Computer Modeling in Engineering & Sciences
DOI:10.32604/cmes.2020.010579
Article

Enhancing Embedding-Based Chinese Word Similarity Evaluation with Concepts and Synonyms Knowledge

Fulian Yin, Yanyan Wang, Jianbo Liu* and Meiqi Ji

Communication University of China, Beijing, 100024, China
*Corresponding Author: Jianbo Liu. Email: ljbcuc@163.com
Received: 13 March 2020; Accepted: 08 May 2020

Abstract: Word similarity (WS) is a fundamental and critical task in natural language processing. Existing approaches to WS are mainly to calculate the similarity or relatedness of word pairs based on word embedding obtained by massive and high-quality corpus. However, it may suffer from poor performance for insufficient corpus in some specific fields, and cannot capture rich semantic and sentimental information. To address these above problems, we propose an enhancing embedding-based word similarity evaluation with character-word concepts and synonyms knowledge, namely EWS-CS model, which can provide extra semantic information to enhance word similarity evaluation. The core of our approach contains knowledge encoder and word encoder. In knowledge encoder, we incorporate the semantic knowledge extracted from knowledge resources, including character-word concepts, synonyms and sentiment lexicons, to obtain knowledge representation. Word encoder is to learn enhancing embedding-based word representation from pre-trained model and knowledge representation based on similarity task. Finally, compared with baseline models, the experiments on four similarity evaluation datasets validate the effectiveness of our EWS-CS model in WS task.

Keywords: Word representation; concepts and synonyms knowledge; word similarity; information security

1 Introduction

Currently, information security has become a global problem, and it is important to study and learn about security-related technologies. Especially in the field of text information security, through similarity technology research, we can not only detect information security vulnerabilities, but also effectively prevent text information security problems. Word similarity (WS) aims to measure the relatedness or similarity degree between word pairs [1–3], which is a fundamental and critical component in many tasks, such as information retrieval [4,5], detection of information security [6,7], machine translation [8], semantic disambiguation [9] etc.

Tradition WS methods obtain the similarity of word pairs by using relationship of word pairs in public lexical resources, which provide professional and authoritative knowledge by experts and scholars, such as character-word concept [10,11] and synonym information [11,12]. Afterwards, the embedding approaches

based on corpus get more and more attention to measure WS, including some well-known models, such as continuous bag-of-words (CBOW) and Skip-gram (SG) in Word2Vec [13,14], GloVe [15], and improved methods considering more complex network structures [10,16–19]. However, most of above models obtain excellent results based on massive corpus, and exist serious expression ambiguity.

In order to address these problems, some studies proposed to construct more fine-grained unit of word representation, such as character [20–24], radical (the graphical component of Chinese) [24–26]. Chen et al. [20] and Xu et al. [23] incorporated character information into embedding models to construct word representation models, character-enhanced word embedding model (CWE) and similarity-based character-enhanced word embedding (SCWE), respectively. Yu et al. [24] proposed a joint word embedding (JWE) model by considering Chinese words, characters, and fine-grained sub-character components.

Some studies also captured rich semantic knowledge by incorporating extra information, such as sentimental information [7,27–29], synonym [1] and concept [30] information. Niu et al. [30] combined the lexical concepts in HowNet as prior knowledge to enhance word embedding representation, which realized sense disambiguation for better word similarity. Huang et al. [1] introduced multiple prior knowledge including statistical features or lexicon resources into word embedding to improve the performance of word similarity. However, these methods cannot calculate words not included in training corpus, and just simply used the combination of word similarities calculated by different features, which ignored lexical overlap relationship between different features.

Take the word "骄傲" (pride) as an example, Fig. 1 shows the related words of it in different expert knowledge resources including synonym base CiLin, word concept base HowNet, and character concept base, and top ten related words extracted from Skip-gram model. From the results, first we can know that the different related word sets obtained by prior knowledge resources provide rich semantic information. For example, the synonyms of the word "骄傲" (pride) have different meanings with different sentiment tendencies, "光荣" (glory) has a positive sentiment, however "自满" (complacent) has a negative one. This information is difficult to learn using pre-trained embedding model. Our work is motivated by this idea, we encode knowledge representation of each word by incorporating different semantic knowledge, such as synonym, character-word concept.

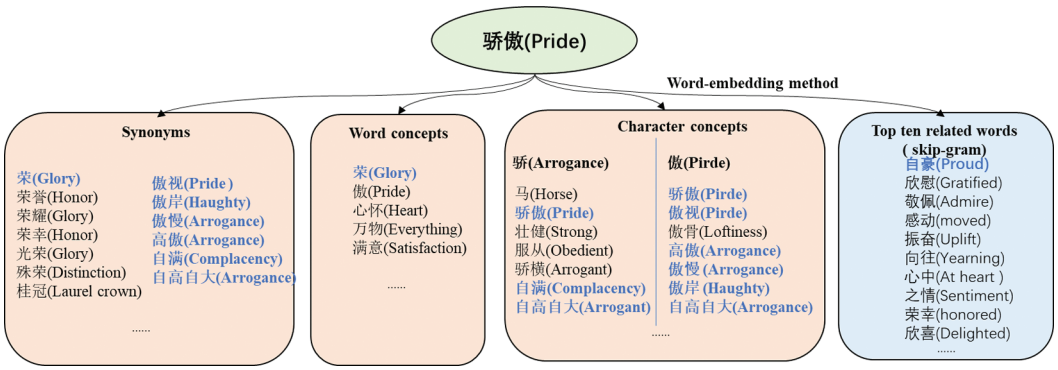

Figure 1: An example of semantic knowledge for word "骄傲" (pride). If a word repeats in multiple lexical resources, it is marked blue color. Because the related words are extracted from different knowledge resources, many words have the nearly similar meaning, which lead to the same English expression

In this paper, we propose an enhancing embedding-based Chinese word similarity evaluation with concepts and synonyms knowledge (EWS-CS), which consists of three major modules i.e., knowledge extraction, knowledge encoder and word encoder. First, we extract related knowledge including concepts, synonyms, to construct related knowledge word set. Then, in knowledge encoder, the core is to encode

the knowledge representation via integrating synonym information from CiLin, character-word level concept from HowNet and sentiment information from lexicons resources to supplement the semantic information under small corpus. Word encoder is to learn enhancing embedding-based word representation from pre-trained model and knowledge representation based on similarity task. The experiments are conducted on four evaluation datasets to validate the effectiveness of our method in WS task. The result shows that our EWS-CS model can improve the stability and adaptability under small corpus.

The rest of the paper is organized as follows: we introduce some methods of word similarity in Section 2, and describe our model in Section 3. Then, we present the results and performance comparisons in Section 4, followed by the conclusions and next research plan in Section 5.

2 Related Work

There are mainly three popular methods for word similarity (WS), including embedding-based method, lexical resource method and hybrid method.

