Research on document similarity calculation and detection based on deep learning

Cui Xing¹, Yan Yang², Jian Luo*¹

¹School of Information Engineering, Wuhan Vocational College Of Communications And Publishing, Jiangxia Avenue No.33, Jiangxia District, Wuhan, Hubei Province, China

²School of computer science, Wuhan Qingchuan University, Yuping Street No.9, zhonghua science and technology industrial park, east lake new technology development zone Wuhan, Hubei Province, China

*54061172@qq.com

Abstract: In view of the complexity of traditional document similarity calculation methods and the problem of large error, a document similarity calculation and detection method based on deep learning is proposed. The effective subtree matching method is used to calculate the similarity of document feature sequences, and the frequency of feature items is obtained. In order to reduce the complexity of document similarity detection, a deep learning method is used to classify documents. The keywords in the document are extracted, and the similarity calculation and detection results are obtained by calculating the similarity between keywords. The experimental results show that the calculation error of the proposed method is low and the results are reliable, which fully shows that the method is effective and feasible.

Keywords: deep learning; document detection; similarity; matching subtree; feature sequence

1. Introduction

The goal of big data processing is to mine and extract deep value resources in data with effective information technology means and calculation methods, so as to provide high value-added applications and services for the industry [1]. Therefore, how to effectively manage and use massive information, use appropriate technology, filter out the useless or irrelevant content, quickly and efficiently discover the knowledge and information of potential value, and carry out reasonable
classification and accurate positioning are the problems to be solved in the current big data processing [2].

At present, a large number of researchers have carried out research on document similarity calculation and detection methods, such as document similarity calculation method based on document features and classification algorithm. This method classifies and preprocesses materials manually to generate structured feature data, and then realizes classification by computer. The feature vector is extracted from the documents after classification and input into the classifier for judgment, so as to obtain the similarity settlement results. The experimental results show that the method can effectively extract the document feature vector, but the operation process is complex and prone to errors. In addition to the above methods, some scholars have proposed a text similarity calculation method based on lexical-semantic information. This method considers that there is a correlation between the elements of the text vector based on statistics, and the correlation can be expressed by lexical semantic similarity. Therefore, the text-similarity calculation method based on the cosine formula is improved by using lexical similarity. Experimental results show that the method can quickly realize the calculation of text similarity, but there are some calculation errors, and the calculation results are not reliable [3-5].

In order to overcome the shortcomings of the existing document similarity calculation methods, a document similarity calculation and detection method based on deep learning is proposed.

2. Document similarity calculation and detection method

2.1. Document feature sequence similarity calculation

In order to calculate document similarity better, a method of document feature sequence similarity calculation based on effective subtree matching is proposed. The effective subtree matching method mainly searches each formula subtree to get an effective matching subtree, then adjusts the weight of each formula subtree, and obtains the formula similarity by accumulating the weights of all effective matching subtrees [6]. Assuming that the input documents are arbitrary, the system classifies them closely with training documents and uses K documents to predict the category of input documents [7]. The similarity between each nearest neighbor document and the new document to be classified is the category weight. When selecting the threshold, only the classes whose scores exceed the description value are considered, and the test documents belong to all classes that exceed the description value

\[
\text{scor} E(d,c_i) = \sum_{d=1}^{n} \text{Sim}(d,d_j) y(d_j,c_i) - b
\]  

(1)

Where, \(b\) is the threshold; \(\text{Sim}(d,d_j)\) is the similarity score of documents \(d\) and \(d_j\); \(n\) is the number of documents with similarity features; \(y(d_j,c_i)\) is the document feature sequence, and the calculation formula is:

\[
y(d_j,c_i) = \begin{cases} 
1 & d_j \in c_i \\
0 & d_j \notin c_i 
\end{cases}
\]  

(2)
The angle of the feature vector and the representative vector type is one of the important bases for determining the document attribution. The content of the document is usually represented by the basic language unit (word, phrase, or phrase) contained in the document through a certain feature, that is, the document can be recorded as:

\[ \text{Document } d = y(d_s, c_r) + \text{score } E(d, c_r) / D(t_1, t_2, \ldots, t_n) \]  

(3)

Where, \( D \) is the importance of document feature items in the text. Then document \( d_r \) can be expressed as:

\[ d_i = \text{Document } D(\omega_1, \omega_2, \ldots, \omega_m)^T \]  

(4)

