World of Computer Science and Information Technology Journal (WCSIT)
ISSN: 2221-0741
Vol. 1, No. 3, 92-95, 2011

Using MI Method for Feature Weighting to Improve Text Classification Performance

Morteza Zahedi
Department of Computer Engineering and IT
Shahrood University of Technology
Shahrood, Iran
zahedi@shahroodut.ac.ir

Aboulfazl Sarkardei
Department of Computer Engineering and IT
Shahrood University of Technology
Shahrood, Iran
ab.sarkardei@shahroodut.ac.ir

Abstract— In text classification, feature weighting is a main step of preprocessing. Commonly used feature weighting methods only consider the distribution of a feature in the documents and do not consider the class information for feature weighting. Mutual Information (MI) method which represents the dependency of a feature in the regarding class, has been previously used for feature selection. The aim of this paper is to show that the use of MI method for feature weighting increases the performance of text classification, in terms of average recall and average precision. While K-nearest neighbor classifier is employed for classification, the average recall is increased about 18% and average precision is increased about 10%. It is shown that the results for average precision and average recall become 91.7% and 89.29% respectively.

Keywords- text classification; mutual information; MI; feature weighting; Hamshahri; K-nearest neighbor.

I. INTRODUCTION

Due to vast availability of texts in digital form and the increasing need to access them in flexible ways, text classification becomes a crucial task. The goal in text classification is to classify the texts into some predefined classes. Depending on the words used in a text, which are considered as the features of the text, we determine the class of each new text belongs to. There are two important issues in text classification, which are feature selection and weighting (1) and classification method (2). Best features for classification are selected based on the discriminating characteristic of those features. Another important issue to be considered in text classification is the classification method. For example, when we have a text that can fall into two different classes, it is not possible to use all kind of classification methods, and on the other hand, a perfect choice of classification method can be very effective in increased speed and accuracy of the classification process. In the past several years, many methods based on machine learning and statistics have been applied to text classification. Among those methods, decision trees [1], k-nearest neighbors (k-NN) [2-5], neural networks [6], Naïve Bayes classifier [7-8] and support vector machines (SVM) [9] are of successful examples.

Feature selection plays an important role as a filter for inappropriate features to reduce the number of input features. There are various methods for feature filtering, e.g. document frequency thresholding (DF), information gain (IG) and term straight (TS), from which DF is used in this paper.

Feature weighting as one of the preprocessing techniques in the text classification process, has a valuable role in achieving both high quality indexing and good classifiers. In this paper, mutual information (MI) that has been previously used for feature selection is used as a feature weighting method in text classification.

MI is a method mostly used in statistical approaches. As it is mentioned earlier, in text classification, MI method is used as a feature weighting tool here. MI value of a feature in a class represents the dependency of that feature in the regarding class, thus indicating the importance of the feature in that class. So to measure the fitness of a feature in a specific class, the MI of that feature will be calculated for that class during the feature selection phase. This value is then used as the weight of the feature.

MI considers the distribution of the features in different classes while weighting each feature in each class. In MI method the value of each feature in each class is calculated and in this paper this value is used for feature weighting

based on class dependency. The aim of this paper is to show that using MI method for feature weighting while K-nearest neighbor classifier is employed for the classification, increases the performance of text classification, in terms of average recall, average precision.

The reminder of the paper is organized as follows: section II describes feature extraction. In this section steps in text classification is explained. Section III discusses feature weighting where some feature weighting methods and MI method that has been previously used for feature selection and now is used as a feature weighting method are explained. Section IV explains evaluation measures used in this paper. In section V, k-NN algorithm is described and section VI explains the experiment details. In this section, a data set that is selected randomly from Hamshahri corpus database is described and the test set and the train set is introduced. In section VII, the results that are obtained by using explained method and the data set, which is described in its previous section, are presented and the paper concludes in section VIII.

II. FEATURE EXTRACTION

In general, text classification can be considered as a process of classifying a text into predetermined classes and usually consists of some steps: preprocessing, indexing and weighting. In the preprocessing step, a text associated with its related characters is changed to a proper representation form for the learning and classification algorithms.

Step 1, preprocessing: The first step is the preprocessing of the datasets, where documents are parsed, non-alphabetic characters and tags including XML are discarded, and stop words (for word features) are eliminated. Stop words are those words that repeat in the text and do not include any useful information. We use the list of 61 stop words. Using a stop-list significantly reduces the feature vector size and the memory requirements of the system [10].

