CHI SQUARE FEATURE EXTRACTION BASED SVMS ARABIC TEXT CATEGORIZATION SYSTEM

Abdelwadood Moh'd A Mesleh
Faculty of Information Systems & Technology, Arab Academy for Banking and Financial Sciences, Amman, Jordan.

Keywords: Text Classification, Arabic Language, SVMs, CHI Square.

Abstract: This paper aims to implement a Support Vector Machines (SVMs) based text classification system for Arabic language articles. This classifier uses CHI square method as a feature selection method in the pre-processing step of the Text Classification system design procedure. Comparing to other classification methods, our classification system shows a high classification effectiveness for Arabic articles term of Macroaveraged F1 = 88.11 and Microaveraged F1 = 90.57.

1 INTRODUCTION

Text Classification (TC) is the task to classify texts to one of predefined categories based on their contents (Manning and Schütze, 1999). It is also referred as Text categorization, document categorization, document classification or topic spotting. And it is one of the important research problems in information retrieval IR, data mining, and natural language processing.

TC has many applications that are becoming increasingly important such as document indexing, document organization, text filtering, word sense disambiguation and web pages hierarchical categorization.

TC research has received much attention (Sebastiani, 2002). It has been studied as a binary classification approach (a binary classifier is designed for each category of interest), a lot of TC training algorithms have been reported in binary classification e.g. Naïve Bayesian method (Yang and Liu, 1999), *k*-nearest neighbours (*k*NN) (Yang and Liu, 1999), support vector machines (SVMs) (Joachims, 1998) etc. On the other hand, it has been studied as a multi classification approach e.g. boosting (Schapire and Singer, 2000), and multiclass SVM (Vapnik 1998). In this paper, we have restricted our study of TC on binary classification methods and in particular to Support Vector Machines (SVMs) classification method for Arabic Language articles.

The rest of this paper is organized as follows. In section 2, the text classification procedure is described. Experiment results and conclusions are discussed in sections 3 and 4 respectively.

2 TEXT CLASSIFICATION PROCEDURE

The TC system design usually compromise three main phases:

- Data pre-processing phase is to make the text documents compact and applicable to train the text classifier.
- The text classifier, the core TC learning algorithm, shall be constructed, learned and tuned using the compact form of the Arabic dataset.
- Then the text classifier shall be evaluated (using some performance measures).

Then the TC system can implement the function of document classification.

The following sections 2.1, 2.2 and 2.3 are devoted to data pre-processing, text classifier and TC performance measures.

2.1 Data Pre-processing

2.1.1 Arabic Data Set

Since there is no publicly available Arabic TC corpus to test the proposed classifier, we have used an in-house collected corpus from online Arabic newspaper archives, including *Al-Jazeera*, *Al-Nahar*,

Moh'd A Mesleh A. (2007).
CHI SQUARE FEATURE EXTRACTION BASED SVMS ARABIC TEXT CATEGORIZATION SYSTEM.
In *Proceedings of the Second International Conference on Software and Data Technologies - PL/DPS/KE/WsMUSE*, pages 235-240
DOI: 10.5220/0001329402350240

Al-hayat, *Al-Ahram*, and *Al-Dostor* as well as a few other specialized websites. The collected corpus contains 1445 documents that vary in length. These documents fall into Nine classification categories that vary in the number of documents. In this Arabic dataset, each document was saved in a separate file within the corresponding category's directory, i.e. this dataset documents are single-labelled.

2.1.2 Arabic Data Set Pre-processing

As mentioned before, this pre-processing aims to transform the Arabic text documents to a form that is suitable for the classification algorithm. This phase processes the Arabic documents according to the following steps: (Benkhalifa, et.al, 2001) and (ElKourdi, et.al 2004).

