PROCEEDINGS OF SCIENCE

Research on Similarity Detection of Massive Text based on Semantic Fingerprint

Xiaolin Jin[1]
Communication University of China
Beijing, 100024, China
E-mail: 15538301358@163.com

Shuwu Zhang[2]
Institute of Automation, Chinese Academy of Sciences
Beijing, 100024, China
E-mail: shuwu.zhang@ia.ac.cn

Jie Liu [3a] **; Hu Guan** [4b]
Institute of Automation, Chinese Academy of Sciences
Beijing, 100024, China
E-mail: [a]jie.liu@ia.ac.cn; [b]hu.guan@ia.ac.cn

In order to find the required information quickly and efficiently in massive texts, this paper proposes a method of combining semantic fingerprint with cosine distance. After text preprocessing for Chinese texts, the Term Frequency-Inverse Document Frequency algorithm is used to extract feature words of the text, and then screen the text initially by the Simhash algorithm, finally compare these candidate texts tby using the cosine distance for the second similarity to extract the most similar texts. Based on a single Simhash algorithm, the proposed method can greatly improve the accuracy and recall under the modified textual environment, and can also meet the needs of massive texts' similarity testing requirements. Therefore, this method of combining semantic fingerprint with cosine distance can effectively make up for the problem of high false positive rate of Simhash algorithm and is more suitable for the similarity detection of massive texts in fact.

ISCC2017
16-17 December 2017
Guangzhou, China

[1]Speaker

This work is part of the research achievements of the Key Laboratory of Digital Rights Services, which is one of the National Science and Science and Standardization Key Labs for Press and Publication Industry.
[2]This work has been supported by the National Key Technology R&D Program of China under Grant No.: 2015BAH49F01.

[3]Correspongding Author
[4]Correspongding Author

PoS(ISCC 2017)009

1.Introduction

With the widespread use of computers and the popularization of the Internet, the era of information explosion has arrived. So, people are increasingly expecting quickly and accurately extracted information they needed in the massive text environment. To solve this problem, we need a text similarity detection technology to quickly and accurately find the most similar text from a huge amount of data.

The methods of computing text similarity both at home and abroad can be divided into four types as follows: similarity calculation based on distance[1], similarity calculation based on vector space, semantic similarity calculation and similarity calculation based on hash algorithm . The distance-based calculation method is suitable for a small amount of text environment. In a mass text, the method is very inefficient. The most commonly used method for calculating vector similarity is the Vector Space Model which was firstly proposed by Gerard Salton et al.in 1969[2]. However, in the traditional space vector model, there is no semantic relation between the words of vectors and each word needs to be vectorized, and the cosine calculation is performed to consume much time of processing of the Central Processing Unit. Based on the semantic similarity calculation method which is divided into the corpus based methods, the methods based on semantic dictionary and the method based on statistical language model[3-8], these methods are more complicated and are easily limited by the size of the corpus. The results of similarity calculation are more easily influenced by the noise of the training data. Therefore, the similarities between words trained by different corpora are different. Hash algorithm is also known as the local sensitive hash algorithm (LSH). The most commonly used hash algorithm is Simhash which was proposed in 2002 by Google's Charikar[9]. This algorithm transforms a document into an n-digit signature and calculate the similarity of the document by comparing the similarity of signature. Simhash processes text quickly and calculates fingerprints that can be easily stored in a database; therefore, it is very suitable for similarity calculation of large amounts of text.

2.Text Similarity Detection Algorithms

2.1TF-IDF

TF-IDF, as a common used weighting technique for information retrieval and information exploration[10], is a statistical method used to assess the importance of a document in a document set.The main idea of TF-IDF is that if a word or phrase appears in an article has a high frequency but low frequency in other articles, it is believed that the word or phrase has an important place in the article to better represent the subject of the article. TF-IDF is actually: TF * IDF.TF represents the term frequency, that is, the number of times in respect of a given word appearing in the file.IDF means the inverse document frequency. If a word in a document appears more but less in other documents, it means that the word better represents the document.

