sets.

As shown in Table 3, frame *Delivery*, frame *Assessing* and frame *Leadership*, which cover more core frame elements in S1 and convey more semantic information, are more important than frame *Calendric_unit*, frame *Subordinates_and_superiors*, frame *Origin* and frame *Military*. Since different frames have different importance, we define two equations to compute the importance of frames.

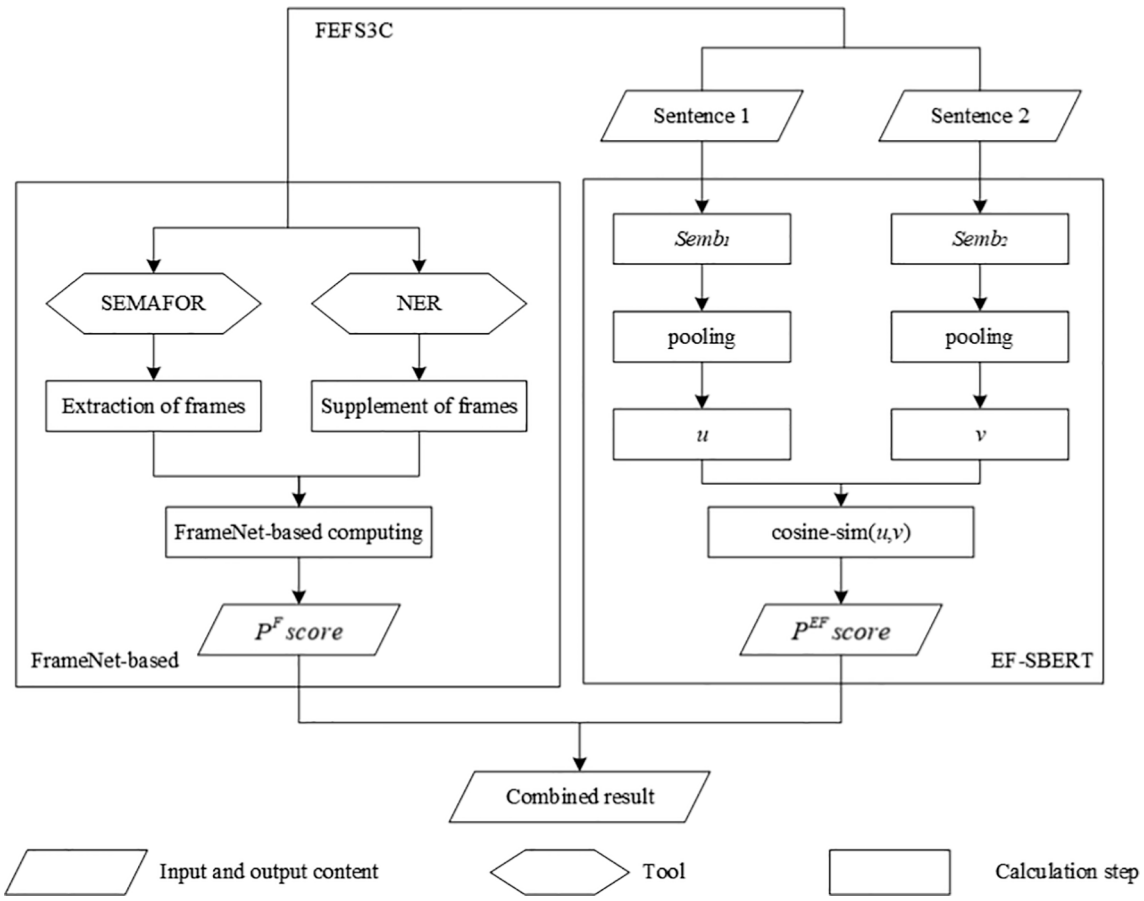

Fig. 4. The integrated working flow of FEFS3C.

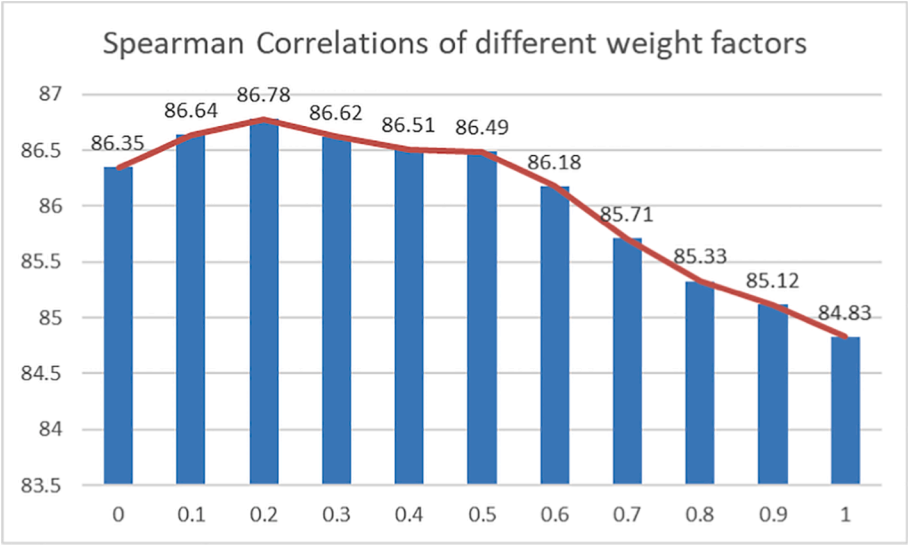

Fig. 5. The influence of weight in FEFS3C.