- Each article in the Arabic data set is processed to remove digits and punctuation marks.
- We have followed (Samir et.al 2005) in the normalization of some Arabic letters such as the normalization of "ء" (hamza) in all its forms to "ا" (alef).
- All the non Arabic texts were filtered.
- Arabic function words (such as "آخر", "أبدا", "أحد" etc.) were removed. The Arabic function words (stop words) are the words that are not useful in IR systems e.g. pronouns and prepositions.
- The vector space representation (Salton, et.al. 1975) is used to represent the Arabic documents.
- We have not done stemming because it is not always beneficial for text categorization, since many terms may be conflated to the same root form (Hofmann, 2003).

In Vector space model (VSM), Term frequency TF concerns with the number of occurrences a term i occurs in document j while inverse document frequency IDF concerns with the term occurrence in a collection of texts and it is calculated by $IDF(i) = \log(N / DF(i))$, Where N is the total number of training documents and DF is the number of documents that term i occurs in.

In information retrieval, it is known that TF makes the frequent terms more important. As a result, TF improves recall. On the other hand, the inverse document frequency IDF makes the terms that are rarely occurring in a collection of text more important. As a result, IDF improves precision.

Using VSM (Salton and Bucklry, 1988) shown that combining TF and IDF to weight terms (*IDF.TF*) gives better performance. In our Arabic dataset, each document feature vector is normalized to unit length and the *IDF.TF* is calculated.

2.1.3 Feature Extraction

In text categorization, we are dealing with a huge feature space. This is why; we need a feature selection mechanism. The most popular feature selection methods are document frequency thresholding (DF) (Yang and Pedersen. 1997), the X^2 statistics (CHI) (Schutze, et.al, 1995), term strength (TS) (Yang and Wilbur, 1996), information gain (IG) (Yang and Pedersen. 1997), and mutual information (MI) (Yang and Pedersen. 1997).

The X^2 statistic (Yang and Pedersen. 1997) measures the lack of independence between the text feature term t and the text category c and can be compared to the X^2 distribution with one degree of freedom to judge the extremeness. Using the two-way contingency table (Table 1) of a term t and a category c, **A** is the number of times t and c co-occur, **B** is the number of times t occurs without c, **C** is the number of times c occurs without t, **D** is the number of times neither c nor t occurs, and N is the total number of documents. The term-goodness measure is defined as follows:

$$X^2 = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)}$$

This X^2 statistic has a natural value of zero if t and c are independent. Among above feature selection methods (Yang and Pedersen. 1997) found (CHI) and (IG) most effective. Unlike (Joachims, 1998) where he has used (IG) in his experiment, we have used CHI as a feature selection method for our Arabic Text categorization system.

Table 1: X^2 statistics two-way contingency table.

A = #(t,c)	C = #(¬t,c)
B = #(t,¬c)	D = #(¬t, ¬c)
N = A + B + C + D	

2.2 Text Classifier

As any classification algorithm, TC algorithm has to be robust and accurate. Numerous classification algorithms have been implemented for TC tasks. Support Vector Machines proves to be best. Many other classical methods (Mitchell 1996) have been investigated in literatures, e.g. k-NN and Naïve Bayes classifiers.

SVMs is briefly presented in section 2.2.1. k-NN and Naïve Bayes classifiers are briefly presented in sections 2.2.2 and 2.2.3 respectively.

2.2.1 Support Vector Machine

Support Vector Machines (SVMs) are binary classifiers, which were originally proposed by (Vapnik, 1995). SVMs (Joachims. 1998) and other kernel based methods e.g. (Hofmann, 2000 and Takamura, et.al. 2004) have shown empirical successes in the field of TC.

TC empirical results have shown that SVMs classifiers are performing well. Simply because of the following text properties (Thorsten, 1998):

- High dimensional text space: In text documents, we are dealing with a huge number of features. Since SVMs use overfitting protection, which does not necessarily depend on the number of features, SVMs have the potential to handle large number of features.
- Few irrelevant features: We assume that most of the features are irrelevant (feature selection tries to determine them).
- Document vectors are sparse: For each document, the corresponding document vector contains only few entries, which are not zero.
- Most text categorization problems are linearly separable.

