Keyword Extraction From Chinese Text Based On Multidimensional Weighted Features

YANG JIAN
Jiangnan Institute of Computing Technology
Wuxi, 214000, China
yj_jns@163.com

Journal of Digital Information Management

ABSTRACT: *This paper proposed to solve the problems of incomplete coverage and low accuracy in keyword extraction of Chinese text based on intrinsic feature of the Chinese language and an extraction method of multidimensional information weighted eigenvalues. This method combined theoretical analysis and experimental calculation to study the parts of speech, word position, word length, semantic similarity and word co-occurrence frequency in Chinese texts. By combining multidimensional data related to word frequency, word feature values, word similarity and word co-occurrence probability, we calculated that the weighted eigenvalues obtained by comparing precision rate, recall rate and F measure and concluded that the proposed method can give a better measure of the word accuracy than using word frequency or the basic eigenvalue methods alone. The conclusions obtained in this study provide reference values for keyword extraction and text mining.*

Subject Categories and Descriptors
I.2.7 [Artificial intelligence]: Natural Language Processing - Text analysis; **H.2.8 [Database Applications]:** Data mining;

General Terms: Chinese text mining, sentiment analysis

Keywords: Weighted Eigenvalue, Feature Extraction, Text Word Frequency, Keyword Extraction

Received: 24 December 2015, Revised 21 January 2016, Accepted 5 February 2016

1. Introduction

Extensive research on keyword extraction from Chinese text exists in many applications including text mining, information retrieval of text content and text content analysis. For example, Zhang and Chen [1] proposed a Chinese keyword extraction method through word frequency and successive word correlation based on term frequency–inverse document frequency (TF-IDF) and word relevance methods. Wang and Huai [2] detailed the way to extract keywords through Chinese semantics in the semantic-based keyword extraction algorithm. In text retrieval, Wang and Wang [3] suggested matching text by acquiring keywords in the proposed method of Chinese text retrieval based on the content. In the process of text mining, Matsuo [4] et al. employed the same keywords from different text to do text mining based on text word co-occurrence method for keyword extraction. However, while there is a wide range of needs for keyword extraction in China and other countries, traditional methods of extracting feature keywords are inadequate compounded by the big differences that exist between English and Chinese in terms of text structure, grammatical features, language habits and so on. Consequently it is very difficult to simply employ traditional methods to effectively extract keywords from Chinese texts. This paper attempts to solve the accuracy problem in Chinese keyword extraction based on the combination of special structures in Chinese text and related research techniques at home and abroad.

Keyword extractive technology based on eigenvalues [5] is widely used in information retrieval, data mining [6], machine learning, pattern recognition, artificial intelligence, statistics, computational linguistics, computer network technology, information science, network security, behavioral analysis and other fields. In the early 1960s, Salton and others proposed the classical Vector Space Model (VSM) [7], an algebraic model for representing text documents as vector operations in vector space with space similarity corresponding to semantic similarity [8], which

was successfully applied to the famous SMART text retrieval system. When the document is represented as a vector of the document space, the similarity among documents can be measured by calculating the similarity among vectors [1]. When conducting information matching, feature terms $T(T_1, T_2, T_n)$ and their weights W_i represent the target information. These feature items are used to evaluate the relevance of the unknown text to the target sample. The selection of the feature terms and weights is called the feature extraction of the target sample [9].

Traditional methods to extract feature keywords from a text usually start with obtaining the term frequency-inverse document frequency (TF-IDF) vector of the text through a sample [10], followed by the dimension-reduction of word frequency vector, the setting up of the keyword feature weighting function, assigning a weighted value to get new results and finally extracting the keywords through the weighted value to describe the text. However, the text feature factors extracted by this method are fairly ordinary, and do not effectively reveal the individuality and other personal aspects of the text. This lack affects the accuracy of the extracted keywords and related research based on keyword extraction. Therefore, it is necessary to find a technique that can increase the accuracy of keyword extraction.