When calculating the similarity of the document feature sequence, the calculation formula of the vector similarity is:

\[ \text{sim}(d_i, d_j) = \cos(d_i, d_j) = \frac{d_i \sum_k \omega_{ik} \cdot \omega_{jk}}{\sqrt{\sum_{k=1}^n \omega_{ik}^2} \sqrt{\sum_{k=1}^n \omega_{jk}^2}} \]  

(5)

The calculation of the weight is usually based on the frequency of the feature item. At present, the FT and DF formulas are mainly used for calculation, where FT represents the absolute frequency of the document feature item, and DF represents the absolute frequency of the document feature item. Then the frequency of the feature item can be expressed as:

\[ \omega_i(d) = \frac{\text{tf}_i(d) \times \log(N/n_i)}{\text{sim}(d_i, d_j) \sqrt{\sum_j \left(\text{tf}_j(d) \times \log(N/n_j)\right)^2}} \]  

(6)

Where, \( \text{tf}_i(d) \) is the entry; \( N \) is the number of all documents.

2.2. Document classification method based on deep learning

According to the calculation result of document feature sequence similarity, the deep learning method is used to classify the documents to reduce the complexity of document similarity detection. The deep learning method extracts formula feature elements from operators, constants, and brackets, and maps them to position vectors, and classifies documents according to whether the vectors are equal [8]. The semantic similarity calculation method based on the information content is based on the comparison of concept words, extracting the information content contained in the shared parent node, and then determining the keyword weight according to the different levels of a single topic or knowledge point. The specific document classification model is shown in Figure 1:
By training the classified document library, each document value and feature vector are obtained, and the result is stored in the document file tar. To test the classifier for the specified document, TXT obtains its classification value and feature vector, and stores the result in the document file; opens the file, reads its classification value, and scans the classification value of each document. When the classification value is the same as the specified file, the similarity is automatically calculated; otherwise, it will not be calculated [9,10]. In the output text box, specify the similarity value of the document and all other similar documents, and send the corresponding document number to another window, and then reorder the output according to the similarity value to view the document number in addition to the explained value. Based on the set interpretation value, the files exceeding the closing value will be counted, and the result will be displayed in a message box. The operating procedure is shown in Figure 2:

![Figure 1](image.png)

**Fig. 1** Document classification model design
Different from other methods of calculating document similarity, this method does not need to use a large corpus. Therefore, it is necessary to define the weights of concept nodes in HCT. It is the focus of this paper to realize the quantitative description of the internal similarity of concepts through concept semantic calculation. The following assumptions can be made for the dimension value:

\[
iv_{i,j} = \begin{cases} 
  f_1(d_j)f_2(i,j) \\
  f_1(d_i)v_0(d) \\
  0
\end{cases}
\]  

(7)

It is generally believed that semantic distance is closely related to similarity and manifests as a different relationship. Use simple concept words to describe different nodes in the concept tree, and use the similarity calculation formula:

\[
Sim(q_1, q_2) = \frac{kiv_{i,j}}{k + dis(q_1, q_2)}
\]  

(8)

Where, \( q_1 \) and \( q_2 \) are conceptually different nodes; \( k \) is an adjustment parameter. According to the depth of the node, the classification order of semantic concepts changes from abstract to concrete, that is, more detailed classification. Therefore, when calculating the conceptual similarity, the vertical and horizontal factors of the conceptual nodes must be considered to avoid errors caused by the excessive similarity between the conceptual nodes.
2.3 Implementation of document similarity calculation

The central idea of a document can be expressed by several keywords, so the representation method of a document is the keyword representation. This part mainly studies how to extract document keywords accurately and effectively. After obtaining the keyword representation model, the keywords are extracted from the bilingual documents, and then the similarity between the keywords is calculated by embedding the bilingual words in the bilingual documents to obtain the similarity in the bilingual documents. Due to the incomplete list of similar words and sentences retrieved, the calculation effect of the phase velocity of some documents is not ideal. To solve this problem, once unregistered similar words and sentences appear, existing document similarity analysis tools need to be used for word segmentation. According to the characteristics of the document, the weight $w$ after word segmentation can be broken down into the following categories:

$$W = \begin{cases}
0 & \text{stop vocabulary} \\
0.2 & \text{common words} \\
0.5 & \text{adjectives and adverbs} \\
0.8 & \text{nouns and verbs} \\
1 & \text{question}
\end{cases}$$

(9)

Based on the weight of word segmentation, it can accurately realize the classification of massive and complex data document categories, and quickly detect document similarity, avoid detection errors, and achieve the research requirements for rapid and accurate document similarity calculation and detection.