This is why SVMs based classifiers are working well for TC tasks. However, other kernel methods have outperformed SVMs method e.g. hyper plane-based TOP kernel, (Takamura et. al.2004).

Given the training examples $\{(x_i, y_i)\}_1^l$ with input data $x_i \in R^d$ and the corresponding binary class label $y_i \in \{-1, +1\}$. In SVMs, a separating hyperplane with the largest margin $f(x) = w.x + b$ is constructed on the condition that the hyperplane discriminates all the training examples correctly. (Figure 1 shows the distance between the hyperplane and its nearest vectors). This condition is relaxed in the non-separable case). To insure that all the training examples are classified correctly $y_i(x_i.w+b)-1 \geq 0$ must hold for the nearest examples. Two margin-boundary hyperplanes are formed by the nearest positive examples and the nearest negative examples. Let d be the distance between these two margin-boundary hyperplanes, and $\bar{x}$ be a vector on the margin-boundary hyperplane formed by the nearest negative examples. Then equations (1) and (2) are hold.

$$-1 \times (\bar{x}.w + b) - 1 = 0, \quad (1)$$

$$+1 \times ((\bar{x} + dw/|w|).w + b) - 1 = 0. \quad (2)$$

Noting that the margin is half of the distance d and computed as $d/2 = 1/|w|$. It is clear that maximizing the margin is equivalent to minimizing the norm of w.

So far, we have shown a general framework for SVMs. SVMs classifier is dealing with two different cases: the separable case and the non-separable case.

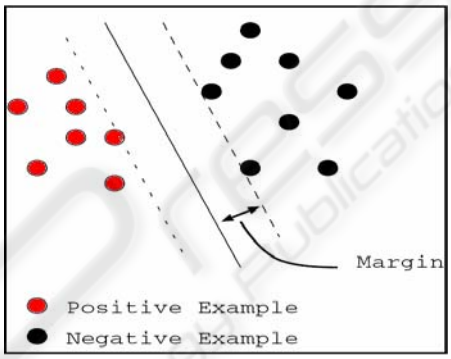

Figure 1: Support Vector Machines.

In the separable case, where the training data is linearly separable, the norm $|w|$ minimization is accomplished according to equation (3):

$$\begin{aligned} \min. \quad & \frac{1}{2}|w|^2 \\ s.t. \quad & \forall i, y_i(x_i.w + b) - 1 \geq 0 \end{aligned} \quad (3)$$

In the non-separable case, where real data is usually not linearly separable, the norm $|w|$ is minimized by equation (4):

$$\begin{aligned} \min. \quad & \frac{1}{2}|w|^2 + C\sum\nolimits_i \xi_i, \\ s.t. \quad & \forall i, y_i(x_i.w + b) - 1 + \xi_i \geq 0, \\ & \forall i, \xi_i \geq 0. \end{aligned} \quad (4)$$

where $\xi_i, (\forall i)$ are slack variables, which are introduced to enable the non-separable problems to be solved (Cristianini et al. 2000), in this case we allow few examples to penetrate into the margin or even into the other side of the hyperplane.

Skipping the details of using the Lagrangian theory, equations (3) and (4) are converted to dual problem as shown in equations (5) and (6), where

α_i is a Lagrange multiplier, C is a user-given constant.

$$\begin{aligned} \max. \quad & \sum_i \alpha i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i . x_j \\ s.t. \quad & \sum_i \alpha_i y_i = 0,\ \forall i, \alpha_i \geq 0. \end{aligned} \tag{5}$$

$$\begin{aligned} \max. \quad & \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i . x_j \\ s.t. \quad & \sum_i \alpha_i y_i = 0, \\ & \forall i, 0 \leq \alpha_i \leq C \end{aligned} \tag{6}$$

Because dual problems in equations (5) and (6) have quadratic forms, they can be solved more easily than the primal optimization problems in equation (3) and (4). Solution can be done by a general purpose optimization package like MATLAB optimization toolbox. As a result we obtain equation (7) which is used to classify examples according to its sign, where $\alpha_i^* (\forall i)$ and b^* are real numbers.

