Duplicate Short Text Detection Based on Word2vec

Jin Gao[1*], Yahao He[2], Xiaoyan Zhang[1] and Yamei Xia[1]
[1]*School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*
gjgj@bupt.edu.cn
[2]*School of Software , Tsinghua University, Beijing 100084, China*
hyh15@mails.tsinghua.edu.cn

Abstract—**In modern life, people own new social relationship, watch news, create e-commerce transactions and have entertainment online. Light blogs and short comments become more and more popular. The traditional duplicate long text detection algorithms are hard to be applied in the current situations, so more effective duplicate detection algorithm for short text is needed. Based on the bag-of-words model Word2vec, this paper proposes a kind of duplicate detection algorithm with semantic embedded for short text. Words are embedded into vectors which are as input elements in Simhash algorithm to acquire 64 bits sequence, then compare two sequences with Hamming distances and return result filtered by preset threshold value. Subsequently, a more superior improvement is proposed where we add weighted idea into. The results are compared with the unweighted Word2vec method and the traditional TF-IDF method. Experiments are carried out on the SICK corpus, and its result shows that the weighted Word2vec method achieves higher accuracy and recall rate.**

Keywords-short text; duplicate detection; TF-IDF; Word2vec; semantic similarity; Simhash

I. INTRODUCTION

In recent years, with the development of internet technology, text is the main form of internet information. There is a large amount of short texts generated every day in social, news, shopping and other fields. The cost of information replication is so low that it is easy to generate a large number of duplicate and redundant data [1]. Not only does a large amount of short text duplicate information make it difficult to retrieve information, but also it increases the storage burden. In the field of data analysis, it will also affect the effect of data mining algorithms. The traditional text duplicate detection method is not effective in solving the problem of short text. Therefore, short text duplicate detection becomes a hot research problem.

II. RELATED WORK

In order to solve the problem of text duplicate, firstly, researchers proposed a method for matching the longest string [2-4]. Monostori et al. [5] proposed a prototype MDR that used the model of suffix tree to match the maximum length substring. YAP [6] that is similar to the MDR method is also to match the maximum length substring. However, this method has high complexity of time and space, which is not practical. Border's shingling algorithm [7] tried to transform documents into set operations, but it also has very high complexity of time and space, so it is not suitable for practical applications. Subsequently, the researchers proposed a more practical vector space model(VSM) [8], The basic thoughts of this method are as follows: firstly, compute the word embedded, transform it into a vector, then, use different measurement methods to calculate the vector distance [9-12]. The advantage of this method is that it is convenient and high-speed. It has been applied in industry and has achieved good results.

Word embedding has developed rapidly and has been widely used. The most widely used method that is used to represent document is based on bag-of-words. The model was originally assumed for a text that ignore the word order and syntax, which is just as a collection of text words. Every word is independent, don't rely on other words and can't be influenced by the context. This method only calculate the occurrences of each word in the text and the number of times that the word appears in the whole text collections. TF-IDF [13] is the most classical method of text similarity measurement, the method takes into account both word frequency and reverse text frequency, which retains important words and ignores common words. Later, a lot of improvements have been made to TF-IDF, such as adding weight to word item, modifying the TF-IDF computed functions, combining other methods and so on. The above models based on bag-of-words method work well on traditional document classification, but often lead to high and sparse matrix dimensions. Thus these models can't represent the semantics of a document well. For example, when the document vector is expressed by the bag-of-words model, the structure similarity of journalist and reporter is zero, actually, the two words are often synonymous. Thus the bag-of-words model is difficult to represent the semantic information of the document. With the development of deep learning, Word vector representation method based on neural networks for feature extraction has attracted more and more attention. In 2013, Mikolov et al. [14] proposed Word2vec model that is used to calculate the word vector, which transforms a word into a lower dimensional vector by the context information of the word, and the more similar words are more close to each other in vector space [15-16]. This method is very successful in the field of natural language processing. It has been widely used in text clustering, searching synonyms, emotional classification and so on. Good semantic representation effect has been achieved.

