ATLANTIS PRESS

Advances in Intelligent Systems Research, volume 133

2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE2016)

Optimized TF-IDF Algorithm with the Adaptive Weight of Position of Word

Jie Chen*, Cai Chen and Yi Liang

Faculty of Information Technology, Beijing University of Technology Beijing, China

*Corresponding author

***Abstract*—The classical TF-IDF algorithm only considers the weight of the term frequency and the inverse document frequency, without considering the weights of other feature of word. After the author analyzing summary of Chinese expression habits, an adaptive weight of position of word algorithm based on TF-IDF is proposed in this paper, which can be called TF-IDF-AP algorithm. The TF-IDF-AP algorithm can dynamically determine the weight of position of word according to the position of word. This paper introduced the vector space model (VSM) and designed comparative experiment under the scene of Chinese document clustering. The results show that the F-measure of TF-IDF-AP algorithm has been improved by 12.9% comparing with the classical TF-IDF algorithm.**

Keywords-text feature extraction; adaptive weight; weight of position; Term Frequency-Inverse Document Frequency(TF-IDF)

I. INTRODUCTION

With the rapid development of the Internet, real-time online automatic text categorization technology with high-efficiency, as an important part of text mining, is helpful to improve the efficiency of text information retrieval based on content, there are significant practical means and huge application prospect for improving the efficient, real-time and accurate usage of information. And the research on feature reduction which is the core of automatic text categorization technology has become one of the hot areas of text mining. The key to information classification is extract theme features of huge amounts of information effectively.

And key word on the Internet not only plays an important role on public opinion analysis, but also has a very high commercial value. So in the field of massive social network data analysis, it is an urgent task to extract the valuable commercial word quickly and precisely in a cloud computing environment.

Traditional extraction algorithm only considers the relationship between the key words and the number of it appearing in texts, and ignores the key words' distribution in a certain category, so that decreased the accuracy of word extraction.

The major contributions of this work are follows:

a) We have analyzed the expressions of Chinese habits, and find a characteristic situation that the position of the key words would perform the theme of these articles accurately.

b) In the traditional methods, single word frequency weighting factor weights and the problem of insufficient ability of text characteristic, the traditional feature vector doesn't consider the semantic and context of terms .

c) In view of the above problems, we proposed the adaptive weight for the position of the key word and introduced the Vector Space Model follows the traditional TF - IDF algorithm.

II. RELATED WORK

A. Current Status of the Research

In the document categorization task, many statistical classification methods and machine learning techniques have been used, including VSM is a traditional and effective document categorization model and it becomes a general method to sort, retrieve and categorize documents since it was proposed by Salton et al.[1] and Turney[2] surveys the use of VSMs for semantic processing of text, Naive Bayes algorithms[3], decision trees[4], Term Frequency and Inverse Documentation Frequency algorithm(TF-IDF)[5].Usually, the methods for extract theme features mainly through the extraction of key words and to give these words from papers in suitable weighting. In recent years, the key weight calculation method can be divided into three parts: the weight of the Boolean calculation and the method which weight calculation based on word frequency [6, 7].

Among these methods, which is used commonly is that based on words Frequency. Salton and Clement[5] proposed TF-IDF algorithm and demonstrates the effectiveness of in the field of information retrieval[8]. Although the TF-IDF algorithm has been used in all kinds of areas, many researchers found that TF - IDF algorithm only considers the weight of the frequency of key words and the weight of frequency of reverse text, without considering the other weights of key words, if only use the TF - IDF algorithm to extract key words.

There is a deviation will inevitably cause the results of the information classification, if we consider TF-IDF algorithm separately, it also has the limitations. The scientific researchers to try to do a lot of research work to improve the TF-IDF algorithm:

Hua-Meng[9] proposed the key distribution and information entropy weights follows the traditional TF - IDF algorithm. Literature [10] suggested a similarity measurement, which is based on TF-IDF method, and analyzes similarity between important terms in text documents. Literature [11] improved term weighting algorithm named Document Triage-Term Frequency-Inverse Document Frequency (DT-TF-IDF)

was proposed by introducing document scores and users annotation to TF-IDF and giving a greater weight to annotated term. In [9, 10, and 11] will greatly increase the algorithm time complexity, some of them also need analysis tools with the help of third party tools.