Kernel methods (Vapnik, 1995) are often combined with SVMs to compensate their limited separating ability. In kernel methods, the dot products in equations (6) and (7) are replaced with more general inner products $K(x_i, x)$, called the kernel function. This means that the feature vectors are mapped into a higher dimensional space and linearly separated there. The significant advantage is that only the general inner products of two vectors are needed. This leads to a relatively small computational overhead. On the hand, the crucial issues for SVMs are choosing the right kernel function and the parameter tuning.

$$f(x) = \sum_i \alpha_i^* y_i x_i . x + b^* \tag{7}$$

2.2.2 k-NN Classifier

k-NN classifier (Manning and Schütze, 1999), which constructs k-nearest neighbors as a basis for a decision to assign a category for a document, shows a very good performance on text categorization tasks for Arabic texts Language (Al-Shalabi et.al, 2006). It worth pointing that k-NN uses cosine as a similarity metric.

2.2.3 Naïve Bayes Classifier

Naïve Bayes classifier (Manning and Schütze, 1999) uses a probabilistic model of text. It achieves good performance results on TC task for Arabic texts (Elkourdi et.al 2004).

2.3 TC Performance Measures

Text classification performance is always considered in terms of computational efficiency and categorization effectiveness. When categorizing a large number of documents into many categories, the computational efficiency of the TC system shall be considered, this includes: feature selection method and the classifier learning algorithm.

TC effectiveness (Baeza-Yates and Ribeiro-Neto, 1999) is measured in terms of Precision, Recall and F1 Measures. Denote the precision, recall and F1 measures for a class C_i by P_i, R_i and F_i, respectively. We have:

$$\mathrm{P_i} = \frac{TP_i}{TP_i + FP_i}, \tag{8}$$

$$\mathrm{R_i} = \frac{TP_i}{TP_i + FN_i}, \tag{9}$$

$$\mathrm{F_i} = \frac{2\mathrm{P_i}\mathrm{R_i}}{\mathrm{R_i} + \mathrm{P_i}} = \frac{2TP_i}{FP_i + FN_i + 2TP_i}. \tag{10}$$

where TP_i: true positives; the set of documents that both the classifier and the previous judgments (as recorded in the test set) classify under C_i, FP_i: false positives; the set of documents that the classifier classifies under C_i, but the test set indicates that they do not belong to C_i, TN_i: true negatives; both the classifier and the test set agree that the documents in $\mathrm{TN_i}$ do not belong to C_i and FN_i: false negatives; the classifier does not classify the documents in $\mathrm{FN_i}$ under C_i, but the test set indicates that they should be classified under.

To evaluate the classification performance for each category, precision, recall, and F1 measure in equation (8-10) are used. To evaluate the average performance over many categories, the macro-averaging F1 and micro-averaging F1 are used and defined in equations (11) and (12) respectively.

$$\mathrm{F_1^M} = 2[\sum_{i=1}^{N} \mathrm{R_i} \sum_{i=1}^{N} \mathrm{P_i}] / N[\sum_{i=1}^{N} \mathrm{R_i} + \sum_{i=1}^{N} \mathrm{P_i}], \tag{11}$$

$$\mathrm{F_1^\mu} = 2\sum_{i=1}^{N} TP_i / [\sum_{i=1}^{N} FP_i + \sum_{i=1}^{N} FN_i + 2\sum_{i=1}^{N} TP_i] \tag{12}$$

Macroaveraged F1 treats every category equally, and calculates the global measures as the means of the local measures of all categories. The

Microaveraged F_1 computes an overall global measure by giving different weights to each category's local performance measures based on their numbers of positive documents.