2. Text Feature Extraction Using Weighted Vector Value

In our approach to keyword extraction, the basic eigenvalues of words are calculated according to part of speech, word position and word length after labeling the word segmentation and filtering stop words in the document. An eigenvalue that can better assess word criticality is obtained by combining the calculation of word frequency, word feature values, word similarities and word co-occurrence probabilities.

2.1 Annotation of Basic Word Information through Word Segmentation

The text document to be extracted is defined as D and the document set as DS. Word segmentation uses the high-performance Chinese word segmentation component "word segmentation" [11] that is distributed by Java. While inputting the sample DS, the result of the word segmentation is recorded and the corresponding order of each word and the document number K are traced.

When document *D* is surveyed, the number of words in the title is defined as *TW*. The total paragraphs are P_n (p_i represents the paragraph *i* of the article, where $i \in [1, P_n] \subset N$; p_1 represents the first paragraph and p_n represents the last paragraph). The total number of sentences of a paragraph is $p_i S_n$ ($p_i S_j$ represents the sentence *j* of the paragraph *i*). The total word count ofthe article after removal of the duplicates is Wf_n (wf_i: the number of times a word appeared in the article). After word segmentation, the original result set of word segmentation is W_0.

2.2 Basic Eigenvalues of Filtered Words and Computed Words

The original result set W_0 obtained from section 2.1 is filtered by the thesaurus of the stop words, W_1 and a new set of words is obtained after the removal of stop words from W_0, where stop words refer to the meaningless words contained in the article, such as auxiliaries and the like.

$$W_1 = W_0 - (W_0 \bigcap W') \qquad (1)$$

After processing the word set W_1 by going through each word in the word set, the original vector $V_0(x_i, y_j)$ is generated according to the word's offset, paragraph, sentence, the number of occurrences in the article, disabled or not and document number where: x_i represents a word, i is the longitudinal dimension, y_j represents the data stated above and j is the transverse dimension.

2.2.1 Part of Speech Eigenvalues

Basic word features include the word part of speech (POS) [2], word position [12] and word length [13]. POS is simply divided into noun, verb and adjective categories. The level of importance is defined in accordance with Wang and Huai [2] for different parts of speech. The weight of the noun is 0.8, the average of the noun-verb and adverbial verb is 0.5, and the average of all types of adjectives is about 0.4. The noun has much higher word weight than other POS. Moreover, due to the importance of personal names in an article, a noun is subdivided into personal names and non-names in this paper, with personal names assigned a value of 0.9, and non-names a value of 0.7. In addition, the weight of a verb was set to 0.5, and that of an adjective, 0.3. Word POS weights are set as wv_i, as follows:

$$WV = \{0.9, 0.7, 0.5, 0.3\}, \quad wv_i \in WV \qquad (2)$$

2.2.2 Word Position Eigenvalues

The word position often best reflects the key contents location of the article. In general, the article's title, subtitle, summary, first paragraph, last paragraph, first sentence, last sentence can accurately express the general idea of the article. The weight of word position wl_i indicates how important the word position is in the article, i.e. the importance of the same word varies with its position in the article. The importance of word position can be assessed according to the weight assignment for word position as per Gupta and Lehal [12]. While keywords of significance have a high frequency of appearance in the first section and last section of the article and in other locations, some of them appear in the second paragraph of the article and because keywords could be in other paragraphs, different weights are assigned to each paragraph [14]. Here, the different weights are assigned to the

different positions, the weight of the titles and sub-titles is 0.2 and the one of the text is 0.8. The weights of sentences in text are as shown in Table 1.

The weight allocation of the rest of the text is: the first and last sentence of each paragraph is $0.1/(P_n\text{-}3)\times 0.45$, and for the rest part of the paragraph: $0.1/(P_n\text{-}3)\times 0.1$. *WL* represents the collection of all word weight in the article, *WL*={0.2, 0.1, 0.085, 0.05, 0.045, 0.03, 0.01}, and $wl_l \in WL$

Paragraph	First sentence	Last sentence	Others	Total
First paragraph	0.100	0.100	0.050	0.250
Second paragraph	0.085	0.085	0.030	0.200
Last paragraph	0.100	0.100	0.050	0.250
Others	0.045	0.045	0.010	0.100

Table 1. Sentence weight of each paragraph in the text.