2.1 Embedding-Based Method

The popular WS method currently is to calculate the cosine similarity between the vectors of word pairs based on word embedding model trained by large-scale corpus. Some widely used models include CBOW, SG [13,14], GloVe [15]. The CBOW model predicted the vector representation of the current word through context words, by contrast, the SG model utilized a word to achieve the representation of context word. The GloVe model integrated the global information with local contexts and learned the word representation using matrix decomposition. Most of the subsequent models are basically improved on the basis of above models. Ji et al. [31] proposed a WordRank model, and converted the word vector learning problem into a sorting problem to place the context words with strong relevance at the top of the list. The directional Skip-Gram (DSG) model proposed by Song et al. [18] is an extended model based on SG, which considered the direction factor of context words. It not only predicted its context words, but also clearly pointed out the left or right direction of these words. Sakketou et al. [32] proposed to incorporate the semantic information and the complex relationships of the words by semantic lexicons based on GloVe to improve the similarity calculation task. Peters et al. [17] proposed the ELMo model, which employed a linear combination of layers to represent word vectors based on a bidirectional language model. BERT [16] aimed to pre-train deep two-way representation based on the left and right contexts of all layers. Zhang et al. [19] proposed the ERNIE model, which fused text and knowledge mapping information based on BERT. These methods highlight large-scale corpus to train for word embedding, however some limitations are ignored:

First, the wastage of these models is huge, and it is not effective to the research development.

Second, the distributed hypothesis that similar words have similar distributions is inherently questionable, because some words in the same position are not all synonymous. For example, the distributions of “good” and “bad” are similar, but they are adverse in fact.

Third, training objective and task are inconsistent. The parameters that achieve the state-of-the-art results in training process may not be suitable for similarity tasks.

In addition, the internal information of a word is taken into account, mainly including character feature [20–22,24,33], radicals in Chinese characters [24–26]. Chen et al. [20] proposed character-enhanced word embeddings (CWE) model, which introduced internal character information into word embedding methods to alleviate excessive reliance on the external information. Sun et al. [22] proposed a hybrid model to learn word embedding by simultaneously considering the pixel-level characteristics, character-level characteristics and context characteristics of words.

2.2 Lexical-Based Resource Method

The lexical-based resource approach utilizes synonym, concept relationships between words in different lexical resources to enrich the semantic and sentiment information of words. Commonly used lexical resources include WordNet [2,34,35], CiLin [12,36,37], HowNet [38,39].

WordNet is a lexical database for the English language [11]. It provided a short, summary definition for each synset, which consisted of a group of words with the same meaning. Jimenez et al. [35] exploited the related word set from WordNet graph to calculate word similarity, and achieved the similar effect as the word embedding method. CiLin [12] consists of synonyms and related word of each word. Chen et al. [36] calculated the semantic similarity between words by exploiting the path and depth in CiLin, and then assigned different weights to the edges between the different layers. This method made the value of similarity change dynamically, not limited to fixed value.

At present, the most widely used word conceptual lexicon in Chinese is HowNet [38], which describes the concepts represented by Chinese and English words and reveals the relationship between concepts and their attributes. Liu et al. [38] first explored the calculation of lexical semantic similarity in HowNet. Zhu et al. [39] calculated the word similarity by integrating HowNet and CiLin. They first calculated the single similarity according to the characteristics of each lexical resource, and then obtained the final similarity based on the dynamic weighting strategy. Compared with using a single resource, combination method can include more semantic information and improve the accuracy of word similarity. However, lexical-based resources are not always updated and the timeliness is poor, which lead to a low word coverage.

In addition, there are also some methods by considering sentiment information. Smarandache et al. [2] proposed a fuzzy-based sentiment similarity measurement method, which assigned each word positive, negative and neural sentiment value extracted from SentiWordNet 3.0[1] [40] (an English sentiment lexicon) to construct a fuzzy sentiment vector representation. Then the word similarity was obtained by calculating the vector distance of each word pair. Tang et al. [41] integrated word context and sentiment polarity to construct a hybrid model HyRank. Lan et al. [27] proposed a sentiment word vector learning model based on a convolutional neural network. First, sentimental tags were automatically recognized using emoticons, and then a traditional CNN was extended by using two channels semantics and sentiments. The two were integrated to create a determined word vector (SWV) for similarity calculations.

2.3 Hybrid Method

Currently, word similarity focuses on the combination method, which incorporates multiple semantic information from different knowledge resources since single method exists accuracy and coverage limitations.

A widely used approach incorporate lexical resources into word embedding [29,30,42]. Niu et al. [30] fused the semantic information from HowNet based on the Skip-gram framework, and the different sense weights of the words were calculated based on the context attention mechanism. It could relieve word sense disambiguation. Yan et al. [29] combined word embedding with lexical resource to improve the similarity calculation of retrieval tasks, which solved the gap of synonyms when using lexical-based method.

In addition, Huang et al. [1] incorporated statistical methods, lexicon methods and word embedding methods, and used a variety of mathematical and counter-fitting combination strategies [43] for similarity calculations. Guo et al. [44] proposed a multi-feature fusion similarity algorithm, which adopted prior knowledge features and corpus statistical features. However, these methods generally combine a single internal feature or external feature with word embedding without taking into account the correlation between different knowledge.

[1] https://github.com/aesuli/sentiwordnet

The knowledge information has recently begun to be explored for word similarity, which so far had shown great promise. Inspired by this, we propose an enhancing embedding-based Chinese word similarity evaluation with synonyms and concepts knowledge (ESW-CS). The core of our method is to encode the knowledge representation via integrating synonym information from CiLin, character-word concept from HowNet and sentiment information from lexicons resources to supplement the semantic information under small corpus.

3 Our Enhancing Embedding-Based Chinese Word Similarity Model

Our ESW-CS model consists of three parts, including knowledge extraction, knowledge encoder and word encoder for similarity calculation, which is shown in Fig. 2. First, we extract related knowledge including concepts and synonyms from different knowledge resources, to construct a related word set R_i for a word w_i. Then, we introduce a dual weight method to calculate the importance of n-th related word r_i^n in R_i by combining semantic and sentiment weights. Then, we incorporate related vector v_i^p with pre-trained vector v_i^o of word w_i to obtain the final word representation of w_i. And the vector cosine similarity of each pair of words is calculated as the semantic similarity to obtain the Pearson and Spearman coefficients as the output of the model. Finally, by continuously adjusting the weight parameters of various knowledge during training, the optimal Pearson and Spearman coefficients are obtained and used to evaluate the model.

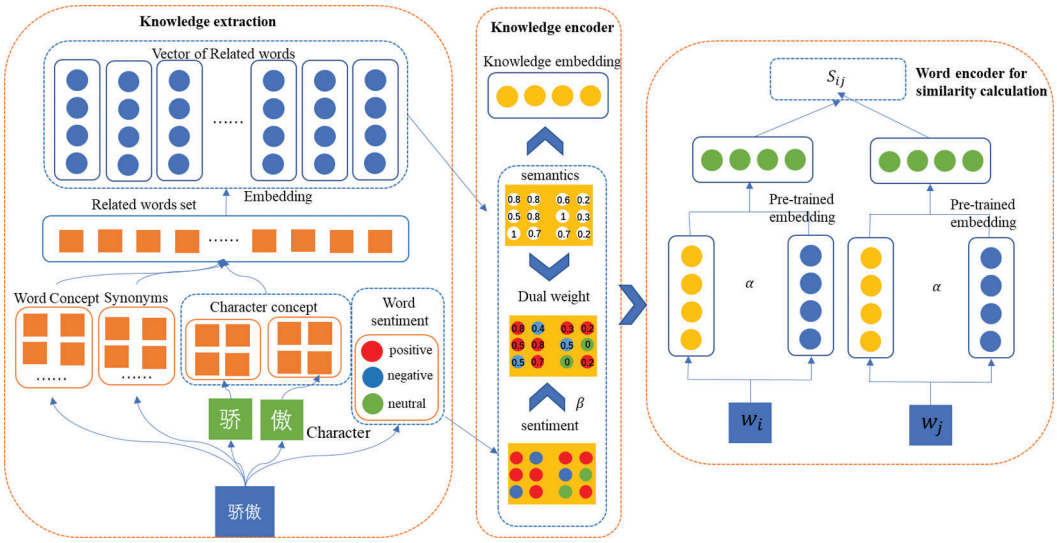

Figure 2: Architecture of our EWS-CS model

3.1 Knowledge Extraction

Lexical knowledge resources are constructed by numerous experts and scholars, which can be considered to provide highly refined and correct information. In our work, we assume that word similarity of a word pair (w_i, w_j) not only relates to the context semantic information, but also has correlation with the semantic knowledge including concept and synonyms from lexical resources. Hence, we extract the candidate knowledge word set R_i' of each word w_i in multiple knowledge resources, such as HowNet and CiLin, among which there may be some poorly related words.

