Journal of Computer Science 3 (6): 430-435, 2007
ISSN 1549-3636

Chi Square Feature Extraction Based Svms Arabic Language Text Categorization System

Abdelwadood Moh'd A MESLEH
Faculty of Information Systems and Technology,
Arab Academy for Banking and Financial Sciences, Amman, Jordan.
Computer Engineering Department, Faculty of Engineering Technology,
Balqa' Applied University, Amman, Jordan

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 system shows a high classification effectiveness for Arabic data set in term of F-measure (F=88.11).

Keywords: Arabic Text Classification, Arabic Text Categorization, CHI Square feature extraction.

INTRODUCTION

Text Classification (TC) is the task to classify texts to one of predefined categories based on their contents[1]. 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 [2]. It can be 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 [3], *k*-nearest neighbours (*k*NN)[3], support vector machines (SVM)[4,5] etc. On the other hand, it has been studied as a multi classification approach e.g. boosting[6], and multiclass SVM[7].

In this paper, we have restricted our study of TC on binary classification methods and in particular to Support Vector Machines (SVM) classification method for Arabic Language text.

TC Procedure: The TC System Design Usually Compromise Three Phases: Data pre-processing, text classification and performance measures: 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 by some performance measures.

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

The following sections are devoted to these three phases

Data Pre-processing:

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, 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 (Table 1) that vary in the number of documents.

In this Arabic dataset, each document file was saved in a separate file within the corresponding category's directory, i.e. this dataset documents are single-labelled.

Representing Arabic dataset Documents: As mentioned before, this representing aims to transform the Arabic text documents to a form that is suitable for the classification algorithm. In this phase, we have followed [8,9] and [10] and processed the Arabic documents according to the following steps:

1. Each article in the Arabic data set is processed to remove the digits and punctuation marks.
2. We have followed [11] in the normalization of some Arabic letters such as the normalization of (hamza) in all its forms to (alef).

3. All the non Arabic texts were filtered.
4. Arabic function words were removed. The Arabic function words (stop words) are the words that are not useful in IR systems e.g. The Arabic prefixes, pronouns, prepositions.
5. Infrequent terms removal: we have ignored those terms that occur less than 4 times in the training data. The vector space representation [12] is used to represent the Arabic documents.

Table1: Arabic Data set

Category	Document Number
Computer	70
Economics	220
Education	68
Engineer	115
Law	97
Medicine	232
Politics	184
Religion	227
Sports	232
Total number of documents	1445

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 [13].

Based on the vector space model (VSM) each term corresponds to a text feature with term frequency $TF = t_{ij}$, the number of times term i occurs in document j, as its value. This TF makes the frequent words *for the document* more important. We have used the inverse document frequency IDF [4] to improve system performance. DF, the number of documents that term i occurs in, is used to calculate $IDF(i)$,

$$IDF = \log(\frac{N}{DF})$$

where N is the total number of training documents. Then the vectors are normalized to unit length. $IDF.TF$ is calculated as a weight for each term – text feature.

Feature selection: In text categorization, we are dealing with a huge feature spaces. This is why; we need a feature selection mechanism. The most popular feature selection methods are document frequency thresholding (DF)[14], the X^2 statistics (CHI) [15], term strength (TS) [16], information gain (IG) [14], and mutual information (MI) [14],

The X^2 statistic [14] 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 2) 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.

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

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

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 [14] found (CHI) and (IG) most effective. Unlike [4] where he has used (IG) in his experiment, we have used CHI as a feature selection method for our Arabic TC.

SVMs TC Classifier: As any classification algorithm, TC algorithms have to be robust and accurate. There are a lot of machine learning based methods that can be implemented for TC tasks; It is obvious that Support Vector Machine (SVM) [4] and other kernel based methods e.g. [17] and [18] 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 [4]:

High dimensional text space: In text documents we are dealing with a huge number of features. Since SVMs use over fitting protection, which does not necessarily depend on the number of features, SVMs have the potential to handle this large number of features.

Few irrelevant features: One way to avoid these high dimensional input spaces is to assume that most of the features are irrelevant. In text categorization there are only very few irrelevant features.

