J Inf Process Syst, Vol.13, No.4, pp.863~875, August 2017
https://doi.org/10.3745/JIPS.02.0067

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

A Text Similarity Measurement Method Based on Singular Value Decomposition and Semantic Relevance

Xu Li*, Chunlong Yao*, Fenglong Fan*, and Xiaoqiang Yu*

Abstract

The traditional text similarity measurement methods based on word frequency vector ignore the semantic relationships between words, which has become the obstacle to text similarity calculation, together with the high-dimensionality and sparsity of document vector. To address the problems, the improved singular value decomposition is used to reduce dimensionality and remove noises of the text representation model. The optimal number of singular values is analyzed and the semantic relevance between words can be calculated in constructed semantic space. An inverted index construction algorithm and the similarity definitions between vectors are proposed to calculate the similarity between two documents on the semantic level. The experimental results on benchmark corpus demonstrate that the proposed method promotes the evaluation metrics of F-measure.

Keywords

Natural Language Processing, Semantic Relevance, Singular Value Decomposition, Text Representation, Text Similarity Measurement

1. Introduction

Text similarity measurement is a real-valued function that quantifies the similarity between two documents, which is an important issue in natural language processing and plays an important role in information retrieval, document categorization, document copy detection, document summarization and machine translation, and so on.

The traditional text similarity measurements based on word frequency vector model commonly use the bag-of-words model. The bag-of-words model disregards collocation information in word strings: the context for each word essentially becomes the entire document in which it appears. Let $D=\{d_1,d_2,\ldots,d_n\}$ be a document set of n documents, where d_1, d_2,…, d_n is individual document. Let the word set $W=\{w_1,w_2,\ldots,w_m\}$ be the feature vector of the document set. Each document d_i, $1\le i\le n$, is represented as $d_i=(a_{i1}, a_{i2}, \ldots, a_{ij}, \ldots, a_{im})^T$, where each a_{ij} denotes the feature value of w_j in the ith document. Each document is modeled as a vector in the m dimensional space R^m, which is why this method is called the VSM (vector space model). The feature value a_{ij} could be word frequency, relative word frequency or TF-IDF (term frequency-inverse document frequency). Cosine similarity,

Manuscript received February 15, 2016; first revision November 11, 2016; second revision February 9, 2017; accepted March 14, 2017.
Corresponding Author: Xu Li (lixu102@aliyun.com)
* School of Information Science and Engineering, Dalian Polytechnic University, Dalian, China (yaocl@dlpu.edu.cn, fanfl@dlpu.edu.cn, tigeryxq@dlpu.edu.cn)

Jaccard similarity and Manhattan distance are usually used to measure the similarity between two vectors [1]. These traditional methods show some limitations as a result of ignoring the sense of the words and the original semantic relations between the words in the feature value calculation of a document vector. Furthermore, the dimensionality of the document vector can be extremely large and the vectors are typically very sparse, i.e., a lot of feature values are zeros. The high-dimensionality and sparsity of document vector have become severe obstacles to text similarity measurement [2,3].

Nowadays terabytes or petabytes of data pour into our computer networks, the World Wide Web, and various data storage devices every day. Text mining has aroused great concerns of the researchers. Due to the complexity of structure and the diversity of ambiguities, researchers face many challenges to systematically extract the valuable information from the tremendous amounts of raw data [4]. Most words have multiple possible senses. For example, *pen* can have two senses: a writing instrument or an enclosure where small children can play. There are two documents regarding *pen*, which respectively refer to writing instrument and enclosure. The traditional VSM-based text similarity measurements are likely to regard them as similar documents. On the other hand, the two documents concerning fruits may be considered dissimilar subjects because the two different words *apple* or *orange* continually occurs in two documents, respectively.