Tab. 1 shows the word concept set (from HowNet), character concept set (from Xinhua online dictionary[2]) and synonym set (from CiLin) constructed by the four example words. It shows that the

[2] http://xh.5156edu.com/

Table 1: Part word set of different features constructed by four sample words

Words	Word concept set		Character concept set	Synonym set
街道 (street)	居民区 (residential area) 地方 (local) 道路 (road)	街 (street)	两边 (both sides) 街巷 (street) 宽阔 (wide) 地方 (place) 商业 (business) 场所 (place) 买 (buy) 卖 (sell) ……	街 (road) 马路 (road) 大街 (road) 街道 (street)
		道 (way)	思想 (Ideological) 方向 (direction) 规律 (law) 道德 (moral) 方法 (method) 路 (way) 道理 (reason) 宗教 (religion) ……	
问题 (problem)	劫难 (disaster) 问 (ask) 实体 (entity) 商讨 (discuss) 辩论 (debate) 提出 (propose)	问 (ask)	问答 (Q&A) 问题 (questions) 问候 (greetings) 询问 (asking) 解答 (answer) 明白 (understand) ……	问题 (question) 答案 (answer) 成绩 (score) 谜底 (answer)
		题 (topic)	内容 (content) 主题 (topic) 讲演 (lecture) 解答 (answer) 考试 (exam) 题材 (subject) 写作 (writing) ……	
挑战 (challenge)	要求 (demand) 指使 (direct) 较量 (contest)	挑 (choose)	扁担 (shoulder pole) 提升 (lift) 两头 (both) 讲求 (stress) 选择 (choose) 挂 (hang) 指使 (instruct) 部件 (part) ……	求战 (fight for war) 离间 (provocation) 搬弄 (fiddle with) 挑拨 (instigate)
		战 (war)	卖力 (struggling) 军 (army) 战绩 (achievement) 争斗 (battle) 战略 (strategy) 战争 (war) 打仗 (warfare) 战术 (tactics) ……	
和平 (peace)	弱 (weak) 群体 (group) 境况 (situation)	和 (and)	谐调 (harmony) 和睦 (harmony) 相安 (mutual security) 和谐 (Harmonious) ……	妥洽 (mediation) 妥协 (compromise) 温和 (mild) 和平 (peace) 和谐 (harmony) 安定 (stability) 愉快 (happy)
		平 (flat)	水面 (water surface) 平地 (flat) 静止 (static) 平面 (plane) 倾斜 (tilt) 平行 (parallel) 一样 (same) 凹凸 (concave-convex)) 平原 (plain) 无 (none) ……	

word sets from different lexicon resources have different representation importance for a word. Take the word "街道" (street) as an example, in the word concept set, "道路" (road) is more important for the representation of words compared with "居民区" (residential area) and "地方" (local). Similarly, in the character concept, "街巷" (street) is more meaningful for word representation than "两边" (both sides). Therefore, we will assign a scoring function to select some words with strong correlation to construct the related word set for each word, which is shown in Fig. 3.

First, we use pre-trained word embedding to obtain D candidate word representations for R_i', $\boldsymbol{v}(w_{Rm})$ and $\boldsymbol{v}(w_{Rk})$ represent the pre-trained words' vectors of the m-th and k-th candidate words w_{Rm} and w_{Rk}, and we calculate the relevance $R(w_{Rm}, w_{Rk})$ of each pair of candidate words based on the vector representation:

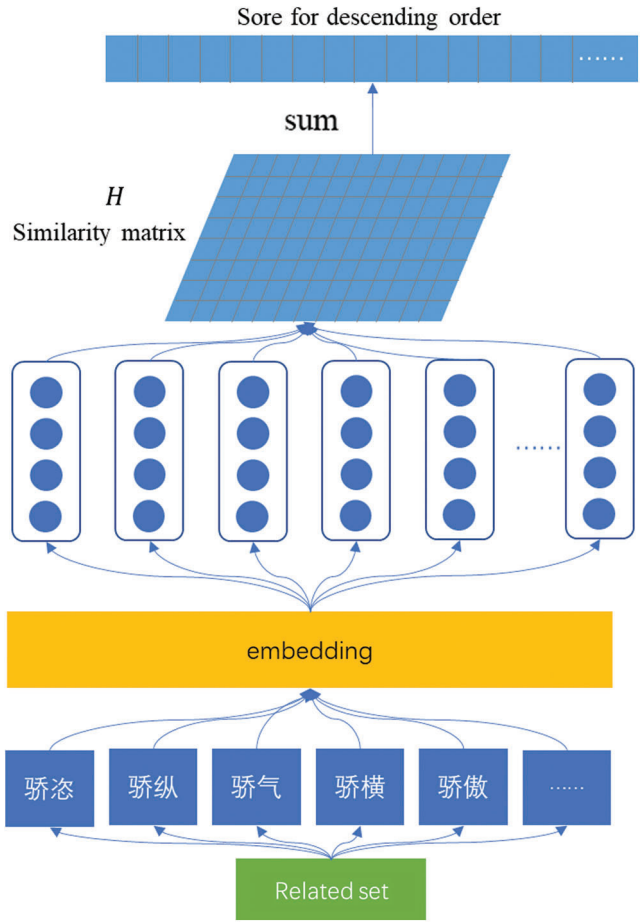

Figure 3: Knowledge extraction model (These Chinese words have similar English expression, "pride")

$$R(w_{Rm}, w_{Rk}) = \frac{\boldsymbol{v}(w_{Rm})\boldsymbol{v}(w_{Rk})}{||\boldsymbol{v}(w_{Rm})|| \times ||\boldsymbol{v}(w_{Rk})||} \tag{1}$$

Then, the similarity matrix H is constructed, $R(w_{Rm}, w_{Rk})$ represents the similarity degree corresponding to the m-th row and k-th column in the matrix H. We score the importance of candidate words, which obtained to sum the matrix by rows. The score $S(w_{Rm})$ of the m-th word is defined as follows:

$$S(w_{Rm}) = \sum_{k=1}^{D} R(w_{Rm}, w_{Rk}) \tag{2}$$

Finally, we select some candidate words with the highest scores to construct the knowledge set R_i.

3.2 Knowledge Encoder

Currently, most of knowledge resources is generally universal, especially for a single Chinese character word "大" (big) shown in Fig. 4, which has 12 meanings in HowNet, and some of meanings are more important than others, such as "龄大" (age old), "高于正常" (above normal). Therefore, it is worthy of study to measure the importance of different meanings or related words for R_i to achieve knowledge representation.