At the same time, there are also some researchers[12] proposes methods for computing weight of text characteristic items based on multiple factors weighting . The introduction of weight of words' position, but it is to endow them with different positions of key through a fixed percentage of the weight, lack of adaptability.

In conclusion, through summarizing the predecessors' research experience, this paper proposes an Adaptive weight based on word's position which follows the TF-IDF algorithm, namely TF-IDF-AP algorithm (the Adaptive weight of the Position of the words), the algorithm's advantages: it is neither need third party analysis tools, nor significantly increase the time complexity of the algorithm, as well as good adaptability.

B. The Classical TF-IDF Algorithm

TF-IDF algorithm represents the importance of the word, Term Frequency (TF) and Inverse Document Frequency (IDF) are associated with the word importance.

TF represents the number of times of a word appearing in a document. The importance of word ti in a document can be expressed as:

$$TF(word) = \frac{Count(word)}{\sum_{i=0}^{n} Count(word_i)} \tag{1}$$

In formula (1), $Count(word)$ presents the number of occurrences of the word in the document. The denominator is the sum of the number of occurrences about all the words in the documents.

IDF is a measure of the word ability to distinguish between categories. IDF of a word can be obtained by the total number of documents that contain the word divided by the total number of documents after the quotient logarithmic.

$$IDF(word) = log\left(\frac{Count(docs)}{Count(word, docs)} + 0.01\right) \tag{2}$$

In formula (2), $Count(docs)$ is the total number of documents. The denominator is the total number of documents that contain the word. Calculated inverse document frequency indicates that the number of documents that contain a word fewer, $Count(word, docs)$ smaller, thus indicating that the word has a very good class discriminative.

$$Weight(word) = TF(word) * IDF(word) \tag{3}$$

Formula (1)'s result is multiplied by the result of Formula (2) to obtain the result of Formula (3), which represents the weight of words.

III. THE TF-IDF-AP ALGORITHM

In the Chinese expression, we often propose the topic of the document in the title, abstract, the first paragraph, and restate the topic at the end of the document, therefore these words which at the start or end of the document can describe the topic more than other words of the document. If you can effectively extract these words which at the start or end of the document, you will have great help for the determination of the document topic.

If the context of each word is analyzed in detail, the algorithm's time efficiency will increase dramatically, so this paper only considered the position of first occurrence of a key word and the position of last occurrence of a keyword in a document.

The formula for the position of first occurrence of a key word can be described as:

$$FirstPosition(word) = \frac{FPBeforeCount(word)+1}{\sum_{i=0}^{n} Count(word_i) - FPBeforeCount(word)} \tag{4}$$

In formula (4), the meaning of $FPBeforeCount(word)$ is the number of all the words that appear before the position (not including this word) in the first appearance of the key word. $\sum_{i=0}^{n} Count(word_i)$ is the total number of words in the document.

The formula for the position of last occurrence of a key word can be described as:

$$LastPosition(word) = \frac{LPAfterCount(word)+1}{\sum_{i=0}^{n} Count(word_i) - LPAfterCount(word)} \tag{5}$$

In formula (5), the meaning of $LPBeforeCount(word)$ is the number of all the words that appear after the position (not including this word) in the last appearance of the key word. The denominator is the total number of words in the document.

In summary, the formula for the adaptive weight of key word's position can be described:

$$PositionWeight(word) = \frac{1}{FirstPosition(word) + LastPosition(word)} \tag{6}$$

The meaning of formula (6) is that the more front position of the key word which appears for the first time is, i.e. the value of *FirstPosition(word)* smaller is, the more likely it is the word in the document title, abstract or first paragraph, and the more after position of the key word which appears for the last time is, i.e. the value of *LastPosition(word)* smaller is, the more likely it is the word in the document conclusion or last paragraph.

Therefore, the formula for the weight of the key word is optimized, which can be described as:

$$Weight(word, doc) = \frac{TF * IDF * PositionWeight}{\sqrt{\sum_{word \in doc} [TF * IDF * PositionWeight]^2}} \quad (7)$$

In formula (7), the TF is the occurrence frequency of key word, the IDF is the inverse document frequency of key word and the *PositionWeight* is the adaptive weight of key word's position.

IV. EXPERIMENT

A. Experimental Environment

Experiments used C++ as the implementation language of the algorithm, used Studio Visual 2013 as the development environment. The simplified version of Chinese corpus of Sogou lab was used as the experimental data, which can be called "SogouT"[13]. The corpus contained nine categories of topics, each category has 1990 documents, a total of 17910 documents, these topics include finance, IT, health, sports, tourism, education, recruitment, culture and military.