To solve the polysemous and synonym problems, the knowledge-based similarity measurements use the knowledge in the specific fields and calculate the text similarity by recognizing the synonyms, semantic redundancies and textual entailments in the documents [5]. The establishment of the knowledge base is a complex and ambitious project, and therefore the knowledge base is generally replaced by the comprehensive word dictionary, such as WordNet or HowNet, in the existing research works [6]. The text similarity measurements organized all the words to form a semantic network and calculated the similarity between words by examining the network edges, densities, node depths and link types between them [7,8]. The word similarity is expanded to paragraph similarity counting, and then the paragraph similarity counting is expanded to article similarity counting. Based on the above the idea of the similarity measurements between words, Wang and Hodges [9] used the word sense disambiguation method to improve clustering. The F-measure of this approach had some improvements over the traditional ones. However, the high dimensional feature space not only consumes the time of clustering algorithm but also reduces the accuracy of the algorithm. Based on the clustering technology, Aliguliyev [10] proposed a similarity measurement between sentences and applied it to automatic text summarization. Gang et al. [11] introduced ontology to recalculate and reorder the relevance between documents for the results returned by the search engine. However, this approach required the interactions with users in order to obtain an accurate result. Hotho et al. [12] analyzed the concepts of words, synonyms and hyponymy words by WordNet and replaced the word frequency vector with the new one where each word in the document was extended to synonyms or hyponymy words. However, there are lacks of reduction the dimensionality of word-document matrix and the similarity measurement definition between two documents in the approach. Bellegarda [13] exploited latent semantic information in statistical language modeling and employed singular value decomposition (SVD) to reduce the dimensionality of the word-document matrix. However, this approach did not analyze the effect of the number of selected singular values on clustering accuracy.

A novel text similarity measurement based on singular value decomposition and semantic relevance is proposed in this paper. TF-IDF method is used to build an associated matrix of co-occurrences between words and the word-document matrix is performed SVD. If the decomposed space dimension

s is too small, the word-document matrix is compressed too much to represent the original semantics. If *s* is too large, the effect of the dimensionality reduction is not ideal and a lot of noises retained. Therefore, the optimal number of singular values is analyzed in the proposed approach. The words which have the greater semantic relevance with other words and the larger TF-IDF feature values are more important in text similarity measurement. These words should be weighted in text similarity calculation. Thus, the decomposed singular vectors are mapped onto a semantic space, in which the Euclidean distances that reflects the semantic closeness between words are calculated. An inverted index of semantic relevance between words and the weighting factor between document vectors are proposed in the paper. The similarity definition combined semantic and positional information with TF-IDF is given. Compared with the traditional TF-IDF and the method proposed by Hotho et al. [12], the experiments show that the proposed approach makes a significant improvement in clustering.

The contributions of this paper are as follows. (1) The optimal value selection method of singular value number is proposed in the improved singular value decomposition, which not only retains most of the original semantic information but also overcomes the curse of dimensionality and noises. (2) An inverted index construction algorithm for semantic relevance and the similarity definition between document vectors are proposed, which provide the basis and guidance for semantic similarity calculation between documents. (3) The experimental results on benchmark corpus demonstrate that the proposed method promotes the evaluation metrics of F-Measure.

The rest of this paper is organized as follows. Proposed approach is described in Section 2. Experimental results and discussions are presented in Section 3. Finally, concluding remarks are given in Section 4.

2. Proposed Text Similarity Measurement

2.1 Association Matrix Construction

Vector space model is an algebraic model for representing text documents as vectors of identifiers. Preprocessing is necessary before converting document into vector. Preprocessing includes segmentation, deleting stop words and function words, and processing the special words such as name, place name, etc. The word set $W=\{w_j|j=1,2,\ldots,m\}$ is obtained after the completion of document pretreatment. In this paper, TF-IDF method is used to calculate the feature value of each word in the document. Each document d_i, $1\le i\le n$, is represented as $d_i=(a_{i1},a_{i2},\ldots,a_{ij},\ldots,a_{im})^{\mathrm{T}}$, where each a_{ij}, $1\le i\le n$ $1\le j\le m$, denotes the feature value of word w_j in the ith document. TF-IDF is a combination of word frequency and inverse document frequency. Word frequency is based on the assumption: the weight of a word that occurs in a document is simply proportional to the word frequency. Typically, the words which have the greater frequency are more related to the topic of the document. Inverse document frequency is a measure of how much information the word provides, which is incorporated which diminishes the weight of words that occur very frequently in the document set and increases the weight of words that occur rarely. According to the information theory, IDF actually is a cross-entropy of word probability distribution in the certain conditions [14]. The formula for TF-IDF is given below:

$$\text{tf-idf}_i(w_j)=\text{tf}_i(w_j)\times\text{idf}(w_j)=\frac{t_{ij}}{\sum_{j=1}^{m}t_{ij}}\times\log\frac{n}{|\{i:w_j\in d_i\}|} \tag{1}$$

where

t_{ij} number of times word w_j occurs in document d_i;

$\sum_{j=1}^{m} t_{i,j}$ sum of the number of times each word occurs in document d_i;

n total number of documents in the corpus;

$|\{i:w_j \in d_i\}|$ number of documents where the word w_j appears.