So TF-IDF formula is shown as follows: $\sum_k i, n_{k,j} * log \frac{(D)}{(\{j : t_i \in d_j\})}$;

$$tfidf_{i,j} = tf_{i,j} * idf_{i,j} = \frac{n_{i,j}}{i}$$

(2.1)

2.2 Simhash Algorithm

The Simhash algorithm is an algorithm proposed by Charikar in 2002, which is currently regarded as the best and most effective web content deduplication algorithm and repetitive data deduplication algorithm[11-14]. The Simhash algorithm is essentially a hash algorithm with locally sensitive features. The algorithm maps feature the vector of a text to a binary vector of bits of a given dimension and uses this binary hash value to represent the textual content. For an article, as shown in the Figure 1, the keyword is extracted after text preprocessing, then the keywords are changed into Hash by Hash algorithm. If the hash value is 1, the feature vector is 1; when the Hash value is 0, the vector is -1. According to the weight of words in the feature vector based on the weight vector is multiplied by the value of the word. The vectors of all the words in a document are accumulated, which also is the full-text principal vector. Then, the component greater than or equal to zero is mapped to 1, and the mapping smaller than 0 is 0 to achieve dimensionality reduction. The Simhash value is also the fingerprint of the text. Finally the text can be obtained directly by comparing fingerprints between different texts.

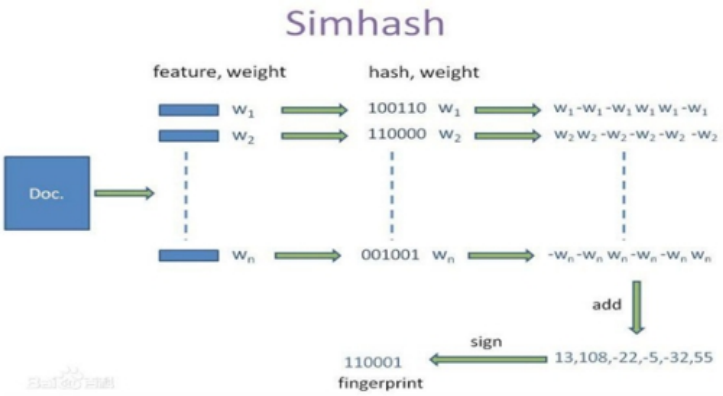

Figure 1: Simhash Algorithm Specific Flow Chart

3.Semantic Fingerprint based Mass Text Similarity Detection Technology

The proposed text similarity detection algorithm is mainly based on the combination of Simhash algorithm and cosine distance. The Simhash algorithm mainly reduces the text storage space by mapping high-dimensional text feature vectors into a unique binary text fingerprint value, and then calculates the degree of text similarity by comparing the Hamming distance of semantic fingerprints between texts. However, many similar texts have deletions, modifications and additions, the similarity detection by Simhash algorithm often leads to misjudgment or omission. The algorithm can only get a rough similarity screening and the high similarity does not mean that the text must be similar. In order to solve this problem, as shown in Figure 2, this paper firstly screened the relevant texts by Simhash algorithm, and then compared the similarity by cosine similarity algorithm to find the most similar text more quickly.

In this paper, the TF-IDF algorithm is used to extract the text feature words, in addition to using another indicator to improve accuracy in the feature weights-word part of speech. From the part of speech, nouns characterize more features of a document; so their weight should be the highest, the verb second, the adjective second, and the rest of the words have the lowest weight [15].