The main methods of text similarity measurement are summarized as follows: One is to compute the similarity between vectors by calculating euclidean distance or cosine

978-1-5386-0497-7/17/$31.00 ©2017 IEEE

similarity between vectors. The other one is Simhash and Hamming distance method. In 2012，Google’s Charikar [17] proposed a Simhash algorithm, This method transforms a document into a n bits signature and acquires the similarity between the two original documents by calculating the Hamming distance of the two signatures. The latter approach has made a great breakthrough in time efficiency through hash. It is a local sensitive hashing method, which is widely used in large-scale text duplicate detection.

Based on previous research work, we design our method according to the characteristics of short text, which contains fewer words, extracts features difficultly and is lack of word standardization [18]. We use Word2vec model weighted by TF-IDF to obtain more accurate word vector, then obtain characteristics of the text by Simhash. Finally, calculate Hamming distance to determine whether the text is duplicate.

III. DUPLICATE DETECTION MODEL BASED ON WORD2VEC WEIGHTED BY TF-IDF

The short text is short and contains fewer words, it is not only necessary to consider the completely repeated text, but also the semantic similarity can be determined as duplicate text. Therefore, this paper adopt deep-learning Word2vec model, which accurately characterizes the similarity between words and represents the short text at a deeper level. This paper use Word2vec [19] model weighted by TF-IDF [20] and Simhash [21] method to conduct experiments of duplicate detection of short text. First, the word vector is generated according to the Word2vec model, and the word vector is weighted by the TF-IDF value of the word. Then short text feature vector is generated according to the principle of Simhash. The distance between the vector and the pivot vector that is defined with 1 is calculated by the cosine similarity method. The feature word represented by 0 and 1 is generated. Finally, By calculating the Hamming distance between the feature characters of short text [22] determines whether the text is duplicate.

The research model of this paper is shown in figure 1.

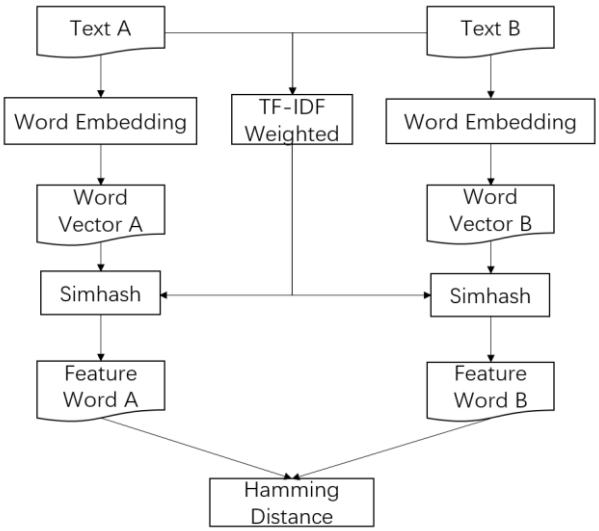

Figure 1. The model of duplicate detection method

1) Preprocess short text, each word is put into the Word2vec model for training, and the word vector W is obtained.

$$W = \begin{bmatrix} w_1 & \cdots & w_n \end{bmatrix} \tag{1}$$

2) The TF-IDF value of the word is normalized to λ, the word vector W in step (1) is weighted by λ, then obtain W_1.

$$W_1 = \begin{bmatrix} \lambda w_1 & \cdots & \lambda w_n \end{bmatrix} \tag{2}$$

3) Define a $m \times n$ matrix of 1 as the pivot A, for each word vector W_1 in step (2), the vector W_2 representing the feature of the short text is obtained according to the Simhash method.

$$W_2 = \begin{pmatrix} a_{11} + \lambda w_1 & \cdots & a_{1n} + \lambda w_n \\ \vdots & \ddots & \vdots \\ a_{m1} + \lambda w_m & \cdots & a_{mn} + \lambda w_n \end{pmatrix} \tag{3}$$