大(big)	
1. 龄大(age old)	7. 重要(important)
2. 年龄 生物(age organisms)	8. 高声(high voice)
3. 高于正常(above normal)	9. 最 距离 (maximum distance)
4. 高等(higher)	10. 长辈(elders)
5. 尺寸 物质(size substance)	11. 严重(serious)
6. 大(big)	12. 强(strong)

Figure 4: Different meanings for the example character "大" (big)

In this section, we propose a dual weight method to assign importance for each word in R_i by considering semantic and sentiment weights. The former one is obtained by calculated the cosine similarity from the vectors between the w_i and n-th related word r_i^n to highlight the different semantic importance of the r_i^n. The latter one is based on sentiment lexicons to assign correlation weight for each related word. Considering that some words have no pre-trained vectors, we design two strategies to obtain a dual weight, which is shown in Fig. 5.

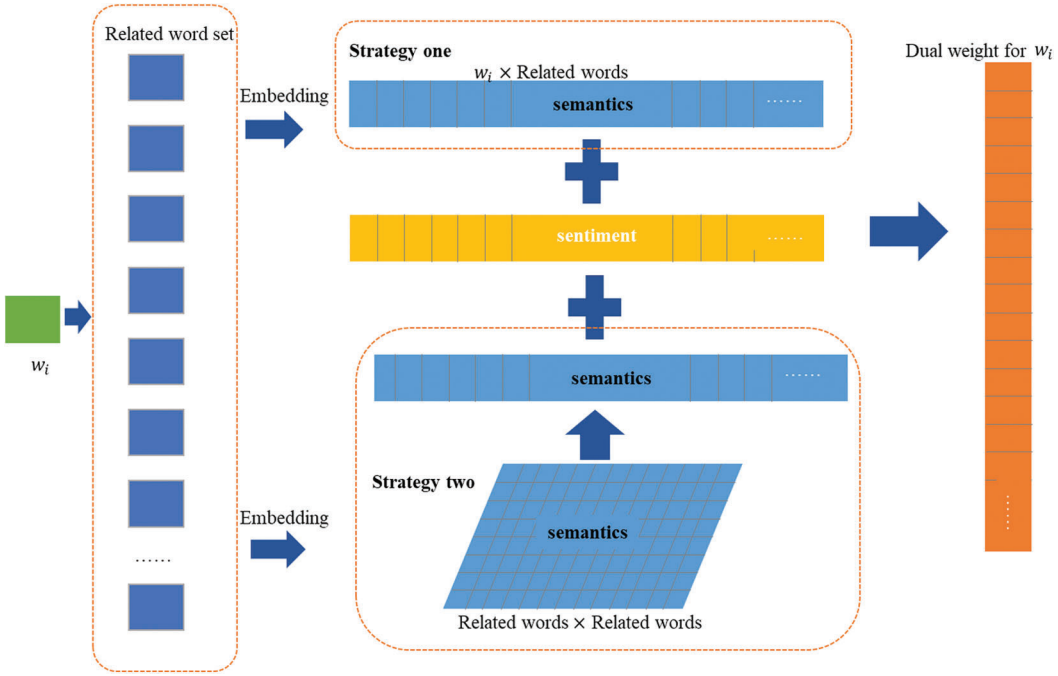

Figure 5: A dual weight mechanism

3.2.1 Strategy One

If w_i has a pre-trained vector $\boldsymbol{v_i}$, we use the cosine similarity between vectors of w_i and each related word r_i^n to determine importance of related words, defined as

$$W_{sem}\left(r_i^n\right) = \frac{\boldsymbol{v}(w_i)\boldsymbol{v}\left(r_i^n\right)}{||\boldsymbol{v}(w_i)|| \times ||\boldsymbol{v}(r_i^n)||} \tag{3}$$

where $W_{sem}\left(r_i^n\right)$ is the semantic weight of r_i^n, $\boldsymbol{v}\left(r_i^n\right)$represents the vector of n-th related word r_i^n.

3.2.2 Strategy Two

If w_i has no pre-trained vector $\boldsymbol{v_i}$, we build a semantic matrix S_i, and each element uses the correlation between vectors of the m-th r_i^m and n-th related word r_i^n to determine importance of related words.

$$S_i = \begin{bmatrix} s_i^{1,1} & \cdots & s_i^{N,1} \\ \cdots & \cdots & \cdots \\ s_i^{1,N} & \cdots & s_i^{N,N} \end{bmatrix} \tag{4}$$

$$s_i^{n,m} = \frac{\boldsymbol{v}(r_i^m)\boldsymbol{v}(r_i^n)}{||\boldsymbol{v}(r_i^m)|| \times ||\boldsymbol{v}(r_i^n)||} \tag{5}$$

Finally, the matrix S_i is summed by rows to get the semantic weight of each related word:

$$W_{sem}(r_i^n) = \sum_{n=1}^{N} s_i^{n,m} \tag{6}$$

For incorporating sentiment information, we query and get the sentiment value representation of each word in R_i via sentiment lexicons, and then build the corresponding sentiment set $s_{set}(R_i) = \{sen(r_i^1), \ldots, sen(r_i^n), \ldots, sen(r_i^N)\}$. The n-th word's sentimental polarity is defined as

$$sen(r_i^n) = \begin{cases} 1, v(r_i^n) = positive \\ -1, v(r_i^n) = negative \\ 0, v(r_i^n) = neutral \end{cases} \tag{7}$$

where N is the total number in R_i, $v(r_i^n)$ means the sentiment polarity of r_i^n, 1, –1 and 0 indicate positive, negative and neutral sentiments, respectively. By comparing the sentiment value $sen(w_i)$ and $sen(r_i^n)$, the sentiment weight of r_i^n can be obtained:

$$W_{sen}(r_i^n) = \begin{cases} 1, sen(w_i)sen(r_i^n) = 1 \\ \beta, sen(w_i)sen(r_i^n) = -1 \\ 0, sen(w_i)sen(r_i^n) = 0 \end{cases} \tag{8}$$

where β represents the sentimental weight when two words belong to different sentimental polarities. We will explore the optimal value in the experiments. Finally, we can get the dual importance weight of each word in R_i by integrating semantic and sentimental weights.

$$W_{ss}(r_i^n) = W_{sem}(r_i^n) \times W_{sen}(r_i^n) \tag{9}$$

Finally, the corresponding knowledge vector v_i^k is defined as:

$$v_i^k = \sum_{n=1}^{N} W_{ss}(r_i^n) v_o(r_i^n) \tag{10}$$

where $v_o(r_i^n)$ represents pre-trained word representation of the n-th related word r_i^n.

3.3 Word Encoder and Similarity Calculation

After obtaining the knowledge vector, we combine the word's contextual semantic vector to get an updated word representation $\hat{\boldsymbol{v}}(w_i)$.

$$\hat{\boldsymbol{v}}(w_i) = (1 - \alpha)v_i^k + \alpha v_i^o \tag{11}$$

where α represents the harmonic weight of the original semantic features with respect to w_i, which is used to adjust the proportion of knowledge vector v_i^k and context semantic vector v_i^o. Then the similarity between any two words w_i and w_j can be calculated by the cosine similarity.

$$s(w_i, w_j) = \frac{\hat{\boldsymbol{v}}(w_i)\hat{\boldsymbol{v}}(w_j)}{||\hat{\boldsymbol{v}}(w_i)|| \times ||\hat{\boldsymbol{v}}(w_j)||} \tag{12}$$

4 Word Similarity Experiments and Analysis

4.1 Experiment Setting

4.1.1 Dataset

Training corpus: The corpus obtaining the pre-trained word embedding comes from Sogou News [45], including 1.1 million news and an average of 223 words per news, about 300 million tokens in total. In the experiments, 1 million news are randomly selected to obtain pre-trained model. In addition, in order to increase the diversity of data and verify the broad applicability of our model, we also use some pre-trained word vectors[3] provided by Li et al. [46], which were trained on multiple corpora including Baidu Encyclopedia, Wikipedia, People's Daily News, Financial News and Literature based on Skip-gram model. As for the pre-trained embedding settings, window size is five, negative sampling is five, iteration is five, low frequency word is ten, dimension of vector is 300 and we only use the pre-embedding with word feature.

