Text Clustering Based on the Improved TFIDF by the Iterative Algorithm

Xingheng Wang
School of Information Science and Technology,
East China Normal University,
Shanghai, China
wangxh@cc.ecnu.edu.cn

Jun Cao
Shanghai International Studies University Library,
Shanghai, China
caojunuse@163.com

Yao Liu, Shi Gao, Xue Deng
School of Information Science and Technology,
East China Normal University,
Shanghai, China
liuyao928@sina.com,
garret130@hotmail.com,
msndxy@hotmail.com

***Abstract*—Text clustering, an important part of the machine learning and pattern recognition, has extensive applications in the field of natural language processing. In this paper, a method is given to improve the classic TFIDF algorithm on its shortcomings. This paper classifies the text through Naive Bayesian classifier. And uses the iterative algorithm to optimize the selection of feature words, and then to optimize the classification ceaselessly. Experimental results show that the algorithm has preferable efficiency in feature-select and can increase classification accuracy.**

Keywords- TFIDF; text clustering; VSM; Naive Bayesian; iterative algorithm

I. INTRODUCTION

Text clustering is defined as a collection of text automatically grouped into different categories. Text clustering, an important part of the machine learning and pattern recognition, has extensive applications in the field of natural language processing. For example, Text clustering can be used in the pre-processing step of multi-document summarization, cluster the search engine results, help users to quickly locate the information they need, can also be used for automatic text classification. Document clustering mainly based on well-known cluster hypothesis: the documents in the same category will have more similarity than the documents in the different category. This naturally raises a question: how to measure the similarity between the two texts? Then the text clustering research problem will convert into text similarity research problem.

Text similarity is usually measured in different algorithms after the texts have extracted the text features on analyzing the text. Text features are often described by a model - Vector Space Model, which is often used in the natural language processing. Gerard Salton presented the vector space model theory in the 1960s. The usually generation of vector space model: first process the text by word segmentation, then select the feature words and calculate the weight of feature words, build an N-dimensional vector space at last. One of the keys is how to measure and calculate the weight of the feature words.

The feature word weight is the feature word's weight percentage in a document, used to indicate the feature word's important level in a document. There are many ways in feature word weight calculating. Such as Boolean weight, term frequency weight, square root function, logarithmic function, entropy function, TFIDF weight, TFC weight, LTC weight, etc. The way which you choose will affect the overall performance of the text clustering. TFIDF weight has been welcomed by researchers in different field which because its algorithm is relatively simple and has a relatively higher precision and recall rate. So the author uses the TFIDF weight in this paper.

The concept of IDF is first presented by Karen Sparck Jones in 1972 in [1]. But she did not explain the origin of IDF algorithm theoretically. In the same year Robinson made an initial interpretation of IDF. Salton had discussed TFIDF's application in the information retrieval field by writing articles and books several times, and presented the TFIDF algorithm in [2]. Salton had demonstrated that the TFIDF formula's effectiveness in the field of information retrieval in [3]. After that time TFIDF is increasingly widespread applied in different field. In this process, many scholars presented a lot of methods to improve the classic TFIDF algorithm. Bong Chih How and Narayanan K had presented an algorithm called Category Term Descriptor to improve the TFIDF algorithm in [4]. Roberto Basils presented the $TF \times IWF \times IWF$ formula in 1999 in [5].

In this paper, the algorithm is based on the theory of vector space model, uses the improved TFIDF algorithm to calculate the feature words weight, classifies the documentation set using vector cosine similarity algorithm and Naive Bayesian classifier, and then keeps using the iterative algorithm to obtain optimal classification.

II. THE TRADITIONAL TFIDF ALGORITHM

TFIDF is a method used to calculate the weight of a word or phrase in a document in the field of natural language processing. The main idea of TFIDF: If the term frequency of a word or phrase in a document is high, and this word or phrase rarely appear in other documents, then we believe that the word or phrase has a good ability in categories distinguishing. TFIDF is the product of Term Frequency, TF and Inverse Document Frequency, IDF. IDF formula is as follows:

978-1-4673-2365-9/12/$31.00 ©2012 IEEE

$$IDF_i = \log\left[\frac{N}{n_i}\right] \quad (1)$$

Where, N is the total number of documents, n_i is the number of documents containing the word t_i in the documentation set.

TFIDF formula is as follows:

$$TFIDF_{ij} = TF_{ij} \times \log\left[\frac{N}{n_i}\right] \quad (2)$$

In (2), TF reflects the activity of word t_i in the document d_j, and as we can see, the bigger the TF, the more significance the word t_i for the document d_j. And IDF reflects the prevalence of word t_i in the documentation set, the bigger the IDF, the more unique and more obvious ability in categories distinguishing the word t_i is. So, TFIDF, the synt hesis of TF and IDF can reflect the ability of a word to express a document.