4) the vector is simplified according to the cosine similarity calculation method. The cosine similarity formula is as follows.

$$similarity = \cos\theta = \frac{A \cdot B}{\|A\|_2 \|B\|_2} \tag{4}$$

According to the text features W_2 obtained in step (3) and pivot B as follows.

$$W_2 = \begin{bmatrix} W_1 \\ \vdots \\ W_m \end{bmatrix} \tag{5}$$

$$B = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \tag{6}$$

Each W_m calculates the cosine similarity with B, and obtains the $m \times 1$ matrix W_3.

$$W_3 = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix} \tag{7}$$

5) According to the result of W_3 in step (4), less than 0 is expressed as 0 and greater than 0 is expressed as 1. Then the feature word represented by m bits of 0 and 1 is obtained .

6) According to the Hamming distance between m bits feature word, when the Hamming distance is less than a threshold value, the text is determined duplicate.

IV. EXPERIMENT AND RESULT ANALYSIS

A. Dataset Preprocess

In this paper, test dataset consists of long texts and short texts. 10,000 Wikipedia were long text test sets as well as 10,000 SICK corpus were short text test set. For long text, we generate test set by randomly remove some sentences, for short text, because the SICK corpus has been scored by its relatedness, we only need to define a threshold to split whole corpus into two parts, the first one is duplicate as its score is larger than threshold, another one isn't according to its score. The document format of the processing results contains four attributes, e.g., pair id, first text, second text, similarity. Each attribute is separated by a tab. The results of the specific treatment are shown in table 1.

TABLE I. THE RESULT OF SHORT TEXT PREPROCESSING

pair_id	*text_A*	*text_B*	*isSimilar*
1	A skilled person is riding a bicycle on one wheel	A person is riding the bicycle on one wheel	Y
2	A person is riding the bicycle on one wheel	A man in a black jacket is doing tricks on a motorbike	N
3	Children in red shirts are playing in the leaves	Children in red shirts are sleeping in the leaves	N

B. Experiment Evaluation

In this paper, the classical evaluation methodology: accuracy rate A, recall rate R, and $F1$ score value are used to evaluate the experimental results [23]. The accuracy rate is the ratio of the amount of duplicate text detected to text detected, and the recall rate is the ratio of the amount of duplicate text detected to whole amount of duplicate text. The $F1$ value takes into account both the accuracy rate and the recall rate so it's a better metric to reflect whether an algorithm is efficient or not. They are written as follows:

$$A = \frac{TP + TN}{TP + FN + FP + TN} \tag{8}$$

$$R = \frac{TP}{TP + FN} \tag{9}$$

$$F1 = \frac{2 \times R \times A}{R + A} \tag{10}$$

TP is the amount of duplicate texts detected as duplicate texts correctly , FP is the amount of duplicate texts detected as non-duplicate texts wrongly, TN is the amount of non-duplicate texts detected as non-duplicate correctly, FN is the amount of non-duplicate texts detected as duplicate texts wrongly.

C. Result Analysis

The experiment is carried out on two data sets, long texts and short texts. Each kind of experiment is compared with three methods. The first is the Word2vec algorithm model which is weighted by TF-IDF, and the second is pure Word2vec algorithm model process, the third is the traditional TF-IDF algorithm model. We use the Hamming distance to measure the similarity of the text to determine whether the text is duplicate or not.

The experimental process and experimental parameters are as follows: First, Word2vec is used to embed text to the vector of 300 dimensions, and then the word vector is weighted by normalized TF-IDF. Then, each word vector is processed according to Simhash thought, and get 64×300 word matrix. After simplifying with cosine similarity, 64-bit words of zeroes and ones are produced, and finally calculate the Hamming distance to detect the duplicate text.

