International Journal of Scientific Engineering and Technology ISSN:2277-1581
Volume No.5 Issue No.5, pp: 296-300 01 May 2016

A New Algorithm For Document Classification Based On Weighting Features And Files

Mahbubeh Ziaee
Sama Community College, Islamic Azad University, Kazeroon Branch, Kazeroon, Iran
m_z8465@yahoo.com

Abstract— With regard to the increasing amount of information in the present world, there is increasing need for new powerful instruments for changing data to useful knowledge. One of the vital ways of controlling and managing data is classifying texts. This article presents an algorithm for classifying documents. It has capabilities such as quality control of created classification based on feedback from F evaluation measure, weighting features based on the classes, assigning weight to each file in all classes and transferring file to a class that has the most weight. This procedure deletes the redundancy words with high quality due to improvement in classes. Finally we evaluate the algorithm, that is, first, the influence of different early random classifications are studied, then the influence of different weighing methods TFCRF،TFRF،TFIDF and the proposed weighing method is investigated on the output of the proposed classification algorithm . Finally, the proposed algorithm is compared with other algorithms. The results show that all mentioned cases collectively increase quality and accuracy of the classification.

Keywords— documents classification, weighting features, retrieving documents.

I. Introduction

In the present world, it is not the shortage of the information that creates a problem rather it is the shortage of the knowledge that can be achieved from this information. Text analysis or getting knowledge from the text was first introduced in [1]. Classification of text documents into two or three predetermined classes based on the content is called document classification. Classification of documents is a supervised learning process.

If we have a set of texts $D = \{(d1, y1),...,(di , yi),...(dn , yn)\}$ in a way that n is the number of texts and $d_i = [W_{i,1},...,W_{i,k},..W_{i,|d_i|}]$ is the ith text of this set, *Wi,k is the kth word of ith text and yi refers to the class to which the text belongs(that is yi* $\in$ *C in a way that C* $\in$ $\{C1,C2,...,C|c|\}$ is the collection of predefined classes in the system). The purpose in text classification is the derivation of F relation function in a way that $yi \in f(di)$. [1]

The classification of texts based on different methods has been done for English language. Yang and Liu classified texts based on the root frequency vectors in 1999. [2] Joachims classified texts by using Support Vector Machine (SVM) [3]. At the same year Apte ,et,al used decision making tree for classifying texts. [4] In 1992 Creecy used KNN nearest neighbor categorization in order to categorize texts. [5] Koller used Bayes categorizing in 1997. [6] For more information on the ways of categorizing texts refer to [7]. Among the abovementioned methods, SVM is one of the best methods. [8].

Giving weight to the characteristics of the texts that we want to classify is very important and results in a proper classification of the documents. [9] The present weighting methods mostly have been used in the area of retrieving information for the first time. If in the weighting features method, we pay attention not only to the way features are distributed in different documents, but also the way they are distributed in different classes, it will have a very desirable influence on improving the classification of the documents. Among these methods are TFRF and TFCRF [9].

The structure of the article is as the followings: chapter two describes some of the methods of weighting features briefly. Chapter three involves implementation and description of the proposed weighting method, methodology and algorithm. Chapter four includes the analysis of the results and finally chapter five involves conclusion.

II. A review of feature weighting methods

We can classify common weighting methods based on TF, IDF, compound TFIDF methods and finally special methods of document classification [9]. In this section, we will briefly describe the function of each one. The common feature of all these algorithms is that determining key words is done based on the analysis of a set of related texts [11]. For more information on weighting methods refer to [7].

1.2. TF- based methods

In these methods, weight is a functional feature of different feature distribution in each document di ∈ D. This is a simple and practical method and if the feature exists, its weight equals the number of di in the document tk and the iteration of that feature in the related document.

$$w_{kj} = tf(t_k, d_j) = \begin{cases} \#(t_k, d_j) & t_k \in \text{vector of } d_j \\ 0 & t_k \notin \text{vector of } d_j \end{cases} \quad (1)$$

In which # (tk , di) is the number of each iteration of feature tk in the document di.