Step 2, filtering: Since some features are appeared in the text either rarely or more than usual, a threshold value is often used for removing those features [11]. In this paper, we have used document frequency (DF) threshold method which is used for feature selection because it has been shown that DF threshold is the simplest method with the lowest cost in computation, especially when the computation cost of these measures are too expensive [12].

Step 3, feature weighting: One of the main preprocessing steps for having a precise text classifier is feature weighting. Commonly used feature weighting methods, such as TF and IDF-based methods, only consider the distribution of a feature in the documents discarding class information. In this paper, MI method that is commonly used for feature selection in text classification is now considered as a weighting method in which the weight of each feature in class X shows the power of that feature to discriminate class X from the other classes.

III. FEATURE WEIGHTING

One of the main preprocessing steps for having an accurate and fast text classifier is feature weighting. Commonly used feature weighting methods only consider the distribution of a feature in the documents and do not consider class information for the weights of the features. Several methods were reported for feature weighting to be based on such as term frequency (TF) [13-14], inverse document frequency (IDF) [15-18], category concept and other concepts [19-20]. As an example, the weight of a feature in IDF-based weighting methods has an inverse relationship with the number of documents containing that feature. When the total number of documents containing a specific feature increases, the capability of that feature in discriminating documents from each other and so the weight of it decreases. Although this is a right assumption in information retrieval (IR) domain, it needs some modifications for being used in domains of text categorization. As a matter of fact, when the number of documents containing a specific feature t_k increases and most of those documents belong to class C_j, feature t_k not only is not inappropriate feature for that class instead is one of the powerful features for discriminating class C_j from the other classes. Hence, feature t_k should produce a high weight in class C_j. On the other hand, if the number of classes except class C_j containing a feature t_k increases, the weight of feature t_k in class C_j should decrease. Consequently, the IDF factor used in feature weighting methods needs some modifications to consider these two aspects.

In this paper, MI method that is commonly used for feature selection in text classification is now considered as a weighting method in which the weight of each features in class X shows the power of that feature to discriminate class X from the other classes.

In first step, TF weighting method is evaluated and used for MI method and after that, in second step, the weights that are produced with MI method is used in k-NN algorithm.

Since in MI method there are weightings for each feature in each class, for using k-NN algorithm and MI method together the k-NN algorithm need some changes. In other words, in train set, the weights based on class dependency are allocated and after that, before the distance of a test set is calculated from a train set, initially, the test sample is weighted based on the class the train sample belongs to. This process is applied to all test samples before calculating the Euclidian distance between each two pair of train and test samples. The class of the train sample which has the minimum distance from the test sample is considered as the class of the test sample.

TF weighting method is one of the simplest methods used for feature weighting in which the weight of feature t_k in document d_i is equal to the frequency of that feature in the vector of the document as shown in (1).

$$W_{ki} = tf(t_k, d_i) \begin{cases} \#(t_k, d_i) & t_k \in \text{vector of } d_i \\ 0 & t_k \notin \text{vector od } d_i \end{cases} \quad (1)$$

where $\#(t_k, d_i)$ is the frequency of feature t_k in document d_i.

In MI method, the weight of feature t_k in class c_i is evaluated by using (2):

$$MI(t_{k,}c_i) = \log \frac{A*N}{(A+C)*(A+B)} \quad (2)$$

where A is equal to the frequency of t_k in class c_i and B is equal to the frequency of t_k in other classes and C is the number of features in class c_i except t_k and N is the number of documents.

$MI(t_{k,}c_i)$ shows the power of t_k in class c_i thus by using MI method for feature weighting, dimension of weighting matrix is equal to the number of classes multiplied by the number of features.

IV. Evaluation Measures

Precision and recall measures are widely used for evaluation of classification tasks. They are defined as follows in (3) and (4):

$$\text{Precision} = \frac{\text{Correct assignments}}{\text{total number of assignments}} = \frac{TP}{TP+FP} \quad (3)$$

$$\text{Recall} = \frac{\text{Correct assignments}}{\text{total number of correct assignments}} = \frac{TP}{TP+FN} \quad (4)$$

where TP is the number of documents correctly assigned to a category. FP is the number of documents incorrectly assigned to a category. FN is the number of documents incorrectly omitted from a category.

