Document Similarity Detection for Project Development Using Fused Interactive Attention Mechanisms

Chao Zhang[1], Ying Zhang[2], Gang Yang[3], Fan Hu[4]
State Grid Shanxi Electric Power Company, Taiyuan, 030021, China[1, 4]
State Grid Shanxi Electric Power Research Institute, Taiyuan, 030001, China[2, 3]

***Abstract*—This study introduces a novel multi-feature fusion model aimed at improving text similarity calculation in scientific and technological projects. The primary objective is to enhance the accuracy and efficiency of assessing text similarities, particularly in evaluating originality and identifying duplications in project submissions. To overcome the limitations of traditional text similarity methods (e.g., Vector Space Models, Latent Dirichlet Allocation, and TF-IDF) in capturing complex semantic and structural features, a hybrid model is proposed. The model combines word embeddings (word2vec and cw2vec), a Bi-LSTM network, and a multi-perspective convolutional neural network (MP-CNN) for effective feature extraction. Additionally, a fusion attention mechanism and interactive attention are incorporated to improve the extraction of semantic, contextual, and structural information. Experimental evaluation on two benchmark datasets demonstrates that the proposed model achieves an average precision of 0.75, a recall of 0.71, and an F1-score of 0.73, outperforming traditional methods (LDA, TF-IDF, Word2vec+Cosine) and deep learning baselines (Siamese-LSTM, MP-CNN) by more than 10% on average. These results confirm that the proposed architecture effectively balances semantic relevance and structural integrity, yielding superior similarity detection performance. The integration of advanced deep learning components—Bi-LSTM, MP-CNN, and attention mechanisms—substantially improves both the accuracy and efficiency of similarity evaluation, providing a more reliable and objective approach for scientific project assessment.**

Keywords*—*Text similarity; multi-feature fusion model; word2vec; cw2vec; MP-CNN; fusion attention mechanism; semantic extraction; project evaluation

I. Introduction

The duplication of scientific and technological projects is not only related to the smooth implementation of China's prospective research, but also has a profound impact on the orderly development of China's economy and culture. In order to promote the rational use of scientific research resources and funding, the scientific review of scientific and technological projects has become an important part of the scientific and technological program management system [1]. Therefore, in order to ensure the smooth development of original and innovative scientific research, project reviewers need to make more accurate judgments on the duplicity and similarity of the applied projects [2]. However, the duplicity review of scientific and technological projects is a complex process, with the year-on-year growth in the scale of scientific and technological project declaration, the project review methods used at this stage have been difficult to meet the current demand for originality review, and often need to be repeated by experts in various fields based on their own scientific research experience on the project and the related literature and patented technology screening, not only checking and checking for new accuracy fails to meet the requirements, but also seriously limits the efficiency of scientific and technological project review, resulting in the scientific and technological project review efficiency, which leads to a more accurate judgment of duplication and similarity. The review efficiency of science and technology projects has been seriously limited, which has hindered the management of science and technology programs.

Most of the traditional text similarity calculation methods apply literal repetition or probability models. For example, the traditional Vector Space Model1 (VSM) utilizes the theory of statistics to measure the similarity between texts based on the probability distribution of words. In [7], the authors utilize VSM to calculate the similarity between texts, adding keywords to avoid the removal of valid features. In recent years, a variety of methods have been proposed to enhance textual similarity detection. Traditional statistical approaches such as VSM, TF-IDF, and Latent Dirichlet Allocation (LDA) have laid the foundation for lexical and probabilistic modeling of text. However, these methods often fail to capture deep semantic and contextual information, leading to lower accuracy when dealing with complex or domain-specific documents. To overcome these limitations, deep learning-based approaches have been introduced. For instance, Siamese-LSTM models [3] learn sequence-level semantics through shared weight encoding, while CNN-based models such as MP-CNN [4] focus on extracting structural and local contextual features from paired sentences. Attention-based architectures like ABCNN [5] further refine feature alignment between text pairs, enabling improved sentence-level similarity detection. Despite these advances, existing models typically rely on single-level representations, either word-level or character-level, and rarely consider the integration of both semantic and structural perspectives. In contrast, our proposed model introduces a comprehensive multi-feature fusion strategy that simultaneously leverages word embeddings (word2vec and cw2vec), sequential encoding via Bi-LSTM, and multi-perspective convolution through MP-CNN, enhanced with fused and interactive attention mechanisms. This design enables the model to capture semantic relevance, structural integrity, and contextual dependencies in

an integrated framework, thereby addressing the key limitations of previous methods.

With the development of machine learning and deep learning technology, more and more researchers use deep learning technology to build models to study related tasks in the field of natural language processing. Vani Kl [6] et al. built a model for detecting plagiarism of academic ideas by taking plagiarism of ideas as a research object to address the increasing academic misconduct in the field of research and education. Velásquez [7] et al. proposed a plagiarism detection system called Document Copy Detector 3.0 (DOcument COpy Detector 3.0, DOCODE 3.0) to address the problem of academic plagiarism in educational institutions. Ehsan N et al. [8] addressed the problem of plagiarism being difficult to detect in cross-linguistic systems, and established a localized plagiarism detection model based on topic word retrieval and segment similarity assessment of hetero-linguistic sources as a research object. Arts S[9] et al. analyzed the limitations of the United States Patent Classification System (USPCS) in the detection of patent technology similarity, improved the text matching algorithm of the system, and proposed a text matching-based patent technology similarity detection algorithm. A text matching-based similarity analysis model for patented technologies. Sutoyo [10] et al. proposed a document plagiarism detection method based on a K-member grammar model with a screening algorithm and evaluated and selected the performance of the K-member grammar model's K-value with sliding window calculation. Scholar Choi S P M [11] et al. proposed an information retrieval-based text similarity detection algorithm that is capable of handling multilingual source documents and seamlessly integrates with existing learning management systems. The algorithm identifies potentially plagiarized phrases by employing information retrieval and sequence matching techniques, with parametric control to minimize false positives and negatives. Empirical evidence shows that the algorithm not only accurately and quickly identifies documents suspected of plagiarism, but also quantifies and visualizes the severity of plagiarism in data, thus providing scholars with a good aid in reviewing and assessing plagiarism.

Many researchers have achieved good results by not performing the pre-training task without text in order to take into account more comprehensive and underlying textual information. In [12], the authors does not carry out the pre-training of words, directly convolves the vectors represented by the unique hot code, fully mines and predicts the contextual information with the most original data, adds unsupervised region embedding, efficiently expresses the features of the text, and finally achieves better results on various tasks. Literature [13] utilizes convolutional neural networks to directly train and learn directly from the underlying characters of the text without text pre-processing, and applies the extracted features in various tasks of text processing, and achieves better results on large datasets. Literature [14] used fine-grained character-level text input to the convolutional neural network and then processed it using Long Short-Term Memory (LSTM), followed by applying the extracted features to a variety of languages to achieve better results, indicating that the model is able to obtain semantic information from the character-level input.