$$\beta = \frac{coreFEsCount}{\sum_{i=1}^{|frame|} FEsCount_i} \tag{4}$$

β is a probability that is determined by *coreFEsCount* and the number of frame elements contained in one entire sentence S. Within one frame, *coreFEsCount* indicates the number of core frame elements and *FEsCount* indicates the number of all frame elements. $|frame|$ is the number of frames in the entire sentence S.

$$frameWeight = \frac{e^{\beta}}{\sum_{i=1}^{|frame|} e^{\beta_i}} \tag{5}$$

frameWeight is the importance of a frame. We calculate the importance of a frame by normalization.

Based on Equation (2)-(5), Equation (6) is defined to compute the similarity score of two comparing sentences.

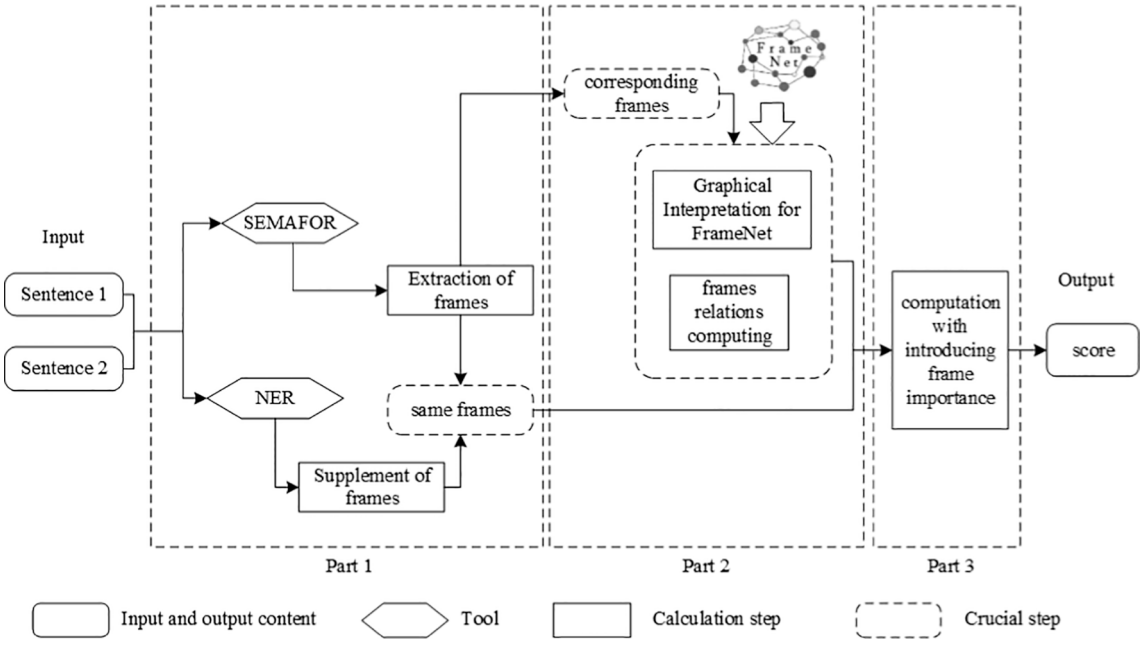

Fig. 6. The architecture of sentence similarity computation based on FrameNet.

Table 3
Frame importance heat map.

Sentence	Frame importance heat map
S1	Moseley and a senior aide delivered their summary assessments Subordinates_and_superiors Delivery Assessing to about 300 American and allied military officers on Thursday. Origin Millitary Leadership Calendric_unit
S2	GeNERal Moseley and a senior aide presented their summary Subordinates_and_superiors Presence assessments at an internal briefing for American and allied military Assessing Origin Millitary officers at Nellis Air Force Base in Nevada on Thursday. Leadership LOCATION STATE_OR_PROVINCE Calendric_unit

$$Score = \sum_{i=1}^{|frameSame|} minFrameWeight_i + \frac{PathScore}{Maximun(|frameOne|, |frameTwo|)} \quad (6)$$

Equation (6) includes two parts. One part is the cumulative sum of the frame importance of same frames in two sentences. $minFrameWeight_i$ means *frameWeight* takes the minimum value of the frame importance of same frames in two comparing sentences. $|frameSame|$ is the number of same frames between two sentences. The other part is *PathScore* counts for the max number of frames evoked by two sentences. $Maximum(|frameOne|, |frameTwo|)$ selects a bigger value between $|frameOne|$ and $|frameTwo|$.