2.2.3 Word length eigenvalues

The words with larger lengths are often content-oriented. Thus, longer terms are assigned greater weight during the weight assignment process for word length. Word length is defined as l, and the weight of the individual word as follows:

$$wd_i = \frac{l}{l+1} \tag{3}$$

According to the respective weight of part of speech, word position, word length, the basic eigenvalues of an individual word is:

$$wk_i = wv_i \times wl_i \times wd_i \tag{4}$$

2.3 Word Frequency Calculated Based on TF-IDF Algorithm

TF represents a positive word frequency, indicating the word's ability to describe the content of the document. IDF represents the inverse document frequency, signifying the word's ability to distinguish the document from other documents. Here is the equation of TF-IDF:

$$tfidf(t_i, d_i) = tf(t_i, d_i) \times \log\left(\frac{K}{K(t_i)+1}\right) \tag{5}$$

Where $K(t_i)$ is the number of documents that contain this *word and K is the total number of documents. After tfidf* (t_i, d_i) is obtained, the weight of *tfidf* (t_i, d_i) is iteratively modified by the word frequency in the document, thus the change of the word characteristics is shown.

A new set of W_2 is obtained through the frequency calculation and processing text set W_1, representing the result of W_1, after de-duplication. The corresponding vector V_1, which is the dimension-reduction of V_0, is generated.

After *tfidf* (t_i, d_i) is calculated, wf_i is normalized to make the weight between [0, 1] as follows:

$$F_i(d) = \frac{tfidf(t_i, d_i)}{\sqrt{\sum_{i=1}^{n} tfidf^2(t_i, d_i)}} \tag{6}$$

The weight (4) calculated from section 2.1 is substituted into the calculation of the weight of word frequency to obtain this:

$$F_i(d) = \frac{tfidf(t_i, d_i)}{\sqrt{\sum_{i=1}^{n} tfidf^2(t_i, d_i)}} \times \sum_{i=1}^{n} \frac{wk_i}{wf_i} \tag{7}$$

In the above equation, n represents the number of elements of the document set W_2.

2.4 Calculation of Semantic Similarity

Word similarity in the document [16-18] is usually described by the semantic distance [19] between words and a common method is to calculate the semantic distance between two words by a synonym dictionary with *HIT IR-Lab Tongyici Cilin* (Extended) [20] used in this paper. In the instruction of the extended version, code c_i is described as $tc_i = X_{i1}\ X_{i2}\ X_{i3}\ X_{i4}\ X_{i5}\ F_i$, where the 5-level codes describe the categories, classes, subclasses, word group and atomic word group. 1-bit flag is "=", "#" or "@", where "=" means synonymous; "#" means similar, relevant words; and "@" indicates self-enclosed, independent words that have no synonyms or related words in the dictionary.

According to the definition of semantic distance, coding distance and semantic similarity for *HIT IR-Lab Tongyici Cilin* referred to Wang and Huai [2], and the conceptual hierarchies stated in Oliva et al [21], the related definitions in this article are as follows:

Definition 1 Semantic Distance and Coding Distance: The number of code for word w_i is m, $c_{11}, c_{12}, \ldots c_{1m}$ respectively, and the number of code for word w_2 is n, $c_{11}, c_{12}, \ldots c_{1m}$ respectively. Then the semantic distance *dis* (w_1, w_2) and coding distance *dis* (c_1, c_2) of words and are defined as:

$$\mathrm{dis}(w_1, w_2) = \min_{i=1,2.....m;\ j=1,2....n} \mathrm{dis}(c_{1i}, c_{2j}) \tag{8}$$

$$dis(c_1, c_2) = \begin{cases} 0 & (a) \\ weights[5] \times init\_dis & (b) \\ weights[i-1] \times init\_dis & (c) \end{cases} \tag{9}$$

