A Novel Improved TFIDF Algorithm

KeHua Yang
College Of Information Science and Engineering
Hunan University
ChangSha, China
khyang@hnu.edu.cn

Dan Ye
College Of Information Science and Engineering
Hunan University
ChangSha, China
dancy@hnu.edu.cn

***Abstract*—Feature weighting algorithm has a great effect on the accuracy of text categorization. The classical Term Frequency and Inverse Documentation Frequency algorithm (TFIDF) ignores the semantic relationships between terms in the document set, thus to influence the accuracy of term weight calculation. To calculate the weight of words more correctly, the paper introduces the semantic association between words and proposed a new improved algorithm (S-TFIDFIGE) combined with semantic, information entropy and information gain. Experimental results show that the proposed method has better classification results than the traditional TFIDF and other improved algorithms.**

Keywords-text categorization; TFIDF; semantics; information entropy; information gain

I. INTRODUCTION

Text categorization, the assignment of natural language documents to one or more predefined categories based on their content [1-2] is a important technique for organizing and processing vast quantities of text. It has a wide range of applications in information retrieval, information filtering and search engine. The vector space model (VSM) is the commonly used model for text expression. In vector space model, a multi-dimension vector he similarity between two documents can be evaluated by cosine value of the two vectors. Thus, term weighting calculation can directly affect the accuracy rate of text classification [3].

At present, the common weight calculation methods include Boolean weighting, Word frequency weighting, Entropy weighting and TFIDF weighting, and so on [4]. TFIDF weighting is one of the most widely used algorithms among them, but it ignores the proportion of distribution of terms among and inside categories, and so on. In recent years, aiming at the defects of TFIDF, the researchers have done a lot of improvements work. After studied and analyzed those improved algorithms in detail, the paper introduces into the semantic analysis between words and proposes an improved algorithm S-TFIDFIGE which combines semantic, information entropy and information gain, and the paper also makes an experimental comparison with other improved algorithms.

II. RELATED WORK

A. Traditional TFIDF

In 1973, Salton proposed TFIDF in the literature for the first time[5], the main idea of TFIDF is that the more often a term appear in a particular text, the stronger category distinguishing ability it will have, so we should give a higher weight to the term. While a term has a broader presence in the dataset, its category distinguishing ability will be lower and the term should be given a lower weight. At present, TFIDF is widely applied to the information retrieval area, and the classical formula is as follows:

$$w_{ij} = tf_{ij} \times idf_j = \frac{tf_{ij} \times lb\left(N / n_j + 0.01\right)}{\sqrt{\sum_{j=1}^{n}\left(tf_{ij}^{\,2} \times lb\left(N / n_j + 0.01\right)^2\right)}} \qquad (1)$$

In (1), tf_{ij} represents the term frequency of term j in document i, n_j represents the number of documents where term j appears.

B. Improved TFIDF Approach

a) Skewed data distribution

Distribution of classes of document sets are often skewed, which will seriously affect the weight of term. Bong Chih How and Narayanan K [6] adopted TFIDF in a category perspective instead of document perspective and used the Category Term Descriptor (CTD) to improve TFIDF.

b) Intra- and Inter-class distributing Deviation

Traditional TFIDF considers the document collection as a whole and ignores the proportion of distribution of terms among and inside categories. Y. Zhang and X. Chen [7] applied the information gain to remedy the defect of TFIDF and proposed a TFIDF algorithm based on information gain (TFIDFIG). On the basis of TFIDFIG, X. Li and H. Li [8] introduced information entropy and put forward a TFIDF algorithm based on information gain and information entropy(TFIDFIGE) which further improved the accuracy of text classification result.

c) Other Improvements

In addition, some researchers replace IDF with different selection function. Roberto Basili[9] proposed TF*IWF*IWF in 1999, he used IWF*IWF to represent the second factor, thus improved the accuracy of text classification.