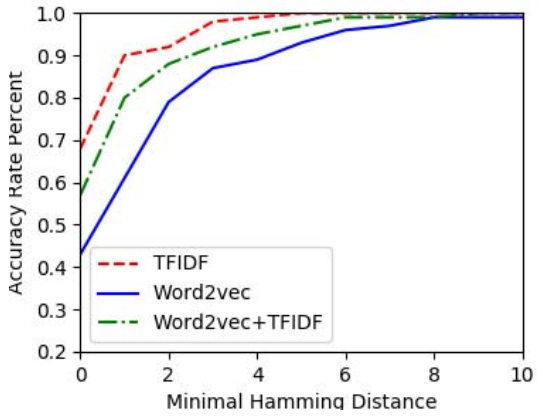

Figure 2. The change of Hamming distance of long text on the effect of accuracy rate

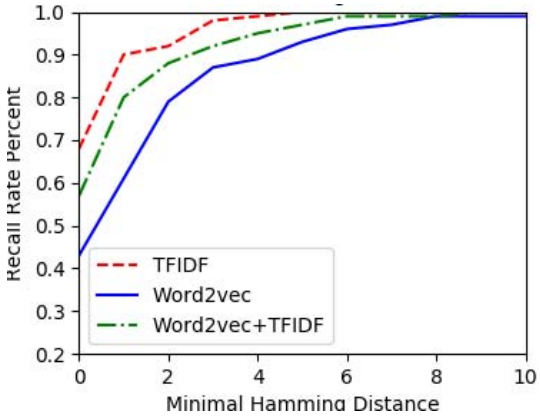

Figure 3. The change of Hamming distance of long text on the effect of recall rate

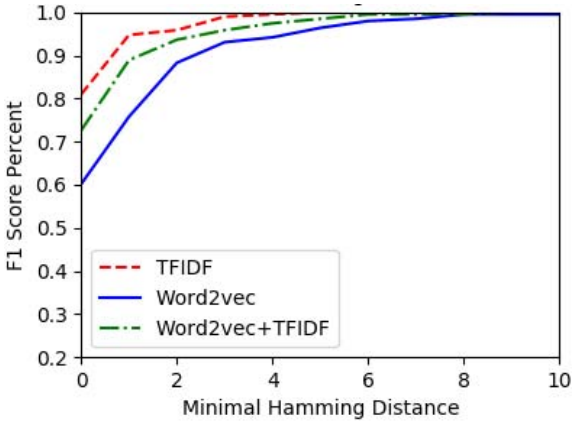

Figure 4. The change of Hamming distance of long text on the effect of F1

As shown in Figure 2,3 and 4, when Hamming distance is 6, accuracy rate, recall rate and F1 score rate are all over 99% when applying weighted Word2vec method, which is very close to the result of traditional TF-IDF method. So weighted Word2vec method is proved to be efficient.

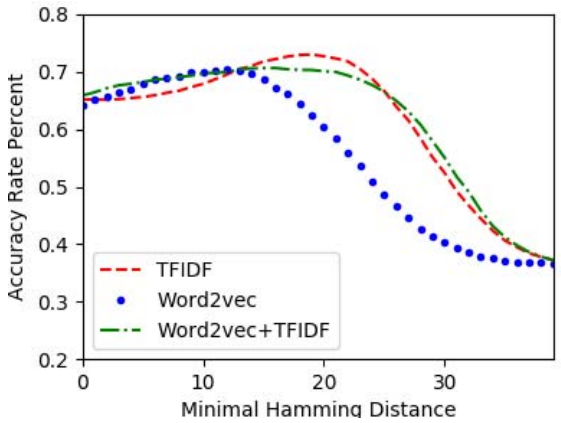

Figure 5. The change of the Hamming distance of short text on the effect of accuracy rate

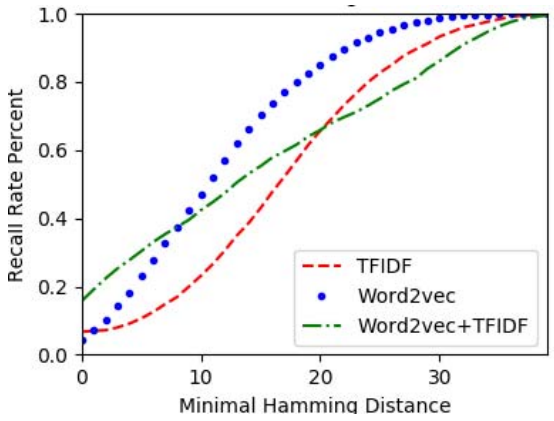

Figure 6. The change of the Hamming distance of short text on the effect of recall rate