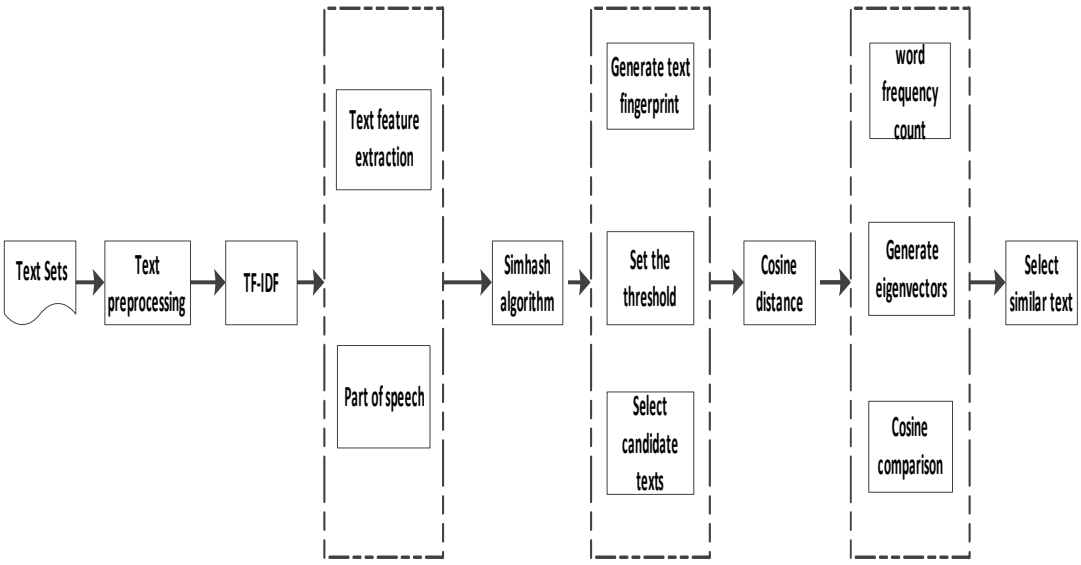

Figure 2: Mass Text Similarity Detection Technology Framework

4.Experimental Verification

This chapter verifies the validity of the method based on the combination of semantic fingerprint and the cosine distance for detecting the mass text fast similarity by experiments.

4.1Experimental Processing

The main process of massive text similarity detection is shown in Figure 3. The specific steps are as follows:

The first step: to use massive text segmentation and stop words to carry on the pretreatment, the use of TF-IDF to extract the feature of text;

The second step: to use Simhash algorithm to generate a binary text digital fingerprint, the establishment of the text fingerprint database and text feature thesaurus;

The third step: to detect text of the two steps of the operation, get the binary fingerprint and feature words;

The fourth step: firstly, to detect the text and text fingerprint in database of Hamming distance comparison, if less than or equal to the specified threshold of cosine distance, otherwise output similarity;

The fifth step: to calculate the features of word frequency in the threshold, generate the feature vector and compare the cosine distance, finally output of similarity;

The sixth step: to output the maximum similarity of the text, then the maximum similarity text as the detected text of the most similar;