Sometimes, in order to standardize the formula, often using the normalized TFIDF formula:

$$TFC_{ij} = \frac{TF_{ij} \times \log\left[N/n_i\right]}{\sum_{i=1}^{M}\left[TF_{ij} \times \log(N/n_i)\right]^2} \quad (3)$$

In (3), M is the size of feature word set.

The main idea of IDF: If the document contains fewer word or phrase t_i, means n_i is smaller in (1) and IDF is bigger, then the word or phrase t_i has a better ability in categories distinguishing. TFIDF simply believe that the word with smaller frequency in documentation set is more important, the word with bigger frequency in documentation set is more useless, obviously, this is not entirely correct.

TFIDF does not take the distribution of the word t_i between inter-class and within-class into account. For example, if the number of document in a certain class of document C_i which contains the word t_i is m_i, while the number of document in other class which contains the word t_i is k_i, it is clear that $n_i = m_i + k_i$, n_i is the number of document which contains the word t_i. The bigger m_i is, the bigger n_i is, according to (1), IDF will be smaller, shows that the word or phrase t_i has a less ability in categories distinguishing. But in fact, if a word or phrase appears frequently in a class of documents, then the word or phrase can be a good representative of this class of documents' text features. This word or phrase should be given a higher feature weight, and be selected as the feature word to distinguish from other class of documents. This is the inadequacy of the IDF.

III. THE IMPROVED TFIDF ALGORITHM

This paper presents an improved TFIDF algorithm aiming at the shortage of traditional TFIDF algorithm based on the iterative algorithm.

Improved algorithm is divided into the following steps:

1) Primary feature words selection

In this step, the author uses the open-source tool ICTCLAS which developed by Institute of Computing Technology Chinese Academy of Sciences for Chinese word segmentation and part of speech tagging. The correct rate of this tool is up to 97.58%. After this, we need to extract feature words. The author of this paper uses TF as the primary feature word weight. In order to facilitate statistics, we select the top 100 words order by TF weight as the candidate feature words. In this process, the algorithm will remove the word which does not meet the requirements of the part of speech, such as prepositions.

2) Advanced feature words selection

In this step, the algorithm will remove the word which contains redundant or implied, meaningless information. Then the left words will constitute the feature words set.

3) Vector space generation

In this step, use the theory of Vector Space Model to generate the vector space.

4) Text classification.

The algorithm will classify the documentation set using vector cosine similarity algorithm and Naive Bayesian classifier. After this we need to calculate each feature word's distribution in different class of document. How to find the feature word's distribution in each class of document? For example, if the number of document in a certain class of document C_i which contains the feature word t_i is m_i, while the number of documents in other classes which contains the feature word t_i is k_i, it is clear that $n_i = m_i + k_i$, n_i is the number of documents which contains the word t_i. Then we can use $\frac{m_i}{m_i + k_i}$ as the distribution of the feature word t_i in the class of document C_i. We can use 70% as the threshold value. If the distribution, $\frac{m_i}{m_i + k_i}$ is bigger than 70%, then the word t_i can be a good representative of this class of documents' text features. Where, we must not only eliminate the negative impact of m_i in the traditional TFIDF formula, but also need

to increase the positive impact of m_i in feature weight of word t_i. The traditional IDF formula is as follows:

$$IDF_i = \log\left[\frac{N}{n_i}\right] \quad (4)$$

This formula can also be as follows:

$$IDF_i = \log\left[\frac{N}{m_i + K_i}\right] \quad (5)$$

If $\frac{m_i}{m_i + k_i}$ is bigger than 70%, we can use the improved IDF formula which is as follows:

$$IDF_i = \log\left[\frac{N}{K_i}\right] \quad (6)$$

Where, the improved TFIDF formula is as follows:

$$TFIDF_{ij} = TF_{ij} \times \log\left[\frac{N}{K_i}\right] \quad (7)$$

Then we can get the improved TFIDF values according the feature word's distribution by improved TFIDF formula.

5) Iterative algorithm

The improved algorithm uses the constantly updated TFIDF weight value which calculated by the improved TFIDF formula as the feature word weight. The algorithm will select new feature words order by new TFIDF in this step. The algorithm constantly repeats the above 3) 4) 5) steps as a cycle, until convergence to obtain a particular target number of the feature words. The target number can be a certain percentage of candidate feature words' number or the total words' number.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Data and tool

In this paper, the author uses the Chinese web page classification training data set developed by Computer Networks and distributed Systems Laboratory of Peking University as the experimental data. The training data set include a total of 15571 web pages. There are 12 classifications in the data set.

The author uses Weka to demonstrate the effectiveness of the algorithm. Weka is a software which integrates a large number of machine learning algorithms usually used in data mining field.

The author has done two experiments and compared the result of the two experiments.