Secondly, to address the problem that entity relationships in the text of scientific and technological project declaration are difficult to be extracted effectively, an entity relationship extraction algorithm based on entity group co-occurrence rate is further proposed to realize high-quality entity relationship extraction in the text of scientific and technological project declaration; thirdly, to address the demand for similarity calculation of scientific and technological project declaration text and the structural characteristics of the text, a text matching model based on polytunnels is proposed to realize the comprehensive evaluation of semantic relevance of different components of the text. Finally, to address the problem of limited accuracy caused by only detecting the text as a whole or artificially setting the weights of each check item in the declaration text of scientific and technological projects, a semi-structured text similarity assessment method combining graph structural similarity and text matching degree is designed.

The remainder of this study is organized as follows: Section II presents the text preprocessing model and details the feature extraction process using word2vec and cw2vec embeddings. Section III introduces the proposed text similarity calculation model based on fused and interactive attention mechanisms. Section IV describes the experimental setup, datasets, and evaluation metrics, followed by a discussion of comparative results. Finally, Section V concludes the study and outlines directions for future work, including model optimization and potential generalization to other languages.

II. Text Preprocessing Model

Deep learning technology has achieved better results in various kinds of tasks, such as text similarity calculation, intelligent translation, sentiment classification, semantic analysis, etc., and has attracted many researchers at home and abroad. Deep learning utilizes multi-layer neural networks to extract deep features in the text, and in the field of natural language processing, it mainly uses convolutional neural networks and recurrent neural networks to extract text features, and due to the fact that the recurrent neural networks have memory units, the recurrent neural networks are more effective than the convolutional neural networks in various tasks in the field of natural language processing. As a whole, the text of a scientific and technological project declaration consists of structured document structure and semi-structured text data, and as a kind of text data with special text features, this text type has a big difference from free text data in terms of syntax, wording, and article organization, thus presenting obvious text structural features and semantic features.

Word vectors are the prerequisite for the calculation of semantic similarity of text, so it is necessary to pre-process and pre-train the text to obtain high-quality word vectors, the process is shown in Fig. 1. Preprocessing is to segment the text in the corpus and training set and remove the stop words in the text, and pre-training is to convert the word sequences into feature vectors that can be recognized by the computer.

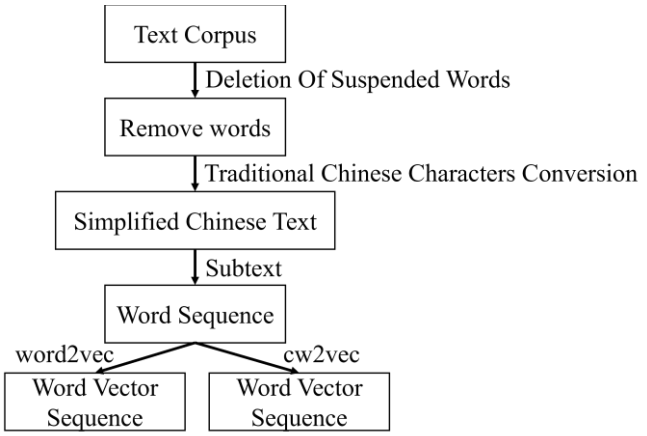

Fig. 1. Text preprocessing model.

A. Text Segmentation

Chinese word segmentation is the process of splitting a Chinese text into words according to semantic criteria and combining the results into a new sequence. Since English words are separated by spaces, each word can be used as a semantic unit, so English text segmentation is less difficult. On the other hand, Chinese text consists of consecutive Chinese characters connected together, and there is no separating mark between each word, so the segmentation of Chinese text is a basic and important step, and the accuracy of this process has a large impact on the subsequent related tasks and an accurate and fast segmentation algorithm is needed before the model training. On the issue of selecting a text segmentation model, the word labeling-based conditional random field model is the most used segmentation model, which is a model that uses word construction rules, and has a higher recall rate for unregistered words, but at the same time, it also generates more segmentation errors. Segmentation models using word annotations also require the addition of a complex denoising process in subsequent tasks. For the binary syntactic participle model, it only recalls the words present in the word list, and combined with the new word discovery algorithm, it can effectively alleviate the recall problem of high-frequency unregistered words. Assume that the sentence $\boldsymbol{T = w_1 w_2 w_3 \ldots w_n}$ has completed the disambiguation operation, where w is the n words composing sentence T, sentence T is changed to the original sentence $\boldsymbol{S = c_1 c_2 c_3 \ldots c_n}$, after passing through the noise channel without disambiguation, where $\boldsymbol{c}$ is the Chinese character in the sentence. Calculation using the Bayesian formula gives in Eq. (1):

$$\boldsymbol{P(T \mid S) = \frac{P(T)P(S|T)}{P(S)} \propto P(T) = P(w_1 w_2 w_3 \ldots w_n)}$$

$$= P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1 w_2)\ldots P(w_n \mid w_1 w_2 \ldots w_{n-1}) \quad (1)$$

Assume that the current word is only related to the previous work:

$$P(T) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_2)\ldots P(w_n \mid w_{n-1}) \quad (2)$$

This is the binary grammatical disambiguation model [Eq. (2)]. Add the interpolation smoothing calculation, as shown in Eq. (3):

$$P(w_n \mid w_{n-1}) = \alpha P(w_n \mid w_{n-1}) + (1-a)P(w_n) \quad (3)$$

where, $\boldsymbol{\alpha}$ is taken as 0.7. Jieba is one of the most active Chinese word splitter tools in China with a large number of users, which provides various functions such as word splitting, keyword, extraction, word labeling and so on. The participle function mainly has three modes: search engine mode, full mode, and exact mode, which can be applied in search engine, text, sentence and other scenarios that need participle. And the exact model is most suitable for use in the task of text analysis, therefore, the experiments of the text selected jieba participle tool to assist in participle.

B. Word-CW2VEC Model

Representing text using computer-recognizable word vectors can effectively alleviate the situation where the text is not trainable due to its variable length, large data dimension, and complex structure. Utilizing trainable word vectors can preserve similar features at the semantic level, mapping high-dimensional text data to low dimensions while avoiding dimensional catastrophe. Bengio et al. [15] used a neural network language model to train the data, learning the feature representations of the words as word vectors through the hidden layer, which learns the semantic information through the neural network.