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 problems. However, other kernel methods have outperformed SVMs linear kernel method e.g. [18].

Support Vector Machines (SVMs) are binary classifiers, which were originally proposed by [19].

SVMs have achieved high accuracy in various tasks, such as object recognition [20].

Suppose a set of ordered pairs consisting of a feature vector and its label is given:

$$(x_1, y_1), (x_2, y_2), ..., (x_l, y_l) \qquad (1)$$

$$\forall i, x_i \in R^d, y_i \in \{-1, +1\}$$

In SVMs, a separating hyper plane with the largest margin $f(x) = w.x + b$ (The distance between the hyper plane and its nearest vectors, see Figure 1) is constructed on the condition that the hyper plane discriminates all the training examples correctly (however, this condition will be relaxed in non-separable case).

To insure that all the training examples are classified correctly $yi(xi.w + b) - 1 \geq 0$ must hold for the nearest examples. Two margin-boundary hyper planes are formed by the nearest positive examples and the nearest negative examples. Let d be the distance between these two margin-boundary hyper planes, and $\overline{x}$ be a vector on the margin-boundary hyper plane formed by the nearest negative examples. Then the following equations are hold:

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

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

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 the general framework for SVMs.

SVMs classifier is formulated in two different cases: the separable case and the non-separable case.

In the separable case, where the training data is linearly separable, the norm $|w|$ minimization is accomplished according to equation (2):

$$\min. \quad \frac{1}{2}|w|^2 \qquad (2)$$

$$s.t. \quad \forall i, y_i(x_i.w + b) - 1 \geq 0$$

In the non-separable case, where real data is usually not linearly separable, the norm is minimized by equation (3):

$$\min. \quad \frac{1}{2}|w|^2 + C\sum_i \xi_i, \qquad (3)$$

$$s.t. \quad \forall i, y_i(x_i.w + b) - 1 + \xi_i \geq 0,$$

$$\forall i, \xi_i \geq 0.$$

where $\xi_i, (\forall i)$ are slack variables, which are introduced to enable the non-separable problems to be solved [21], in this case we allow few examples to penetrate into the margin or even into the other side of the hyper plane.

Skipping the details of using the Lagrangian theory, equations (2) and (3) are converted to dual problem as shown in equations (4) and (5), where α_i is a Lagrange multiplier, C is a user-given constant.

Because dual problems have quadratic forms, they can be solved more easily than the primal optimization problems in equation (2) and (3). Solution can be done by any general purpose optimization package like MATLAB optimization toolbox

$$\max. \quad \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i.x_j \qquad (5)$$

$$s.t. \quad \sum_i \alpha_i y_i = 0,$$

$$\forall i, 0 \leq \alpha_i \leq C.$$

$$\max. \quad \sum_i \alpha i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i.x_j \qquad (4)$$

$$s.t. \quad \sum_i \alpha_i y_i = 0,$$

$$\forall i, \alpha_i \geq 0.$$

As a result we obtain equation (6) which is used to classify examples according to its sign, where $\alpha_i^* (\forall i)$ and b^* are real numbers.

$$f(x) = \sum_i \alpha_i^* y_i x_i.x + b^* \qquad (6)$$

Since SVMs are linear classifiers, their separating ability is limited. To compensate for this limitation, the *kernel method* is usually combined with SVMs [19].

In the kernel method, the dot products in (5) and (6) are replaced with more general inner products $K(x_i, x)$, called the kernel function. The polynomial kernel and the Radial Basic Function kernel (Gaussian) are often used. This means that the feature vectors are mapped into a higher dimensional space and linearly separated there. In this process, 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.

Other TC Classifiers:

$$\text{F-measure} = \frac{2(\text{Precision x Recall})}{(\text{Precision + Recall})}$$

Many other TC classifiers [22] have been investigated in literatures:

k-NN Classifier: k-NN classifier [1], a generalization of the nearest neighbor rule, constructs k nearest neighbors as a basis for a decision to assign a category for a document. *k* -nearest neighbor classifiers shows a very good performance on text categorization tasks for English Language [23]. It worth pointing that k-NN uses cosine as a similarity metric.

