Improving Feature Vector by Words' Position and Sequence for Text Classification

Poonia Taheri Makhsoos
Computer Engineering Faculty, Iran University of Science and Technology, Iran
po_taheri@comp.iust.ac.ir

Mohammad Reza Kangavari
Computer Engineering Faculty, Iran University of Science and Technology, Iran
kangavari@iust.ac.ir

and

Hamid Reza Shayegh
Software Engineering Department, Tarbiat Modares University, Iran
shayegh@modares.ac.ir

***Abstract*– Automatic document classification is an important topic in computer science because of its various applications in data mining and information technology. Most of these applications need a fast and robust method with low error rate and achievement of this goal is still a challenge. Document classification includes different parts such as text processing, feature extraction, feature vector construction and final classification. Thus improvement in each part should lead to better results in document classification.**

In this paper, we propose a novel method that report better results in text classification by improvement of text feature vector. In this method we focus on two important elements that can highly demonstrate differences between documents' content. One of these elements is the position of each word (after removing stop words and doing word stemming) in document and the other is their sequences. We achieve to these goals without involving in problems of phrases and native language processing.

We test this method on INEX and Reuters datasets and compare this with other existing methods such as [GuoDong2007], [Miller2000], [Zhao & Grisman2005], [Kambhatla2004] and [Roth and Wih2002]. Our method approximately improves precision of classification reported on these datasets about 3.5 %.

***Keywords*– Feature Vector, Data Mining, Text Classification, Features**

I. INTRODUCTION

In the future, electronic documents will be one of the main tools in writing communications. Then, books, papers, magazines and etc. will be only regarded as historical tools [1]. The main problem in nowadays communications is not lack or shortage of information, but, is the lack of extraction and analyzing methods for automatic understanding. Only, if a human editor tracks all of current flows or read the text manually, he can know the type of a document or document flow. This method is not useful for high information volumes and in modern information systems with high complexity [2].

Document classification, i.e. putting documents in some known classes based on context, is one of the important problems in datamining [2]. Realtime arrangement of E-mails or files in hierarchical folders [1], detection of document topic [2], structural search and finding documents according to user favorites [3], are some applications in document classification. Using them in information systems, such as document management systems is crucial.

In this paper, we propose a new method for adding location and arrangement of words (or grams) in text to the feature vector that consequently leads to improvement of document classification, and thus better result is reached.

In section 2 the basic process of text classification and related definitions is presented. Then in section 3 different methods of feature extraction, feature vector construction and so on will be presented as related works. In section 4 we propose our method for extraction of position and arrangement of words or grams. The experimental results are shown in section 5 and finally a conclusion of our work is presented in section 6.

II. DOCUMENT CLASSIFICATION PROCESS AND DIFINATION CONCEPTIONS AND EXPRESSIONS

The E-document classification method has three main parts:

- *Text analyzing and extraction of suitable features from document.*
- *Feature vector construction for each document. Features are contributed with different weights.*
- *Using of classification algorithms on feature vectors in both training and test phases.*

Fig. 1 shows the whole process of document classification. In the first phase training documents arrive into feature extraction section and necessary features are extracted. These features must have useful information about document, such that both distinction and similarity of documents can be recognized [2, 4]. Then we organize feature vectors based on extracted features. In this section it may be necessary to reduce feature vector length for better classification process with better results and less complexity. In the next section, the feature vector will be constructed for test documents. Then, these vectors will be attributed to available classes compared to training vectors and best suitable class based on classification method (SVM, K-NN, mlp, etc) is selected.