Word2vec is a Google open-source tool for calculating word vectors. The two models CBOW and Skip-Gram, used in the tool for calculating the generated word vectors are proposed by Mikolov et al. [24]. The models simplify the Neural Network Language Model (NNLM) and design an accelerated training strategy to allow the model to be efficiently trained on massive training sets. The word vectors trained on a large amount of text data are able to represent the semantic relationships between words and tap into deep features. Because of their good performance and performance, the two models CBOW and Skip-Gram are widely used and have achieved good results on many natural language processing tasks. CBOW's core concept is to use words within a specified distance around a central word as context to model and predict the likelihood of the central word using a linear model. The architecture is shown in Fig. 2. The CBOW model is improved from NNLM by abandoning the strategy of a nonlinear hidden layer and vector splicing, which affects the training efficiency of NNLM, and mapping the word vectors to the same location.

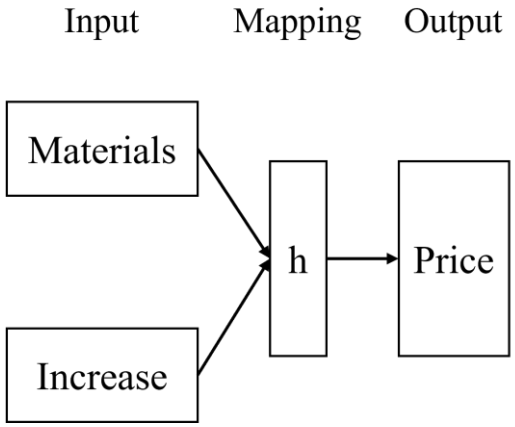

Fig. 2. CBOW model.

For a text matrix $W = (w_1, w_2, \ldots, w_n)$, CBOW utilizes the mapping layer e to sum the word vectors in the context c expressed as Eq. (4):

$$\boldsymbol{h} = \frac{\mathbf{1}}{\boldsymbol{n-1}} \sum_{\boldsymbol{w_i \in c}} \boldsymbol{e(w_i)} \tag{4}$$

The center word is then predicted, and the weights are constantly updated by maximizing the conditional probability with the contextual representation h, maximizing the conditional probability as:

$$\boldsymbol{L} = \sum_{\boldsymbol{(w,c)\in D}} \boldsymbol{log\, P(w \mid c)} \tag{5}$$

$$\boldsymbol{P(w \mid c)} = \frac{\boldsymbol{exp(e'(w)^T h)}}{\sum_{\boldsymbol{w' \in V}} \boldsymbol{exp\,(e'(w')^T h)}} \tag{6}$$

where, $\boldsymbol{e}$ is the mapping function, w is the center word, and $\boldsymbol{c}$ is the contextual information [Eq. (5) and Eq. (6)].

CW2VEC is a method proposed by Cao [16] and others to decompose Chinese text strokes in order to extract deeper and finer-grained information, and utilize Chinese strokes for model training and feature representation. The n-gram language model is utilized to mine the associations and semantic information on text morphology, and the sliding window is changed by adjusting the size of n to extract the semantic information of different granularity of strokes. The processing flow of the n-gram model based on strokes is shown in Fig. 3.

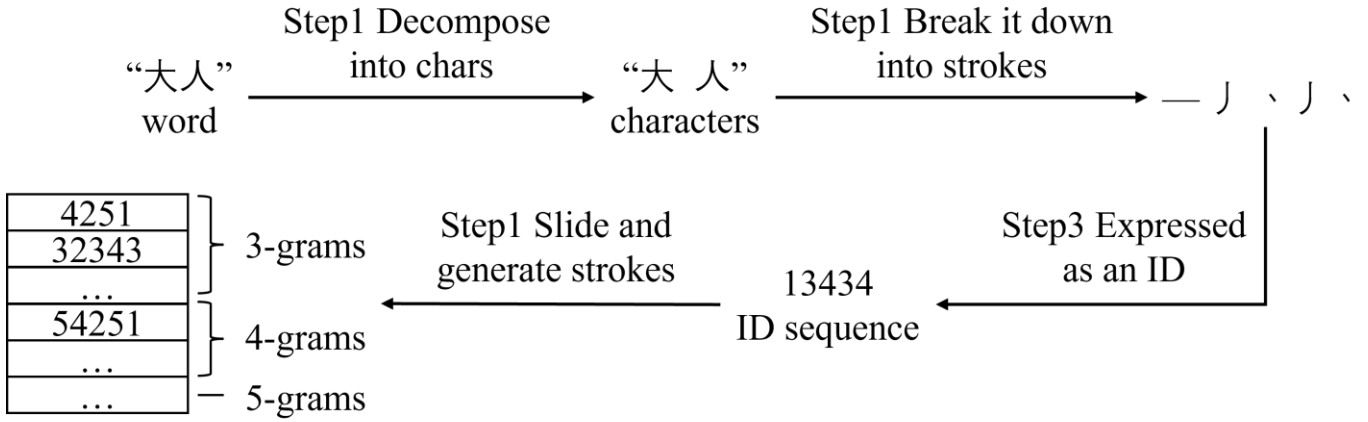

Fig. 3. Stroke-based n-gram modeling.

In Fig. 4, the model structure of cw2vec is shown; the basic idea is similar to Skip-Gram, both of them use the center word to predict the context information. The difference is that cw2vec uses the n-gram language model of strokes to represent the context information.

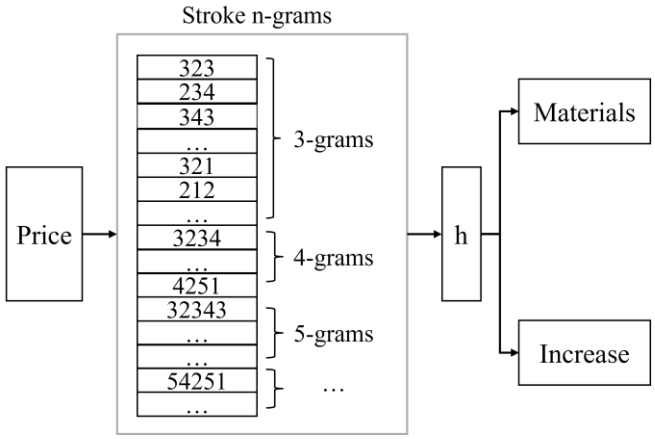

Fig. 4. CW2VEC model structure.