We take S1 and S2 in Table 3 as examples to show the process of computing sentence similarity using FrameNet-based approach and illustrate that FrameNet-based approach leads to more interpretable results. As we can see in Table 3, *frameSame* of S1 and S2 has six elements (i.e. *Subordinates_and_superiors*, *Assessing*, *Origin*, *Military*, *Leadership*, *Calendric_unit*). Then, $|frameSame|$ is 6 and the result of $\sum_{i=1}^{|frameSame|} minFrameWeight_i$ is 0.83999 by executing Equation (4) and (5). Take *Subordinates_and_superiors* as an example, in Equation (4), *coreFEsCount* is 2, *FEsCount* (of S1) is 76, and *FEsCount* (of S2) is 65. Two corresponding values are 2/76 and 2/65, respectively. The corresponding *frameWeight* values are 0.12252 and 0.18393. The smaller value "0.12252" is taken as *minFrameWeight*.

Next, we single out the elements that appear just in *frameOne* and *frameTwo*, i.e. *frameOneR* and *frameTwoR*. *frameOneR* has one element (i.e. *Delivery*) and *frameTwoR* has three elements (i.e. *Presence*, *LOCATION*, *STATE_OR_PROVINCE*). From GIFN, we can get that there exists a shortest path between *Delivery* in *frameOneR* and *Presence* in *frameTwoR*. Therefore, *frameRel* contains only one element (i.e. < *Delivery*, *Presence* >). Use assignment values of frame relations in Table 1 as *pathWeight*, we calculate *pathValue* between *Delivery* and *Presence*. According to Equation (2) and (3), the result of *PathScore* is 0.45625.

Finally, according to Equation (6), the similarity result between S1 and S2 is 0.8856. With reference to the threshold "0.505", we can conclude that S1 and S2 are similar, and this result is consistent with the manual labeling in the data set MSRP.

3.3. Element focusing Sentence-BERT

To better handle a general linguistic reality that different sentence elements play varying roles in the meaning of a sentence, we propose an adapted version of SBERT, i.e. element focusing SBERT (EF-SBERT). The structure of the EF-SBERT model is identical to that of SBERT, as shown in Fig. 7.

In Fig. 7, there are five layers in the EF-SBERT model architecture.

Layer 1. Input layer. Sentences have complex structures and every sentence consists of critical elements (subject, predicate, object, etc.). NER that identifies entities of specific significance and the Stanford Parser that performs dependency parsing of Sentence One and Sentence Two are used in the first layer to extract key information in two sentences.

Some definitions involved in the first layer are described as follows.

Definition 7. S_{basic} is a complete sentence, which is considered to be the basic part of a sentence.

Definition 8. S_{ef} is the improved part of a sentence, which contains critical aspects of a sentence (primarily from the subject, predicate, and object, among other things), i.e. the information of a sentence after processing.

Table 4 shows examples of both the basic and improved parts of two statements.

Layer 2. Encoder layer. The task of Encode layer is to achieve sentence embeddings. In this layer, BERT plays a major role. It is composed of several transformer encoder layers. The input sentence is mapped into a vector space through the BERT model. Like the work of Devlin et al. (2018), we use two model sizes (BERT-base and BERT-large). BERT-base has 12 layers, 768 hidden sizes and 12 self-attention heads while BERT-large has 24 layers, 1024 hidden sizes and 16 self-attention heads. We report our experiment results based on these two model sizes in Section 4.3.

In order to obtain sentence embeddings, four parts of two sentences $(S_{basic1}, S_{ef1} and S_{basic2}, S_{ef2})$ from the Input layer are sent to BERT models, as seen in Fig. 7.

Layer 3. Computing sentence embeddings. Through Encoder layer, we get four sentence embeddings. In order to obtain two complete sentence embeddings of Sentence One and Two, we use these four sentence embeddings to compute complete sentence embeddings by using Equation (7). The last output layer takes these two complete sentence embeddings as input.

Some definitions involved Layer 3 are described as follows.

Table 4
Examples of the basic part S_{basic} and the improved part S_{ef} of S3 and S4.

S_{basic}(original sentences)	S_{ef}
S3: The Prime Minister, Junichiro Koizumi, joined the criticism.	Prime Minister joined criticism
S4: Colin Powell, the Secretary of State, said contacts with Iran would not stop.	Colin Powell said contacts not stop

Definition 9. *$Semb_{basic1}$ and $Semb_{basic2}$ are sentence embeddings of the basic parts of One and Two.*

Definition 10. $Semb_{ef1}$ and $Semb_{ef2}$ are sentence embeddings of the improved parts of One and Two.

Definition 11. W_{ef} is a weight factor that is used to alter the embeddings of two sections, the basic and the improved.

Definition 12. *Semb* is the complete sentence embedding defined as follows.

$$Semb = Semb_{basic} + Semb_{ef} * W_{ef} \tag{7}$$

when $W_{ef} = 0$, EF-SBERT equals SBERT. In FEFS3C, we tune the value of W_{ef} so as to obtain good results. According to specific adjustment results on a validation set, we assigned the value of W_{ef} as "0.2". The specific process of adjusting the value of W_{ef} is described as follows.