Among the TF methods are ITF, Sparck, normTF, Log TF. [9]

2.2. IDF- based methods

In these methods, weighting features is a function of distribution of feature tk in a set of document D. The main idea of weighting in this group is as the following: if we have less documents that have tk feature, tk is more suitable feature for differentiating documents from each other; therefore, it must

ISSN:2277-1581
01 May 2016

have more weight. This method was first introduced in [9] as the following:

$$w_{ki} = idf(t_k, d_i) = \log \frac{|D|}{|D(t_k)|} \quad (2)$$

In which |D| is the whole number of documents and |D(tk)| is a number of documents of D that contain feature tk. It is obvious that Wki decreases by increasing |D(tk)|.

2.3. TFIDF- based Methods

These methods which were introduced in the field of retrieving information for the first time were used in classifying documents for weighting features. TFIDF method was the result of combining methods based on TF and methods based on IDF [9] and it is calculated based on the following relation:

$$w_{ki} = tfidf(t_k, d_i) = tf(t_k, d_i) * idf(t_k, d_i) \quad (3)$$

2.4. Methods Based on the Information of Classes

This group of feature weighting methods are not limited to distribution of feature tk in the D set and benefit from distribution of feature tk in predefined classes Cj C.

2.4.1. TFRF Method

In this method feature weighting is done based on the classes. In fact, one kind of weighting feature is in the area of document classification.[9] A relation factor rf for each feature tk in the class Cj is defined in the following relation:

$$rf(t_k, c_j) = \log\left(2 + \frac{|D(t_k, c_j)|}{\sum_{m=1, m\neq j}^{|C|} |D(t_k, c_m)|}\right) \quad (4)$$

In which |D(tk,cm)| ، Cj εC are the number of documents of D set and Cj class that have tk feature and $\sum_{m=1, m\neq j}^{|C|} |D(t_k, C_m)|$ are the sum of the number of the documents of D set and a class other than CJ that has tk feature. Based on the formula, we can understand that factor rf has direct relation with the number of documents that have tk feature and belongs to Cj and has adverse relation with the number of documents that have tk feature and belongs to a class other than Cj. Therefore, the weight of tk feature in the document di after normalization can be achieved from the following relation:

$$w_{ki} = TFRF(t_k, d_i) = \frac{tf(t_k, d_i) * rf(t_k, c_{d_i})}{\sqrt{\sum_k \left(tf(t_k, d_i)\right)^2 * (rf(t_k, c_{d_i}))^2}}$$

In which $C_{d_i} \in C$ is the class of document di.

2.4.2. TFCRF method

Factor rf in the above method is regarded independent of the number of existing documents in each class. In TFCRF, this subject is considered. Instead of rf two factors of positive RF and negative RF are defined. Positive RF shows the ratio of a number of documents of Cj class that have tk feature to the entire documents of that class and negative RF shows the ratio of the sum of the number of documents from a class other than Cj that have tk feature to the entire sum of documents of classes other than Cj and it will be defined as the following:

$$positiveRF(t_k, c_i) = \frac{|D(t_k, c_j)|}{|D(c_j)|} \quad (6)$$

$$negativeRF(t_k, c_i) = \frac{\sum_{m=1, m\neq j}^{|C|} |D(t_k, c_m)|}{\sum_{m=1, m\neq j}^{|C|} |D(c_m)|} \quad (7)$$

In which $|D(cj)|$ is the number of documents of Cj class and $|D(tk,cj)|$ is the number of documents from D set and class Cj that have tk feature. From the above relations, the amount of value of relation factor of each class (crfValue) can be defined in this way:

$$crfValue(t_k, c_i) = \frac{positiveRF(t_k, cj)}{negativeRF(t_k, c_j)} \quad (8)$$

To remove the effect of the length of the document on the accuracy and efficiency of classifier, normalization is used to limit the weight of features in the domain (0,1). Weighting formula TFCRF for weighting feature of tk in the document di is provided as the followings:

$$w_{ki} = TFCRF(t_k, d_i) = \frac{\log(tf(t_k, d_i) * crfValue(t_k, c_{d_i}))}{\sqrt{\sum_k (\log(tf(t_k, d_i) * crfValue(t_k, c_{d_i})))^2}} \quad (9)$$

III. The Proposed Algorithm

In this section an algorithm is provided for classifying the documents. First, the algorithm, flowchart and then a description of the stages are provided.