The $m \times n$ word-document matrix D resulting from the above feature extraction defines two vector representations for the words and the documents. The matrix may be sparse. Therefore the smoothing is required for a zero probability or low probability events. A small constant α is defined, and the association matrix A between words and documents is constructed as follows:

$$A = \frac{\alpha}{n} \cdot I + (1-\alpha)D \tag{2}$$

where I denotes the identity matrix.

2.2 Improved Singular Value Decomposition

Each word w_j can be uniquely associated with a row vector of dimension n, and each document d_i can be uniquely associated with a column vector of dimension m. Unfortunately these vector representations are unpractical for two reasons. First, the dimension m and n can be extremely large; second, the two spaces are distinct from one other. To address the issues, it is useful to employ singular value decomposition. The nature of the singular value decomposition may be considered the subspace rotation. We regard the word-document matrix as a transformation in high dimensional space. The good description of the transformation lies in its major changes characterization in the direction [15]. By applying singular value decomposition, we can obtain a descending singular values sequence, which corresponds to a significant degree of changes in the direction from high to low. Thus, we can approximately describe the word-document matrix by characterizing a number of the most important changes in the direction of the transformation. According to the matrix theory, for any $m \times n$ matrix A can be employed the singular value decomposition.

$$A = UEV^T \tag{3}$$

where

U ($m \times r$) left singular matrix with row vectors u_j ($1 \le j \le m$);

E ($r \times r$) diagonal matrix of singular values $e_1 \ge e_2 \ge \ldots \ge e_r > 0$;

V ($n \times r$) right singular matrix with row vectors v_i ($1 \le i \le n$);

$r << \min(m,n)$ order of the decomposition.

On the basis of decomposing the matrix A into the three matrix products, latent semantic analysis chooses the s ($s << r$) largest singular values from E and discards other singular values to reduce the dimensionality. The matrix E becomes the $s \times s$ diagonal matrix of singular values E_s. The m-s, n-s

column vectors are removed from U and V respectively to obtain U_s and V_s. U_s and V_s are column-orthonormal matrices. $U_s U_s^T = V_s V_s^T = I$. The column vectors of U_s and V_s each define an orthonormal basis for the space of dimension s spanned by the u_j's and v_i's. Furthermore, it can be shown that the matrix A_s is the rank-s approximation to the word-document matrix A.

$$A_s = U_s E_s V_s^T \tag{4}$$

The row vectors of A_s are projected onto the orthonormal basis formed by the column vectors of the right singular matrix V_s, or, equivalently, the row vectors of V_s^T. This defines a new representation for the words, in terms of their coordinates in this projection, namely, the rows of $U_s E_s$. In essence, the row vector $u_j E_s$ characterizes the position of word w_j in the underlying s-dimensional space. Similarly, the column vectors of A_s are projected onto the orthonormal basis formed by the column vectors of the left singular matrix U_s. The coordinates of the documents in this space are, therefore, given by the columns of $E_s V_s^T$. This in turn means that the column vector $E_s v_i^T$, or, equivalently, the row vector $v_i E_s$, characterize the position of document d_i in s dimensions. We refer to each of the m scaled vectors $u_j E_s$ as a word vector, uniquely associated with word w_j in the vocabulary, and each of the n scaled vectors $v_i E_s$ as a document vector, uniquely associated with document d_i in the corpus.

By employing singular value decomposition and choosing the rank-s approximation matrix, the "synonymous" noises contained in the original word-document matrix are removed and the semantic relations between word and document are highlighted. On the other hand, the word-document vector space is reduced and the accuracy of text clustering is improved. However, the value of s directly affects dimensionality reduction and text clustering. The value of s should be small enough to remove the noises that should not be retained; meanwhile, the value of s should be large enough to preserve the main framework in the underlying semantic structure. If s reaches a certain value, the reduced semantic space not only retains most of the original semantic information but also removes most of the noises, and then the value is the optimal number of singular values and it should be selected. As is well known, the matrix A_s is the approximation to the word-document matrix A. In order to make the difference between the two matrices as small as possible, the appropriate value of s should to be selected so that their F-norm is as small as possible. According to the above theory, the optimal value selection of s is transformed into finding the minimum value of the following formula.

$$D = \left\| A - A_s \right\|_F = \sum_{i=1}^{n} \sum_{j=1}^{m} (a_{ij} - a_{ij}')^2 \tag{5}$$

where D is the F-norm, a_{ij} and a_{ij}' are the values of ith row and jth column of the A and A_s respectively.