Training data: In order to train the parameters α and β in our method, we use the SimLex-999 translated dataset [47], which contains 999 word pairs and corresponding similarity score translated from English, to train and predict the similarity of each word pair. Then, we calculate correlation between predicted similarity sequence and the standard similarity sequence.

Evaluation data: The purpose of our work is to construct word representation model for calculating similarity of Chinese words. At present, there are four evaluation datasets commonly used in Chinese, namely WordSim-240 [20], WordSim-296 [48], MC30 [11] and RG35 [33], all of which are word pairs with similarity scores. The details of evaluation dataset are shown in Tab. 2.

Table 2: Statistic details of datasets

Dataset	Number of word pairs	Interval of similarity score
WordSim-240	240	0.15–9.2
WordSim-296	296	0.26–4.98
MC30	30	0.02–0.98
RG35	35	0.0125–0.97
SimLex-999	999	0.23–9.8

4.1.2 Metrics

In order to evaluate the effectiveness of our proposed method, we use the Spearman (ρ) and Pearson (r) rank correlation coefficient, which are both widely applied in word similarity task. As for each evaluation dataset $D = \{(w_1^1, w_1^2, X_1), \ldots, (w_n^1, w_n^2, X_n), \ldots, (w_{N1}^1, w_N^2, X_N)\}$, N represents the total number of word pairs, (w_n^1, w_n^2, X_n) is the n-th word pair, w_n^1 and w_n^2 indicate the two words in n-th word pair, X_n is the n-th gold-standard similarity score. Through our ESW-CS model, we can predict the similarity Y_n of the n-th word pair, and then get two sequences $X = \{X_1, \ldots, X_n, \ldots, X_N\}$ and $Y = \{Y_1, \ldots, Y_n, \ldots, Y_N\}$.

[3] https://github.com/Embedding/Chinese-Word-Vectors

The key to the evaluation of the similarity task is to find the correlation between the two sequences. The Pearson (r) is defined as:

$$r = \frac{\sum_n (X_n - \bar{X})(Y_n - \bar{Y})}{\sqrt{\sum_n (X_n - \bar{X})^2}\sqrt{\sum_n (Y_n - \bar{Y})^2}} \tag{13}$$

where $\bar{X}$ and $\bar{Y}$ are the average value of two sequences X and Y.

The Spearman correlation coefficient (ρ) is defined as

$$\rho = 1 - \frac{6\sum_{n=1}^{N}(R_{X_n} - R_{Y_n})^2}{N(N^2-1)} \tag{14}$$

where R_{X_n} and R_{Y_n} are the rank of X_n in X and the rank of Y_n in Y, respectively.

4.1.3 Parameter Settings

In our experiments, pre-trained word embedding is 100 dimensions. We select SG model as basic pre-trained method. The parameters are followed by [13,14], window is 5, min count of word is 20, negative is 3, sample is 0.001. The other contrast experiment models use the same parameters. In our Section 4.2 similarity comparison experiments, we set $\alpha = 0.2$, $\beta = 0.1$, and the specific inquiry experiments are set in Section 4.3. In order to avoid the occasional case of our experiments, each evaluation dataset is trained five times to obtain the average result.

In order to obtain the sentiment of each word, we integrate multiple Chinese sentiment lexicon resources, including HowNet[4], DUTIR[5], NTUSD[6] and sentiment lexicon from Tsinghua University [49].

4.2 Word Similarity Experiments

We evaluate our model based on concepts and synonyms knowledge on the task of word similarity. To present the effectiveness for word similarity, we compare and analyze the performance of our model to the following state-of-art models, which are widely used in Chinese word similarity:

Lexical-based method: The commonly used resources in Chinese are HowNet, a word concept resource and CiLin a synonym resource. HowNet provides the concept set of each word, and then calculates the similarity of the two words based on the path relationship between the concept word set of the word pair [39]. CiLin contains synonyms and related words for each word, and then calculates word similarity according to the path relationship between the synonyms and related words of the word pairs [39].

Word embedding method: We apply the wide word embedding models, including CBOW, SG and Glove to obtain the vectors of word pairs, and then utilize them on word similarity task. In addition, we also compared with some of improved embedding methods, such as CWE [20], SCWE [23], JWE [24]. JWE was proposed by Yu et al. [24] to learn the joint embedding of Chinese words, characters and fine-grained sub-character components. SCWE considered the Chinese word and internal structure character to learn the word embedding [23]. CWE method was proposed to obtain multiple-prototype character embedding for word similarity task [20].

Hybrid method of word embedding and lexicons: Niu et al. [30] proposed a sememe attention over target model (SAT) to incorporated word concepts from HowNet into word embedding representation learning for word similarity task. Sememes are used to describe the meaning of word, and each sememe has different importance to the meaning of the word.

[4]Hownet http://www.keenage.com/html/e_index.html

[5]Sentiment Ontology http://ir.dlut.edu.cn/EmotionOntologyDownload

[6]Lexicon from National Taiwan University https://down.itsvse.com/amp/16003.html

Our EWS-CS model: We propose an enhancing embedding-based Chinese Word similarity evaluation with concepts and synonyms knowledge, and take sentiment information of words into considerations. Concepts feature includes character-word concepts from HowNet and Xinhua online dictionary, and synonyms feature contains synonyms from CiLin.

The evaluation results of our EWS-CS model and baseline methods on word similarity datasets are shown in Tab. 3. From the results, our model outperforms other baseline models and we can observe that:

1. The performance of lexicon-based methods is very unstable, which performs well on MC30 and RG35 with a small amount of word pairs, especially using synonym information from CiLin, but does not perform well in WordSim-240 and WordSim-296 with more word pairs. MC30 and RG35 have a good performance since most word pairs of them can extract related concepts and synonyms through the knowledge resources. However, many words in the WordSim-240 and WordSim-296 evaluation data sets cannot be matched in the knowledge resources to lead poor results. This reflects the shortcomings of the lexicon-based method that the similarity of a word pair not in the lexicons or knowledge base resources cannot be obtained.
2. The evaluation indices of word embedding-based methods fluctuate little. The overall performance of small evaluation datasets (MC30 and RG35) are better than that of large data sets (WordSim-240 and WordSim-296). On the one hand, it shows that different word embedding methods can capture the semantics in the corpus to a certain extent, on the other hand, it also reflects that the word embedding method can no longer further improve the word similarity effect.
3. Our model outstrips other state-of-the-art baseline methods, including lexicon-based method, word embedding-based methods, hybrid methods. Compared with the lexicon-based method, the performance of our model is improved significantly when there are many word pairs, such as WordSim-240 and WordSim-296. And the Spearman correlation coefficient (ρ) improved by more than 50%. Compared with the word embedding-based methods, the improvement is obvious in the case where word pair is small (MC30 and RG35), and the Spearman correlation coefficient (ρ) is increased by more than 20%. In the whole, the EWS-CS model has achieved outstanding results in WordSim-240 and WordSim-296, indicating that our method with synonym and character-level concept knowledge can effectively represent words.