In this paper, average precision and recall measures are used and their equations are as following in (5) and (6):

$$p^A = \frac{\sum_{j=1}^{|c|} p(c_j)}{|c|} \quad (5)$$

$$r^A = \frac{\sum_{j=1}^{|c|} r(c_j)}{|c|} \quad (6)$$

In the equations above, |C| is referred to the number of classes.

V. K-Nearest Neighbor Method

K-NN which is a modified form of nearest neighbor classifier stands for k-nearest neighbor classification. Considering an arbitrary input document which has to be classified, the system ranks it into the class of its most similar document among the all training documents, i.e. nearest neighbor method. In order to avoid some usual mistakes in nearest neighbor classifier, we use the categories of the k top-ranking neighbors for category indication of the input document. It has to be mentioned that the similarity of each neighbor candidate to the new document which has to be classified is the weight of each category it belongs to. Due to the use of k-NN, the sum of category weights over the k top-ranking nearest neighbors is used for each category.

VI. Exprimenting the Proposed Method

In our experiments the Hamshahri corpus database is used, a collection of 190,206 articles covering the following subject categories: politics, city news, economics, reports, editorials, literature, sciences, society, foreign news, sports and etc. To evaluate the proposed feature weighting method, 603 random articles from five categories have been selected as our document dataset. The name of categories and the number of articles in each category can be seen in Table I.

Vector space model is used for document indexing in this experiment. Stop words, tags, punctuations and numbers have been removed. The number of unique features (vocabulary) is 13516.

Document frequency (DF) threshold method is also used for feature selection because it has been shown that DF threshold is the simplest method with the lowest cost in computation, when the computation cost of these measures are too expensive. In the next step, with defining a threshold, those features that are considerable lower than that threshold are removed. After the above step, the number of unique features (vocabulary) is reduced to 6165.

This document database, D, is partitioned into a training set (TrainD) of 402 documents and a testing set (TestD) of 201 documents. In this step, based on the MI weighting method, the weights are allocated to each of the documents in TrainD classes while weights are also allocated to each of the documents in TestD in k-NN algorithm.

TABLE I. Name of categories and the number of articles

Numbers	Category name
109	Literature and Art
113	Miscellaneous, Happenings
120	Economy, Bank and Bourse
130	Social
131	Politics

VII. Experimental Results

In this paper, the introduced database is used to evaluate MI method as a feature weighting tool. The average accuracy and average recall of the approach is evaluated using k-NN classifier and TF weighting method. As shown in Table II, the average accuracy is 71% and average recall is 81%. In the second part of the process, MI method is applied to the data as a weighting method and the data is then classified using k-NN algorithm. After the second processing step is applied, as it can be seen in Table III, the MI method applied to the same data increases both the average accuracy and average recall to 91% and 89%, respectively.

TABLE II. Obtained results of k-NN method

Method	Average Recall	Average Precision
1-NN	0.7105	0.814

TABLE III. Obtained results of k-NN method by using MI method

Method	Average Recall	Average Precision
1-NN	0.8929	0.9171

As it can be seen in Table II and Table III, by using MI method for feature weighting, performance of text classification in terms of average recall and average precision is increased.

VIII. CONCLUSION

In this paper we have tried to introduce MI method as a weighting method while K-nearest neighbor is employed for the classification. MI method considers the distribution of the features in different classes and weights each feature in each class. In MI method, the value of each feature in each class is produced and in this paper this value is used for weighting features based on class dependency. The obtained results indicates that proposed method is able to particularly increase the performance of the text classification in terms of average recall and average precision.

IX. FUTURE WORKS

In this paper, by evaluating the MI method as a feature weighting tool, the increased performance of text classification is observed in terms of average accuracy and average recall. Yet, the position of the word in the text is not considered in the weighting process, so all the words are considered with the same positional value. This research is expandable by considering the position of the words in the text as a weighting feature in the text classification combined with MI method. This new approach can also be tested on a more comprehensive database for a more precise result. Another expansion to this algorithm is to find a solution for texts that fall into two or more classes at the same time. And finally, to reduce the number of features, one can use genetic algorithm as a feature selector for the proposed text classifier.

ACKNOWLEDGEMENT

The authors want to thanks Mr. Ali Reza Manashty for his kind support and for editing this article for publishing.