(a), (b) conditions in the above equation (10) are $c_1 = c_2$ and $F_1 = F_2 \neq$ "#"; (c) condition is c_1 and c_2 are coded differently starting layer i . *init_dis* is the initial distance value

and it is 10 in this paper. Weight number array weights equals $[t_1, t_2, t_3, t_4, t_5, t_F]$, where $t_1 > t_2 > t_3 > t_4 > t_5 > t_F$. Weight is defined here as $[1.0, 0.5, 0.25, 0.125, 0.06, 0.03]$

Definition 2: Semantic Similarity: the similarity $sim(w_1, w_2)$ of the word w_1 and word w_2 is:

$$sim(w_1, w_2) = \frac{\alpha}{dis(w_1, w_2) + \alpha} \quad (10)$$

Where α as an adjustable parameter indicates the distance between words with similarity of 0.5, in this paper α = 5.

Substituting the above equation (11) of weight of semantic similarity into the calculation of the weight of the words, this can be obtained:

$$F_i(d) = F_i(d) \times \frac{1}{Wf_n} \times \sum_{j=1}^{Wf_n} sim(w_i, w_j) \quad (11)$$

2.5 Calculation of Word Co-Occurrence Frequency

In the Chinese text, the meaning of a sentence is expressed by the relations between the words. The word relations are simply shown by the word co-occurrence in the same sentence. The word co-occurrence [22] refers to the relations between words in the same sentence and in the same document.

Let the word frequency of word w_i and word w_j be wf_i and wf_j respectively, and the frequency of occurrence in the same sentence as wf_{ij}. There is no repeat count inside the sentence, and the co-occurrence probability is:

$$p_{ij} = \frac{wf_{ij}}{wf_{ii} + wf_{jj} - wf_{ij}} = \frac{wf_{ij}}{wf_i + wf_j - wf_{ij}} \quad (12)$$

Through the co-occurrence frequency calculation of word by word in the document, the co-occurrence probability matrix V_p can be obtained. It is a Wf_n order symmetric matrix, and Wf_n is the feature number of the document. Combining with the weight equation (12) in section 2.4, the following weight equation can be obtained:

$$F_i(d) = F_i(d) \times \sum_{j=1}^{Wf_n} pij \quad (13)$$

2.6 Extraction of Feature Keywords from Text

Word feature, word co-occurrence probability in a sentence, word interconnection semantics and word frequency of overall article are all integrated into the extraction of keywords. Through these methods, a more accurate technique to extract keywords is proposed. The steps of calculation are as follows:

Input sample document set DS and test the document D.

Output feature keywords of the testing document D.

(1) Process a sample document set DS.

(2) Perform Chinese word segmentation to the document D, record the feature of relevant words and word frequency, recognize part of speech of the words, and obtain relevant word set.

(3) Filter useless words through the thesaurus, label part of speech including adjectives, verbs, and nouns (people and places) to get word collection, and calculate word weight according to the definition of word POS, word position and word length.

(4) Calculate word frequency to get weighted value of word frequency.

(5) Calculate word similarity in the word set and annotating weights.

(6) Calculate weights of word co-occurrence frequency.

(7) By calculating equations (1) - (13), the eigenvalue vector V and the word collection W composed of eigenvalues are obtained. V and W are used to describe the target document.

(8) Sort words according to the weights of the word collection W, and output M feature keywords with largest eigenvalues, where M is as follows:

$$M = \frac{TW + \sum_{i=1}^{n} TW_i}{n+1} \quad (14)$$

M is the average number of words in the title of all the documents, with a range from 3 to 8. When M is less than 3, it is set to 3. When is larger than 8, it is set to 8.

3. Results

3.1 Data Collection

In this study, the sample data is from the official website of People's Daily http://www.people.com.cn/ with only text content retained for the selected article. Of a total of 323 articles, 213 articles are labeled for sample document set DS and the other 110 ones are labeled D for the test document set. The sample document set covers a variety of topics, including social, entertainment, technology and finance. They are evenly distributed to avoid skewed data due to a single document category.