First of all, a list of values over 1.0 is selected to test. Then, we turn to set values between 0.1 and 1.0. Finally, we limit the range of W_{ef} value and select the best value by evaluating the performance. The adjustment results of W_{ef} is shown in Fig. 8. Based on the feedback from our tests, we set W_{ef} to "0.2" with limiting W_{ef} between 0.1 and 1.0.

Layer 4. Pooling layer. In this layer, pooling operations are applied to the output of two BERT networks to generate two fixed-sized sentence

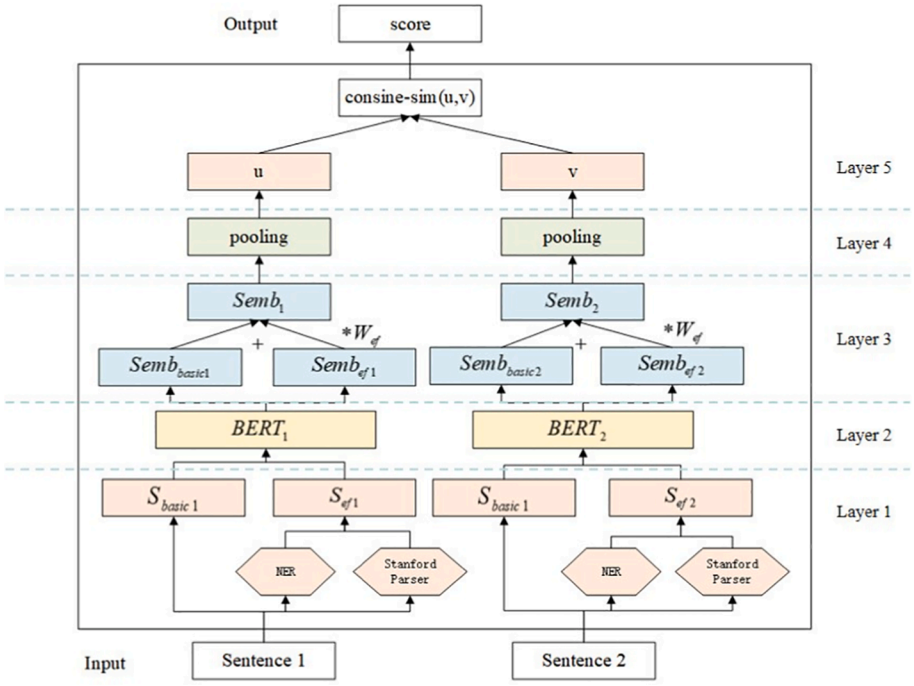

Fig. 7. EF-SBERT model architecture.

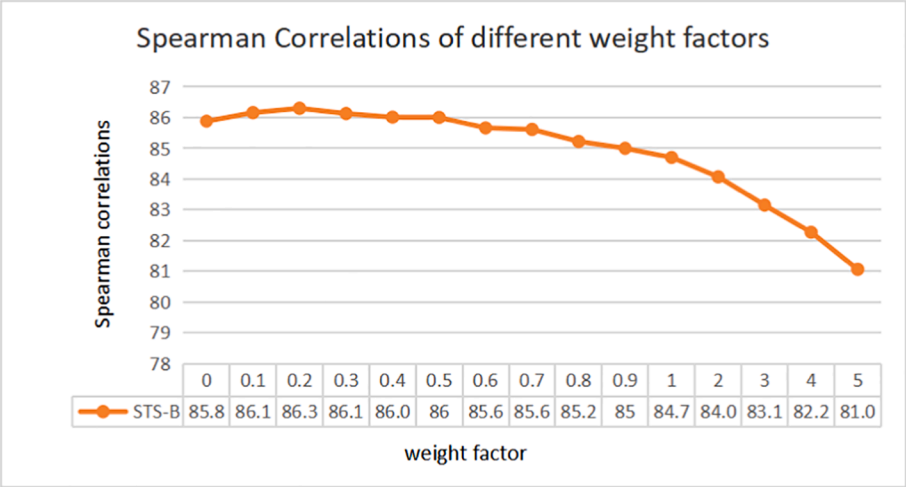

Fig. 8. The specific adjustment process of..W_{ef}

embeddings "u", "v", in the same way as in Reimers & Gurevych (2019).

Layer 5. Output layer. In the last layer, EF-SBERT computes the cosine-similarity between two sentence embeddings "u", "v" and finally gets a computation result.

4. Experiments

In this section, we first introduce experiment goals and the data sets used in experiments. Then, second subsection shows evaluation metrics. Next, the third subsection provides details of experiment results, after which the discussion of results is given.

4.1. Experiment goals and data sets

In total, we designed three different experiments and used two common data sets in experiments. The purpose of three experiments can be outlined as follows. Two common data sets used in experiments are also briefly introduced.

(i) In order to verify whether the introduction of frame importance can lead to an improvement performance of EFS3C in our previous work, we designed and executed the first experiment. In view of the fact that the data set we used in our previous work is **Microsoft Research Paragraph Corpus (MSRP)** (Das & Smith, 2009; Dolan, Quirk, Brockett, & Dolan, 2004), to better compare with our previous work and verify the performance, we first use MSRP for comparison experiments.