4.3 Applicability of Our Model in Different Corpora

In order to strengthen the diversity of data samples and further verify the applicability of our model to different text, we used pre-trained vectors based on Skip-gram model from different corpora including Baidu Encyclopedia, Wikipedia, People's Daily News, Financial News and Literature. It can be known from Section 4.2 that the performances of WordSim-240 and WordSim-296 are relatively stable, so we choose WordSim-240 and WordSim-296 two evaluation data sets for verification in this section. From Tab. 4, our model performs significantly better than the pre-trained Skip-gram model in different corpora.

Specifically, we can observe that:

1. Different corpus: On the whole, as the size of corpus increases, the task performance gets better and better for different corpora. Although the size of Baidu Encyclopedia is higher than Financial News, the effect is similar. The possible reason is that financial news is more professional and the quality of corpus is better than Baidu Encyclopedia.
2. Different evaluation data sets: Although the overall effect of our model is better than the pre-trained Skip-gram model, the results of the WordSim-296 data set are more improved compared with WordSim-240. Most of the word pairs contained in WordSim-240 are related words, but there are many similar words in WordSim-296 data set. From this perspective, due to the incorporating synonym information in our model, the word pairs in WordSim-296 can be better supplemented with semantic knowledge, so that the results are improved more.

Table 3: Experimental results using different models

Methods	WordSim-240		WordSim-296		MC30		RG35	
	ρ	r	ρ	r	ρ	r	ρ	r
Lexicon-based methods								
HowNet	0.0089	−0.0537	0.2573	0.1908	0.6831	0.7785	0.1394	−0.0101
CiLin	0.0427	0.1620	0.3013	0.4037	0.8240	**0.8564**	**0.7939**	**0.8606**
Word embedding-based methods								
Skip-gram	0.5545	0.5519	0.5945	0.5955	0.6156	0.6253	0.5933	0.6572
CBOW	0.51345	0.5155	0.51345	0.5155	0.5601	0.5715	0.5804	0.6380
CWE	0.5217	0.5304	0.5577	0.5616	0.6503	0.6290	0.6432	0.6497
SCWE	0.5292	0.5354	0.5513	0.5596	0.6258	0.6126	0.5851	0.5910
JWE	0.5109	0.5069	0.6450	0.6529	0.7058	0.6840	0.6904	0.7488
Hybrid methods								
SAT	0.5127	0.5224	0.5534	0.5532	0.6163	0.6092	0.4313	0.4634
Our model	**0.5601**	**0.5794**	**0.6464**	**0.6531**	**0.8428**	0.8254	0.6947	0.7699

Table 4: Experimental results using different corpora

Corpora	Size	Models	WordSim-240		WordSim-296	
			ρ	r	ρ	r
Literature	511 M	Skip-gram	0.5088	0.5418	0.5894	0.6094
		Our model	**0.5173**	**0.5864**	**0.6597**	**0.6810**
Wikipedia	960 M	Skip-gram	0.5343	0.5471	0.5703	0.5768
		Our model	**0.5371**	**0.5792**	**0.6328**	**0.6466**
People's Daily News	972 M	Skip-gram	0.5151	0.5319	0.5854	0.5851
		Our model	**0.5702**	**0.5951**	**0.6931**	**0.6823**
Financial News	1.24 G	Skip-gram	0.5710	0.5665	0.5255	0.5486
		Our model	**0.5824**	**0.5702**	**0.5952**	**0.6034**
Baidu Encyclopedia	1.69 G	Skip-gram	0.5607	0.5773	0.6004	0.5946
		Our model	**0.5729**	**0.5996**	**0.6736**	**0.6679**

4.4 Parameter Tuning and Determination

There are two parameters in the EWS-CS model, namely the knowledge and context semantic harmony coefficient α, and the sentiment similarity β of antonyms in sentiment assignment. In order to determine the importance of the two parameters in similarity, we use the SimLex-999 translated dataset for training based on Skip-gram as basic pre-trained method, and each parameter is trained by 5 times, and the result is averaged. Figs. 6 and 7 show the performance of the parameters under different values.

Fig. 6 shows the Spearman's rank correlation coefficient (a) and Pearson correlation coefficient (b) with different α values, respectively. Each boxplot indicates the performance of word similarity under different β

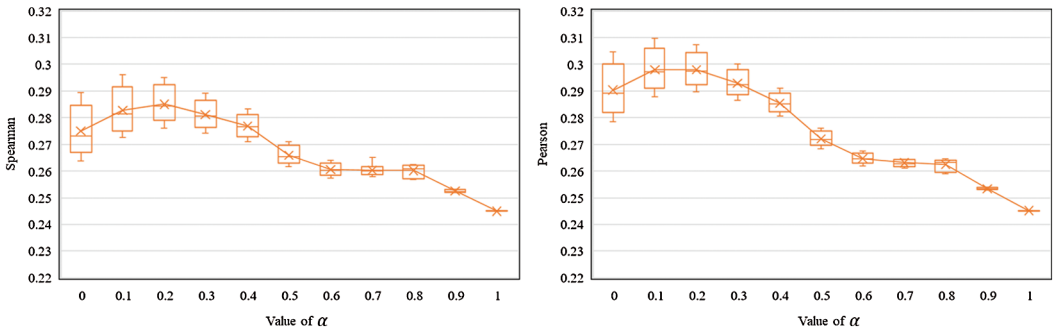

Figure 6: The performance of the SimLex-999 translated dataset under different α, (a) represents the result of Spearman coefficient and (b) represents the result of Pearson coefficient

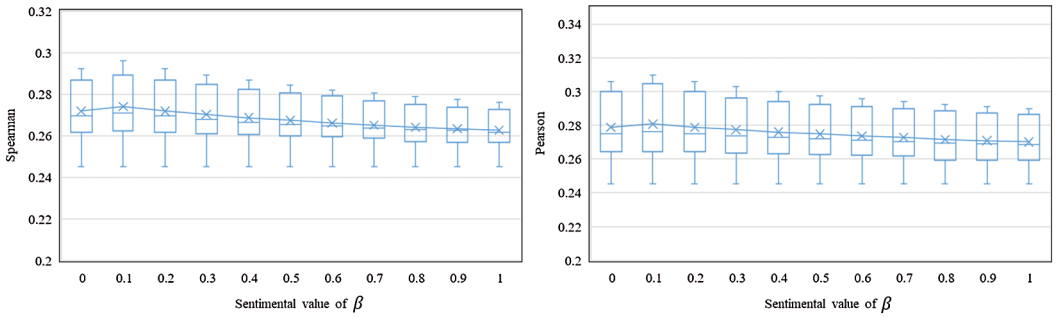

Figure 7: The performance of the SimLex-999 translated dataset under different β, (a) represents the result of Spearman coefficient and (b) represents the result of Pearson coefficient

when α is fixed. The area of the boxplot represents the fluctuation range of the final result with different β when α is fixed. It can be seen that with the increase of α, the area of the boxplot is getting smaller and smaller, which shows that the larger the value of α is, the smaller the fluctuation of β in the result becomes. As for the Spearman's rank correlation coefficient, the experimental results illustrate that the values increase first and then decrease with increasing α. This reflects our proposed model to incorporate character-word concepts and synonyms, which can effectively capture more semantic knowledge, thereby obtaining a better word semantic representation vector. As for the Pearson coefficient, it has a similar law to Spearman's rank correlation coefficient, but the overall fluctuation is slightly higher, which illustrates that the model of integrating concepts and synonyms knowledge into word vector can effectively improve the performance. Especial, the evaluation achieves best result, when $\alpha = 0.2$, so α will be set to 0.2.