The context vector of the center word w is the sum of the feature vectors derived from the stroke n-gram model, computed as Eq. (7):

$$\boldsymbol{h} = \sum_{\boldsymbol{q \in S(w)}} \boldsymbol{e(q)} \tag{7}$$

where, S(w) is the set of stroke features of the center word computed using the n-gram model. The model is trained using the conditional probability of maximizing the center word with the words in the context, with the following Eq. (8) and Eq. (9):

$$\boldsymbol{P(w \mid w_j)} = \frac{\mathbf{exp}(\boldsymbol{e'(w)^T e(w_j)})}{\sum_{\boldsymbol{w/wV}} \mathbf{exp}\,(\boldsymbol{e'(w')^T e(w_j)})} \tag{8}$$

$$\boldsymbol{P(w \mid h_i)} = \frac{\mathbf{exp}(\boldsymbol{e'(w)^T h_i})}{\sum_{\boldsymbol{w' \odot V}} \mathbf{exp}\,(\boldsymbol{e'(w')^T h_i})} \tag{9}$$

For deep learning models, inputs are very important, and although it is possible to design structures with high performance, it is difficult for the model to work well if the input information is limited. In the experiments, it was found that word-level-based results are average, which is because the quality of the participle has a great impact on the model. To explore deeper semantic and structural information in the text, this study introduces a word embedding method combining word2vec and cw2vec, integrating three input sources from stroke sequences, word sequences, and word sequences, where the word2vec word embedding is inputted into the model while the cw2vec word embedding based on strokes is used in the other input channel, and then the two results are fused. This method effectively alleviates the problem of poor quality of word separation in the model. To summarize, the text structure feature extraction structure of this study is shown in Fig. 5.

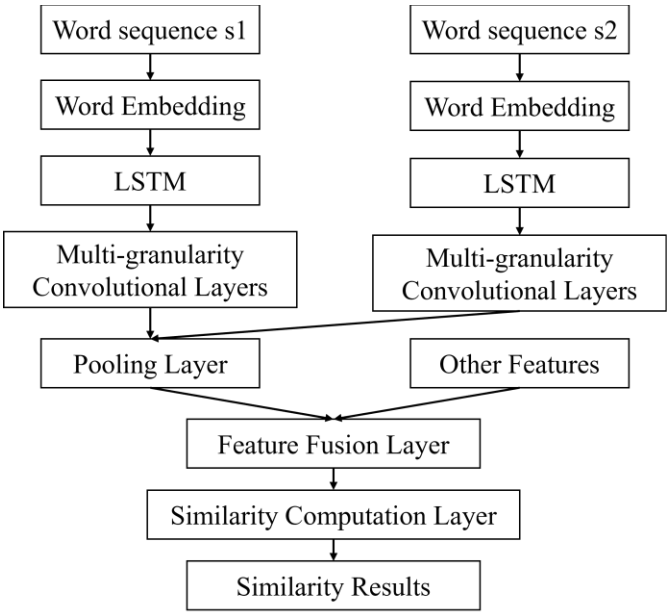

Fig. 5. Text structure feature extraction module.

III. Text Similarity Calculation Model Based on Fused Attention Mechanism

The similarity calculation of text is generally divided into three steps: first of all, the corpus should be preprocessed, including de-duplication, text segmentation, pre-training, and other steps. Then the text representation model is used to extract the feature representation containing semantics in the text, and finally, the similarity is obtained by training the model according to the extracted features.

A. Text Structure Feature Extraction Module Design

In this study, the proposed multi-feature fusion model for text similarity calculation of science and technology projects draws on the idea of Siamese structure, adopts word2vec and cw2vec word embeddings with different granularity as inputs, and jointly extracts the semantic information of the text with a Bi-LSTM network; effectively extracts the structural and word order information of the text through multi-granularity convolution and corresponding pooling; proposes the LSF feature The computation method of LSF features is proposed and proved to be effective. With the advantages of the two models, the extracted features are effectively fused, the semantic features are better preserved, and the deep-level features are mined. Finally, the text similarity calculation module for science and technology projects is designed, and the details of parameter settings are explained.

The whole model is divided into a word embedding module, a text structure feature extraction module (CNN model), a text semantic information extraction module (Bi-LSTM model), an attention mechanism module, a feature fusion module, and a similarity computation module. The structure of the similarity computation model with CNN fusion attention mechanism is shown in Fig. 6.

In this study, we draw on the MPCNN [17] (Multi-Perspective Convolutional Neural Networks) model to design the network structure, and use the convolution of multi-granularity to explore more deep features hidden in the present.

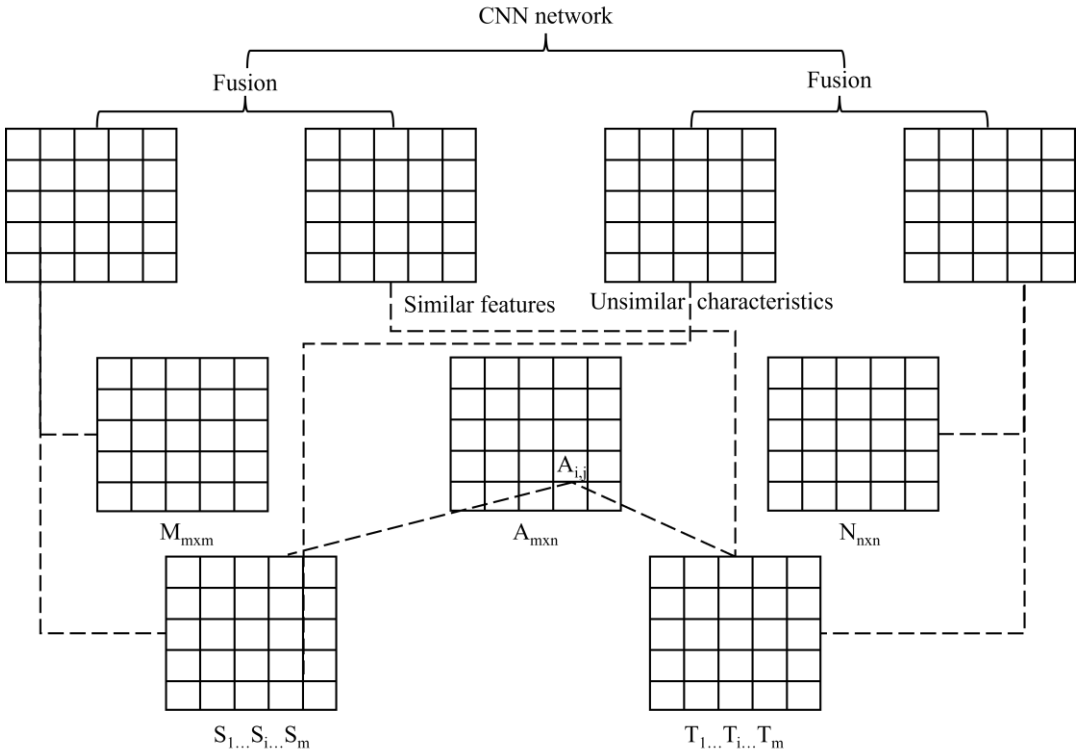

Fig. 6. CNN fusion attention mechanism model.