Vector space model was introduced in the experiment, and each document was replaced by a high dimensional vector. We designed comparative experiment which is the TF-IDF-AP algorithm and the classical TF-IDF algorithm.

B. Evaluating Indicator

Three evaluation indicators of text clustering were introduced in the experiment, they are the Recall(R), precision(P) and F-measure(F1) value[14].

Recall is that the ratio of the number of documents associated with a topic in the cluster to the number of documents in this topic.

Precision is that the ratio of the number of documents associated with a topic in the cluster to the total number of documents in the cluster.

F-measure value is the geometric average of Recall and Precision, which is a comprehensive evaluation indicator of the results of the text clustering. The formula for F-measure value can be described as:

$$F - measure = \frac{2RP}{R + P} \quad (8)$$

C. Experimental Flow

There are three stages in the experiment: preprocessing, feature extraction and build vectors, clustering.

Experimental flow chart shown in Figure I.

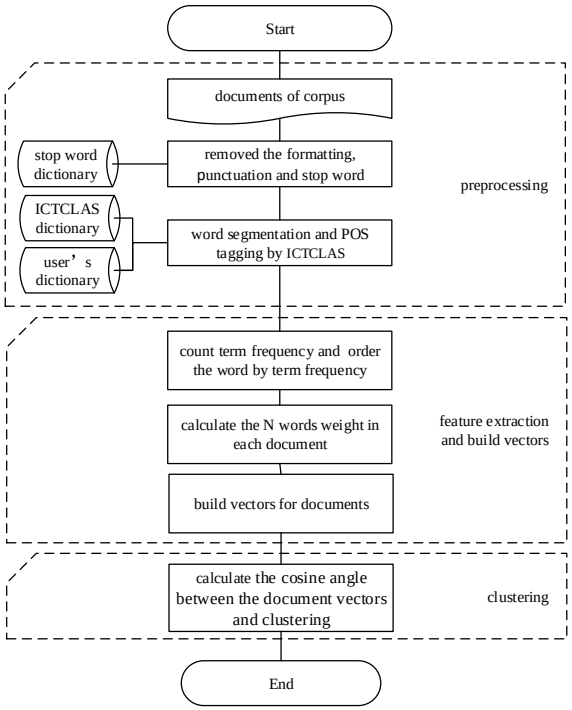

FIGURE I. EXPERIMENTAL FLOW

In the stage of pretreatment, we removed the formatting, punctuation and stop word from the document, and experiments of word segmentation and POS tagging are done by ICTCLAS(Institute of Computing Technology Chinese Lexical Analysis System)[15]. Through the preprocessing , the documents of corpus has become a collection of words tagged POS.

In the stage of feature extraction and building vectors, we count term frequency and accord to the term frequency to order the word. We selected the first N words with the highest frequency and build vectors which taking the N words as the dimension and taking the normalized product of TF, IDF and AP as the weight for documents. Through this stage, each document in the corpus are no longer used text or words to describe, replaced by a vector with N dimensional. The documents of corpus has become a collection of vector.

In the stage of clustering, we used the CURE clustering algorithm and used the cosine angle between the document vectors as the similarity of the document. Finally, the similar documents will be clustered into a cluster.

V. RESULTS AND ANALYSIS

Based on the "SogouT" data set size considerations, this paper's experiments in the N value of 180, as a result of 20 feature words can described clearly the subject information of a king of document.

The recall ratio of TF-IDF algorithm and TF-IDF-AP algorithm is shown in Table I:

TABLE I. RECALL RATIO

Topic	TF-IDF	TF-IDF-AP
finance	72.3%	79.1%
IT	68.4%	78.2%
health	71.8%	79.9%
sports	76.9%	83.8%
tourism	78.6%	84.4%
education	77.3%	83.7%
recruitment	78.6%	85.4%
culture	60.3%	75.9%
military	82.5%	88.3%

The precision of TF-IDF algorithm and TF-IDF- AP algorithm is shown in Table II:

TABLE II. PRECISION

Topic	TF-IDF	TF-IDF-AP
finance	78.6%	90.7%
IT	72.5%	88.3%
health	76.2%	89.7%
sports	81.6%	92.4%
tourism	82.7%	93.2%
education	85.8%	92.9%
recruitment	86.4%	93.6%
culture	75.2%	88.1%
military	69.4%	86.5%