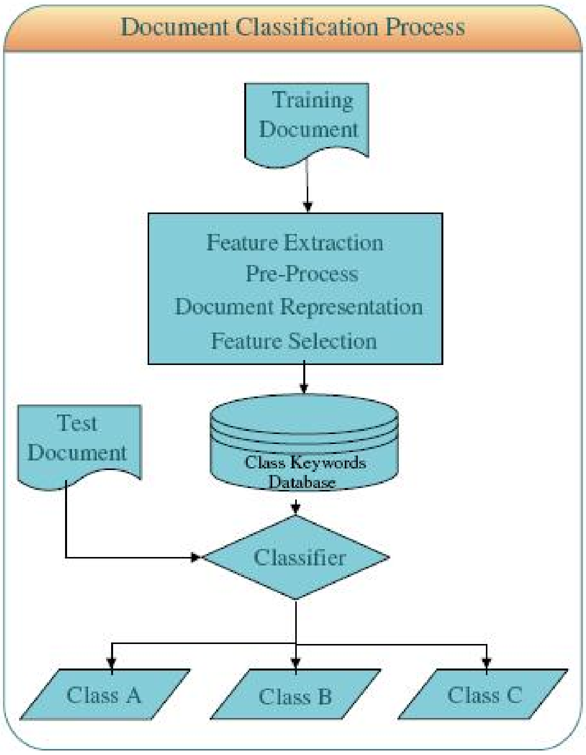

Fig. 1 Basic document classification process

III. FEATURE EXTRACTION AND CLASSIFICATION METHODS

In this section, we introduce different feature extraction and feature vector construction methods.

Document Representation Methods

The most important case in feature vector extraction phase is the type of document representation. It means that which information should be selected from document in order to represent the whole document as a feature vector so that more accuracy in classification is gained.

The proposed document representation methods until now are introduced in below:

- *N-Grams: in this method, all of text characters divide into n-group characters including spaces [6]. This is the simplest way.*
- *Single word: this is the most common text representation. Words of a document can be considered as feature vector elements, weighted by a selective method. The most dramatic problem is that word stemms, consider as different words in vector [2].*
- *Word stems: this method, in fact is the improvement of previous one. The words that are stems together are consider as a unique element and thus weight in unision [7].*
- *Phrase: some researchers in this field believe that if we use similarity between phrase semantics instead of single words, the classification process will be more reliable and more efficient. This method tries to detect phrase elements such as verbs, subjects, and etc. Then compares them with corresponding sections of other texts. This method does not have technical simplicity in many cases [6].*
- *RDR representation: in this method, phrase and expressions in document are related together as some logical rules [6].*

Weighting Feature Vector Elements

After feature extraction based on one of above-mentioned methods, we must find and use a weighting method for weighting features according to selected extraction method. Such important methods are:

- *Weighting method based on TF: in these methods, the weights of features are a function of feature distribution in each document* d_i *∈ D. the weight of each feature is a function of frequency of feature occurrence in selected document. Some of them are: Binary TF, Pure TF, Norm TF, Log TF, ITF and Sparc[5].*
- *The IDF based method: in this method, the feature weights are a function of feature distribution in a set of documents D. some important methods are: classic IDF, TFIDF, NormIDF, etc [5, 8].*
- *Weighting based on classes information: in these methods the distribution functions for each word is considered in all classes, separately. These are: TFRF, TFCRF, LBTF [5, 6, 8]*

Feature Reduction Methods

After feature weighting, we may need a feature reduction phase. It means that we reduce the feature vector dimension for better classification in much less time. Some of feature reduction methods are: DF, IG, MI, TS, X2, and etc. [9, 5].

Feature Vector Analyzing and Classification

After feature vector construction, we can classify documents in known or unknown classes. The most important methods for flat documents (against structural docs) are:

- *Statistical classifiers: these methods, such as Bayesian and regression are the simplest and the weakest methods [9, 5 and 2] that make a decision based on statistics.*
- *Instance based classifiers: these classifiers such as k nearest neighbors are some learner algorithms that only work in test phase and compare feature vector with available vectors without more process [5, 9].*
- *Decision trees: classifiers based on decision trees, contain a tree with nodes labeled by words. The weight of each word is set before any process on this tree [10].*
- *Expert systems: these systems are based on a set of manual rules that are stored in a knowledge base and use some deduction rules for making decisions [11].*

- *Neural networks: these are machine learning methods that make a strong approach for estimation of real functions. These are best method when the problem space is very large [5].*
- *Support vector machines: the main goal in SVMs is finding all plots in n-dimension space that can separate positive and negative samples. SVMs give the best results in document classification processes in many cases [12, 5].*

IV. THE PROPOSED METHOD FOR FEATURE VECTOR IMPROVMENT

In this section, we propose our method that improves feature extraction phase and thus improves classification process.

Basic Idea and Selection Reasons

As mentioned above, more extraction and representation methods are based on words. In all of these methods, each word is considered as a single element and the logical relation between words are not important. We want to consider this relationship between words by a new method.

