Artificial Intelligence and Social Computing, Vol. 163, 2025, 123–128
https://doi.org/10.54941/ahfe1006043

Similarity Calculation of Concepts Based on Feature Distillation

Haitao Wang[1], Lianghong Lin[2], Xinyu Cao[1], and Jianfang Zong[1]

[1]China National Institute of Standardization, Beijing, Beijing, 100191, China
[2]South China Normal University, Guangzhou, Guangdong, 510631, China

ABSTRACT

Aiming at the problem that the accuracy of similarity calculation results is not high due to the lack of semantic information in the standard terminology database, this paper proposes a semantic similarity method based on feature distillation. The method firstly utilizes the FastText model to obtain the word vectors of the text, and then recalculates and weights to get new word vectors using the resources of the standard terminology database, and finally adds the BiLSTM model to further extract the contextual semantic information. The experimental results show that the method effectively integrates domain knowledge, enhances the recognition ability of text semantics, and significantly improves the accuracy of the similarity calculation results between texts in the standard terminology database.

Keywords: Semantic similarity, Terminology, Standardization

INTRODUCTION

Terminology plays an irreplaceable role in standards. Terms in standards should be harmonious and accurate. However, polysemy is constantly occurred. It is particularly important to analysis terms semantically. Accurate semantic similarity calculation helps to improve accuracy and reduce polysemy.

This paper proposes a method to calculate semantic similarity between terms in standards based on conceptual features. The method enhances the ability of model to parse terminology and domain-specific vocabulary by introducing specialized domain information, thus improving the accuracy and reliability of the overall analysis of sentences.

RELATED WORK

With the continuous advancement of Natural Language Processing (NLP) technologies, research on semantic textual similarity (STS) has significantly expanded, particularly in the domain of short text similarity calculation. Scholars worldwide have conducted a large number of studies and proposed numerous models and methods for the problem of text similarity calculation.

While traditional single-method approaches for text semantic similarity calculation have matured, their accuracy often falls short of meeting practical application requirements. As a result, hybrid approaches, which combine the

strengths of multiple methods, have become the main direction for research and development in this field. These hybrid methods integrate two or more techniques to enhance both the efficiency and accuracy of semantic similarity calculations, outperforming individual methods.

Viji and Revathy (2023) proposed a hybrid text similarity model combining Poisson Normal LDA with a Siamese Bi-LSTM and GRU network. The approach uses TF-IDF weighting and Poisson Normal LDA for feature extraction, followed by a deep learning framework to compute similarity scores. Xu et al. (2024) introduced SMSABLC, a text similarity model that integrates polysemy, character order, and contextual semantics. The model employs word embeddings to convert short texts into vectors and employs a multi-head self-attention mechanism to capture word-context relationships. Bidirectional dependencies are modeled using two opposing LSTM networks, while CNN layers extract local character features through convolution and pooling. Ali et al. (2023) proposed a hybrid model, SBiLA, which integrates SBERT, Bi-LSTM, and an attention mechanism. SBERT generates context-aware text embeddings, Bi-LSTM captures long-range dependencies, and the attention mechanism enhances the extraction of key information.

METHOD

The semantic similarity calculation model is shown in Figure 1. Considering that the word vectors generated by the FastText model may not fully capture the specificity and domain relevance of professional terms or domain-specific vocabulary, this study enhances the base word vectors produced by FastText. By incorporating text data that is more specialized and domain-relevant, weighted adjustments are applied to key terms, resulting in more accurate word vector representations. These refined vectors are then integrated with a BiLSTM model to produce the overall sentence vector representation. This process not only effectively improves the semantic expression of specialized terms, but also enhances the differentiation of domain-related words, so that the generated sentence vectors better capture and reflect the domain-specific deep semantic information.

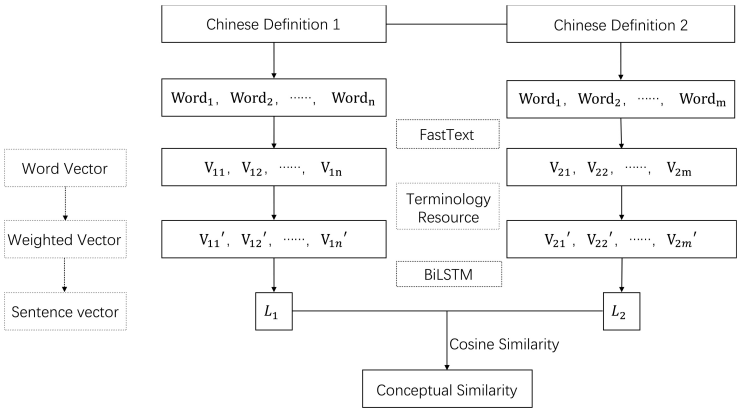

Figure 1: Semantic similarity calculation model.