The structure of CNN-based model for calculating text similarity of science and technology projects is shown in Fig. 6. After the input text data is preprocessed with preprocessing operations such as word splitting and deactivation, the text is trained into word vectors and combined into a text matrix using the word2vec tool. Then the LSTM network is used to further train the word vector representation so that the word vectors contain more semantic information.

This study, inspired by Shen T, proposes an effective fusion of semantic features, contextual structural interaction features, and LSF features extracted from text through the text semantic, structural, and LSF feature extraction modules [18], and the fusion gate feature fusion method is designed, with the following computational equations [Eq. (10) to Eq. (12)]:

$$\boldsymbol{z_t} = \boldsymbol{sigmoid}(\boldsymbol{W_z} \cdot [\boldsymbol{h_{t-1}}; \boldsymbol{x'_t}] + \boldsymbol{b_z}) \quad (10)$$

$$\boldsymbol{r_t} = \boldsymbol{sigmoid}(\boldsymbol{W_r} \cdot [\boldsymbol{h_{t-1}}; \boldsymbol{x'_t}] + \boldsymbol{b_r}) \quad (11)$$

$$\boldsymbol{h_t} = (\boldsymbol{1} - \boldsymbol{z_t}) \times \boldsymbol{h_{t-1}} + \boldsymbol{z_t} \times \tilde{\boldsymbol{h}}_{\boldsymbol{t}} \quad (12)$$

The method fuses $\boldsymbol{h_{t-1}}$ and $\boldsymbol{x_t}$ into $\boldsymbol{h_t}$., where $\boldsymbol{z_t}$ is used to measure the degree of feature fusion, $\boldsymbol{x'_t}$ is the vector obtained by projection transformation of $\boldsymbol{x_t}$, W is the trainable parameter matrix, b is the trainable parameter vector, and the [;] symbols are used to splice the two vectors. Fusion gate method can accelerate the flow of information efficiently, and further integrates the computation results with a layer of multilayer. The fusion gate method can effectively accelerate the information flow, further integrate the computational results with a multilayer perceptual machine, and finally get the vectors with the same dimensions as the features before fusion.

B. Integration of Interactive Attention Mechanisms

In this study, drawing on the ideas of the ABCNN model and the LDC model [19], we introduce the interaction attention mechanism into the CNN model, and propose an improved scheme to construct the interaction attention matrix and take into account the similarities and dissimilarities through the orthogonal decomposition of the matrix. The model structure is shown in Fig. 7:

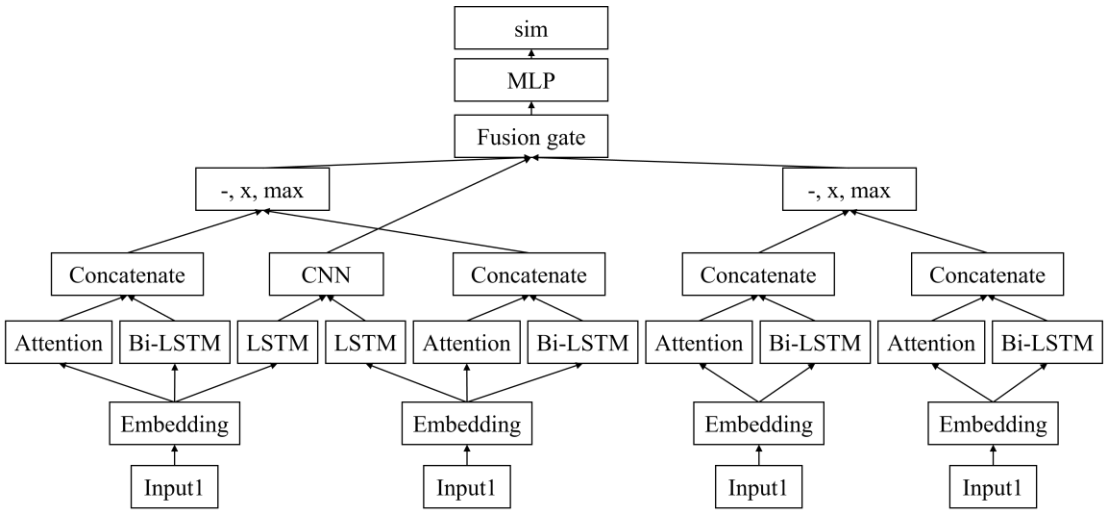

Fig. 7. Multi-feature fusion similarity calculation model.

For text pairs S and T, the text lengths are set as m and n, respectively, in order to obtain the interactive attention vector $\hat{s}_i$ (which implies similar and dissimilar components) of the word vector s_i in the text s, it is necessary to compute the similarity matrix $A_{m\times n}$, which is calculated as Eq. (13):

$$a_{i,j} = \frac{{s_i}^T t_j}{\|s_i\| \cdot \|t_j\|} \quad \forall s_i \in S, t_j \in T \quad (13)$$

where, $a_i, j \in A_{m\times n}$,is an element in $A_{m\times n}$, $s_i (i < m)$ is a word vector in text S, and $t_j (j < n)$ is a word vector in text T. The interactive attention vector of text s can be computed from the text representation matrix T and the similarity matrix $A_{m\times n}$ with the following Eq. (14):

$$\hat{s}_i = f_{match}(s_i, T) = \frac{\sum_{j=k-w}^{k+w} a_{i,j} t_j}{\sum_{j=k-w}^{k+w} a_{i,j}} \quad (14)$$

where, $k = \text{argmax} x_j a_{i,j}$. $\hat{s}_i = f_{match}(s_i, T)$, $\hat{s}_i$ is the ith vector in the interaction attention matrix of the text matrix S. w is a variable parameter, which is represented by w values near the maximum of the similarity matrix, i.e., the local interaction semantics.