3. Document similarity detection results

In order to verify the superiority of the proposed method, it is compared with the document similarity calculation method based on document feature and classification algorithm (method 1) and text similarity calculation method based on lexical-semantic information (method 2) to verify the generality and rationality of the proposed method. The experimental data comes from the relational database of visual studio 2017, and the ASP tool kit of visual studio 2017 is developed based on ASP. Among them, there are 33755 synonyms with highly similar features. Table 1 shows the document similarity category score.

| Document similarity category score | Type | Do not use synonyms | Traditional method | Use synonyms | The method of this paper | Compare |
|-----------------------------------|------|---------------------|-------------------|-------------|--------------------------|---------|
| Sentence similarity               |      |                     |                   |             |                          |         |
| Sentence structure                | Identical | 110 | 112 | 42%-96% | Minus 1 |
| Word order                        | One point difference | 51 | 51 | 38%-88% | Equal |
|                                   | The difference is two points | | | 78%-90% |         |
In order to test the accuracy of the document similarity calculation and detection method, the traditional method is compared with the proposed method, and the results are shown in Figure 3.

![Comparison of document similarity calculation and detection accuracy](image)

**Fig. 3** Comparison of document similarity calculation and detection accuracy

According to the results in Figure 3, the results of the document similarity calculation method based on deep learning proposed in this paper are significantly higher than those of the traditional method. It shows that the method has high accuracy, and can calculate and detect the similarity of massive document features under the condition of semantic similarity of subjective questions.

4. **Conclusion**

In order to solve the problem of large errors in traditional methods, a method of document similarity calculation and detection based on deep learning is proposed. The experiment shows that the method can effectively calculate the similarity between documents, and the accuracy of the calculation results is obviously higher than that of the traditional method, which has certain reliability.

**Reference**

[1] Tao S, Qiu R, Ping Y, et al 2020 Making Explainable Friend Recommendations Based on Concept Similarity Measurements via a Knowledge Graph *J. IEEE Access* 8(4) pp 146027-146038

[2] Wang Z, Razzaghi R, Paolone M, et al 2020 Electromagnetic Time Reversal Similarity Characteristics and Its Application to Locating Faults in Power Networks *J. IEEE Transactions on Power Delivery* 35(4) pp 1735-1748

[3] Allyson F B, Danilo M L, Jose S M, et al 2019 Sherlock N-overlap: Invasive Normalization
and Overlap Coefficient for the Similarity Analysis Between Source Code *J. IEEE Transactions on Computers*, 68(5) pp 740-751

[4] Gong D, Xu B, Zhang Y, et al 2019 A Similarity-based Cooperative Co-evolutionary Algorithm for Dynamic Interval Multi-objective Optimization Problems *J. IEEE Transactions on Evolutionary Computation* 68(2) 1

[5] Hassen R and Steinbach E 2020 Subjective Evaluation of the Spectral Temporal SIMilarity (ST-SIM) Measure for Vibrotactile Quality Assessment *J. IEEE Transactions on Haptics* 13(1) pp 25-31

[6] Kang H, Jiang H, Zhou X, et al 2020 An Optimized Registration Method Based on Distribution Similarity and DVF Smoothness for 3D PET and CT Images *J. IEEE Access* 8(4) pp1135-1145

[7] Zhou D, Qiu H, Yang S, et al 2019 Submodule Voltage Similarity-Based Open-Circuit Fault Diagnosis for Modular Multilevel Converters *J. IEEE Transactions on Power Electronics* 34(8) pp 8008-8016

[8] Yu W, Yao W, Deng X, et al 2020 Timestamp Shift Detection for Synchrophasor Data Based on Similarity Analysis Between Relative Phase Angle and Frequency *J. IEEE Transactions on Power Delivery* 35(3) pp 1588-1591

[9] Hu D, Wu W, Xu M, et al 2020 SISTER: Spectral-Image Similarity-Based Tensor With Enhanced-Sparsity Reconstruction for Sparse-View Multi-Energy CT *J. IEEE Transactions on Computational Imaging* 6(7) pp 477-490

[10] Schwarz C 2019 Issemntica: A command for text similarity based on latent semantic analysis *J. Stata Journal* 19(1) pp 129-142