TABLE I. DATASETS OF THE FIRST EXPERIMENT

Data Set	Number of documents	Number of classes
data set 1	60	6
data set 2	86	6
data set 3	123	6
data set 4	690	6
data set 5	1028	6
data set 6	3682	6

The first experiment uses six data sets all from the same big data set. The six data sets changes in the number of documents and maintain the same number of classes. Shown in Table Ⅰ.

The second experiment also uses six data sets from the same big data set. The six data sets changes in the number of classes and maintain the same number of documents. Shown in Table Ⅱ.

TABLE II. DATA SETS OF THE SECOND EXPERIMENT

Data Set	Number of documents	Number of classes
data set 7	1028	2
data set 8	1028	4
data set 9	1028	6
data set 10	1028	8
data set 11	1028	10
data set 12	1028	12

B. Evaluation Criteria

F-measure is a measure formula of clustering results for assessment. And it's a measure that combines precision and recall rate. The formula is defined as follows:

$$F(i,j) = \frac{2 \times recall(i,j) \times precision(i,j)}{recall(i,j) + precision(i,j)} \quad (8)$$

Where, the precision rate is defined as $precision(i,j) = n_{ij} / n_j$ and the recall rate is defined as $recall(i,j) = n_{ij} / n_j$.

C. Experimental results and analysis

Figure 1 shows the result of the first experiment. The horizontal axis indicates the number of documents and vertical axis represents the F-measure. And Figure 2 shows the result of the second experiment. The horizontal axis indicates the number of classes and vertical axis represents the F-measure.

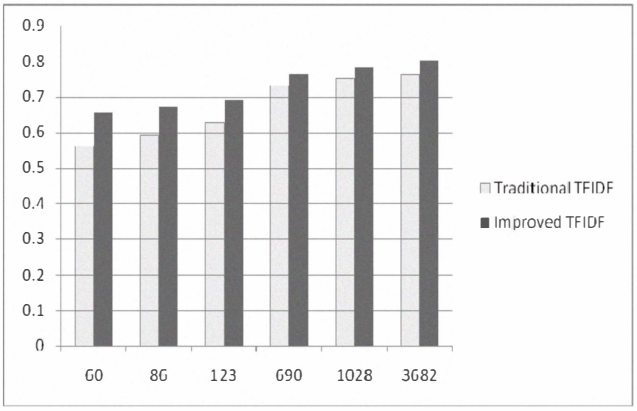

Figure 1. Experimental result 1

From Figure 1, the less the number of documents is, the more effective the improved TFIDF algorithm than the traditional TFIDF algorithm.

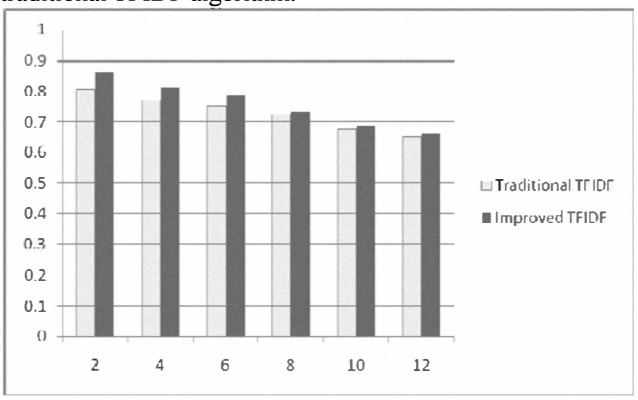

Figure 2. Experimental result 2

From Figure 2, the less the number of classes is, the more effective the improved TFIDF algorithm than the traditional TFIDF algorithm.

V. CONCLUSION

This paper has done an in-depth study on TFIDF algorithm's application in text clustering, the author has proposed an improved TFIDF algorithm based on the iterative algorithm. Experimental results show that the algorithm can increase the text classification accuracy.

The author will further improve the TFIDF formula to make it more reasonable and standardized in my follow-up research.

REFERENCES

[1] JONES K S. A statistical interpretation of term specificity and its application in retrieval[J]. Journal of Documentation, 1972, 28(1): 11-21.

[2] SALTON G, CLEMENT T Y. On the construction effective vocabularies for information retrieval[C]// Proceedings of the 1973 Meeting on Programming Languages and Information Retrieval. New York: ACM, 1973: 11.

[3] SALTON G, FOX E A, WU H. Extended boolean information retrieval[J]. Communications of the ACM, 1983, 26(11): 1022-1036.

[4] HOW B C, NARAYANAN K. An empirical study of feature selection for text categorization based on term weightage[C]// Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence. Washington DC: IEEE Computer Society, 2004: 599-602.

[5] BASILI R, MOSCHITTI A, PAZIENZA M. A text classifier based on linguistic processing[C]// Proceedings of IJCAI-99, Machine Learning for Information Filtering, 1999.