Each vector is orthogonally decomposed into two parts of the geometric space parallel and perpendicular, where parallel is the similar part and perpendicular is the similar part, and where parallel is the similar part and perpendicular is the dissimilar part. The decomposition equation is Eq. (15) and Eq. (16):

$$s_i^+ = \frac{s_i \cdot \hat{s}_i}{\hat{s}_i \cdot \hat{s}_i} \hat{s}_i \quad parallel \quad (15)$$

$$s_i^- = s_i - s_i^+ \quad perpendicular \quad (16)$$

Using the orthogonal decomposition described above, the interactive attention matrix of the text is decomposed into similar and dissimilar matrices, denoted as $S^+ = [s_1^+, \ldots, s_i^+ \ldots, s_m^+]$ and $S^- = [s_1^-, \ldots, s_i^- \ldots, s_m^-]$. Both the dissimilar and similar parts are strongly related to each other, and it is difficult to determine the degree of similarity between similarly shaped but meaningfully different texts if only the similar parts are considered. When the similar and different parts

are considered at the same time, it can be historically good to determine the similarity gui between such texts. Therefore, the model in this study synthesizes the similar component matrix and the different component matrix into one feature vector $\vec{S}$ and $\vec{T}$, and the synthesis function is [Eq. (17) and Eq. (18)]:

$$\vec{S} = f_{comp}(S^+, S^-) \quad (17)$$

$$\vec{T} = f_{comp}(T^+, T^-) \quad (18)$$

where, f_{comp} is a combinatorial function. For the convolution operation, a list of convolution kernels w_o is defined, and each convolution kernel has the shape of $d \times h$, where d is the dimension of the word vector and h is the window size. In Eq. (17) and Eq. (18), each convolution kernel is applied to two channels from similar and dissimilar to generate a feature. The process is shown in the following Eq. (19):

$$c_{o,i} = f\left(w_o * S^+_{[i:i+h]} + w_o * S^-_{[i:i+h]} + b_o\right) \quad (19)$$

where, the $A * B$ operation adds all the elements in B with the corresponding weights in A, $S^+_{[ii+h]}$ and $S^-_{[ii+h]}$ denote the parts from S^+ and S^-, b_o is a bias term, and f is a nonlinear function. Finally, the similar and dissimilar features extracted by the CNN are spliced as extracted features and fused with other features to compute the similarity after the similarity computation layer.

IV. Results and Discussion

A. Experimental Data Collection

The experiments in this study are implemented using the open-source machine learning framework keras, which is rich in documentation, easy to use, and simple to model, and has attracted a large number of developers. In the experiment, an NVIDIA GTX 1050Ti is used as an auxiliary tool for model training, and through the Unified Computing Architecture technology introduced by NVIDA, the graphics card can be used to accelerate the training speed by utilizing APIs on GPUs (General-Purpose GPUs). The jieba segmentation tool is used to segment Chinese words, the Skip-Gram model of the word2vec tool is used to train word vectors, and cw2vec is used to train fine-grained word vectors.

Since there is no publicly available R&D dataset for science and technology projects, two datasets are chosen in this study to validate the experiments. Dataset 1 is the dataset of the Financial Intelligence NLP service tournament of Ant Financial Competition. The dataset is given 100,000 pairs of labeled data, all of which are two paragraphs of customer service and user Q&A, which contain synonymous pairs and non-synonymous pairs, and the algorithm is used to determine whether the same semantics are represented. Dataset 2 is the ChineseSTS similarity training set organized by Xi'an University of Science and Technology (XUST), in which 27,000 pairs of texts are classified into six similarity levels, in which the similarity of completely similar pairs is 5, and the similarity of dissimilar pairs is 0. However, the distribution of the pairs of texts in the training set is not balanced, and 90% of the pairs of texts have similarity of 0 or 5, which makes no sense for the training of the model. Therefore, 6000 text pairs are selected for model training and model performance evaluation. In order to harmonize with dataset 1, the similarity of text pairs in dataset 2 with similarity greater than 2 is set to 1, and the similarity of text pairs with similarity less than or equal to 2 is set to 0, which is used for the training and testing of the model.

B. Evaluation Indicators

In the study of text similarity calculation models for science and technology projects, it is very necessary to evaluate the model, and the evaluation index can measure the goodness of a model. In order to facilitate the evaluation of the performance of the model, this study evaluates the science and technology project text similarity calculation as a binary classification problem, which maps the model prediction results to the set {0,1}, with 0 representing that the two texts are not similar and 1 representing that the two science and technology project texts are similar. The commonly used evaluation metrics for binary classification problems are Precision, Accuracy, Recall, and F1-Measure. The following classes are defined according to the categories predicted by the model and the real categories of the samples:

1) TP (True Positive): model predicts a positive class and the sample is truly labeled as a positive class.

2) TN (True Negative): model predicts a negative category and the sample is truly labeled as a negative category.

3) FP (False Positive): model predicts a positive class, the real labeled as a negative sample.

4) FN (False Negative): model predicts a negative class, the real labeled as a positive class of samples.

where, the total number of samples is the sum of the four classes TP, TN, FP and FN. As shown in Table I, the confusion matrix can be used to describe the relationship between TP, TN, FP, and FN.

TABLE I. Confusion Matrix

	Similarity, positive	**Unsimilar, negative**
Physical resemblance	TP, Positive predicted to be positive	FP, Negative predicted to be positive
Actual dissimilarity	FN, Positive predicted to be negative	TN, Negative predicted to be negative

Accuracy and recall are widely used in the field of statistical classification and can be used to evaluate the accuracy and comprehensiveness of each prediction of a model. Among them, the accuracy rate is the ratio of the number of correct predictions made by the model to the total number of samples, and the equation is as follows [Eq. (20)]:

$$\text{Accuracy} = \frac{TP+TN}{TP+FN+FP+TN} \quad (20)$$

Accuracy evaluates the goodness of the model from the perspective of the model's prediction results, and is the proportion of the samples predicted to be positive classes that are truly labeled as positive classes, which measures the model's checking accuracy, with the following Eq. (21):

$$\text{Precision} = \frac{TP}{TP+FP} \quad (21)$$

Recall analyzes the goodness of the model from the point of view of the labeling results (see Eq. 21), describes the ratio of the number of samples predicted to be positive by the model to the number of samples labeled to be positive, and measures the model's checking rate, the equation is as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{22}$$

In general, recall and precision are negatively correlated (Eq. 22), and the F1 value, which is an indicator of the overall evaluation of the model, is the harmonic mean of recall and precision, with the following Eq. (23):

$$\text{F1} = \frac{2^{*}\text{Precision}^{*}\text{Recall}}{\text{Precision+Recall}} \tag{23}$$

C. Test Results

In order to verify that the model proposed in this study has good results, several models commonly used at present are implemented as comparison models, including traditional similarity calculation models and deep learning-based similarity calculation models.

LDA: The traditional LDA algorithm [20] is used to calculate the similarity of texts.