REFERENCES

[1] DD.Lewis and M.Ringuette "Comparison of two learning algorithms for text categorization", In Proceedings of the third annual symposium on document analysis and information retrieval (pp. 81-93). Las Vegas, NV,1994

[2] TM.Cover, PE.Har."Nearest neighbor pattern classification",IEEE Transaction on Information Theory IT, 13(1), 21–27,1967

[3] STan,"Neighbor-weighted K-nearest neighbor for unbalanced text corpus", Expert System with Applications, 28(4), 667–671,2005

[4] Y.Yang, "An evaluation of statistical approaches to text categorization", Information Retrieval, 1(1), 69-90,1999

[5] Y.Yang and CG. Chut ."An example-based mapping method for text categorization and retrieval", ACM Transactions on Information System, 12(3), 252–277,1994

[6] ED.Wiener, JO.Pedersen and AS.Weigend,"A neural network approach to topic spotting", In Proceedings of SDAIR-95, 4th annual symposium on document analysis and information retrieval (pp. 317–332),1995

[7] DD.Lewis,"Naive Bayes at forty: The independence assumption in information retrieval", In Proceedings of the 10th European conference on machine learning (pp. 4–15). New York: Springer,1998

[8] A.McCallum and K.Nigam," A comparison of event models for naive Bayes text classification", In AAAI-98 workshop on learning for text categorization,1998

[9] T.Joachims,"Text categorization with support vector machines: Learning with many relevant features", In Proceedings of the 10th European conference on machine learning (pp. 137–142). New York: Springer,1998

[10] C.Manning,P.Raghavan and H.Schutze,"Introduction to Information Retrieval", Cambridge University Press,2008

[11] M.Maleki," Optimizing Information Discovery from Semi-Structured XML Documents",Master Thesis,2005

[12] Y.Yang and J.Pedersen,"A comparative study on feature selection in text categorization",International Conference on Machine Learning (ICML97), pp. 412-420,1997

[13] K.Sparck Jones,"Indexing term weighting, Information Storage and Retrieval", vol. 9, pp. 619–633,1973

[14] E.Leopold and J.Kindermann,"Text categorization with support vector machines. how to represent texts in input space", Machine Learning, vol. 46, no. 1-3, pp. 423-444,2002

[15] S.Robertson,"Understanding Inverse Document Frequency: on Theoretical Arguments for IDF", Journal of Documentation, Vol. 5, pp. 503–520,2004

[16] G.Salton and C.Buckley,"Term-weighting approaches in automatic text retrieval, Information Processing and Management", vol. 24, no. 5, pp. 513–52,1988

[17] G.Salton, J.Allan and A.Singhal,"Automatic text decomposition and structuring, Information Processing and Management", vol. 32, no. 2, pp.127–138,1996

[18] J.Zhang and TN.Nguyen,"A new term significance weighting approach", Journal of Intelligent information system, vol. 24. no. 1, pp.61-85,2005

[19] S.Hassan, R.Mihalcea, and C.Banea," Random-walk term weighting for improved text classification", IEEE International Conference on Semantic Computing (ICSC 2007), Irvine, CA, 2007

[20] R.Jin, YC.Joyce and S.Luo," Learn to weight terms in information retrieval using category information", The 22nd International Conference on Machine Learning (ICML2005), Germany, Aug 7-11, 2005

AUTHORS PROFILE

Morteza Zahedi has received the B.Sc. degree in computer engineering (hardware) from Amirkabir University of Technology, Iran, in 1996, the M.Sc. in machine intelligence and robotics from University of Tehran, Iran, in 1998 and the Ph.D. degree in man-machine interaction from RWTH-Aachen University, Germany, in 2007, respectively. He is currently an assistant professor in Department of Computer Engineering and IT at Shahrood University of Technology, Shahrood, Iran. He is the Head of Computer Engineering and IT Department. His research interests include pattern recognition, sign language recognition, image processing and machine vision.

Aboulfazl Sarkardei is a M.Sc. student of artificial intelligence at Shahrood University of Technology, Shahrood, Iran. He also received his B.Sc. degree in software engineering from Shahrood University, Shahrood, Iran, in 2010. He has been researching on network security and text classification 2009. His research interests include pattern recognition, text classification and network security.