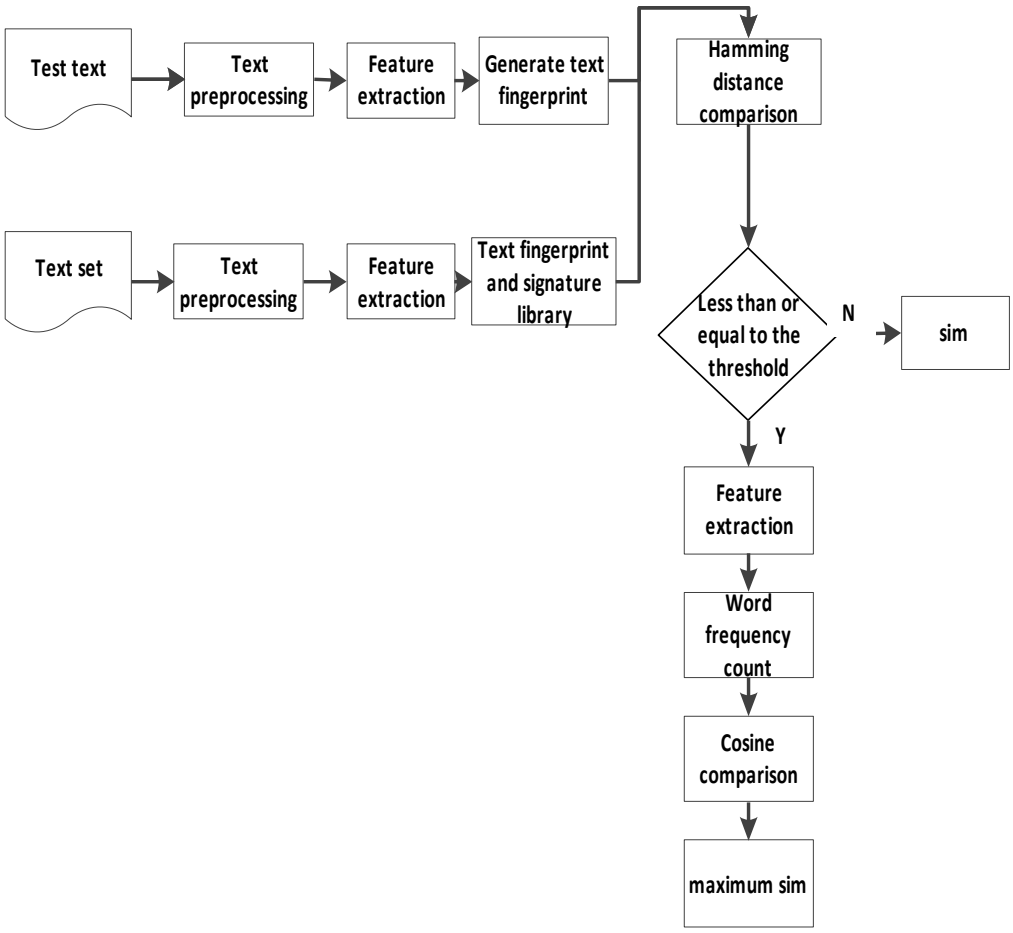

Figure 3: Mass Text Rapid Similarity Detection Steps

4.2Experiments and Results Analysis

The experiment selected part of the THUCNews as experimental data, a total of 219,173 news data, the data mainly include the financial, lottery, real estate, stock. The experimental operating environment is in the CPU for Intel (R) Core i7-4790 @ 3.60GHz, memory 16GB. Python language achieves this algorithm. Python comes with jieba module for word segmentation which can do text keyword extraction, and using Spyder compiler that comes with Anaconda3 runs the program.

The experiment is divided into three parts. In the first experiment, the experimental data were randomly selected from more than 200,000 news texts to find similar texts in 200, 500 and 1,000 texts respectively, and the experiment set the Hamming distance threshold to 16. In other words, texts within 16 are compared for the second degree of similarity. The experiment compared the recall and the accuracy of the Simhash algorithm with the algorithm based on the Simhash algorithm and the cosine distance.

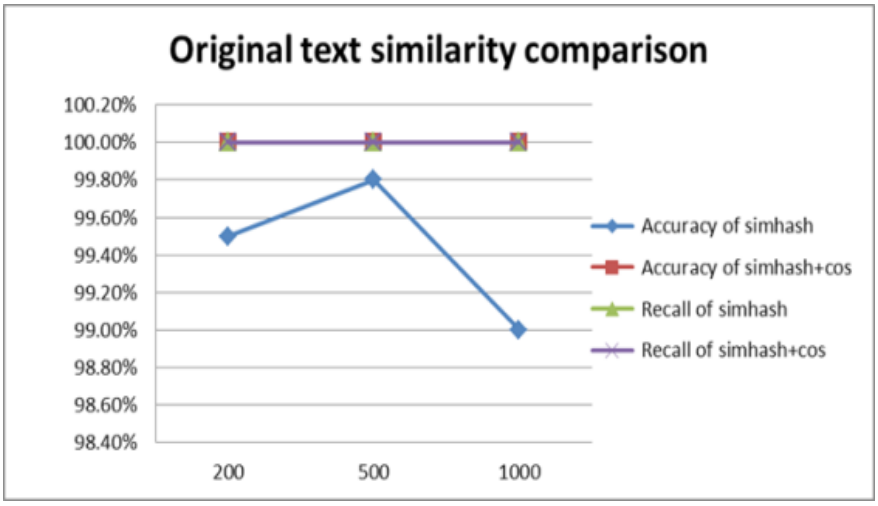

Figure 4: Original Text Recall and Accuracy Comparison

In the second experiment, the experimental data randomly selected 500 news items from more than 200,000 news texts and modified about 20% of these 500 news texts, mainly including deletion, add and position exchange. For the initial deletion, we need to determine a threshold to determine whether the cosine similarity is compared. Therefore, the experiment mainly focuses on the threshold setting of the modified text.