TF-IDF: The four text keywords in the two texts are extracted separately, and the cosine value is calculated as the similarity of the texts using the word embedding representation vectors of the keywords [21];

word2vec-cos: the representation vectors of words are trained using word2vec and combined with the cosine value to calculate the text similarity;

Siamese-LSTM: used to verify the validity of textual semantic features, the input is a text matrix of word vectors trained with the word2vec framework [22];

Siamese-CNN: based on the feature extraction model proposed by Kim, based on the Siamese framework and CNN network, with the input being a matrix representation of the sentence itself and convolved in full dimension;

MP-CNN: used to verify the validity of text structure features, the input is text matrix, and the text similarity is computed after extracting features with different granularity of convolution kernels and multiple pooling methods [23];

Siamese-LSTM-cw2vec: used to validate the effectiveness of cw2vec word embedding, the input of the model is the text matrix with stroke granularity trained with the cw2vec framework;

Our Model: the algorithm proposed in this study, the semantic, contextual structure of the text extracted from the model, the interaction between the text features using fusion gate fusion through the similarity calculation layer to calculate the text similarity.

In order to investigate the performance difference between the text similarity computation model designed and implemented in this study and the mainstream model, and to verify the advantage of multi-feature fusion over single feature, seven sets of comparison tests are set.

The average percentage error MSPE is shown in Fig. 8.

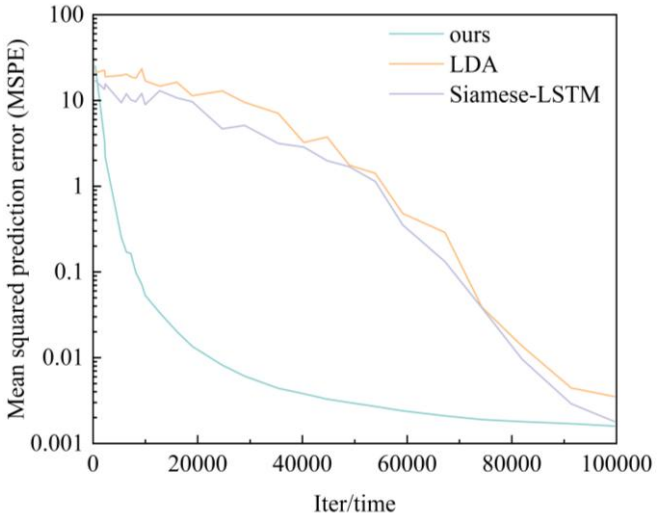

Fig. 8. Model percentage error plot.

TABLE II. COMPARISON OF RESULTS OF DIFFERENT MODELS IN DATASET 1

Model	Precision	Recall	F1
LDA	0.470	0.487	0.478
TF-IDF	0.433	0.468	0.450
Word2vec+cos	0.503	0.524	0.513
Siamese-LSTM	0.554	0.566	0.560
MP-CNN	0.563	0.582	0.572
Siamese-LSTM-cw2vec	0.536	0.572	0.553
Our Model	0.649	0.624	0.636

As can be seen from Table II and Table III, the F1 value of the model proposed for text is improved by about 0.2 compared to the traditional LDA model and the TF-IDF model, and also improved by about 0.1 compared to the Word2vec-cos model. In terms of the text feature representation, pre-trained word embeddings improve the model's ability to represent word features, and the word features trained by the Bi-LSTM network contain higher-order contextual semantic information than the n-gram model with higher order contextual semantic information. To further illustrate the model's computational efficiency, Fig. 9 presents a comparative analysis of time cost across different models. It can be observed that, despite the multi-component architecture, our approach maintains a competitive processing time, demonstrating that the performance gains do not come at the expense of excessive computational overhead. The similarity computation method based on the fusion attention mechanism proposed in this study improves the performance over the existing mainstream methods, which is mainly due to the model's effective fusion of the semantic information of the text and the structural information of the context. By effectively fusing multiple features extracted from the model using the fusion gate to retain as much information as possible from all the features, the performance is better than that of a single feature.

TABLE III. COMPARISON OF RESULTS OF DIFFERENT MODELS IN DATASET 2

Model	Precision	Recall	F1
LDA	0.551	0.568	0.559
TF-IDF	0.579	0.595	0.587
Word2vec-cos	0.594	0.612	0.603
Siamese-LSTM	0.652	0.672	0.662
MP-CNN	0.663	0.657	0.660
Siamese-LSTM-cw2vec	0.682	0.679	0.681
Our Model	0.749	0.714	0.731

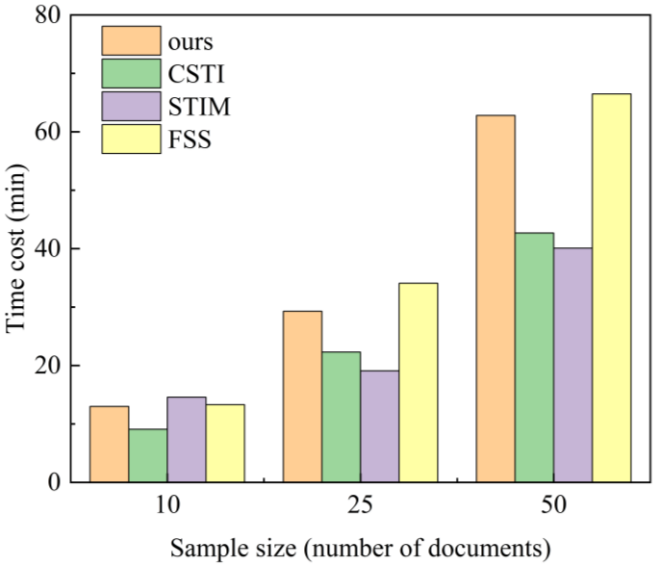

Fig. 9. Comparison chart of the time cost of our method.

Siamese-LSTM model applies the LSTM network with shared weights to the input coding of two texts under the Siamese framework, and the model is easy to train, which is a commonly used method in the field of text similarity computation with better results. However, some of the interaction features are missing, and the introduction of the product of feature vectors, variance, cosine value and Manhattan distance to amplify the dissimilarities of the texts can effectively alleviate the lack of interaction features. The input of the Siamese-LSTM-cw2vec model is the embedding of the stroke granularity of cw2vec, which has a better effect, proves the effectiveness of fine-grained word embedding, and explains that this study proposes the model incorporating cw2vec word embeddings. The MP-CNN model as a single-feature comparison model extracted some features. Word2vec-cos method is less effective because the simple representation of the text as a feature vector with word2vec and then calculate the similarity with the cosine value cannot fully take into account the complex semantic and syntactic information of the Chinese text. LDA is a topic model, which is essentially a bag-of-words model-based LDA is a topic model, which is essentially a model based on bag-of-words model to deal with long text, while the training set constructed in this study has shorter text, mostly belonging to the same type of topics, which can't take advantage of the performance in the experiment, and the performance is not good. The performance of TF-IDF method in the training set is not good because measuring the importance of a word simply by its "word frequency" is not comprehensive enough, and words that affect the semantics may appear more often because of their importance. The reason is that simply measuring the importance of a word by its "word frequency" is not comprehensive enough, and words that affect the semantics of a word because of their importance may appear less frequently, which does not reflect the positional information of the word and does not involve semantic features. It also relies heavily on the corpus, and needs to select a large number of high-quality corpora that match the training task.