The proposed algorithm includes the following stages:

1. Beginning
2. Classify the documents randomly.
3. Calculate F criterion for each class.
4. If F criterion did not increase and algorithm was performed more than once, then go to stage eleven11.
5. Extract the words which exist in each class and then find the etymology of the words by using Porter etymology algorithm.
6. Assign a weight to each word by using the proposed weighting method.
7. Delete the additional words.
8. Calculate the weight of each file in all classes.
9. For all files, if a file in class i has more weight than the other classes, then file should be transferred to class i. (finally a new classification will be created.)
10. Go to the third stage.
11. The end.

Figure 1: The proposed classification algorithm.

ISSN:2277-1581
01 May 2016

1.3. Implementing the proposed algorithm

Among the important features of the algorithm are the followings:

Controlling the quality of the created algorithm based on the feedback from evaluation measure F, that is, after creating new classes, these classes are evaluated and if the created class has improvements in comparison to its previous states, it is inserted in the cycle again, and it is done for all classes. Assigning weight to the files based on the class to which they belong. In this algorithm, a weight is assigned to each feature, the weight of the files is calculated in all classes and if file has more weight in each class, it is transferred to that class. The process of the algorithm results in the removal of the extra words with high quality due to improvements in classes. All the abovementioned cases together increase the quality and accuracy of the classification.

The proposed algorithm once is performed out of the cycle ring and F criterion is produced and then it is inserted in the algorithm cycle.

Flowchart of this algorithm can be shown in the following way:

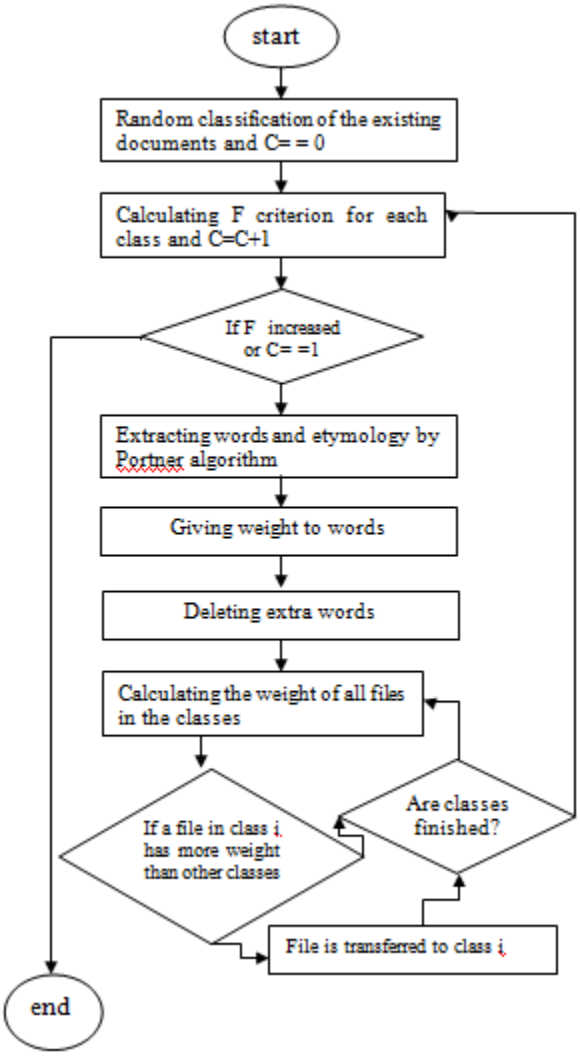


Figure 2: flowchart of the proposed classification algorithm

In general, we can describe the process of the proposed algorithm in the following way:

3.1.1. Early classification of texts, Calculating F evaluation measure for each class

First, we classify the texts on which we want to apply the classification algorithm. This early classification is the entrance of suggested algorithm and then, F is calculated for each class. From the results, we can control the quality of the classes.