One of the suitable features that can present logical relationship between words in different documents is the location of word in document. Consider a document that contains the words: “visit”, “framework”, “country” for example. If these words are in the first paragraph of document, probably this is a document about sport news such as:

“The visit of these treams in the framework of country matches was a draw game”

Now consider another document that contains almost all of these words, but in the last paragraph. Such as:

“In the conclusion of this visit, the basic frameworks of country had been changed”

This document is probably about politics.

This example shows that only with consideration of words, we cannot demonstrate type of document well, but if we use the location feature also, there can be a better decision.

Another suitable feature that extraction of it can improve classification process, is the arrangement of words in each document. Maybe two documents have many similar words that are in the same location (first paragraph for example) but their type of arrangement differ the meaning of documents. Consider these two phrases:

“Open the windows and off the source of heat”

“Windows is not an open source system”

The words in these two texts are very similar. In addition both of them can be in the first paragraph of their documents. But the arrangements of words in phrase are very different and these, make two documents of different types.

In this paper we notice these two important options that do not have mentioned before.

Extraction of Location Features in Documents

The proposed process for extraction of word location in documents as useful features is shown in Fig. 2. This process is:

- *First length of each document is calculated by counting words or grams in text.*

$$\forall d_i \in D, 1 < i < n \quad N_i = \sum_{j=1}^{k} t_k$$

- *The average of document length is calculated for all of training documents.*

$$A_n = \frac{\sum_{i=1}^{n} N_i}{n}$$

- *For each word* t_j*. In the document* d_i*, the location value is computed with this formula:*

$$\forall d_i \in D, 1 < i < n, \qquad \forall 1 < j < k \ \ pt_j = \left[n * \left({}^{A_n}\!/_{N_i}\right)\right]$$

After calculating these values, we must arrange them in feature vectors for all documents. For this, according to experimental results, the best way is to calculate the average number of …. Digits in all documents for each feature and then increase the value of this feature 1/average.

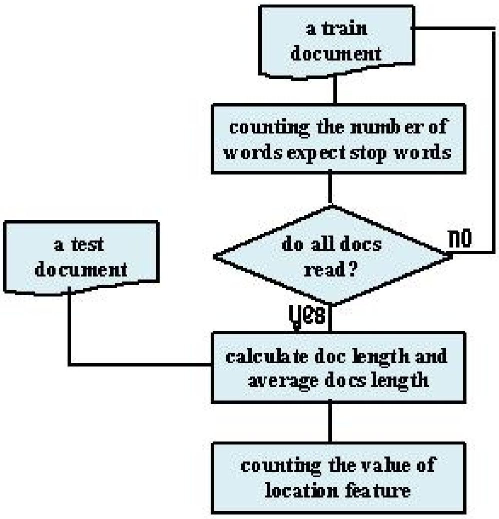
a train document
counting the numb er of words expect stop words
do all docs read?
no
yes
a test document
calculate doc length and average docs length
counting the value of location feature

Fig. 2 Extraction of location feature

Extraction of Word Arrangement in Documents

The essential idea in the proposed method of this section is the information that one can extract from data histograms for each document. In this method, we plot a histogram for each document based on the arrangement of words and based on outcoming histograms, suitable features are added to the feature vector.

The process is that we consider a two dimensional vector. The x-axis is the location of words in feature vector and the y-axis is the location of each word in selected document. After calculating all of locations, we have a histogram that its form shows the type of arraignment of words in document. Fig. 3 show a sample of sequence histogram for three document.

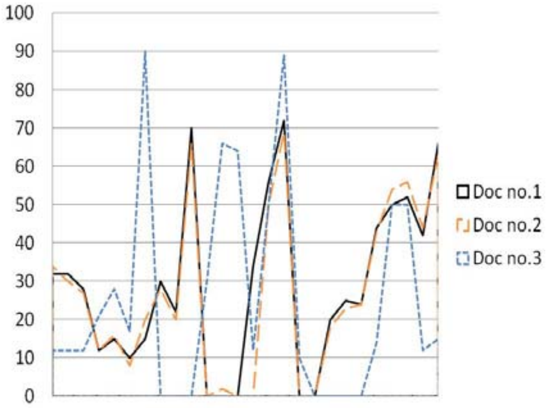

Fig. 3 The word sequence histogram

The Complete Process of the Proposed Method

First, we choose some documents as training data and extract all of words other than stop words (conjunctions, prepositions, etc). Then we use a stemming method and consider stem words as a unique feature. Then we use TFCRF method for weighting words and add features of locations and arrangements of words by using proposed method. Finally, for all training documents feature vectors are produced. At the end we use a classification method such as SVM and K-NN for classification of test documents based on training features.