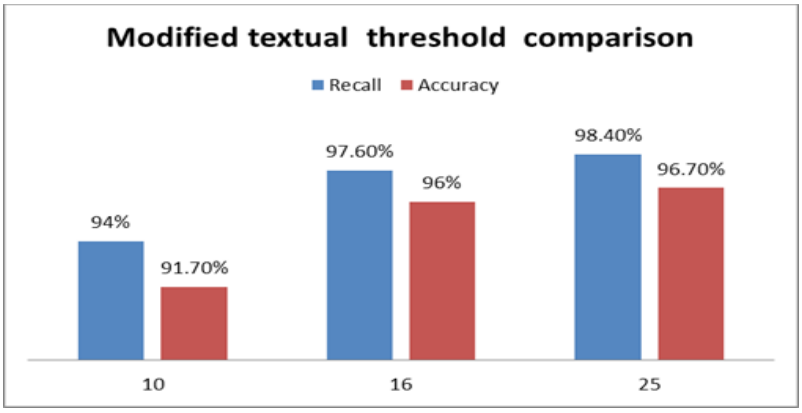

Figure 5: Hamming Distance Threshold Setting

In the finally experiment, the experimental data is the same as that in Experiment 2. In the case of a threshold of 16, the experiment mainly compared the recall and accuracy of the two algorithms under the revised texts.

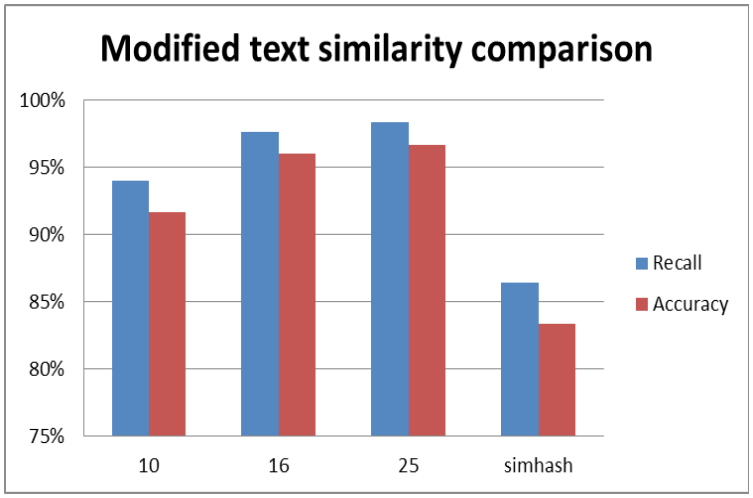

Figure 6: Modified Text Recall and Accuracy Comparison

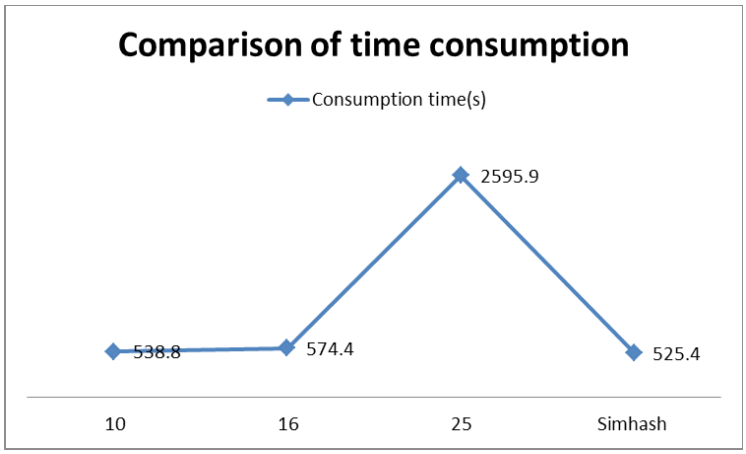

Figure 7: Comparison of Time Consumption

The above experimental comparison showed that the first experiment mainly set the threshold for the initial screening of Simhash algorithm to 16 in the unmodified text, that is, only texts of less than or equal to 16 were compared for the second time. It can be seen from Figure 4 that the original Simhash algorithm had the same recall rate as the algorithm proposed in this paper, but the accuracy of the proposed algorithm is improved. In Experiment 2, in the modified text, the test was done by setting the threshold to 10, 16 and 25. We can see from Figure 5 and Figure 7 that the threshold of 19 did not increase much when compared to 16, but the threshold of 19 consumes much longer than 16, so the threshold was eventually set to 16. In the third experiment, the accuracy of Simhash algorithm was 86.4% and the recall was 83.4% under the modified text, as shown in Figure 6. The accuracy of the algorithm proposed in this paper was 97.6% and the recall rate was 96%. So, it can be seen that there was a great improvement in the accuracy and recall. We also can see from Figure 7 that the threshold of 16 consumed slightly more time than Simhash and the speed was within the acceptable range for fast similarity detection under massive texts.