Word Vector Representation

The FastText model is able to generate high-quality word vectors that effectively capture the semantic information of words and deal with the problem of unregistered words and neologisms through the modelling of subwords. Therefore, for each word w_i after word segmentation, its word vector x_i is computed using the pre-trained FastText model:

$$x_i = FastText\left(w_i\right) \tag{1}$$

Word Vector Weighting

In order to better express the domain characteristics, the thesaurus resource is used to label the important words in the definition text and to compute the vectors of these words. Standard terminology databases typically cover domain proper names and keywords that are often significant and useful in definitions. As shown in Figure 4, this paper identifies whether the words in the text are important terms by consulting a thesaurus, and only the words that exist in the thesaurus are labelled as important terms w_{imp}. For each tagged significant term, the FastText model is used to compute the vector $v\left(S\left(w_{imp}\right)\right)$ of the definition sentence $S\left(w_{imp}\right)$corresponding to that term in the the thesaurus. Specifically, the FastText model generates a vector representation of that defining sentence, which indirectly yields the vector x_{imp} of the vocabulary word:

$$x_{imp} = v\left(S\left(w_{imp}\right)\right) = \frac{1}{\left|S\left(w_{imp}\right)\right|} \sum_{w_j \in w_{imp}} v\left(w_j\right) \tag{2}$$

Finally, a new vector representation of the vocabulary word w_i is obtained by weighting:

$$v\left(w_i\right)' = v\left(w_i\right) + \partial * v\left(w_{imp}\right) \tag{3}$$

Where $\left|S\left(w_{imp}\right)\right|$ denotes the number of words in the defined sentence and ∂ represents the weighting weights.

It is worth noting that if the defining sentence $S\left(w_{imp}\right)$ contains other important words w_{imp_k}, the vectors of these words are also computed using the same method. This computation is recursive, i.e., by computing the vectors of the defining sentences containing important words, the vector representations of these words are obtained step by step. This weighting method can effectively incorporate domain features into the text to ensure that important words are highlighted and accurately represented.

In addition, the process of weighting the vocabulary vectors involves the assignment of weights, which are determined based on the correlation between the words. The correlation is closely related to the national standard number and classification code (China Standard Classification Number, National Standard Classification Number). The Figure 5 shows the specific allocation principles of the weight ∂.

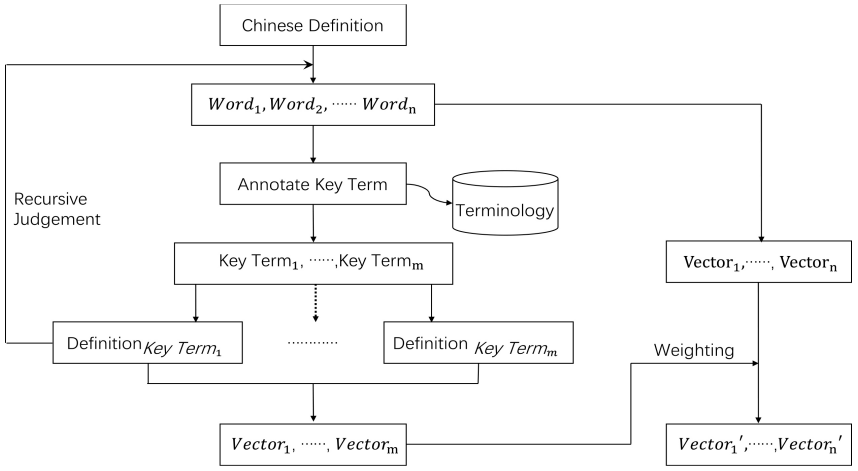

Figure 2: Vector weighting process.

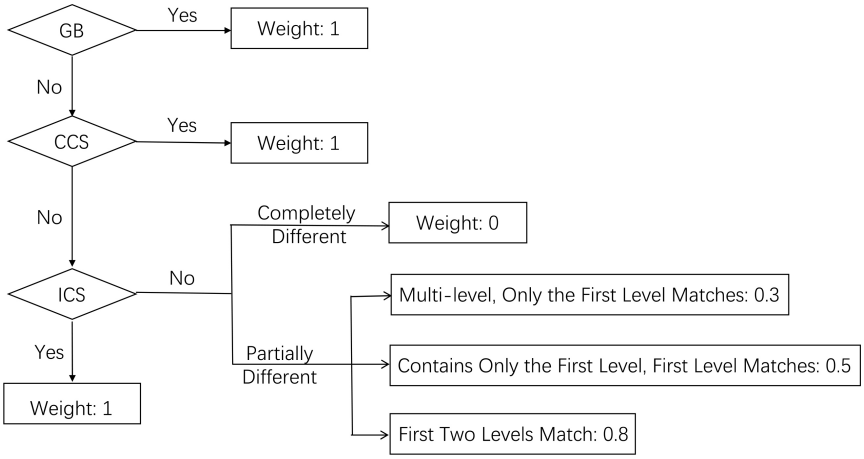

Figure 3: Principles of weight distribution.