3,1,2, Extracting words, Etymology and Giving Weighting to Features

At this stage etymology is performed based on the popular Porter algorithm, and then features are given weighting based on suggested weighting method:

With respect to the fact that methods based on TF and IDF and compound methods TFIDF are mostly introduced for retrieving applications and then are used in document classification area, in these methods, the way feature tk is distributed in Cj □ C is ignored. Thus, these methods need corrective measurements so that weight can also be the functional feature of the class of documents that have that feature. As a result, if methods based on information classification are used, better results will be achieved.

The criterion of rf defined in [12] and also crf [14] provide early solutions for calculating feature weight in classes. Since in these criteria, weight of feature tk in document di has direct relationship with the number of documents from $^{C}d_{i}$ class, and has adverse relationship with the number of documents that are from a class other than $^{C}d_{i}$.

Factors rf and crf pay attention to the existence or lack of a feature in existing document in desired class and don't notice the weight of feature in existing document in that class. However, attention to this factor can result in better weighting to the features. In the suggested method, first, word weight is calculated in the file to which the word belongs, this weight equals the ratio of the number of word iteration in the file to the whole number of files as the followings

$$W(t_i, d_i) = \frac{tf(t_i, di)}{|T(d_i)|} \qquad (1\cdot)$$

I

n the above formula $|T(d_i)|$ equals the whole number of words of file di.

Therefore, to give more precise weighting to features in suggested weighting, positive relation factor shows the ratio of sum of tk weight in the documents that include this feature and are present in Cj class to the whole documents of that class. In fact, positive relation factor gives a weight mean of tk feature in class Cj and negative relation factor shows the ratio of the sum of weights of feature tk in the documents of other classes except Cj that contain tk feature to the whole sum of document classes other than Cj. They are defined as the following:

doi : 10.17950/ijset/v5s5/519

ISSN:2277-1581
01 May 2016

$$positiveRF(t_k, c_i) = \frac{\sum_{t_k \in D} Wt_k(t_k, c_j)}{|D(c_j)|} \quad (11)$$

$$negativeRF(t_k, c_i) = \frac{\sum_{m=1, m\neq j}^{|C|} \sum_{t_k \in D, D \in C_m} Wt_k(t_k, c_m)}{\sum_{m=1, m\neq j}^{|C|} |D(c_m)|} \quad (12)$$

In the above relation $|D(cj)|$ is the number of class documents Cj and $\sum_{t_k \in D} Wt_k(t_k, c_j)$ is sum of tk weight in the documents that include this feature and are present in Cj class.

$\sum_{m=1, m\neq j}^{|C|} \sum_{t_k \in D, D \in C_m} Wt_k(t_k, c_m)$ is the sum of weights of tk feature in other class documents except Ci class that contain feature tk. From the above relations, the weighted crf value is defined as the followings:

$$Weighted\_crf\_Value(t_k, c_i) = \frac{positiveRF(t_k, cj)}{negativeRF(t_k, c_j)} \quad (13)$$

To remove the influence of the length of the document on accuracy and efficiency of classifier, normalization is used to limit feature weight in amplitude (0,1). Weighting formula TF-Weighted-CRF is provided for weighting feature tk in the document di as the following:

$$w_{ki} = TF\_Weighted\_CRF(t_k, d_i) \quad (14)$$

$$w_{ki} = \frac{\log(tf(t_k, d_i) * Weighted\_crf\_Value(t_k, c_{d_i}))}{\sqrt{\sum_k (\log(tf(t_k, d_i) * Weighted\_crf\_Value(t_k, c_{d_i})))^2}}$$

After giving weighting to words based on the abovementioned method, we remove additional words. At this stage, we try to remove the words which are not influential in the main content of the text. Omitting extra words such as preposition, adverbs, adjectives and so on do not negatively influence general content of the text. Then, at the next stage, we assign a weight to each file in all classes.

3.1.3. Calculating the weighting of each file in all classes

To do this, we separate words which are extracted for a class and relate to file i, then we study the rest of the files which exist in this class, we extract words which also exist in file i and we get the weighting sum of the extracted words. In this way the weight of file i is calculated in this class. According to the abovementioned process, we calculate the weight of all files in all classes and then we transfer the file to the class that has a higher weight. Therefore, a new classification is created and F is calculated again and if the amount of F increases, algorithm will enter the cycle and will return to the weighting stage. This will continue up to the time when F is increasing and if F decreases or remains constant, we will choose the classification whose F measure has the best situation and algorithm will end at this point.