III. PROPOSED METHOD

A. Related concepts

Before discussing the improvement of algorithm in this paper, we will give the related concepts about information gain and information entropy.

a) Information entropy

Definition1 Suppose there are n messages whose probabilities are the same, the probability of each message is $p(x)=1/n$, then the information transferred by an message can be described as:

$$I(X)=1b\frac{1}{p(X)}=-lb\big(p(X)\big)=lb(n) \qquad (2)$$

Definition2 Given a probability distribution $P=(p_1,p_2,...,p_n)$, the information entropy of P is as follows.

$$I(P)=-\left(p_1*lbp_1+p_2*lbp_2+...+p_n*lbp_n\right)=-\sum_{k=1}^{n}p_k*lbp_k \qquad (3)$$

If $p_k=0$, then $p_k*lbp_k=0$ .When $p_1=p_2=...=p_n$, $I(P)=1$. Equation (3) shows that the more uniform the probability distribution is, the greater amount of information it will transfer.

b) Information gain

Definition 3 Suppose $I(X)$ refers to the entropy of probability space that a random document belongs to a certain category, $I(X/y)$ refers to the entropy of probability space that a random document belongs to a certain category after word y appears, and the information gain can be described as follows.

$$IG(X/y)=I(X)-I(X/y) \qquad (4)$$

According to the definition3, information gain measures the effect of word y on classification.

B. Semantic Similarity Calculation between Words

Word similarity has different meanings in different applications, here it means the ability of replacement of two words in the text, and we measure it with a value between 0 and 1, the ability of mutual replacement is proportional to this value. This paper takes “HowNet” as a semantic ontology to calculate the semantic similarity between terms. “Concept” and “Sememe” are two basic concepts in HowNet, the sememe refers to the basic unit to describe concept, the concept is a description of lexical semantics, each word can be expressed as a few concepts. Literature [10] elaborates the structure of HowNet. Calculation of Semantic Similarity is as the following:

There are n concepts in word $W_1\left(S_{11},S_{12},\ldots,S_{1n}\right)$ and m concepts in word $W_2\left(S_{21},S_{22},\ldots,S_{2m}\right)$. The similarity between W_1 and W_2 is the maximum of similarity between concepts, the computation formula is as follows:

$$Sim\left(W_1,W_2\right)=\max_{i=1...n,\,j=1...m}Sim\left(S_{1i},S_{2j}\right) \qquad (5)$$

In (5), the formula for computing similarity between two concepts is:

$$Sim\left(S_1,S_2\right)=\sum_{i=1}^{4}\beta_i\prod_{j=1}^{i}Sim_j\left(S_1,S_2\right) \qquad (6)$$

In (6), $Sim_1\left(S_1,S_2\right)$ refers to the similarity of the first sememe expression between two concepts, $Sim_2\left(S_1,S_2\right)$ refers to the similarity of the others independent sememe expression between two concepts, $Sim_3\left(S_1,S_2\right)$ refers to the similarity of relation sememe expression between two concepts, $Sim_4\left(S_1,S_2\right)$ refers to the similarity of symbol sememe expression between two concepts. $\beta_i\left(1\le i\le 4\right)$ is an adjustable parameter, and $\beta_1+\beta_2+\beta_3+\beta_4=1$, $\beta_1\ge\beta_2\ge\beta_3\ge\beta_4$,the value of β_1 is commonly greater than 0.5. From the constraints of $\beta_i\left(1\le i\le 4\right)$, we can see that the role which on the overall similarity from Sim1 to Sim4 decreased gradually. All of the concepts are ultimately represented by sememes, the similarity between two sememes is computed as follows:

$$Sim\left(p_1,p_2\right)=\frac{\alpha}{d+\alpha} \qquad (7)$$

In (7), p1 and p2 refer to two sememes, d represents the path length of p1 and p2 in sememe levels, it is a positive integer, α is an adjustable parameter.

C. Algorithm S-TFIDFIGE

Most of the previous improvements view each feature word as isolated and there are few studies involving lexical semantics. But in fact, there is a semantic relation between feature words. For example, “text” and its synonym “document” are regarded as two irrelevant feature words in traditional TFIDF, actually there will be two similar texts, one of them uses the word “text” frequently, while the word “document” appears in the other text constantly. The classification result according to traditional TFIDF may be incorrect.

Some improved algorithm have considered the proportion of distribution of terms in categories and between categories such as TFIDFIGE, and improved the accuracy of text classification. However, they also ignored the semantic relation between words in documents set. Take a example that a word often appears in certain category and its synonyms occur in other categories frequently, this indicates the word can not represent the category well and should be assigned a small weight because of its weak classification ability, but the weight calculated by TFIDFIGE is large. Meanwhile, some terms have low frequency in certain category, but their synonyms appear frequently in this category, these terms are representative and should be assigned a large weight, but the weight computed by TFIDFIGE is small.