5.Conclusion

This paper proposes a Simhash algorithm in combination with the cosine distance method to improve the accuracy and recall. In this paper, TF-IDF algorithm is firstly used to extract the text features, by using Simhash algorithm, select the candidate texts, and finally the most similar text is selected by the cosine distance. Experiments show that this method has obviously improved the accuracy and recall for the modified texts. This method is suitable to compare the full text similarity. However, for more fine-grained comparison, a more refined algorithm of similarity is needed, which is also deemed as the future work to achieve full-text to sentence-level similarity detection.

References

[1]M.D. Zhu, L.X. Xu, *Cosine similarity query method for uncertain text data* [J]. Journal of Computer Science and Exploitation, 2016, 10: 44-60 (In Chinese)

[2]G. Salton, A. Wong, C.S. Yang, *A vector space modelfor automatic indexing* [J]. Communications of the ACM, 1975, 18 (11): 613 - 620

[3]Z.J. Zhan, L.N. Lian,Y.X. Ping, *Calculation of word similarity based on Baidu encyclopedia* [J] .Computer Science, 2013, 40 (6): 199-202 (In Chinese)

[4]J.L. Tian, Y. Zhao, *The word similarity computation based on tongyicicilin method* [J]. Journal of Jilin University: Information Science Edition, 2012, 28 (6): 602-608 (In Chinese)

[5]S.S. Desai, J.A. Laxminarayana, *WordNet and Semantic Similarity based Approach for Document Clustering*[C]. Computation System and Information Technology for Sustainable Solutions 2016, IEEE, India , 2016:312-317

[6]Hong T. Tu, Tuoi T. Phan, Khu P. Nguyen, *An adaptive Latent Semantic Analysis for text mining*[C]. 2017 International Conference on System Science and Engineering , IEEE, Vietnam , 2017:588-593

[7]Z.Z. WANG, M. He, Y. P. DU, *Text similarity calculation based on LDA topic model* [J] .Computer Science, 2013, 12: 292-232(In Chinese)

[8]A. Mnih, G. Hinton, *A scalable hierarchical distributed language model*[J].Advance in Neural Information Processing System, 2008: 1081–1088

[9]M.S. Charikar, *Similarity estimation techniques from rounding algorithms* [C]. Proceedings of the thiry-fourth annual ACM symposium on Theory of Computing, ACM, USA, 2002:380-388

[10]W. Zhu, W. Zhang, G.Z. Li, *A study of damp-heat syndrome classification using Word2vec and TF-IDF*[C]. IEEE International Conference on Bioinformatics and Biomedicine, IEEE, China, 1415-1420

[11]X. Jiang, Z.J. Wang, Y. Liang, Y.Z. Tao, *Semantic Similarity Detection of Massive Text Based on Semantic Fingerprint* [J]. Computer Knowledge and Technology, 2016, 12 (36): 175-177 (In Chinese)

[12]G.Q. Zhang, W.E. Ge, C.L. He, *Quick Search Optimization Method of Mass Similarity Documents Based on Simhash* [J]. Command Information Systems and Technology, 2015, 6 (2): 60-65 (In Chinese)

[13]C.L. Cheng, L. Chen, J. Xiong, H. Yu, *Research and Improvement of Deduplication Technology Based on Simhash Algorithm* [J]. Journal of Nanjing University of Posts and Telecommunications, 2016, 39 (3): 85-91(In Chinese)

[14]Y. Yi, Y.Z. Zhang, Z.J. Hu, *Research on Large Scale Document Deweighting Based on Simhash* Algorithm [J] .Institute of Information and Communication, 2015, 2: 28-29(In Chinese)

[15]J.C. Jiang, H. Hong, J.C. Cheng, *Automatic Digest Based on Keyword Weight and Sentence Features*[J]. Journal of South China University of Technology. 2010, 38 (7): 50-54(In Chinese)