The F-measure value of TF-IDF algorithm and TF-IDF- AP algorithm is shown in Table III:

TABLE III. F-MEASURE VALUE

Topic	TF-IDF	TF-IDF-AP
finance	75.3%	84.5%
IT	70.4%	82.9%
health	73.9%	84.5%
sports	79.2%	87.9%
tourism	80.6%	88.6%
education	81.3%	88.1%
recruitment	82.3%	89.3%
culture	66.9%	81.5%
military	75.4%	87.4%

As can be seen from the table I, the recall rate of the TF-IDF- AP algorithm which proposed in this paper increased by 10.8%, compared with the classic TF-IDF algorithm. As can be seen from the table II, the precision of the TF-IDF- AP algorithm which proposed in this paper increased by 15.1%, compared with the classic TF-IDF algorithm. As can be seen from the table III, the F-measure value of the TF-IDF- AP algorithm which proposed in this paper increased by 12.9%, compared with the classic TF-IDF algorithm.

VI. CONCLUSION

Based on the summary of the experience from the predecessors' studies, we have analyzed the habits of Chinese, then propose adaptive weight in the position of the key word follows the traditional TF - IDF algorithm, which named TF - IDF - AL algorithm. This algorithm can make full use of the internal characteristics of key documents, overcome single word frequency weighting factor weights and the problem of insufficient ability of text characteristic in the classic TF-IDF algorithm, improving the precision of selection of key and text clustering accuracy. If scenario requires, more weighting factor should be introduced to the TF-IDF algorithm, time complexity of algorithm also should be avoid increase rapidly.

ACKNOWLEDGMENT

This work is supported in part by the NDRC Grant No. Q5025001201502.

REFERENCE

[1] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

[2] Turney P D, Pantel P. From frequency to meaning: Vector space models of semantics [J]. Journal of artificial intelligence research, 2010, 37(1): 141-188.

[3] Russell S J, Norvig P N. Artificial Intelligence: A Modern Approach. Prentice Hall[J]. Artificial Intelligence A Modern Approach, 2009, 15(96):217-218.

[4] Rokach L, Maimon O. Data Mining With Decision Trees: Theory and Applications[M]. World Scientific Publishing Co. Inc. 2014.

[5] Salton G, Yu C T. On the construction of effective vocabularies for information retrieval[M]// Operator algebras, unitary representations, enveloping algebras, and invariant theory :. Birkhauser, 1990:48-60.

[6] Rizomiliotis P. Improving the high order nonlinearity lower bound for Boolean functions with given algebraic immunity[J]. Discrete Applied Mathematics, 2010, 158(18):2049-2055.

[7] Xiong L, Tan L, Zhong M. An Automatic Term Extraction System of Improved C- value Based on Effective Word Frequency[J]. New Technology of Library & Information Service, 2013, 29(3):409-411.

[8] Salton G, Fox E A, Wu H. Extended Boolean Information Retrieval[J]. Communications of the ACM, 1983, 26(11): 1022- 1036.

[9] LI Hua-Meng,LI Hai-Rui,XUE Liang. TFIDF Algorithm Based on Information Gain and Information Entropy[J]. Computer Engineering, 2012, 38(08): 37-40.

[10] Hui H C, SUN Yat Sen University, Guangzhou. A Text Similarity Measurement Combining Word Semantic Information with TF-IDF Method[J]. Chinese Journal of Computers, 2011, 34(5):856-864.

[11] Shokripour R, Anvik J, Kasirun Z M, et al. A time-based approach to automatic bug report assignment[J]. Journal of Systems & Software, 2014, 102:109-122.

[12] Sang S J, Zhou Y. A Method for Computing Weight of Characteristic Item Based on the Length of Word[J]. Computer Knowledge & Technology, 2011.

[13] Sogou corpus [DB/OL] http://www.sogou.com/labs,2008.

[14] Anoual H, Aboutajdine D, Elfkihi S, et al. Features extraction for text detection and localization[J]. I/V Communications and Mobile Network (ISVC, 2010:1-4.

[15] Zhang H P, Yu H K, Xiong D Y, et al. HHMM-based Chineselexical analyzer ICTCLAS[J]. 2003:184-187