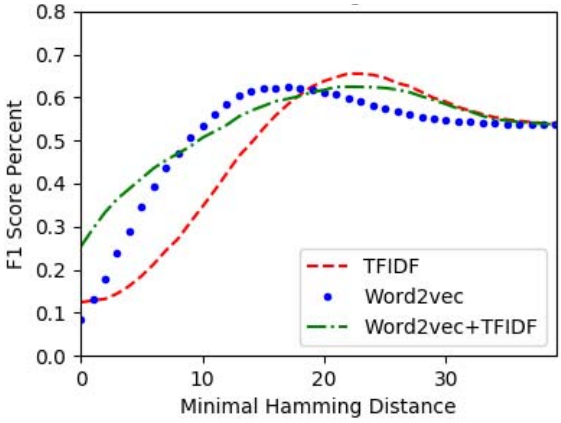

Figure 7. The change of the Hamming distance of short text on the effect of F1

As shown in Figure 5, 6 and 7, when the Hamming distance is within the meaningful range (less than 10), the accuracy rate, recall rate and the F1 value of the experimental results of Word2vec are higher than the traditional TF-IDF algorithm, Weighted Word2vec method is more efficient than the first two, because the method is semantically based, not only through the TF-IDF weighted consideration of the frequency of the word, but also taking into account the synonyms, and other effects, e.g., "He is clever" and "He is smart", the result of method detection is duplicate. Therefore, Word2vec method weighted by TF-IDF is superior to unweighted Word2vec and traditional TF-IDF method in short text data set in this paper.

TABLE II. THE RESULTS OF THREE ALGORITHMS ON LONG TEXT

Method	*Accuracy Rate (%)*	*Recall Rate (%)*	*F1 (%)*
Weighted Word2vec	99.0	99.0	99.4
Word2vec	96.0	96.0	97.9
TF-IDF	100.0	100.0	100.0

As shown in table 2, we show that the comparison result of the three algorithm models on the long text test set when Hamming distance is 6. It shows that efficiency of Word2vec weighted by TF-IDF method is very close to traditional TF-IDF method.

TABLE III. THE RESULTS OF THREE ALGORITHMS ON SHORT TEXT

Method	*Accuracy Rate (%)*	*Recall Rate (%)*	*F1 (%)*
Weighted Word2vec	68.6	33.0	43.6
Word2vec	68.5	27.9	39.4
TF-IDF	65.9	12.6	21.2

As shown in table 3, we show that the comparative result of three algorithm models on the short text test set when Hamming distance is 6. Because short text has fewer feature to distinguish themselves, weighted Word2vec method we proposed produce superior result than previous two methods.

Based on the above experimental results, we have verified that the efficiency of the three models is very high, and the experimental results are very close. It is proved that weighted Word2vec method is reasonable and efficient. As the short text features are less, the semantic-based duplicate detection

method is theoretically higher than the method based on word frequency, and this conclusion has been proved in the short text test set. Therefore, weighted Word2vec method is efficient.

V. CONCLUSION

Short text duplicate detection for natural language processing, data mining and other fields is very significant. This paper not only takes into account the frequency of words contained in the short text, but also takes into account the sematic in the short text. After comparing the experiments, it is proved that the model in this paper is superior. The method in this paper needs to be further improved, the author will further study and explore, to enhance the model.

REFERENCES

[1] Xiang Gao, Bing Li. Study on Chinese Short Text Duplicate Detection[J]. Computer Engineering and Applications, 2014, 50(16).