To solve the problem above mentioned, this paper firstly makes a semantic analysis of the word and groups the words whose semantic similarity is greater than a given threshold, then calculates IDF, information entropy, information gain respectively. The improved S-TFIDFIGE can be described as follows:

Input: words set from preprocessed document set

Output: vector table of the text

Step 1 For each word t in preprocessed words set, we calculate the semantic similarity between t and another word in feature set according to (5), when the value of semantic similarity is greater than α （α is a threshold, the value is 0.8 in our experiments), we add the word to the t ’s similar feature group, finally we count the number of words in the group, and assigns to m.

Step 2 Compute the inverse documentation frequency of term t, the formulas are as follows.

$$\overline{idf} = lb\frac{N}{\bar{n}} \qquad （8）$$

$$\bar{n} = \frac{n_t + \sum_{r=1}^{m} n_i}{m+1} \qquad （9）$$

In (8) and (9), N refers to the total number of texts in document set, n_t refers to the number of texts containing t. $\bar{n}$ refers to the average of the sum total of the number of the text of appearing t and the number of the text of terms of other text semantic similarity with t, The Calculated inverse documentation frequency according to (8) considers the distribution of t and its semantic similarity words in the document set, thus the results are more authentic.

Step 3 Calculate the weighted factor of information entropy in certain category, the formula is:

$$\overline{E} = -\sum_{i=1}^{n} \frac{tf(t_g, d_j)}{tf\left(t_g, C_K\right)} lb \frac{tf\left(t_{g,} d_j\right)}{tf\left(t_g, C_K\right)} \qquad (10)$$

In (10), $tf\left(t_g, d_j\right)$ refers to the frequency that t and the member of its similar feature group appear in the jth text in category C_K, while $tf\left(t_g, C_K\right)$ refers to the frequency that t and the member of its similar feature group appear in category C_K. The formulas are as follows.

$$tf\left(t_g, d_j\right) = tf_j + \sum_{r=1}^{m} tf_{rj} \qquad (11)$$

$$tf\left(t_g, C_K\right) = \sum_{i=1}^{n}\left(tf_i + \sum_{r=1}^{m} tf_{ri}\right) \qquad (12)$$

In (11) and (12), n refers to the total number of texts in category C_K, tf_j refers to the number of times t appears in the jth text in category C_K, tf_{ri} refers to the number of times the rth word in t’ s similar feature group appears in the ith text in category C_K. According to the weighted factor of information entropy, we can conclude when t and its semantic similarity word evenly distribute in certain category, its information entropy $\overline{E}$ takes the maximum value 1 and t has the strongest classification ability. On the contrary, if t and its semantic similarity word only appears in a text in certain category, $\overline{E}$ takes the minimum value 0 and t has the weakest classification ability. Therefore, $\overline{E}$ can well reflect the distribution of t and its semantic similarity word in certain category, and its value is proportional to the classification ability.

Step 4 Calculate the weighted factor of information gain between categories. The formula is:

$$\overline{IG(C,t)} = E(C) - \overline{E\left(C/t\right)} \qquad (13)$$

$$E(C) = -\sum_{i=1}^{u} p\left(C_i\right) * lb\left(p\left(C_i\right)\right) \qquad (14)$$

$$\overline{E(C/t)} = -\sum_{i=1}^{u}\left[\left(p\left(C_i/t\right) + \sum_{r=1}^{m} p\left(C_i/t_r\right)\right) * lb\left(p\left(C_i/t\right) + \sum_{r=1}^{m} p\left(C_i/t_r\right)\right)\right] \qquad (15)$$

In (13) (14) and (15), C refers to the document set, $p\left(C_i\right)$ refers to the probability of C_i, u refers to the number of categories of documents, $p\left(C_i/t\right)$ refers to the probability that term t appears in category C_i, $p\left(C_i / t_j\right)$ refers to the probability that the j^{th} word in t' s similar feature group appears in category C_i. The paper takes term t and the word in its similar feature group as computed objects, then calculates the information gain, thus provides comprehensive information for document classification.

Step 5 Calculate the weight of feature word t, the formula is:

$$w_{it} = tf\left(d_i\right) * \overline{idf} * \overline{E} * \overline{IG\left(C,t\right)} \qquad (16)$$

In (16), w_{it} refers to the weight of t in text d_i, $tf\left(d_i\right)$ refers to the frequency of t in text d_i.

Step 6 Repeat step 1-5 until getting the feature vector tables for all the texts, and each text will be expressed as $\left(w_{i1}, w_{i2}, \ldots, w_{in}\right)$, n refers to the dimension of feature vector, w_{in} refers to the weight of the n^{th} term in text d_i.