MSRP is widely used to evaluate similarity computation methods. The data set includes 5,801 sentence pairs. These sentence pairs are divided into two groups: a train set (4,076) and a test set (1,725). All these sentence pairs have a manually assigned value, 0 or 1, indicating they are similar or not respectively. Our previous works (Liu et al., 2020; Wang et al., 2021) used MSRP to compute sentence similarity and achieved a good performance in the term of F1-score.

(ii) The primary purpose of the second experiment is to see whether using the BERT-based model can improve the performance comparing with traditional sentence similarity methods. Reimers & Gurevych (2019) employed Semantic Textual Similarity Benchmark (STS-B) (Cer, Diab, Agirre, Lopez-Gazpio, & Specia, 2017) to complete sentence similarity tasks, which outperformed other state-of-the-art sentence embeddings methods. In order to better compare with Reimers & Gurevych (2019)'s work, STS-B is used in our experiments.

The STS-B data set is widely used to evaluate supervised STS systems. There are 8,628 sentence pairings in the data set, divided into three categories: captions, news, and forums. Each sentence pair is given a fine-grained gold standard semantic similarity between 0 and 5 and is separated into three sets: a train set (5,749 sentence pairs), a development set (1,500 sentence pairings), and a test set (1,379 sentence pairs).

(iii) The major goal of the third experiment is to see whether FEFS3C can show better results over other independent models. Considering that Reimers & Gurevych (2019)'s work plays a landmark role in sentence similarity computation tasks in recent years, we continue to use the data set STS-B and the evaluation metric "Spearman correlation" as they used to do comprehensive experiments.

4.2. Evaluation metrics

In the comparative experiments with and without introducing into the importance of frames based on FrameNet, we use F1-score (Derczynski, 2016) to evaluate the FrameNet-based approach. F1-score is a statistical evaluation metric often used with MSRP to measure the advantages and disadvantages of methods. It takes into account both precision and recall of a model and can be used as a metric to evaluate the overall effectiveness of a model.

In semantic textual similarity tasks, two commonly used evaluation metrics are Pearson correlation and Spearman correlation. Since Reimers, Beyer, and Gurevych (2016) have showed that Pearson correlation is badly suited for STS, we employed Spearman correlation (ρ) as the verification method, similarly as it is used in Reimers & Gurevych (2019). To evaluate the model based on BERT and the integration approach, we employ Spearman correlation coefficients between predicted similarity and gold standard similarity scores as an evaluation metric, which is the same way used by Li, Zhou, He, Wang, Yang, & Li (2020).

4.3. Experiment results

We divide experiments into three groups.

(i) Experiments based on FrameNet with and without introducing into frame importance.
(ii) Experiments based on BERT without integrating FrameNet.
(iii) Experiments based on FrameNet and EF-SBERT.

4.3.1. Experiments based on FrameNet with and without introducing into frame importance

Table 5 shows the results of similarity computations based on EFS3C in our previous work (Wang et al., 2021) with and without introducing into frame importance. We employ MSRP as the test data set, which has 1,725 sentence pairs. Since dataset MSRP includes two classes and to better compare with our previous work, we use F1-score for the first experiment.

After introducing into frame importance, the similarity testing result has been improved, and the F1-score increases by 2.02%. It is proved that the introduction of frame importance affects sentence similarity to a certain extent.

4.3.2. Experiments using BERT-based model without integrating FrameNet

Table 6 shows the results of sentence similarity computations based on BERT models. STS-B is the test data set. The Spearman's correlation coefficients between predicted similarity and gold standard similarity scores (Li et al., 2020) are presented. The test data includes 1,379 sentence pairs.

As shown in Table 6, BERT-based methods improved by about 9% in the semantic textual similarity tasks, which outperformed methods proposed between 2014 and 2019. It is indicated that BERT-based methods show great advantages.

4.3.3. Experiments based on FrameNet and EF-SBERT

Table 7 shows the testing similarity computation results based on FrameNet and EF-SBERT. The test data set is STS-B. The test data includes 1,379 sentence pairs, which are the test group in STS-B.

We report Spearman correlation in Table 7. FEFS3C integrates FrameNet with EF-SBERT taking contextual issues into consideration. As shown in Table 7, EF-SBERT-large with EFS3C which fails to account for the impact of semantic importance of different frames can slightly reduce the performance, which explains why it is slightly worse than models proposed in the previous work. However, after taking frame importance into consideration, our method performs better than other methods, which proves the superiority of FEFS3C.

Even though the difference between our proposed ensemble method FEFS3C and EF-SBERT is not statistically significant, it is a new attempt to combine a traditional method with a deep learning-based method in sentence similarity computation tasks. Moreover, FEFS3C has better interpretability than other methods on tasks of computing sentence similarity. Considering the example presented in Table 3, the similarity of two sentences is calculated from the semantic level, which cannot be explained by a purely neural network-based approach.

4.4. Discussion

On the basis of our previous work, introducing frame importance into sentence similarity computation improves the performance of the FrameNet-based approach in terms of F1-score. Then we incorporate advanced deep learning models and verify on another data set "STS-B", using integrated learning methods, and achieve good experiment results.