3.2 Comparison of Results

The experimental results are based on the comparison of the manual keyword extraction, word frequency TF-IDF algorithm and the weighted eigenvalue method. When manually extracting keywords, the number of keywords for each article references the number of words in the title, ranging from 3 to 8 keywords. The extracted keywords are evaluated based on (Precision rate), (Recall rate) and (F-measure). The definitions of the three key

indicators are as follows:

$$P = \frac{A \cap B}{B}, R = \frac{A \cap B}{A} \quad (15)$$

$$F = \frac{2P \times R}{P + R} \quad (16)$$

Wherein, A is the manually selected keyword set, B is the keyword set extracted by various methods and P is the ratio of manually selected keywords contained in the keyword set extracted by algorithms. R is the ratio between the number of keywords extracted by the algorithm and the manually selected keywords. F is the harmonic mean of recall rate and precision rate, reflecting the accuracy of extracted keywords. The results are shown in Table 2.

Figure 1 shows how precision rate changes with the number of keywords for TF-IDF, the basic eigenvalue and weighted eigenvalue methods. The precision rate of weighted eigenvalues and TF-IDF displays an increasing trend, while the precision rate of basic eigenvalues reaches maximum when the keyword number is 5. In addition, the precision rate of weighted eigenvalues is more stable when the number of keywords is from 4 to 8.

Number of keywords	Extraction method	Precision rate	Recall rate	F-measure
3	TF-IDF	0.423	0.266	0.327
	BFV	0.567	0.441	0.497
	AFV	0.702	0.572	0.631
4	TF-IDF	0.399	0.298	0.342
	BFV	0.531	0.475	0.502
	AFV	0.638	0.646	0.642
5	TF-IDF	0.361	0.325	0.343
	BFV	0.499	0.522	0.511
	AFV	0.673	0.647	0.660
6	TF-IDF	0.328	0.354	0.341
	BFV	0.469	0.504	0.486
	AFV	0.637	0.655	0.646
7	TF-IDF	0.312	0.357	0.333
	BFV	0.448	0.486	0.467
	AFV	0.567	0.666	0.613
8	TF-IDF	0.292	0.362	0.324
	BFV	0.382	0.447	0.412
	AFV	0.543	0.644	0.590

Table 2. Comparison of experimental data for each method

BFV stands for basic eigenvalues and AFV stands for weighted eigenvalues.

Figure 2 shows how the recall rate changes with the number of keywords for TF-IDF, the basic eigenvalue and weighted eigenvalue methods. According to the graph, the recall rate decreases with the increase in the number of keywords. However, the recall rate of weighted eigenvalues is higher than the other two methods.