We can conclude from the description above that algorithm S-TFIDFIGE which is based on semantics, information entropy and information gain considers not only the distribution of term t throughout the text set and the inter-class and intra-class distribution, but the distribution of semantic similarity words.

IV. Experimental Result and Analysis

A. Experiment Description

The experimental data in the paper is TanCorp, a corpus for Chinese text classification is collected and processed by Songbo Tan [11]. TanCorp is divided into 12 categories, After removing the stopping words, we randomly selected 150 samples per category among 12 categories as experimental corpus. The experiment uses cross validation, that is, dividing the dataset into tree equal parts, taking two parts of them as training data and other one part as testing data in turn, then we use information gain method to select 500 features. Next, we use the S-TFIDFIGE proposed in the paper to calculate the weight of terms in training texts, then express all of training texts as vectors. As we don't know which category the testing text belongs to, we use the traditional TFIDF to calculate weight and express all of testing texts as vectors. Finally we use the KNN (K Nearest Neighbor) as text classifier [12] to conduct the experiment.

B. Evaluation Criterion

The commonly used indicators in text classification, precision(P), recall(R) and F1 are used to evaluate the experimental result. The F1 combines recall and precision ,as to a certain category:

$$\Pr ecision = \frac{number \quad of \quad correct \quad positive \quad predictions}{number \quad of \quad positive \quad predictions} \qquad (17)$$

$$\operatorname{Re} call = \frac{number \quad of \quad correct \quad positive \quad predictions}{number \quad of \quad positive \quad examples} \qquad (18)$$

$$F_1 = \frac{2 * \Pr ecision * \operatorname{Re} call}{\left(\Pr ecision + \operatorname{Re} call\right)} \qquad (19)$$

Using these averages, we can observe the effect of different kinds of weighting algorithm on a text classification system.

C. Comparison and Analysis

The experiment takes traditional TFIDF, Improved approach to weighting terms using information gain (TFIDFIG), TFIDF Algorithm Based on Information Gain and Information Entropy (TFIDFGE) and S-TFIDFIGE proposed in the paper respectively to calculate the weight of terms, then classify the text in testing dataset with KNN classifier. The experimental result is measured and contrasted by precision, recall and F1 measure. Table 1, Table 2 and Table 3 are corresponding to precision, recall and F1 measure in these four algorithms respectively.

TABLE I. THE PRECISION OF DIFFERENT METHODS

Category	**Weighting Algorithm**			
	TFIDF	*TFIDFIG*	*TFIDFIGE*	*S-TFIDFIGE*
Sports	0.8234	0.8391	0.8663	0.8786
Entertainment	0.8957	0.9178	0.9344	0.9689
Autocar	0.5534	0.5789	0.6278	0.7214
Property	0.7426	0.7653	0.7768	0.8096
Finance	0.7929	0.8364	0.8717	0.8683
Education	0.6986	0.7343	0.7625	0.7864
Talent	0.7501	0.7743	0.8026	0.8398
Computer	0.8331	0.8587	0.8846	0.9068
Technology	0.9389	0.9534	0.9745	0.9896
Art	0.6547	0.6973	0.7536	0.8459
Region	0.6569	0.6742	0.7084	0.7985
Hygiene	0.6725	0.7067	0.7353	0.7686
Avg(P)	0.7511	0.7780	0.8082	0.8485

From Table 1, we can see our improved S-TFIDFIGE weighting method has the best precision, and the precision has greatly increased at each category, especially at autocar and art. The average precision of S-TFIDFIGE is 84.85%, which is approximately 9.74% higher than that of TFIDF, 7.05% higher than that of TFIDFIG and 4.03% higher than that of TFIDFIGE.

TABLE II. THE RECALL OF DIFFERENT METHODS

Category	**Weighting Algorithm**			
	TFIDF	*TFIDFIG*	*TFIDFIGE*	*S-TFIDFIGE*
Sports	0.8097	0.8749	0.8663	0.9368
Entertainment	0.9833	0.9807	0.9215	0.9264
Autocar	0.6387	0.6731	0.7335	0.7846
Property	0.6122	0.6501	0.7253	0.8389
Finance	0.9228	0.9135	0.9343	0.9561
Education	0.8653	0.8823	0.9041	0.9185
Talent	0.8947	0.8836	0.8765	0.8754
Computer	0.6502	0.7284	0.7931	0.8873
Technology	0.9054	0.9288	0.9613	0.9887
Art	0.7274	0.7526	0.8192	0.8663
Region	0.6821	0.7343	0.8687	0.9036
Hygiene	0.8321	0.8246	0.7962	0.8078
Avg(R)	0.7935	0.8189	0.8508	0.8925

Table 2 shows that the improved algorithm S-TFIDFIGE is superior to TFIDF, TFIDFIG and TFIDFIGE at recall. Compared with other three algorithms, the average of recall ratio improves by 9.9%、7.36%、4.17% respectively.