Fig. 7 shows the Spearman's rank correlation coefficient (a) and Pearson correlation coefficient (b) with different β values, respectively. Each boxplot indicates the performance of word similarity under different α when β is fixed. The area of each boxplot is larger, and the area slightly decreases as β increases, indicating that when β is fixed, different α has a greater impact on the result, and as β increases, this effect decreases slightly. Fig. 7 also shows that sentimental similarity of antonyms has slight little effect on our method. When the antonym's sentimental similarity β equals 0.1, the overall result is relatively optimal.

4.5 Case Analysis

In order to better understand the quality of our proposed model, we conduct a case analysis in Tab. 5 to illustrate the similarity for 10 pairs of words under different methods. Our model outperforms other methods in three aspects:

1. For some similar words, such as lines 1, 2, 3, the performance of our method is very close to the standard result of artificial labeling, which is evidently better than other methods. The reason is that our model integrates synonym and sentiment information, which can better represent the synonymy of words.
2. For some related words, such as lines 4, 5, 6, because our model considers the character-word level concepts of words, it complements the relevance of words, making our proposed model outperform existing methods.
3. For some word pairs that are unrelated, such as lines 8, 9, 10, our method considers the dual superposition of semantics and sentiment knowledge to update the word representation, making our method significantly better than existing methods.

In summary, our method has good results for various types of word similarity calculations. However, for some word pairs, such as line 7 "日本" (Japan) and "南京大屠杀" (Nanjing massacre), due to the influence of history and other factors, the above similarity methods have not achieved good performance.

Table 5: The examples of similarities for the 10 word-pairs calculated by different methods

No.	Word pairs	Standard	Our model	SAT	SG	JWE	SCWE	CWE
1	类型, 种类 (Type, Category)	0.8481	0.8053	0.5351	0.4857	0.5945	0.5637	0.5694
2	消费者, 顾客 (Consumer, Customers)	0.8400	0.8379	0.7584	0.7240	0.6574	0.7199	0.5890
3	街道, 大街 (Street, Street)	0.7978	0.8568	0.6244	0.5472	0.5440	0.5392	0.6054
4	银行, 钱 (Bank, Money)	0.8584	0.5384	0.4220	0.3198	0.3178	0.3233	0.2383
5	演唱会, 歌手 (Concert, Singer)	0.8345	0.8317	0.6571	0.6830	0.5458	0.5745	0.6225
6	死亡, 囚犯 (Dead, Prisoner)	0.5119	0.4718	0.4208	0.2757	0.2035	0.2379	0.6574
7	日本, 南京大屠杀 (Japan, Nanjing massacre)	0.8950	0.5795	0.4809	0.4218	0.3152	0.2726	0.1249
8	和平, 气氛 (Peace, Atmosphere)	0.4728	0.3797	0.3185	0.2402	0.1187	0.0993	0.1599
9	问题, 挑战 (Problem, Challenges)	0.4480	0.4865	0.3607	0.3293	0.2766	0.2919	0.1440
10	发现, 太空 (Discovery, Space)	0.4111	0.4168	0.1683	0.1185	0.0157	0.0357	0.0495

5 Conclusion

Similarity calculation is a basic task in natural language processing, which is of great significance for information retrieval and information security detection. We propose an enhancing embedding-based word similarity evaluation method, which highly emphasizes on synonyms and character-word level concepts knowledge. Different from traditional methods to calculate similarity within a single feature, in this paper, we first construct a knowledge related word set to enrich semantic information for each word, and then obtain the knowledge representation utilizing semantic and sentimental information to enhance

the word embedding and distinguish the significance of different knowledge. In our work, we break the boundary between multi-features and consider synonyms, character-word level concepts and sentiment knowledge, which achieves excellent word representation. Experiments on similarity task have validated the effectiveness of our proposed model, which not only improves the performance of the word similarity under small samples, but also increases the stability of result.

Of course, there are still some issues worthy of further study in the similarity calculation based on small samples.

First, synonyms and related words are confused as conducting the word similarity task. Since similar and related word pairs are essentially different, similar words mean that they can be replaced in the same position by each other, and related words indicate that they have certain associations and appear in each other's context. Therefore, different word connotations lead to higher requirements for WS task, which makes the study consider the correlation and similar relationship of word pairs as a key issue to improve the accuracy of similarity calculation.

Second, the problem of low vocabulary coverage cannot be ignored under small sample corpus. It is not prominent when using large corpus because of the wide range of vocabulary. In this paper, we incorporate the concept of character in the case of small corpus. Although this problem is alleviated, it is still incompetent for some words without character concept. Therefore, studying the word similarity calculation of vocabulary to achieve more coverage is still a key issue.

Funding Statement: This work is supported by the National Natural Science Foundation of China (No. 61801440), the High-quality and Cutting-edge Disciplines Construction Project for Universities in Beijing (Internet Information, Communication University of China), State Key Laboratory of Media Convergence and Communication (Communication University of China), and the Fundamental Research Funds for the Central Universities.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Huang, D., Pei, J., Zhang, C., Huang, K., Ma, J. (2018). Incorporating prior knowledge into word embedding for Chinese word similarity measurement. *ACM Transactions on Asian and Low-Resource Language Information Processing, 17(3),* 57–81.
2. Smarandache, F., Colhon, M., Vladutescu, S., Negrea, X. (2019). Word-level neutrosophic sentiment similarity. *Applied Soft Computing, 80,* 167–176. DOI 10.1016/j.asoc.2019.03.034.
3. Yin, Y., Zeng, J. L., Wang, H. J., Wu, K. Q., Luo, B. et al. (2019). A lexical resource-constrained topic model for word relatedness. *IEEE Access, 7,* 55261–55268. DOI 10.1109/ACCESS.2019.2909104.
4. Madonsela, S. (2019). African Wordnet as a tool to identify semantic relatedness and semantic similarity. *South African Journal of African Languages, 39(2),* 185–190. DOI 10.1080/02572117.2019.1618020.
5. Mahdaouy, A. E., Ouatik, S. E. A., Gaussier, E. (2019). Word-embedding-based pseudo-relevance feedback for Arabic information retrieval. *Journal of Information Science, 45(4),* 429–442. DOI 10.1177/0165551518792210.
6. Akram, J., Shi, Z., Mumtaz, M., Luo, P. (2020). DroidSD: an efficient indexed based android applications similarity detection tool. *Journal of Information Science and Engineering, 36(1),* 13–29.
7. Taheri, R., Ghahramani, M., Javidan, R., Shojafarbc, M., Pooranianc, Z. et al. (2020). Similarity-based Android malware detection using Hamming distance of static binary features. *Future Generation Computer Systems, 105,* 230–247. DOI 10.1016/j.future.2019.11.034.
8. Glavas, G., Franco-Salvador, M., Ponzetto, S. P., Rosso, P. (2018). A resource-light method for cross-lingual semantic textual similarity. *Knowledge-Based Systems, 143,* 1–9. DOI 10.1016/j.knosys.2017.11.041.