The testing results presented in the three subsections above show that FEFS3C has achieved better performance. There are three main reasons for our good performance. First, we use frame semantics to analyze sentences and compare sentences in fine granularity, which not only makes results interpretable, but also improves the performance of sentence similarity computation method. Second, EF-SBERT based on deep learning technologies adopts bi-encoder structure, which makes good use of the characteristics of sentence pairs. Last but not least, ensemble method FEFS3C exhibits apparent advantages in sentence similarity computation, which obtains better predictive performance than independent methods.

However, despite experiment results of the method have improved, the improvement is not significant. One of the main reasons is that different words may evoke the same frame, which makes semantics of frames deviate. The other reason is that although the EF-SBERT model is powerful and can capture more semantic information, it only enhances the representation of a sentence without changing its original structure, which leads to the improvement slightly. Hence, as stated in Xue, Jiang, Neri, and Liang (2021), it is important to design an appropriate network structure to further improve the performance in the sentence similarity computation task.

Table 5
Comparison of similarity test results with and without introducing into frame importance.

Experiments	F1-score
EFS3C (without introducing) (Wang et al., 2021)	80.30%
EFS3C (with introducing)	82.32%

Table 6
The results of sentence similarity computations based on BERT models.

Model		Spearman (ρ)
Previous methods proposed between 2014 and 2019	Avg. GloVe embeddings (Pennington, Socher, & Manning, 2014)	58.02
	Avg. BERT embeddings (Zhang, Kishore, Wu, Weinberger, & Artzi, 2019)	46.35
	USE (Conneau, Kiela, Schwenk, Barrault, & Bordes, 2017)	74.92
	InferSent-Glove (Williams, Nangia, & Bowman, 2017)	75.30
	Skip-thought (Kiros, Zhu, Salakhutdinov, Zemel, Urtasun, Torralba, & Fidler, 2015)	76.70
BERT-based methods	BERT-base (Devlin et al., 2018)	85.80
	SBERT-base (Yin, Zhang, Zhu, Liu, & Yao, 2020)	85.76
	EF-SBERT-base	86.30

Table 7
Evaluation on the STS benchmark test set.

Model	Spearman
Trained on STS-B dataset	
BERT-base (Devlin et al., 2018)	85.80
SBERT-base (Yin et al., 2020)	85.76
EF-SBERT-base	86.30
BERT-large (Devlin et al., 2018)	86.50
SBERT-large (Yin et al., 2020)	86.35
FEFS3C: EF-SBERT-large + EFS3C	86.33
FEFS3C: EF-SBERT-large + EFS3C with introducing into frame importance	86.78

5. Related work

In the realm of natural language processing, there are numerous ways for computing sentence similarity. Methods based on word co-occurrence, methods based on a lexical database, and methods based on neural networks are the three key types of related study (Li, McLean, Bandar, O'shea, & Crockett, 2006).

In Information Retrieval (IR) systems (Boyce, Boyce, Meadow, Kraft, Kraft, & Meadow, 2017), methods based on word co-occurrence are extensively used. These approaches are based on the notion that two sentences that are similar have more terms in common. For instance, Peking University Computing Language presented the formula $2c/(m+n)$, where m and n represent the number of words in two sentences, respectively, and c represents the number of identical words in two sentences (Wang, Chi, Chang, & Bai, 2005). A lexical dictionary is used in one variation of word co-occurrence methods (Okazaki, Matsuo, Matsumura, & Ishizuka, 2003) to determine the similarity of a pair of

terms obtained from two sentences being compared. Another variation is the pattern matching method (Chiang & Yu, 2005), which uses a basic pattern matching algorithm to compute sentence similarity. Word co-occurrence approaches are straightforward, but they ignore the word order of sentences and fail to appreciate the meaning of the word in context (Li et al., 2006).

Methods based on a lexical database use a predetermined word hierarchy to compute sentence similarity (Pawar & Mago, 2019). The latent semantic analysis (LSA) (Foltz, Kintsch, & Landauer, 1998; Landauer, Laham, Rehder, & Schreiner, 1997; Dennis, Landauer, Kintsch, & Quesada, 2003) and the Hyperspace Analogues to Language (HAL) (Burgess, Livesay, & Lund, 1998) are two well-known classical lexical database methodologies. Both of these two approaches use lexical database information to capture the meaning of a word or sentence. In Gabrilovich and Markovitch (2007), Gabrilovich et al. (2017) computed sentence similarity using Explicit Semantic Analysis (ESA) based on Wikipedia concepts. Based on word alignments across a larger corpus, Sultan, Bethard, and Sumner (2015) computed sentence similarity using word-alignment models. However, methods based on a lexical database have one limitation that they neglect the true meaning of words. Another limitation is that the information content of different corpora varies a lot, which may affect the similarity.