IV. Analysis and Evaluation

To study the efficiency of the suggested algorithm from the existing documents in dataset SFU-Review-Corpus-Raw, four classifications of computer, music, machine and book were chosen and each class contains 50 text files. The results of the algorithm were evaluated with different preliminary classifications. The method which was used for giving weighting to features is our suggested method and the results are as the followings:

performance	Computer early	Computer final	Book early	Book final	Music early	Music final	Machine early	Machine final
1	0.97	0.97	0.97	0.98	0.95	1	0.96	0.96
2	0.89	0.91	0.79	0.97	0.85	0.94	0.80	0.93
3	0.59	0.89	0.58	0.89	0.55	0.83	0.62	0.84
4	0.23	0.78	0.28	0.89	0.45	0.78	0.34	0.76

As it can be seen from table one, with different random classification, algorithm again has high efficiency in classifying documents. By having better early classification, certainly, we will achieve better results in this algorithm. However, if an early random classification has low F measure, again, algorithm produces \\\\acceptable results. Since in this algorithm, the quality of the created classifications is controlled, moreover, a weight is calculated for text files and file is transferred to a better class. Finally, the new created classification replaces early random classification; therefore, the influence of early random classification reduces. To better study the results, we calculate the mean of F measure, the results are as the followings:

Graph 1: Different states of performing suggested algorithm with different random classification

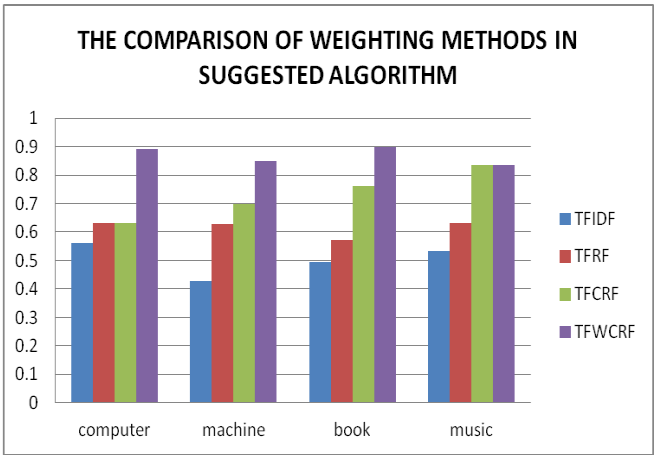
As we can see from graph 1, suggested algorithm produces good results even in the early random classification which contains low F. Now, we evaluate the influence of different weighting methods on the output of the algorithm. The results are as the followings (graph 2):

Table 2: the comparison of the influence of different weighting methods on the output of the suggested algorithm for document classification

Weighting methods	computer	machine	book	music
TFIDF	0.561	0.429	0.494	0.532
TFRF	0.632	0.629	0.573	0.631
TFCRF	0.632	0.698	0.763	0.834
TFWCRF	0.893	0.849	0.899	0.834

doi : 10.17950/ijset/v5s5/519

International Journal of Scientific Engineering and Technology
Volume No.5 Issue No.5, pp: 296-300
ISSN:2277-1581
01 May 2016



Graph 2: the results of the comparison of weighting methods in suggested algorithm for text classification in different classes.

Finally, the suggested algorithm for text classification (CFW) was compared with SVM, K-NN, C 4.5, Bayes on Reuters dataset. Evaluation criterion $\frac{precision}{recall}$ is used and the results show the high capability and potential of this suggested algorithm. (graph 3)

Graph 3: the results obtained from the comparison of suggested classification methods with suggested method on Reuter's dataset.