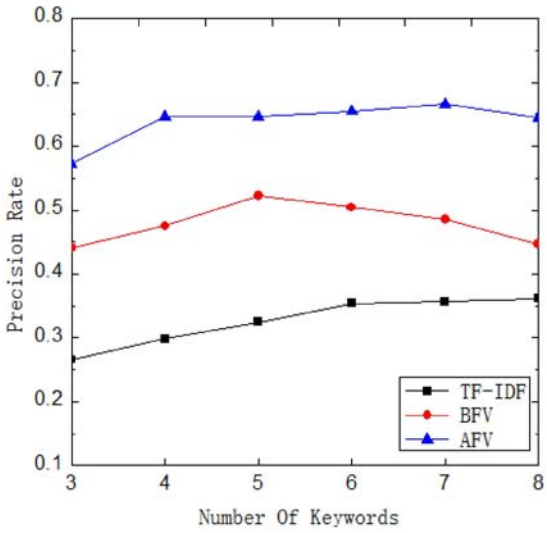

Figure 1. Precision rate changes with the number of keywords

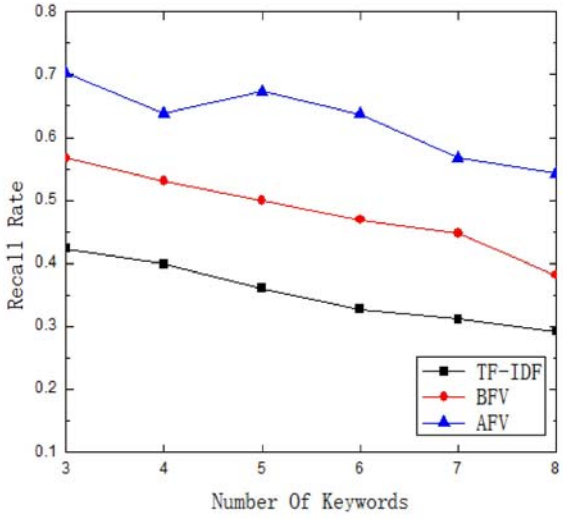

Figure 2. Recall rate changes with the number of keywords

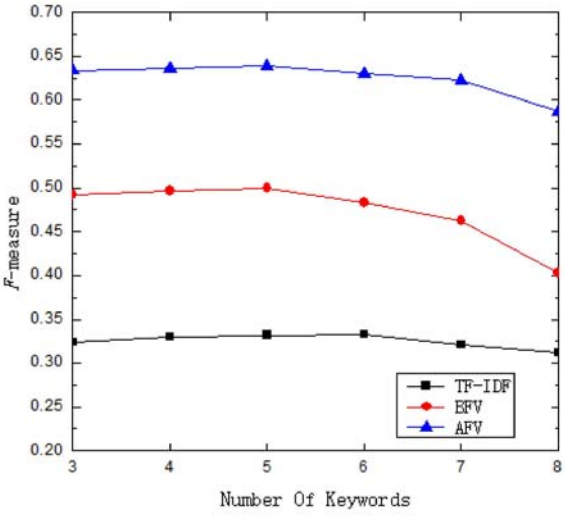

Figure 3. F-measure changes with the number of keywords

Figure 3 shows how F-measure changes with the number of keywords for TF-IDF, the basic eigenvalue and weighted eigenvalue methods. It is the comprehensive reflection of recall rate and precision rate, a better assessment of the accuracy of the data. As it can be seen, the F-measure of weighted eigenvalues is much higher than the other two methods.

According to Figure 3, the F-measure of the weighted eigenvalue method is significantly higher than the other two methods, indicating higher relevance and effectiveness to extraction the keywords from documents. It can also be seen that F-measure reaches the peak value when the number of keywords is 5 for all three methods. This is because the weight of each keyword is relatively larger with fewer keywords and the weight relatively smaller with more keywords. As shown in Figure 3, the F-measure of the weighted eigenvalue method used in this study shows no significant change for all numbers, while only a small drop in the magnitude of the curve can be observed as the number of keywords increases from 6 to 8.

4. Conclusion

In order to improve the accuracy of keyword extraction from Chinese text and compensate for the shortcomings of the traditional word frequency method, this paper proposes a weighted eigenvalues keyword extraction method based on the combination of multidimensional information. By combining the word frequency and the weighted eigenvalues method the accuracy of keyword extraction is improved through the quantization of the weighted eigenvalues. In the experiment, the recall rate, precision rate and F-measure of the extracted keywords using the traditional word frequency method, basic eigenvalue method and the weighted eigenvalue method proposed in this paper are statistically tested and horizontally compared with the following conclusions drawn from the experimental results:

(1) The accuracy of the keywords extracted by using the weighted eigenvalue method which combines word frequency, word position and word length as information, is significantly higher than using the word frequency or basic eigenvalue method alone.

(2) The precision rate of the weighted eigenvalue method proposed in this paper is twice as high as that of the word frequency method and 1.5 times of the basic eigenvalue method. The recall rate is 1.6 times the word frequency method and 1.2 times the basic eigenvalue method. The F-measure is about 1.9 times that of the word frequency method and 1.3 times more than the basic eigenvalue method. For the precision rate, recall rate and F-measure, the weighted eigenvalue method is significantly higher than the word frequency and the basic eigenvalue method.

Based on the comparison of the numerical difference and the shape of the numeric curve of the three methods discussed, it was been shown that the weighted eigenvalue method can effectively improve the precision rate and accuracy rate of keyword extraction. The following problems need further study: an in-depth study of semantic correlations in Chinese text; exploration of the combination of experiences in text mining and machine learning and research on how artificial intelligence can help in text mining.