Recently, there have been many studies about using models based on neural networks for computing sentence similarity, which have produced significant improvements. He, Gimpel, and Lin (2015) proposed a convolutional neural network-based model for computing sentence similarity. Another revolutionary model proposed by Tai, Socher, and Manning (2015) used Tree-LSTMs to model sentences and performed well on semantic relatedness prediction of two sentences. For matching issues, Lu and Li (2013) proposed a deep neural network that can sufficiently explore the nonlinearity and hierarchy of the matched texts. Despite improvements, the majority of neural network methods concentrate on the representation of each sentence rather on comparable or shared features of sentence pairs (Wang, Yu, Zhang, Gong, Xu, Wang, Zhang, & Zhang, 2017).

Comparing with sentence similarity computation methods above, FEFS3C has demonstrated encouraging outcomes over other independent models. The methodology in this paper makes use of frame semantics which overcomes the shortcomings of computing similarity only for keywords, fully considering the semantic information of sentences, and incorporates powerful deep learning models to boost the execution efficiency of the method.

6. Conclusion and future work

Previous approaches of sentence similarity computation mainly target keywords and structures of sentences. These methods have been severely affected by the incomplete description of sentence semantic and lacked of interpretablity (Ru, Zhiqiang, Shuanghong, Jiye, & Baker, 2013). In this paper, we propose FEFS3C by integrating FrameNet with EF-SBERT on semantic textual similarity. First of all, based on frame semantics, we take the frame as the basic unit, considering the importance of different frames and compute the similarity by combining the semantic relationship between frames. Next, we compute the similarity between text pairs using advanced BERT-based models, highlighting the great advantages of deep learning technologies. Finally, two models are combined by a hyper-parameter. The results of the experiments reveal that FEFS3C can boost performance to a degree.

In the future, we hope to expand the application of FEFS3C, gradually expanding to practical applications such as defect analysis, information retrieval, and text mining, etc.

CRediT authorship contribution statement

Tiexin Wang: Conceptualization, Methodology. **Hui Shi:** Writing – original draft. **Wenjing Liu:** Data curation. **Xinhua Yan:** Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by National Science Foundation of China (NSFC) under Grant No. 61872182.

References

Atkinson-Abutridy, J., Mellish, C., & Aitken, S. (2004). Combining information extraction with genetic algorithms for text mining. *IEEE Intelligent Systems, 19*(3), 22–30.
Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998). August). The berkeley framenet project. In *In 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1* (pp. 86–90).
Boyce, B. R., Boyce, B. R., Meadow, C. T., Kraft, D. H., Kraft, D. H., & Meadow, C. T. (2017). *Text information retrieval systems*. Elsevier.
Burgess, C., Livesay, K., & Lund, K. (1998). Explorations in context space: Words, sentences, discourse. *Discourse Processes, 25*(2–3), 211–257.
Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., & Specia, L. (2017). Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. arXiv preprint arXiv:1708.00055.
Chandrasekaran, D., & Mago, V. (2021). Evolution of Semantic Similarity—A Survey. *ACM Computing Surveys (CSUR), 54*(2), 1–37.
Chiang, J. H., & Yu, H. C. (2005). Literature extraction of protein functions using sentence pattern mining. *IEEE Transactions on Knowledge and Data Engineering, 17*(8), 1088–1098.
Chopra, S., Hadsell, R., & LeCun, Y. (2005, June). Learning a similarity metric discriminatively, with application to face verification. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) (Vol. 1, pp. 539-546). IEEE.
Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364.
Das, D., & Smith, N. A. (2009). August). Paraphrase identification as probabilistic quasi-synchronous recognition. In *In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* (pp. 468–476).
Dennis, S., Landauer, T., Kintsch, W., & Quesada, J. (2003). Introduction to latent semantic analysis. In *In 25th Annual Meeting of the Cognitive Science Society* (p. (p. 25).).
Derczynski, L. (2016). May). Complementarity, F-score, and NLP Evaluation. In *In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (pp. 261–266).
Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv: 1810.04805.
Dolan, W., Quirk, C., Brockett, C., & Dolan, B. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources.
Foltz, P. W., Kintsch, W., & Landauer, T. K. (1998). The measurement of textual coherence with latent semantic analysis. *Discourse processes, 25*(2–3), 285–307.
Gabrilovich, E., & Markovitch, S. (2007). January). Computing semantic relatedness using Wikipedia-based explicit semantic analysis. *In IJcAI, 7*, 1606–1611.
He, H., Gimpel, K., & Lin, J. (2015). September). Multi-perspective sentence similarity modeling with convolutional neural networks. In *In Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1576–1586).
Huang, X., & Hu, Q. (2009). July). A bayesian learning approach to promoting diversity in ranking for biomedical information retrieval. In *In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* (pp. 307–314).
Huang, Y. H., Lee, S. R., Ma, M. Y., Chen, Y. H., Yu, Y. W., & Chen, Y. S. (2019). EmotionX-IDEA: Emotion BERT–an Affectional Model for Conversation. arXiv preprint arXiv:1908.06264.
Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Skip-thought vectors. In Advances in neural information processing systems (pp. 3294-3302).
Kshirsagar, M., Thomson, S., Schneider, N., Carbonell, J. G., Smith, N. A., & Dyer, C. (2015). July). Frame-semantic role labeling with heterogeneous annotations. In *In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* (pp. 218–224).
Landauer, T. K., Laham, D., Rehder, B., & Schreiner, M. E. (1997). August). How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans. In *In Proceedings of the 19th annual meeting of the Cognitive Science Society* (pp. 412–417).
Li, B., Zhou, H., He, J., Wang, M., Yang, Y., & Li, L. (2020). On the sentence embeddings from pre-trained language models. arXiv preprint arXiv:2011.05864.