The distribution rule of singular values is found in the singular value decomposition. The singular values are arranged in descending order and their decreasing speeds are different. The decreasing speeds of the singular values in the front of row are faster and the decreasing speeds of the ones in the behind of row tend to be slow. There is a turning point and the descending trend of the singular values is relatively gentle after the turning point. The small singular values can be ignored on the premise that the semantic information of the original matrix should be retained correctly. According to the deduction, we use the turning point in decreasing trend of the singular values to select the optimal value of s in the proposed method.

2.3 Relevance Weighting Factor of Vectors

The words which have the greater relevance with other ones and the larger TF-IDF feature values are more important in the corpus. Thus, these words should be weighted when the similarity between vectors is calculated. An inverted index of semantic relevance between words is proposed in the paper, which also be used to calculate the relevance weighting factor between document vectors. According to the above illustration for singular value decomposition, we know that the row vector u_jE_s characterizes the position of word w_j in the underlying s-dimensional semantic space, for $1\leq j\leq m$. Therefore, we conclude that a natural metric to consider for the "closeness" between words is the Euclidean distance between the ith and jth rows of the matrix U_sE_s, namely, u_iE_s and u_jE_s, for any $1\leq i, j\leq m$. If the semantic relevance between words is larger than a predefined threshold, the relevance information should be added to the inverted list. An inverted index construction algorithm is as follows.

Input: the document vectors $\overline{v}_i$ and $\overline{v}_j$, a relevance threshold σ

Output: the relevance weighting factor between vectors $\overline{v}_i$ and $\overline{v}_j$, the maximum semantic relevance of each word

Begin

Step 1: An inverted index initialization.

Step 2: Traversing all the associated words in the vector $\overline{v}_j$ with the first word w_1 in the vector $\overline{v}_i$. The formula for calculating the semantic relevance between word w_k and word w_l is given as follows, for any $1\leq k, l\leq m$.

$$c(w_k, w_l) = \cos^{-1}((u_kE_s - u_lE_s)(u_kE_s - u_lE_s)^{\mathrm{T}})^{\frac{1}{2}} \tag{6}$$

If the calculated relevance is larger than the predefined threshold, an inverted item w_{i1} is created and the tuple (*word*, *relevance*) is added to the inverted list.

Step 3: Repeating step 2 from the two word w_2 to the last word in the vector $\overline{v}_i$.

Step 4: The inverted index storage mechanism of the vector $\overline{v}_i$ is shown in Fig. 1.

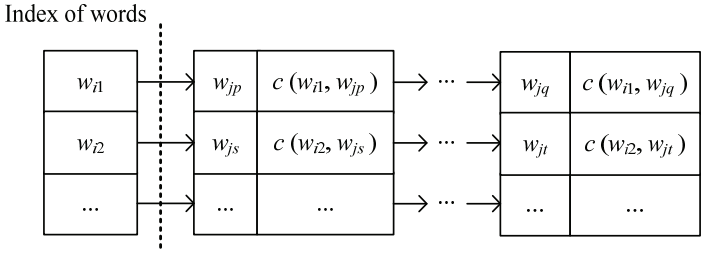

Fig. 1. Inverted index storage mechanism.

Step 5: All the words in the vector $\overline{v}_j$ associated with the words in vector $\overline{v}_i$ are found by the above steps. However, due to the associated words of each one are different, it is necessary to calculate again the associated words of each word in the vector $\overline{v}_j$ and the semantic relevance between them. According to the expression (6), we can know that $c(w_{ik}, w_{jl}) = c(w_{jl}, w_{ik})$. Therefore, the inverted

index of the vector $\overline{v}_j$ is obtained by the transforming the inverted index of the vector $\overline{v}_i$. Searching each word of the vector $\overline{v}_j$ in the inverted list of the vector $\overline{v}_i$. If a word of the vector $\overline{v}_j$ appears in the inverted list, a new inverted item of the word is created and the semantic relevance between words is extracted which is an integral part of the inverted index of the vector $\overline{v}_j$. For example, we may know there has a close semantic relevance between the word w_{i1} in the vector $\overline{v}_i$ and the word w_{jp} in the vector $\overline{v}_j$ from Fig. 1. When traversing the inverted list of the vector $\overline{v}_i$, a new inverted item w_{jp} is created and the tuple (w_{i1},$c(w_{i1}, w_{jp})$) is extracted and added to the inverted list of vector $\overline{v}_j$.