References

[1] Zhang, K., Chen, H., Liu, S. (2015).Application of vector similarity method in multi-plan optimization. *In: IEEE International Conference on Information & Automation,* p. 335-338. Lijiang, Yunnan, China: IEEE, August. 2015.

[2] Wang, LX., Huai, XY. (2012). Semantic-based Keyword Extraction Algorithm for Chinese Text. *Computer Engineering*, 38 (01) 1-4.

[3] Wang, J., Wang, XF., (2012). Chinese Text Retrieval Method Based on Content. *Computer Systems and Applications,* 21 (9) 214-216.

[4] Matsuo, Y., Ishizuka, M. (2004). Keyword Extraction From a Single Document Using Word Co-occurrence Statistical Information. *International Journal on Artifical Intelligence Tools*, 13 (1) 157-169.

[5] Rose, S., Engel, D., Cramer, N. (2010). Automatic keyword extraction from individual documents. Automatic keyword extraction from individual documents. *Text Mining*, John Wiley & Sons, Ltd, p 1-20.

[6] Zhong, N., Li, Y., Wu, S T. (2012). Effective pattern discovery for text mining. *Knowledge and Data Engineering*, 24 (1) 30-44.

[7] Arguello J. (2013). Vector Space Model. http://ils.unc.edu/ courses/ 2013_fall/inls509_001/lectures/06-VectorSpaceModel.pdf

[8] Bär, D., Biemann, C., Gurevych, I. (2012). Ukp: Computing semantic textual similarity by combining multiple content similarity measures. *In: International workshop on First Joint Conference on Lexical and Computational Semantics,* p. 435-440. Montreal, Canada: Association for Computational Linguistics, June 2012.

[9] Ferjani, F., Elloumi, S., Jaoua, A. (2012). Formal context coverage based on isolated labels: An efficient solution for text feature extraction. *Information Sciences*, 188(188) 198-214.

[10] Yatsko V., Dixit, S., Agrawal A J. (2013). TF*IDF Revisited. *Intelligence*, 16 (4) 2-3.

[11] Mo, JW, Zheng, Y, Shou, ZY, Zhang, SL. (2013). Improved Chinese word segmentation method based on dictionary. *Computer Engineering and Design*, 34 (5) 1802-1807.

[12] Gupta, V, Lehal, G S. (2010). A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence*, 2 (3) 258-268.

[13] Li, JY., Li, PF., Zhu, QM. (2011). An Improved TFIDF-Based Approach to Extract Key Words From Web Pages. *Computer Applications and Software*, 28 (5) 25-27.

[14] Rodriguez, J., Cortes, P. (2012). Weighting Factor Design. *Predictive Control of Power Converters and Electrical Drives*, John Wiley & Sons, Ltd, p. 163-176.

[15] Zhang, Y., Zhang, XD. (2011). Improved Feature Weight Algorithm. *Computer Engineering*, 37 (5) 210-212.

[16] Liu, Q, Gu, X. (2010). Study on HowNet-based word similarity algorithm. *Journal of Chinese Information Processing*, 24 (6) 31-36.

[17] Zhang, Y Y., Xie, Q, Ding, Q L. (2010). Chinese Keyword Extraction Algorithm Based on Synonym Chains. *Computer Engineering*, 36 (19) 93-95.

[18] Sheng, Z C., Tao, X P. (2011). Semantic Similarity Computing Method Based on Wikipedia. *Computer Engineering*, 37 (7) 193-195.

[19] Oliva, J., Serrano, J I., del Castillom, M D. (2011). SyMSS: A syntax-based measure for short-text semantic similarity. *Data & Knowledge Engineering*, 70 (4) 390-405.

[20] Zhang, GD., Zhang, HX. (2012). Classification algorithm based on semantics and text feature weighting. *Application Research of Computers*, 29 (12) 4476-4478.

[21] Oliva, J., Serrano, J I., del Castillo, M D. (2011). SyMSS: A syntax-based measure for short-text semantic similarity. *Data & Knowledge Engineering*, 70 (4) 390-405.

[22] Bullinaria, J A., Levy, J P. (2012). Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD. *Behavior research methods*, 44 (3) 890-907.