Li, Y., McLean, D., Bandar, Z. A., O'shea, J. D., & Crockett, K. (2006). Sentence similarity based on semantic nets and corpus statistics. *IEEE transactions on knowledge and data engineering, 18*(8), 1138–1150.

Liu, W., Wang, T., Yang, Z., & Cao, J. (2020). In *November). A Context-Aware Computing Method of Sentence Similarity Based on Frame Semantics* (pp. 114–126). Cham: Springer.

Lu, Z., & Li, H. (2013). A deep architecture for matching short texts. *Advances in neural information processing systems, 26*, 1367–1375.

Okazaki, N., Matsuo, Y., Matsumura, N., & Ishizuka, M. (2003). Sentence extraction by spreading activation through sentence similarity. *IEICE TRANSACTIONS on Information and Systems, 86*(9), 1686–1694.

O'Neill, D., Xue, B., & Zhang, M. (2021). Evolutionary Neural Architecture Search for High-Dimensional Skip-Connection Structures on DenseNet Style Networks. *IEEE Transactions on Evolutionary Computation*.

Pawar, A., & Mago, V. (2019). Challenging the boundaries of unsupervised learning for semantic similarity. *IEEE Access, 7*, 16291–16308.

Pennington, J., Socher, R., & Manning, C. D. (2014). October). Glove: Global vectors for word representation. In *In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).

Petruck, M. R., & Ellsworth, M. (2018). June). Representing spatial relations in FrameNet. In *In Proceedings of the First International Workshop on Spatial Language Understanding* (pp. 41–45).

Reimers, N., Beyer, P., & Gurevych, I. (2016). December). Task-oriented intrinsic evaluation of semantic textual similarity. In *In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 87–96).

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

Ru, L., Zhiqiang, W., Shuanghong, L., Jiye, L., & Baker, C. (2013). Chinese sentence similarity computing based on frame semantic parsing. *Journal of Computer Research and Development, 50*(8), 1728.

Ruppenhofer, J., Ellsworth, M., Schwarzer-Petruck, M., Johnson, C. R., & Scheffczyk, J. (2006). FrameNet II: Extended theory and practice.

Seni, G., & Elder, J. F. (2010). Ensemble methods in data mining: Improving accuracy through combining predictions. *Synthesis lectures on data mining and knowledge discovery, 2*(1), 1–126.

Sultan, M. A., Bethard, S., & Sumner, T. (2015, June). Dls@ cu: Sentence similarity from word alignment and semantic vector composition. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015) (pp. 148-153).

Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075.

Wan, S., Dras, M., Dale, R., & Paris, C. (2006). November). Using dependency-based features to take the'para-farce'out of paraphrase. In *In Proceedings of the Australasian language technology workshop 2006* (pp. 131–138).

Wang, J., Yu, L., Zhang, W., Gong, Y., Xu, Y., Wang, B., … Zhang, D. (2017). August). Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval* (pp. 515–524).

Wang, R., Chi, Z., Chang, B., & Bai, X. (2005). Chinese sentence similarity measure based on word sequence length and word weight. *Jisuanji Gongcheng/Computer Engineering, 31*(13), 142–144.

Wang, T., Liu, W., Yang, Z., & Cao, J. (2021). Frame semantic extension and similarity computation for english sentences. *Journal of Chinese Computer Systems, 42*(10), 2059–2064.

Wang, T., Truptil, S., & Benaben, F. (2017). An automatic model-to-model mapping and transformation methodology to serve model-based systems engineering. *Information Systems and e-Business Management, 15*(2), 323–376.

Williams, A., Nangia, N., & Bowman, S. R. (2017). A broad-coverage challenge corpus for sentence understanding through inference. arXiv preprint arXiv:1704.05426.

Xue, Y., Jiang, P., Neri, F., & Liang, J. (2021). A Multi-Objective Evolutionary Approach Based on Graph-in-Graph for Neural Architecture Search of Convolutional Neural Networks. *International Journal of Neural Systems, 31*(09), 2150035.

Xue, Y., Wang, Y., Liang, J., & Slowik, A. (2021). A self-adaptive mutation neural architecture search algorithm based on blocks. *IEEE Computational Intelligence Magazine, 16*(3), 67–78.

Yih, S. W. T., Chang, M. W., Meek, C., & Pastusiak, A. (2013). Question answering using enhanced lexical semantic models.

Yin, X., Zhang, W., Zhu, W., Liu, S., & Yao, T. (2020). Improving sentence representations via component focusing. *Applied Sciences, 10*(3), 958.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675.

Zhang, X., Sun, X., & Wang, H. (2018, April). Duplicate question identification by integrating framenet with neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).