Step 6: The relevance weighting factor between vectors is defined as follows.

$$wf = 1 + \frac{1}{2} \times \left[\frac{\sum_{k \in \Lambda_i} tfidf(w_{ik})}{\sum_{k=1}^{n} tfidf(w_{ik})} + \frac{\sum_{l \in \Lambda_j} tfidf(w_{jl})}{\sum_{l=1}^{n} tfidf(w_{jl})} \right] \tag{7}$$

where $tfidf(w_{ik})$ denotes the TF-IDF value of the word w_{ik}. The set Λ_i and Λ_j in the expression (7) are defined as follows.

$$\Lambda_i = \left\{ k : 1 \le k \le m, \max_{1 \le l \le m} \left\{ c(w_{ik}, w_{jl}) \right\} \ge \sigma \right\} \tag{8}$$

$$\Lambda_j = \left\{ l : 1 \le l \le m, \max_{1 \le k \le m} \left\{ c(w_{jl}, w_{ik}) \right\} \ge \sigma \right\} \tag{9}$$

If the semantic relevance between the word w_{ik} in the vector $\overline{v}_i$ and the word w_{jl} in the vector $\overline{v}_j$ is larger than the threshold σ, the word w_{ik} is put into the set Λ_i. Similarly, each element contained in the set Λ_j is selected respectively.

Step 7: The relevance weighting factor between two vectors and the maximum relevance $\max_{1 \le l \le m} c\left(w_{ik}, w_{jl}\right)$ of each word are returned.

End

2.4 Text Similarity Measurement

In the vector space model, each document is represented as a feature value vector. The cosine of the angle between two feature value vectors usually be used to measure the similarity between two documents. Let $\overline{v}_i = (a_{i1}, a_{i2}, ..., a_{ik}, ..., a_{im})$ and $\overline{v}_j = (a_{j1}, a_{j2}, ..., a_{jl}, ..., a_{jm})$ be the feature value vectors of the ith document and the jth document respectively, where each a_{ik} ($1 \le k \le m$) denotes the feature value of w_k in the ith document. The feature value calculation is given as follows.

$$a_{ik} = tfidf_i(w_{ik}) \times pwf(w_{ik}) \times \max_{1 \le l \le m} c\left(w_{ik}, w_{jl}\right) \tag{10}$$

where $tfidf_i(w_{ik})$ denotes the TF-IDF of word w_{ik} in the ith document, $pwf(w_{ik})$ denotes the position weighting factor of word w_{ik}, $\max_{1\le l\le m} c(w_{ik}, w_{jl})$ denotes the maximum semantic relevance of word w_{ik} with other words.

Word position in the document is an important factor in the text similarity measurement. The words in the title are more important than the ones in the main body of document. Even in the main body, the words at the beginning and end of documents are more important than the ones in the middle of documents. Therefore, the words appeared in the title and the important positions should be weighted to improve the accuracy of clustering. The word position weighting factor is calculated as follows.

$$pwf(w_{ik}) = 1 + \log_2(1 + n(w_{ik})) \tag{11}$$

where $n(w_{ik})$ denotes the total number of word w_{ik} occurs in the title, abstract, keywords, conclusions. The words which occur more times in the above positions are more important for text clustering, and the position weighting factors are larger.

The formula for the similarity between the feature value vectors and text similarity between the documents are given as follows.

$$VectSim(\overline{v}_i, \overline{v}_j) = \frac{\overline{v}_i \overline{v}_j}{\| \overline{v}_i \| \| \overline{v}_j \|} \tag{12}$$

$$TextSim(\overline{v}_i, \overline{v}_j) = wf \times VectSim(\overline{v}_i, \overline{v}_j) \tag{13}$$

3. Experiments

Our experiment uses Reuters-21578 and 20 Newsgroups datasets which are widely used in text processing. There are significant differences between the size of text, the number of clustering and text distribution in these datasets. The format of the text in the Reuters-21578 dataset is SGM rather than plain text, thus these documents need be pretreated. Three text subsets of each dataset are selected in our experiment, namely R1, R2, R3 from the Reuters-21578 dataset and N1, N2, N3 from the 20 Newsgroups dataset. Each document in the dataset is divided into one or more specific classes in advance. Table 1 shows the characteristics of each data subset in the experiment.

Table 1. Characteristics of the experimental data

Name	Total number of documents	Number of clusters	Minimum number of documents in a cluster	Maximum number of documents in a cluster	Average number of documents in a cluster
R1	100	8	9	16	13
R2	300	8	30	57	38
R3	500	8	51	78	63
N1	200	10	15	25	20
N2	500	10	40	60	50
N3	1,000	10	80	120	100

K-means (KM) and bisecting K-means (BKM) clustering algorithm provided in the CLUTO software package [16] are employed in the experiment in order to verify the efficiency of the proposed text similarity measurement.