3 EXPRIMENT RESULTS

In our experiments, we have used the mentioned Arabic dataset for training and testing our Arabic text classifier. Following the majority of text classification publications, the Arabic stop words were removed, non Arabic letters were filtered out, symbols and digits were removed. But as mentioned before we have not applied a stemming process. For each text category (Table 2), one third of the articles were randomly specified and used for testing and the remaining articles were used for training the Arabic classifier. While conducting many experiments, we have tuned the X^2 feature extraction method to achieve the best Macroaveraged F_1-measure. The best results were achieved when extracting the top 162 terms for each classification category. We have noted that increasing the number of terms does not enhance the effectiveness the TC, on the other hand it makes the training process slower. The performance is negatively affected when decreasing the number of term for each category. While conducting some other experiments, and using the X^2 scores, we tried to tune the number of selected CHI Square terms (in this case, unequal number of terms is selected for each classification category), but we could not achieve better results than those achieved using the 162 mentioned terms for each classification category. We have used an SVM package, TinySVM tool which can be downloaded from http://chasen.org/~taku/. The soft-margin parameter C is set to 1.0 (other values of C shown no significant changes in results). The results of our Arabic classifier in term of Precision and Recall for the Nine categories are shown in Table 3, the Macroaveraged F_1 score is 88.11, and the Microaveraged F_1 score is 90.57. For comparison purpose, we have used the same pre-processing steps to implemented Naïve Bayes and k-NN classifiers. It is obvious that our proposed classifier outperforms the Naïve Bayes and k-NN classifiers as shown in Table 4. Following (Samir et.al 2005) in the usage of light stemming to improve to performance of Arabic TCs, we have used (Larkey et.al 2002) stemmer to remove the suffixes and prefixes from the Arabic index terms. Unfortunately, we have concluded that light stemming does not improve the performance of our Arabic TC classifier, the Macroavaraged F_1-measure drops to 87.1. As mentioned before, the stemming is not always beneficial for text categorization problems (Hofmann, 2003). This may justify the Macroaveraged F_1-measure light drop.

Table 2: Arabic dataset.

Category	Training texts	Testing texts	Document Number
Computer	47	23	70
Economics	147	73	220
Education	45	22	68
Engineering	77	38	115
Law	65	32	97
Medicine	155	77	232
Politics	123	61	184
Religion	152	75	227
Sports	155	77	232
Corpus total number of articles			1445

Table 3: SVMs results for the Nine categories.

Category	Precision	Recall
Computer	78.57143	68.75
Economics	93.02326	71.42857
Education	85.71429	85.71429
Engineering	97.36842	97.36842
Law	92.85714	81.25
Medicine	95.06173	98.71795
Politics	90	76.27119
Religion	96.1039	98.66667
Sports	100	85.71429
Macroaverage F_1-measure = 88.11		
Microaverage F_1-measure = 90.57		

Table 4: Performance comparison.

Classifier Type	Macroaveraged F_1 -measure	Microaveraged F_1-measure
X^2 feature extraction based SVMs	88.11	90.57
Naïve Bayes	82.09	84.54
k-NN	75.62	72.72

4 CONCLUSIONS

We have investigated the performance of X^2 feature extraction method and the usage of SVMs classifier for TC tasks for Arabic language articles. Our classifier achieved practically accepted results and comparable research results. In regard to X^2, we

like to deeply investigate the relation between A, B, C and D values when dealing with small categories like (*Computer* and *Education)*. For this particular category, we have played with the X^2 and the classifier parameters, but we could not enhance the Recall or the Precision values. The investigation of other feature selection algorithms remains for future works. And Building a bigger Arabic Language TC Corpus shall be considered as well in our future research.