TABLE III. THE F1 OF DIFFERENT METHODS

Category	**Weighting Algorithm**			
	TFIDF	*TFIDFIG*	*TFIDFIGE*	*S-TFIDFIGE*
Sports	0.8165	0.8566	0.8762	0.9068
Entertainment	0.9375	0.9482	0.9279	0.9472
Autocar	0.5930	0.6225	0.6765	0.7517
Property	0.6711	0.7030	0.7502	0.8335
Finance	0.8529	0.8732	0.9019	0.9101
Education	0.7731	0.8015	0.8273	0.8473
Talent	0.8160	0.8253	0.8379	0.8572
Computer	0.7304	0.7882	0.8364	0.8969
Technology	0.9218	0.9409	0.9679	0.9891
Art	0.6891	0.7239	0.7850	0.8560
Region	0.6693	0.7030	0.7763	0.8478
Hygiene	0.7438	0.7611	0.7645	0.7877
Avg(F1)	0.7679	0.7956	0.8273	0.8693

From experiment results on Table 3, we can see our improved algorithm S-TFIDFIGE has the best F1, and the F1 has greatly increased compared with other methods. TFIDFIGE have better results than TFIDFIG and the traditional TFIDF has the worst performance at F1.

In conclusion, S-TFIDFIGE is superior to other algorithms at precision, recall and F1, this shows the our improved S-TFIDFIGE is consistent with the theoretical demonstration and the semantic relation between feature words has a great effect on weight, the introduction of semantic relation in weight calculation formula results in the improvement of final classification result.

V. CONCLUSION

The paper firstly analyzes previous improvement methods in detail, then introduces the semantic association between words and proposed the new S-TFIDFIGE algorithm which can make up for the defect of the lack of semantic information in statistical method. Extensive experiments have been carried out to assess the effectiveness of the proposed S-TFIDFIGE term weighting strategy in the field of text categorization, and the result has shown that the new strategy further improves the performance of the text classifiers than other improvement weighting strategy.

ACKNOWLEDGMENT

This research was financially supported by the growth of young teachers of Hunan University

REFERENCES

[1] F. Sebastiani, "Machine learning in automated text categorization," ACM computing surveys (CSUR), vol. 34, pp. 1-47, 2002.

[2] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization," in Proceedings of the seventh international conference on Information and knowledge management, 1998, pp. 148-155.

[3] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," Communications of the ACM, vol. 18, pp. 613-620, 1975.

[4] K. Aas and L. Eikvil, "Text categorisation: A survey," Raport NR, vol. 941, 1999.

[5] G. Salton, E. A. Fox, and H. Wu, "Extended Boolean information retrieval," Communications of the ACM, vol. 26, pp. 1022-1036, 1983.

[6] B. C. How and K. Narayanan, "An empirical study of feature selection for text categorization based on term weightage," in Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, 2004, pp. 599-602.

[7] Y. Zhang, X. Chen, and Z. Xiong, "Improved approach to weighting terms using information gain," Computer Engineering and Applications, vol. 43, pp. 159-161, 2008.

[8] X. Li, H. Li, L. Xue, and G. He, "TFIDF Algorithm Based on Information Gain and Information Entropy," Computer Engineering, vol. 38, pp. 37-40, 2012.

[9] R. Basili, A. Moschitti, and M. T. Pazienza, "A text classifier based on linguistic processing," 1999.

[10] Q. Liu and S. Li, "The Calculation of Semantic Similarity between Words Based on HowNet," Chinese Computational Linguistics, vol. 7, pp. 59-76, 2002.

[11] S. Tan, X. Cheng, M. M. Ghanem, B. Wang, and H. Xu, "A novel refinement approach for text categorization," in Proceedings of the 14th ACM international conference on Information and knowledge management, 2005, pp. 469-476.

[12] Y. Yang and X. Liu, "A re-examination of text categorization methods," in Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, 1999, pp. 42-49.