Sentence Vector Representation

Since FastText is not enough to extract the contextual semantics, the model in this paper is supplemented by the BiLSTM model, which is a combination of a forward LSTM and a backward LSTM. The main feature is that it is able to extract the preceding information during the mapping process between the input and output sequences. The bi-directional structure will provide the output sequence with complete past and future context information for each time node, so that the output sequence can be extracted to the global semantic features of the sentence. The weighted word vectors $[x_1', x_2', \ldots, x_m', \ldots, x_n']$ are fed into the BiLSTM model to compute the vector representation of the whole sentence v_s.

$$v_s = \frac{1}{n}\sum_{t=1}^{n}\left[h_t^{(f)}, h_t^{(b)}\right] \tag{4}$$

The ${h_t}^{(f)}$ and ${h_t}^{(b)}$ represent the forward and backward hidden states of the model output at time t, respectively.

Similarity Calculation

Cosine Similarity measures the degree of similarity between two vectors, and reflects the similarity by calculating the cosine of the angle between the vectors. For two vectors v_{s_i} and v_{s_j}, their cosine similarity is defined as:

$$Similarity = \frac{v_{s_i} \cdot v_{s_j}}{\left\|v_{s_i}\right\| \cdot \left\|v_{s_j}\right\|} \tag{5}$$

Where $v_{s_i} \cdot v_{s_j}$ denotes the dot product of the vectors, and $\left\|v_{s_i}\right\|$ and $\left\|v_{s_j}\right\|$ denote the number of norms of the vectors v_{s_i} and v_{s_j} respectively. The value ranges between $[-1, 1]$, the closer the value is to 1 means the more similar the two vectors are, implying the higher similarity between the texts.

EXPERIMENTS AND RESULTS

Dataset

This experimental dataset is taken from the terminology database, which contains 7,050 data entries. Each terminology entry includes the following information: term number, Chinese term, English term, other preferred Chinese term, other preferred English term, licensed Chinese term, licensed English term, Chinese definition, source, formula, figure, table, note, reference, example, Chinese name of rejected term, English name of rejected term, and the corresponding terminology standard title information (including the national standard number, the International Standard Classification Number, the Chinese Standard Classification Number).

Comparison Experiments

In order to verify the effectiveness of the model in this paper, the model is compared with other models in this paper. The results of the comparison experiments are shown in Table 1, from which it can be seen that the accuracy of the method proposed in this paper is 5.27% higher than the BERT model method and 16.85% higher than the synonym forest method. It proves the effectiveness of the method.

Table 1: The result of comparison experiment.

	Method	Accuracy
1	Ontology-based: Synonym Forest	69.47%
2	Bert model	81.05%
3	Proposed model	86.32%

Ablation Experiments

In order to verify the effectiveness of each module in the model of this paper, we conduct ablation experiments, and the specific experimental results are shown in Table 2. It can be seen that after adding BiLSTM, the performance of the model is improved by about 2.11%, which is enough to prove that the method of further extracting the contextual semantic information of word vectors through BiLSTM is effective; from rows 2 and 4, it can be seen that after adopting the word vector weighting method, the performance of the model is also improved by about 3.16% and 2.11%, which proves the effectiveness of the word vector weighting method.

Table 2: The result of ablation experiments.

Method	FastText	BiLSTM	Weighted	Accuracy
1	√			82.11%
2	√		√	85.26%
3	√	√		84.21%
4	√	√	√	86.32%

CONCLUSION

The method proposed in this paper recalculates the words within the same domain and assigns different weights to them by combining the categorization information in the thesaurus during the vector generation process. This weighting process not only improves the accuracy of sentence vectors, but also highlights the importance of the vocabulary in the sentence, thus improving the model's ability to recognize and analyse term-defined sentences. The improved model can more accurately reflect the sentence vectors defined in a specific domain, thus improving the accuracy and effectiveness of semantic similarity analysis.

ACKNOWLEDGMENT

This research is supported by grants from National Language Commission (ZDI145-84) and China National Institute of Standardization (522024Y-11418).

REFERENCES

Ali Z., Aziz A., Ali A., et al. (2023). Fine-Tuned training method for semantic text similarity measurement using SBERT, Bi-LSTM and Attention Network. International Conference on Machine Vision, Image Processing and Imaging Technology (MVIPIT). IEEE, pp. 134–140.

Viji, D., Revathy, S. (2023). A hybrid approach of Poisson distribution LDA with deep Siamese Bi-LSTM and GRU model for semantic similarity prediction for text data. Multimedia Tools and Applications. Vol. 82, pp. 37221–37248.

Xu Y., Peng Y., Wang H., et al. (2024). A short text similarity calculation method based on deep learning, U.P.B. Sci. Bull., Series C, Vol. 86, No. 1, pp. 2286–3540.