V. CONCLUSION

The present study proposes a multi-feature fusion model for text similarity calculation in scientific and technological projects, effectively integrating semantic, contextual, and structural features to enhance accuracy. Compared with traditional similarity calculation methods, the proposed model achieves significant performance improvements, demonstrating its effectiveness in preserving both semantic meaning and contextual relationships. The combination of word2vec and cw2vec embeddings, together with Bi-LSTM and multi-perspective convolutional neural networks (MP-CNN), enables comprehensive feature extraction and fusion at multiple linguistic levels. Furthermore, the incorporation of fusion and interactive attention mechanisms enhances the model's capability to capture both shared and distinctive patterns across texts, thereby improving overall similarity detection performance.

Experimental results on two benchmark datasets indicate that the proposed model consistently outperforms traditional approaches such as LDA and TF-IDF, as well as advanced deep learning models including Siamese-LSTM and MP-CNN. An average F1-score improvement exceeding 10% over baseline methods highlights the model's ability to capture nuanced semantic dependencies and structural correlations.

Nevertheless, the multi-component architecture introduces additional computational complexity. To address this challenge, future work should focus on optimizing parameter efficiency through lightweight attention mechanisms, dimensionality reduction strategies, and GPU-based parallelization techniques, thereby improving the feasibility of real-time and large-scale deployment. In addition, it is valuable to investigate the model's generalization across multiple languages and domains. Given that cw2vec is primarily designed for Chinese text, subsequent research may incorporate multilingual embeddings (e.g., mBERT or cross-lingual transfer learning methods) to extend applicability to other linguistic contexts and scientific disciplines.

In summary, the proposed multi-feature fusion framework effectively balances semantic relevance, structural integrity, and contextual comprehension. With continued optimization and multilingual adaptation, the model has the potential to evolve into a scalable, domain-independent solution for intelligent and real-time text similarity assessment.

REFERENCES

[1] Chandrasekaran, Dhivya, and Vijay Mago. "Evolution of semantic similarity—a survey." ACM Computing Surveys (CSUR) 54.2 (2021): 1-37.

[2] Pearce, Joshua M. "Economic savings for scientific free and open source technology: A review." HardwareX 8 (2020).

[3] Shih, Chin-Hong, et al. "Investigating siamese lstm networks for text categorization." 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, 2017.

[4] Ding, Peng, et al. "A novel discrimination structure for assessing text semantic similarity." Journal of Internet Technology 23.4 (2022): 709-717.

[5] Yin, Wenpeng, et al. "Abcnn: Attention-based convolutional neural network for modeling sentence pairs." Transactions of the Association for computational linguistics 4 (2016): 259-272. Vani K, Gupta D, Detection of Idea Plagiarism Using Syntax-semantic ConceptExtractions with Genetic Algorithm[J]. Expert Systems with Applications, 2017, 73 :11-26.

[6] Vel¡squez JD, Covacevich Y, Molina F,et al. DOCODE 3.0 (DOcument COpyDEtector): A System for Plagiarism Detection by Applying an Information FusionProcess from Multiple Documental Data Sources[J. Information Fusion, 2016, 27:64-75.

[7] Ehsan N, Shakery A. Candidate Document Retrieval for Cross-lingual PlagiarismDetection Using Two-level Proximity Information[J]. Information Processing &Management,2016,52(6): 1004-1017.

[8] Arts S, Cassiman B, Gomez JC. Text Matching to Measure Patent Similarity[J]Strategic Management Journal,2018,39(1):62-84.

[9] Sutoyo R, Ramadhani I, Ardiatma A D, et al. Detecting Documents Plagiarism UsingWinnowing Algorithm and K-gram Method{C}//2017 EEE International Conferenceon Cybernetics and Computational Intelligence (CyberneticsCom). Phuket: IEEE,2017:67-72.

[10] Choi S P M, Lam S S. IChecker: An Efcient Plagiarism Detection Tool for LearningManagement SystemsJ. International Journal of Systems and Service-OrientedEngineering(IJSSOE),2016,6(3): 16-31.

[11] Johnson R, Zhang T. Semi-supervised convolutional neural networks for text categorization via region embedding[C].Proceedings of the 28th International Conference on Neural Information Processing Systems,2015:919-927

[12] Kim Y, Jernite Y, Sontag D, et al. Character-Aware Neural Language Models[C].The 30th AAAI Conference on Artificial Intelligence,2016:2741-2749

[13] Turian J, Ratinov L, Bengio Y. Word representations:A simple and general method for semi-supervised learning[C].Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics,2010:384-394.

[14] Bengio Y, Ducharme R, Vincent P,et al.A neural probabilistic language model[J]. Journal of machine learning research,2003,3(2):1137-1155.

[15] Cao S, Lu W, Zhou J, et al.cw2vec:Learning chinese word embeddings with stroke n-gram information[C].The 32th AAAI Conference on Artificial Intelligence, 2018:5053-5061.

[16] He H, Gimpel K, Lin J. Multi-perspective sentence similarity modeling with convolutional neural networks[C].Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015:1576-1586.

[17] Shen T, Zhou T, Long G, et al. Disan: Directional self-attention network for rnn/cnn-free language understanding[C].The 32th AAAI Conference on Artificial Intelligence.2018

[18] Wang Z, Mi H, Ittycheriah A. Sentence similarity learning by lexical decomposition and composition[C].International Conference on Computational Linguistics, 2016:1340-1349

[19] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J].Journal of machine Learning research,2003,3(1):993-1022

[20] Grootendorst, Maarten. "BERTopic: Neural topic modeling with a class-based TF-IDF procedure." arXiv preprint arXiv:2203.05794 (2022).

[21] Turian J, Ratinov L, Bengio Y. Word representations: A simple and general method for semi-supervised learning[C].Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics,2010:384-394

[22] Yin W, Schütze H, Xiang B, et al. Abcnn: Attention-based convolutional neural network for modeling sentence pairs[J].Transactions of the Association for Computational Linguistics,2016,4(1):259-272.

[23] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems 26 (2013).