Given the true class label of each document, we use F-measure to evaluate various solutions. F-measure considers both the precision and the recall of the test to compute the score. Given a set of labeled documents belonging to i classes, we assume the clustering algorithm to partition them into j clusters. Let n be the size of the document set and n_i be the size of class i. Let n_j be the size of cluster j and n_{ij} be the number of documents belonging to both class i and cluster j. Then the recall, precision, and F-measure are as follows.

$$P(i,j)=\frac{n_{ij}}{n_j}, R(i,j)=\frac{n_{ij}}{n_i} \tag{14}$$

$$F(i,j)=\frac{2\times P(i,j)\times R(i,j)}{P(i,j)+R(i,j)} \tag{15}$$

where P is the precision of cluster j and R is the recall of cluster j.

The F-measure of all of the clusters is the sum of the F-measures of each class weighted by its size:

$$F=\sum_i \frac{n_i}{n}\max_j F(i,j) \tag{16}$$

An F-measure reaches its best value at 1 and worst at 0.

The different values of s from 30 to 100 are first selected in the singular value decomposition and the corresponding values of D are calculated according to the formula (5). The value of D determines the approximation degree between the original matrix A and the matrix A_s. The fitting results between s and D on the R3 and N2 datasets are shown in Fig. 2. It can be seen from the figure, the difference between the approximate matrix and the original matrix gradually becomes small when the value of s is increased. The value of D is the smallest if and only if $s\approx\min(m, n)$. The larger the value of s is, the more complete the semantic information remains, with the higher dimensionality of the corresponding text representation model and more noises however. Therefore, the dimensionality reduction and the noises removal should be premised on preserving the original semantics.

The first 300 singular values are extracted from the singular value matrix on R3 dataset. The changes of the singular values are shown in Fig. 3 when the value of s is gradually increased. It can be seen from the figure the decline rate of the singular values remained on the whole unchanged and the changes of the singular values occur in a very small interval after the turning point s≈50. We draw the following conclusion: the approximation matrix basically preserves the semantic information of the original document when the space dimensionality s=50. Therefore, the optimal value of s is selected to be 50.

The optimal relevance threshold σ between words is confirmed in the next experiment. Fig. 4 demonstrates the F-measures of different relevance thresholds σ=0.6–0.85 in the use of BKM clustering algorithm.

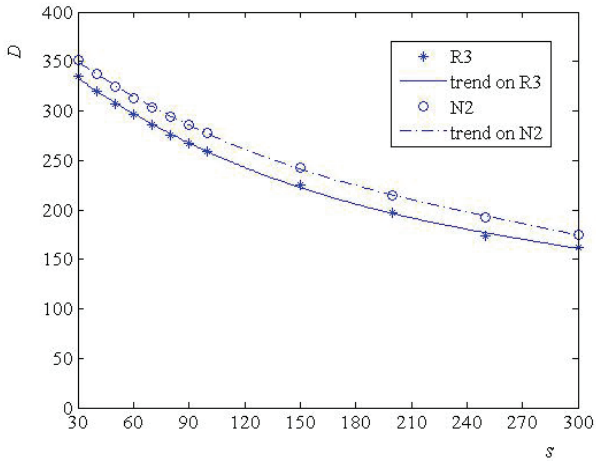

Fig. 2. The relationship between *s* and *D*.

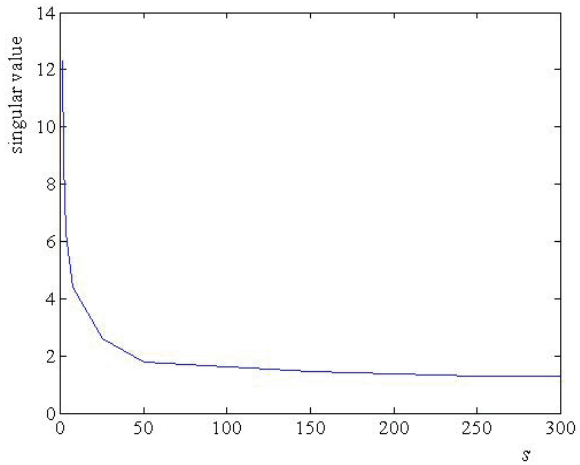

Fig. 3. The relationship between *s* and the singular values.