9. Orkphol, K., Yang, W. (2019). Word sense disambiguation using cosine similarity collaborates with Word2vec and WordNet. *Future Internet, 11(5),* 114. DOI 10.3390/fi11050114.
10. Liu, N. F., Gardner, M., Belinkov, Y., Peters, M., Smith, N. A. (2019). Linguistic knowledge and transferability of contextual representations. *arXiv Preprint*, arXiv: 1903.08855.
11. Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM, 38(11),* 39–41. DOI 10.1145/219717.219748.
12. Mei, J., Zhu, Y., Gao, Y., Yin, H. (1983). *Tongyici cilin*. Shanghai: Shanghai Lexicon Publishing Company.
13. Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). Efficient estimation of word representations in vector space. *Proceedings of ICLR Workshop Papers,* Scottsdale, Arizona, arXiv:1301.3781v1.
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems,* pp. 3111–3119.
15. Pennington, J., Socher, R., Manning, C. (2014). Glove: global vectors for word representation. *Proceedings of the Empirical Methods in Natural Language Processing,* pp. 1532–1543, Doha, Qatar.
16. 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.
17. Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C. et al. (2018). Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics,* pp. 2227–2237, New Orleans, Louisiana.
18. Song, Y., Shi, S. M., Li, J., Zhang, H. S. (2018). Directional skip-gram: explicitly distinguishing left and right context for word embeddings. *Proceedings of NAACL-HLT,* pp. 175–180, New Orleans, Louisiana.
19. Zhang, Z. Y., Han, X., Liu, Z. Y., Jiang, X., Sun, M. S. et al. (2019). ERNIE: enhanced language representation with informative entities. *arXiv Preprint,* arXiv: 1905.07129.
20. Chen, X., Xu, L., Liu, Z., Sun, M., Luan, H. (2015). Joint learning of character and word embeddings. *IJCAI'15: Proceedings of the 24th International Conference on Artificial Intelligence,* pp. 1236–1242, Buenos Aires, Argentina.
21. Ma, B., Qi, Q., Liao, J. X., Sun, H. F., Wang, J. (2020). Learning Chinese word embeddings from character structural information. *Computer Speech & Language, 60,* 101031. DOI 10.1016/j.csl.2019.101031.
22. Sun, C., Qiu, X., Huang, X. (2019). VCWE: visual character-enhanced word embeddings. *Proceedings of NAACL,* pp. 2710–2719, Minneapolis, Minnesota.
23. Xu, J., Liu, J., Zhang, L., Li, Z., Chen, H. (2016). Improve Chinese word embeddings by exploiting internal structure. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* pp. 1041–1050, San Diego, California.
24. Yu, J., Jian, X., Xin, H., Song, Y. (2017). Joint embeddings of Chinese words, characters, and fine-grained subcharacter components. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing,* pp. 286–291, Copenhagen, Denmark.
25. Shi, X. L., Zhai, J. J., Yang, X. D., Xie, Z. H., Liu, C. (2015). Radical embedding: delving deeper to Chinese radicals. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2,* 594–598, Beijing, China.
26. Yin, R. C., Wang, Q., Li, P., Li, R., Wang, B. (2016). Multi-granularity Chinese word embedding. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,* pp. 981–986, Austin, Texas.
27. Lan, M., Zhang, Z., Lu, Y., Wu, J. (2016). Three convolutional neural networkbased models for learning sentiment word vectors towards sentiment analysis. *Proceedings of the International Joint Conference on Neural Networks,* pp. 3172–3179, Vancouver, BC.
28. Vo, A. D., Nguyen, Q. P., Ock, C. Y. (2020). Semantic and syntactic analysis in learning representation based on a sentiment analysis model. *Applied Intelligence, 50(3),* 663–680. DOI 10.1007/s10489-019-01540-2.
29. Yan, F., Fan, Q., Lu, M. (2018). Improving semantic similarity retrieval with word embeddings. *Concurrency and Computation: Practice and Experience, 30(23),* e4489. DOI 10.1002/cpe.4489.

30. Niu, Y., Xie, R., Liu, Z., Sun, M. (2017). Improved word representation learning with sememes. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 1,* pp. 2049–2058, Vancouver, Canada.
31. Ji, S., Yun, H., Yanardag, P., Matsushima, S., Vishwanathan, S. V. N. (2016). Wordrank: learning word embeddings via robust ranking. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,* pp. 658–668, Austin, Texas.
32. Sakketou, F., Ampazis, N. (2020). A constrained optimization algorithm for learning GloVe embeddings with semantic lexicons. *Knowledge-Based Systems, 195,* 105628. DOI 10.1016/j.knosys.2020.105628.
33. Rubenstein, H., Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM, 8 (10),* 627–633. DOI 10.1145/365628.365657.
34. Atoum, I., Otoom, A. (2016). Efficient hybrid semantic text similarity using wordnet and a corpus. *International Journal of Advanced Computer Science & Applications, 7(9),* 124–130.
35. Jimenez, S., Gonzalez, F. A., Gelbukh, A., Duenas, G. (2019). WordNet-Based word representation rivaling neural word embedding for lexical similarity and sentiment analysis. *IEEE Computational Intelligence Magazine, 14(2),* 41–53. DOI 10.1109/MCI.2019.2901085.
36. Chen, H., Li, F., Zhu, X., Ma, R. (2016). A path and depth-based approach to word semantic similarity calcalation in CiLin. *Journal of Chinese Information Processing, 30(5),* 80–88.
37. Peng, Q., Zhu, X. H., Hen, Y. S., Sun, L., Li, F. (2018). IC-based approach for calculating word semantic similarity in CiLin. *Application Research of Computers, 5(2),* 400–404.
38. Liu, Q., Li, S. (2002). Word similarity computing based on HowNet. *Computational Linguistics and Chinese Language Processing, 7(2),* 59–76.
39. Zhu, X., Ma, R. C., Sun, L., Chen, H. C. (2016). Word semantic similarity computation based on HowNet and CiLin. *Journal of Chinese Information Processing, 30(4),* 29–36.
40. Baccianella, S., Esuli, A., Sebastiani, F. (2010). SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10),* pp. 2200–2204, Valletta, Malta.
41. Tang, D., Wei, F., Qin, B., Yang, N., Liu, T. et al. (2016). Sentiment embeddings with applications to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering, 28(2),* 496–509. DOI 10.1109/TKDE.2015.2489653.
42. Yu, H., An, J., Yoon, J., Kim, H., Ko, Y. (2020). Simple methods to overcome the limitations of general word representations in natural language processing tasks. *Computer Speech & Language, 59,* 91–113. DOI 10.1016/j.csl.2019.04.009.
43. Mrksic, N., Seaghdha, D. O., Thomson, B., Gasic, M., Rojas-Barahona, L. M. et al. (2016). Counter-fitting word vectors to linguistic constraints. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* pp. 142–148, San Diego, California.
44. Guo, C. J., Pan, F., Zuo, Y. (2018). Word similarity algorithm based on multi-features. *Journal of Interdisciplinary Mathematics, 21(5),* 1067–1072. DOI 10.1080/09720502.2018.1493031.
45. Wang, C., Zhang, M., Ma, S., Ru, L. (2008). Automatic online news issue construction in web environment. *17th International World Wide Web Conference,* pp. 457–466, Beijing.
46. Li, S., Zhao, Z., Hu, R., Li, W., Liu, T. et al. (2020). Analogical reasoning on Chinese morphological and semantic relations. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics,* pp. 138–143, Melbourne, Australia.
47. Hill, F., Reichart, R., Korhonen, A. (2015). Simlex-999: evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics, 41(4),* 665–695. DOI 10.1162/COLI_a_00237.
48. Jin, P., Wu, Y. (2012). Semeval-2012 task 4: evaluating Chinese word similarity. *SemEval,* pp. 374–377.
49. Li, J., Sun, M. (2007). Experimental study on sentiment classification of chinese review using machine learning techniques. *Proceeding of IEEE NLPKE,* pp. 393–400.