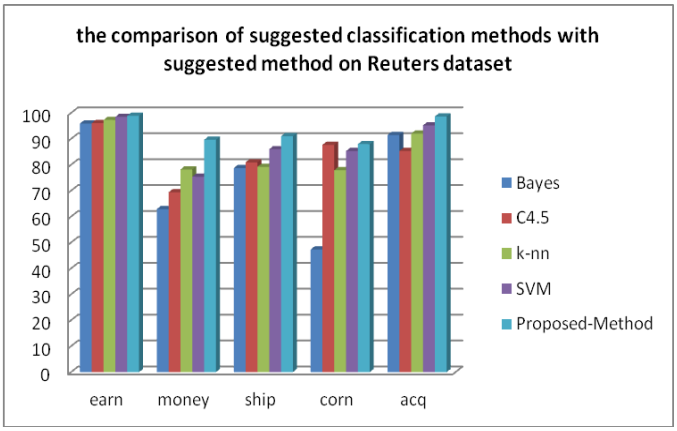


Table 3: the comparison of the document classification methods

Classification methods	earn	money	ship	corn	acq
Bayes	95.9	62.9	78.7	47.3	91.5
C4.5	96.1	69.4	80.9	87.7	85.3
k-nn	97.3	78.2	79.2	77.9	92
SVM	98.5	75.4	86	85.3	95.2
Proposed-Method	98.9	89.65	91.02	87.98	98.637

v. Conclusion

The results show high efficiency of suggested weighting method and high capability of suggested algorithm in text classification. In text classification, if we have classification view in preprocess section, the results obtained from preprocess stage results in improved classification. Since provided algorithm enters iteration and development cycle and is performed several times with different classes it results in excellent efficiency. Also, file replacement and its transfer to more appropriate classes are among good characteristics of suggested algorithm and more satisfactory results have been achieved.

Moreover for the future research, it is suggested that Anthologies be used for achieving better results, it means, feature concepts can be used for document classification in order to get conceptual classification.

References

i. *Feldman, R. and Dagan, I.; Kdt - knowledge discovery in texts; In Proc. of the First Int. Conf. on Knowledge Discovery (KDD); pages 112–117 ; 1995.*

ii. *Zhuge, H. et al ; An Automatic Semantic Relationships Discovery Approach ; The 13th International World Wide Web Conference (WWW2004) ; 2004.*

iii. *Joachims, T. ; Text Categorization with Support Vector Machines: Learning with Many Relevant Features ; In European Conference on Machine Learning (ECML) ; 1998.*

iv. *Apte, C., Damerau, F., Weiss, S. ; Text Mining with Decision Rules and Decision Trees ; The Conference on Automated Learning and Discovery, Workshop 6:Learning from Text and the Web ; 1998.*

v. *Creecy, R.M. et al., ; Trading MIPS and Memory for Knowledge Engineering: Classifying Census Returns on the Connection Machine ; Communications of the ACM, Vol. 35, No. 8, pp. 48–63 ; 1992.*

vi. *S. B. Kiml, K. S. Han, H. C. Riml and S. H. Myaeng, ; Some effectivetechniques for naive Bayes text classification ; IEEE Transactions onKnowledge and Data Engineering, vol. 18, pp. 1447-1466, Nov ; 2006.*

vii. *Aggarwal, Ch. C. ; Xiang Zhai, Ch. ; Mining Text Data ; Springer ; 2012.*

viii. *Ramakrishna Murty, M. ; Murty, JVR . ; PrasadReddy PVGD ; Text document classification based on least square support vector machine with singular value decomposition ; published IJCA, Vol 27,No-7 ;2011.*

ix. *Maleki, M. ; Abdollahzadeh, A. ; TFCRF: A Novel Feature Weighting Method Based on Class Information in Text Categorization (revised version); accepted in the 31st Annual Conference of the German Classification Society on Data Analysis, Machine Learning, and Applications (GFKL 2007) ; 2007.*

x. *Robertson, A.M. and Willett,P., ; An Upperbound to the Performance of Ranked-Output Searching:Optimal Weighting of Query Terms Using a Genetic Algorithm ; Journal of ocumentation, Vol. 52, pp.405–420 ; 1996.*

xi. *Gupta, V. ; Gurpreet, S. ; A Survey of Text Mining Techniques and Applications ; Journal of Emerging technologies in web intelligence ; vol,1 no1; 2009*