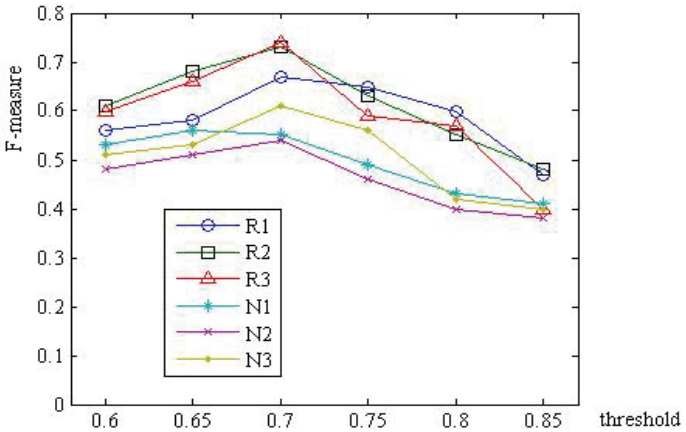

Fig. 4. The clustering results of the different relevance thresholds between words.

We can see from this figure that F-measure tends large when the relevance threshold between words is increased. The reason is that the discrimination between documents is bigger and the clustering effect is getting better with the improvement of the relevance threshold between words. The relevance threshold σ=0.7 reaches the best clustering effect. From the above figure, we can also see that F-measure declines rapidly after the relevance threshold is larger than 0.7. The reason is that the proportion of words whose maximum semantic relevance over 0.7 in the selected datasets is small. The guiding effect of the relevance weighting factor between vectors is declined, thus the overall F-measure is reduced.

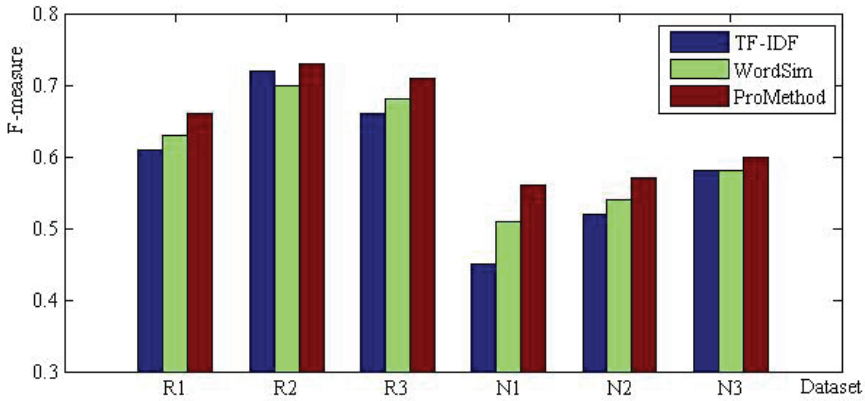

Fig. 5. The clustering results of different methods using KM algorithm.

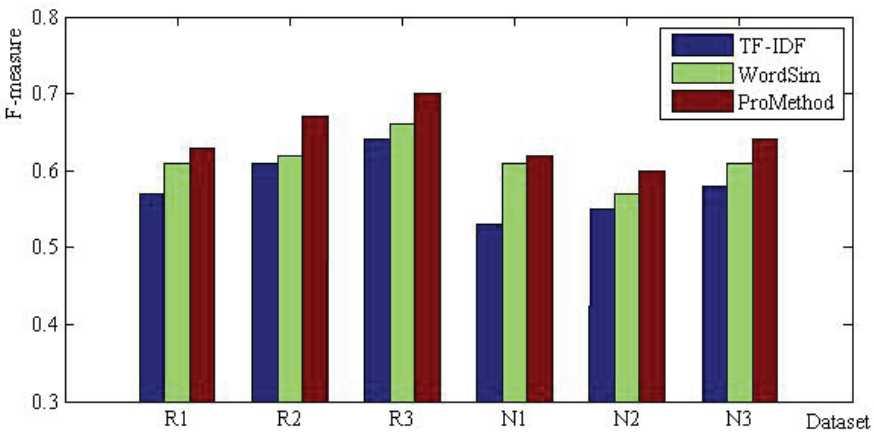

Fig. 6. The clustering results of different methods using BKM algorithm.

The proposed method is compared clustering competence with the traditional TF-IDF method and the WordSim method proposed by Hotho et al. [12]. The relevance threshold between words σ=0.7 is used in the experiments. The WordSim method uses WordNet to integrate background knowledge into the text document representation. The clustering results of the three methods using KM and BKM algorithms are shown in Figs. 5 and 6. As shown, our method performs better than others. The results demonstrate that the semantic relevance calculation between words and the weighted position information can effectively improve the accuracy of text similarity measurement.