Fig. 4 shows the pseudo code of complete classification process.

```
For All Training Documents do
{
  Extract all words or grams of a document
  Removes stop words
  Find stemming words & merge them
  Weight words based on a weighting method
  Construct feature vectors
  Extract position number of all words in a document
  Updates feature vectors
  Construct ordering histogram for each document
  Convert histogram to numbering values
  Update feature vector
}
For a test document do
{
  Construct feature vector with updates
  Compare feature vector with training feature space
based on a classification method
  Guess class of document}
```

Fig. 4 Pseudo code of complete classification process

V. EXPERIMENTS AND RESULTS

We test our method on INEX dataset that is considered in some IEEE papers. So comparison of our results with other similar methods is possible.

Measurement Criteria

To evaluation of our method performance, we use three criteria: precision, recall and accuracy. Table 1 shows different aspects of results that may be produced. We use this information for calculating the criteria. The rows show results of classification process for each class and the columns show real information.

The formulas are shown below:

- *Accuracy criterion: This value shows the accuracy of classification method. It means how this method can detect the classes of document more accurate.*

$$Ac(c_j)=\frac{TP(c_j)+TN(c_j)}{TP(c_j)+FP(c_j)+TN(c_j)+FN(c_j)}$$

TABLE I
ALL POSSIBLE CLASSIFICATION STATUS

Real belonging to C_i		Class C_i	
no	Yes		
FP	TP	yes	The prediction to belonging to class C_i
TN	FN	no	

- *Precision criterion: this formula shows the number of correct decisions for each class.*

$$\Pr(c_j)=\frac{TP(c_j)}{TP(c_j)+FP(c_j)}$$

- *Recall criterion: this formula shows the number of decisions for each class.*

$$\mathrm{Re}(c_j)=\frac{TP(c_j)}{TP(c_j)+FN(c_j)}$$

Experimental Results for Different Sample Numbers and Different Feature Vector Lengths.

Table 2 shows the results of experiments for different methods and different feature numbers. Fig. 5 shows another presentation of these results.

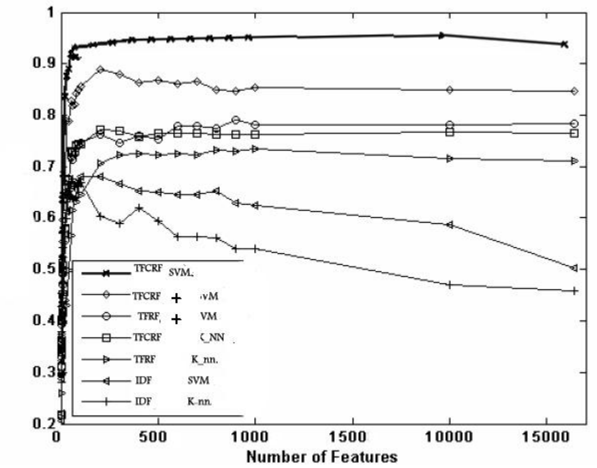

Fig. 5 The results of different classification algorithms

As mentioned below, the proposed method is tested for 800 samples using a feature vector with a 10000 element length; we gain to 96.6% for accuracy.