[2] Cong Geng, Dejun Xue. Study on Chinese Document Duplicate Detection Method[J]. Modern Library and Information Technology, 2007(6).

[3] Yujuan Cao, Zhendong Zhao, Fang Zhao. Approximate Web Detection Algorithm Based on Concept and Semantic Web[J]. Journal of Software, 2011, 22(8).

[4] Junpeng Bao, Junyi Shen, Xiaodong Liu. A Survey of Natural Language Reproduction Detection[J]. Journal of Software, 2003, 14(10).

[5] Monostori K, Zaslavsky A, Schmidt H.MatchDetectReveal: finding overlapping and similar digital documents[C/OL]// Proceedings of the Information Resources Management Association International Conference(IRMA2000), 2000.

[6] Wise MJ.YAP3:Improved detection of similarities in computer programs and other texts[C/OL].Proceedings of the SIGCSE'96.1996:130-134.

[7] A. Z. Border, S. C. Glassman, M. S. Manasse, et al. Syntactic clustering of the web. Computer Networks, 29(1157-1166): 8–13,1997.

[8] Kulkarni R V, Forster A, Venayagamoorthy G K. Com-putational Inteligence in Wireles Sensor Networks:A Survey[J].IEEE Communications Surveys & Tutorials, 2011, 13(1):68-96.

[9] Dai L, Chang Y, Shen Z.An Optimal Task Scheduling Algorithm in Wireles Sensor Networks [J]. International Journal of Computers Communications & Control, 2011, 6(1):101-112.

[10] Zhang Kuo, Xu Hui, Tang Jie, et a1. Keyword extraction using support vector machine[C]//Proceedings of the 7th International Conference on Web—Age Information Management, Hong Kong, China, 2006 : 85–96.

[11] Xiao xiao, Xu Qihua. A Comparative Research on the Classification and Regression Based on SVM and BP[J]. The Journal of New Industrialization, 2014, 4(5): 48–53.

[12] Yuan Ailing, Qi Wei, Qian Xu. Text Classification with a SVM based on Manifold Regularization[J], Software, 2013, 34(2): 65–68.

[13] Joachims T.A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization [M].Springer US,1997:143-151.

[14] Mikolov T, Sutskever I, Chen K, et al.Distributed Representa-tions of Words and Phrases and their Compositionality [J].Ad- vances in Neural Information Procesing Systems, 2013, 26: 3111-3119.

[15] Mikolov T, Chen K, Corado G, et al.Eficient Estimation of Word Representations in Vector Space [C]//ICLR 2013.

[16] Luhn. H. P. The Automatic Creation of Literature Abstracts[J]. IBM Joumal of Research Development, 1958, 2(2): 159.

[17] Charikar M S .Similarity estimation techniques from rounding algorithms [C]//Proceedings of the thir-fourth annual ACM symposium on Theory of Compu - ting .ACM , 2002 :380 - 388.

[18] Guo W, Chen Y, Chen G .Dynamic Task Scheduling Strategy with Game Theory in Wireles Sensor Net- works[J]. New Mathematics and Natural Computation, 2014, 10(3):211-224.

[19] Yinhao Lu. A Method of Computing Similarity of Sentence Based on Word2Vector and Editing Distance[J]. Computer Knowledge and Technology, 2017, 13(5).

[20] Ming Tang, Lei Zhu, Xianchn Zou. A Document Vector Representation Based on Word2Vec[J]. Computer Science,2016, 43(6):214-217.

[21] Sadowski C , Lcvin G.Simhash:Hash-based similarity deteetion[R].Technical report, Google,2007.

[22] Schell, S.V,and Gardner,W.A, 1990.Signal selective high-resolution direction finding multipath.In Proc.IEEE Int.Cone Acoust., Speech, Signal Process., PP:2667-2670.

[23] Chunlin Chen, Lin Chen, Jing Xiong. Research and Improvement of Deduplication Technology Based on Simhash Algorithm[J]. Journal of Nanjing University of Posts and Telecommunications, 2016, 36(3):85-91.