4. Conclusions

We have presented a novel text similarity measurement method in this paper. Text representation model is performed singular value decomposition and the optimal number of the singular values is selected. Semantic relevance between words is analyzed and the semantic similarity between document vectors is defined. Experimental results have shown that the proposed method not only overcomes the curse of dimensionality and noises but also works more effectively than TF-IDF and WordSim method. In future work, we will introduce the semantic and structural information of text to further improve the accuracy of text similarity measurement.

References

[1] N. K. Nagwani, "A comment on "a similarity measure for text classification and clustering"," *IEEE Transactions on Knowledge and Data Engineering,* vol. 27, no. 1, pp. 2589-2590, 2015.

[2] A. Awajan, "Semantic similarity based approach for reducing Arabic texts dimensionality," *International Journal of Speech Technology*, vol. 19, no. 2, pp. 191-201, 2016.

[3] L. Xu, S. Sun and Q. Wang, "Text similarity algorithm based on semantic vector space model," in *Proceedings of the 15th International Conference on Computer and Information Science*, Okayama, Japan, 2016, pp. 1-4.

[4] R. lonescu and M. Popescu, *Knowledge Transfer between Computer Vision and Text Mining: Similarity-Based Learning Approaches*. Cham: Springer, 2016.

[5] E. Blanco and D. Moldovan, "A semantic logic-based approach to determine textual similarity," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 23, no. 4, pp. 683-693, 2015.

[6] M. Shirakawa, K. Nakayama, T. Hara, and S. Nishio, "Wikipedia-based semantic similarity measurements for noisy short texts using extended naive Bayes," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 2, pp. 205-219, 2015.

[7] H. Z. Liu and P. F. Wang, "Accessing text semantic similarity using ontology," *Journal of Software*, vol. 9, no. 2, pp. 490-497, 2014.

[8] W. Song, C. H. Li, and S. C. Park, "Genetic algorithm for text clustering using ontology and evaluating the validity of various semantic similarity measures," *Expert Systems with Applications*, vol. 36, no. 5, pp. 9095-9104, 2009.

[9] Y. Wang and J. Hodges, "Document clustering with semantic analysis," in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, Kauia, HI, 2006, pp. 54-63.

[10] R. M. Aliguliyev, "A new sentence similarity measure and sentence based extractive technique for automatic text summarization," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7764-7772, 2009.

[11] L. Gang, C. Zheng and L. Zhang, "Text information retrieval based on concept semantic similarity," in *Proceedings of the 5th International Conference on Semantics, Knowledge and Grid*, Zhuhai, China, 2009, pp. 356-360.

[12] A. Hotho, S. Staab, and G. Stumme, "Ontologies improves text document clustering," in *Proceedings of the 3rd IEEE International Conference on Data Mining*, Melbourne, FL, 2003, pp. 541-544.

[13] R. J. Bellegarda, "Exploiting latent semantic information in statistical language modeling," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1279-1296, 2000.

[14] C. Buck and P. Koehn, "Quick and reliable document alignment via TF/IDF-weighted cosine distance," in *Proceedings of the 1st Conference on Machine Translation,* Berlin, Germany, 2016, pp. 672-678.

[15] A. Mirzal, "Clustering and latent semantic indexing aspects of the singular value decomposition," *International Journal of Information and Decision Sciences*, vol. 8, no. 1, pp. 53-72, 2016.

[16] G. Karypis, "CLUTO: a clustering toolkit," 2006 [Online]. Available: http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview.

Xu Li

She received her B.S. degree in computer science and technology from University of science and technology Anshan, China, in 2003, and her M.E. and Ph.D. degrees in computer application technology from Yanshan University, China, in 2006 and 2010, respectively. Her current research interests include natural language processing and machine learning.

Chunlong Yao

He received B.S. and M.S. degrees in computer application technology from Northeast Heavy Machinery Institute, Qiqihar, China, in 1994 and 1997, respectively. He received his Ph.D. degree in computer software and theory from Harbin Institute of Technology, Harbin, China, in 2005. His current research interests include database, data mining and intelligent information system.

Fenglong Fan

He received B.S. and M.S. degrees in computer and its application from Dalian Maritime University, China, in 1995 and 2002, respectively. His current research interests include database and intelligent information system.

Xiaoqiang Yu

He received B.S. degree in computer science and technology from Northeast Normal University, China, in 1997, and his M.S. degree in computer application technology from Dalian Maritime University in 2004. His current research interests include enterprise information system, genetic algorithm and data mining.