ACKNOWLEDGEMENTS

Many thanks to Dr. Ghassan Kannaan for providing the TC Arabic dataset and thanks to Dr. Nevin Darwish for emailing me her paper in TC. And thanks to Dr. Asim El Shiekh for Financial support.

REFERENCES

Manning, C., Schütze, H., (1999). Foundations of Statistical Natural Language Processing. MIT Press.

Sebastiani, F., (2002). Machine Learning in Automated Text Categorization. ACM Computing Surveys, 34 (1), 1-47.

Yang, Y., & Liu, X., (1999). A re-examination of text categorization methods. 22nd Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR'99), 42-49.

Joachims, T., (1998). Text categorization with support vector machines: Learning with many relevant features. Proceedings of the 10th European Conference on Machine Learning, pages 137–142

Schapire, R. & Singer, Y., (2000). BoosTexter: A boosting-based system for text categorization. Machine Learning, 39, No.2/3.

Vapnik, V., (1998). Statistical learning theory, John Wiley & Sons, Inc., N.Y.

Benkhalifa, M., Mouradi, A., Bouyakhf, H., (2001). Integrating WordNet knowledge to supplement training data in semi-supervised agglomerative hierarchical clustering for text categorization. International Journal of Intelligent Systems. 16 (8): 929-947.

Elkourdi, M., Bensaid, A., & Rachidi, T., (2004). Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm. Proceedings of COLING 20th Workshop on Computational Approaches to Arabic Script-based Languages, Geneva, August 23rd-27th .2004, 51-58.

Samir, A., Ata, W., & Darwish, N., (2005), A New Technique for Automatic Text Categorization for Arabic Documents, 5th IBIMA Conference (The internet & information technology in modern organizations), December 13-15, 2005, Cairo, Egypt.

Salton, G,. Wong A., & Yang S., (1975). A Vector Space Model for Automatic Indexing. Communications of the ACM, 18(11), pp. 613-620.

Hofmann, H., (2003). Introduction to Machine Learning, Draft Version 1.1.5, November 10, 2003.

Salton, G., & Buckley, C., (1988). Term weighting approaches in automatic text retrieval. Information Processing and Management, 24 (5), 513-523.

Yang, Y., & Pedersen, J., (1997). A comparative study on feature selection in text categorization. In J. D. H. Fisher, editor, The 14th International Conference on Machine Learning (ICML'97), 412-420. Morgan Kaufmann.

Schutze, H., Hull, D., & Pedersen, J., (1995). A comparison of classifiers and document representations for the routing problem. Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 229-237.

Yang, Y., & Wilbur, J., (1996). Using corpus statistics to remove redundant words in text categorization. Journal of the American Society for Information Science, 47(5), 357-369.

Mitchell, T., (1996). Machine Learning, New York, McGraw Hill .

Vapnik, V., (1995). The Nature of Statistical Learning Theory. Springer-Verlag Berlin.

Hofmann, T., (2000). Learning the similarity of documents: An information geometric approach to document retrieval and categorization. Advances in Neural Information Processing Systems, 12, 914–920.

Takamura, H., Matsumoto, Y., & Yamada, H., (2004). Modeling Category Structures with a Kernel Function. Proceedings of Computational Natural Language Learning. *Proceedings of CoNLL-2004*, Boston, MA, USA, 57-64.

Cristianini, N., & Shawe-Taylor, J., (2000). An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press.

Al-Shalabi, R., Kanaan, G., & Gharaibeh, M., (2006). Arabic text categorization using kNN Algorithm, Proceeding of the 4th International Multiconference on Computer Science and Information Technology, volume 4, Amman, Jordan. Retrieved March 20, 2007, from http://csit2006.asu.edu.jo/proceedings.

Baeza-Yates, R., & Rieiro-Neto, B., (1999). Modern Information Retrieval. Addison-Wesley & ACM Press.

Larkey, L., Ballesteros, L., & Connell, M., (2002). Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis. Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2002), Tampere, Finland, August 11-15, 2002, 275-282.