TABLE II
THE RESULT OF DIFFERENT CLASSIFICATION ALGORITHMS FOR DIFFERENT FEATURE VECTORES AND DIFFERENT SAMPLES

Feature Vector Length	Method	Sample Number	Accuracy	Precision	Recall
10	K-NN + TFCRF	400	25%	14%	40%
		800	33%	17%	34%
	SVM + TFRF	400	37%	16%	32%
		800	42%	19%	25%
	SVM + TFCRF	400	68%	19%	55%
		800	59%	20%	63%
	SVM + TFCRF + proposed method	400	88%	19%	25%
		800	81%	20%	34%
100	K-NN + TFCRF	400	44%	33%	39%
		800	47%	37%	51%
	SVM + TFRF	400	29%	36%	66%
		800	67%	39%	81%
	SVM + TFCRF	400	55%	36%	50%
		800	63%	41%	61%
	SVM + TFCRF + proposed method	400	89%	38%	71%
		800	89%	42%	73%
1000	K-NN + TFCRF	400	88%	61%	83%
		800	83%	66%	92%
	SVM + TFRF	400	85%	62%	88%
		800	84%	68%	80%
	SVM + TFCRF	400	89%	75%	89%
		800	93%	82%	95%
	SVM + TFCRF + proposed method	400	95%	78%	91%
		800	91%	91%	96%

10000	K-NN + TFCRF	400	82%	62%	78%
		800	81%	68%	73%
	SVM + TFRF	400	84%	71%	82%
		800	81%	72%	86%
	SVM + TFCRF	400	93%	81%	98%
		800	96%	83%	95%
	SVM + TFCRF + proposed method	400	93%	88%	92%
		800	97%	96%	93%

VI. CONCLUSION

In this paper, we propose a method for improvement of feature vector in document classification. For this goal we extract the location and arrangement of words by a new method and add these values in feature vectors. According to experimental results we get 96% accuracy on INEX dataset with more than 500 samples and feature vector with 5000 element length in average. This results, is better about 2% to 3.5% than similar methods.

The main advantage of this method is that it considers some aspects of documents that are not noticed in previous works.

REFRENCES

[1] M. Maleki, Master Thesis in software engineering, 2005, Amir Kabir University

[2] A. Jalali, F. Oroumchian, Rich document representation for document clustering, RIAO2004, Coupling approaches, coupling media and coupling languages for information retrieval avignon (Vaucluse), France, 2004, pp. 800–808.A. Author 1 and B. Author 2, “Title of the journal paper” *IEEE Trans. Antennas and Propagation*, Vol. 55, No. 1, pp. 12-23, 2007.

[3] E.H. Han, G. Karypis, Centroid-based Document Classification: Analysis and Experimental Results, Springer, 2000.

[4] A. Bratko, B. Filipic, “A Study of Approaches to Semi-structured Document Classification,” Technical Report IJS-DP-9015, Department of Intelligent Systems, Jozef Stefan Institute, November 2004.

[5] F. Raja, M. keikha, F. Oroumchian, M. Rahgozar, Using rich document representation in XML information retrieval, vol. Initiative on the evaluation of XML retrieval (INEX), 2006.

[6] A. AleAhmad, P. Hakimian, F. Oroumchian, N-gram and local context analysis for persian text retrieval, International Symposium on Signal Processing and its Applications (ISSPA2007), Sharjah,United Arab Emirates (UAE), 2007, pp. 12–15.

[7] Z.H. Deng, S.W. Tang, D.Q. Yang, M.Z.h. Li-Yu Li, K.Q. Xie, “A Comparative Study on Feature Weight in Text Categorization,” 6th Asia Pacific Web Conference, Hangzhou, China, April 14-17, 2004.

[8] E. Glover, K. Tsioutsiouliklis, S. Lawrence, D. Pennock, and G. Flake, “Using Web Structure for Classifying and Describing Web,” *11th International World Wide Web Conference (WWW07)*, Honolulu, Hawaii, 2007.

[9] M. Nicholas and P. J. Clarkson (2000), Web-Based Knowledge Management for Distributed Design, *IEEE Intelligent Systems*, pp. 40-47.

[10] W. Lam and C. Y. Ho, “Using a Generalized Instance Set for Automatic Text Categorization,” *21st ACM International Conference on Research and Development in Information Retrieval (SIGIR98*), pp. 81 89, Melbourne, AU, 1998.

[11] M. Lan, S.Y. Sung, H.B. Low, .C.L. Tan, “A Comparative Study on Term Weighting Schemes for Text Categorization,” *IEEE International Conference on Neural Networks (IJCNN05),* pp. 546-551, 2005.

[12] E. Leopold, J. Kindermann, “Text Categorization with Support Vector Machines. How to Represent Texts in Input Space?,” *Journal of Machine Learning*, vol. 46, no. 1-3, pp. 423-444, 20