Naïve Bayes classifier: The main idea of the naïve Bayes classifier [23] is to use a probabilistic model of text. The probabilities of positive and negative examples are computed.

Performance measures: TC 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 is measured in terms of precision and recall [24]. Precision and Recall are defined as follows: [23].

$$\text{recall} = \frac{\text{a}}{(\text{a} + \text{c})} \qquad a + c > 0$$

$$\text{precision} = \frac{a}{(a + b)} \qquad a + b > 0$$

where a counts the assigned and correct cases, b counts the assigned and incorrect cases, c counts the not assigned but incorrect cases and d counts the not assigned and correct cases.

A two-way contingency table (Table 3) contains a, b, c and d .

Table 3: A contingency table for measure performance

	YES is correct	NO is correct
Assigned YES	a	b
Assigned NO	c	d

The values of precision and recall often depend on parameter tuning; there is a trade-off between them. This is why we use other measures that combined both of the precision and recall: the F-measure which is defined as follows:

To evaluate the performance across categories, F-measure is averaged. There are two kinds of averaged values, namely, micro average and macro average [23].

RESULTS

In our experiment, we have used the mentioned Arabic data for training and testing the TC classifier. Following the majority of text classification publications, we have removed the Arabic stop words, filter out the non Arabic letters, symbols and removed the digits. But as mentioned before we have not applied a stemming process. We have used one third of the Arabic data set for testing the classifier and two thirds for training the TC classifier as shown in (Table 4).

Table 4: The categories and their sizes of Arabic data set

Category	Training texts	Testing texts
Computer	47	23
Economics	147	73
Education	45	22
Engineering	77	38
Law	65	32
Medicine	155	77
Politics	123	61
Religion	152	75
Sports	155	77

We have used an SVM package, TinySVM 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 classifier in term of Precision, Recall and F-measure for the nine categories are shown in (Table 5).

Table 5: SVMs classifier results for the nine categories

Category	Precision	Recall	F-measure
Computer	78.57143	68.75	73.33333
Economics	93.02326	71.42857	80.80808
Education	85.71429	85.71429	85.71429
Engineering	97.36842	97.36842	97.36842
Law	92.85714	81.25	86.66667
Medicine	95.06173	98.71795	96.85535
Politics	90	76.27119	82.56881
Religion	96.1039	98.66667	97.36842
Sports	100	85.71429	92.30769
	Macro-Average		**88.11012**

The Macro averaged F-measure is 88.11, our X^2 feature extraction based SVM classifier outperforms the Naïve Bayes and kNN classifiers (which are implemented for result comparisons) as shown in Table 6.

While conducting many experiments, we have tuned the X^2 feature extraction method to achieve the best Macro averaged F-measure. The best results were achieved when extracting the top 162 terms for each classification category. We have noted that increasing the terms number 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 term number for each category.

Table 6: F-measure results comparison

Classifier Method	F-measure
X^2 feature extraction based SVMs Classifier	88.11
Naïve Bayes classifier	84.54
k-NN classifier	72.72

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.

Following [11] in the usage of light stemming to improve to performance of Arabic TCs, we have used [25] 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 CHI square feature extraction based SVMs classifier, the F-measure drops to 87.1. As mentioned before, the stemming is not always beneficial for text categorization problems [13]. This may justify the averaged F-measure light drop.

CONCLUSION

We have investigated the performance of CHI statistics as a feature extraction method, and the usage of SVMs classifier for TC tasks for Arabic language articles. We have 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 in CHI algorithm when dealing with small categories like *Computer*. 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.

ACKNOWLEDGMENT

Many thanks to Dr. Ghassan Kannaan (Yarmouk University, Jordan) for providing the TC Arabic dataset and thanks to Dr. Nevin Darwish (Cairo University, Computer Engineering Dept., Egypt) for emailing me her TC paper [11]. Many thanks to Dr. Tariq Almugrabi for providing many related books, papers and software.

REFERENCES

1. Manning, C., and H. Schütze, 1999, Foundations of Statistical Natural Language Processing. MIT Press.
2. Sebastiani F., 2002 Machine Learning in Automated Text Categorization. ACM Computing Surveys, Vol. 34, No. 1, pp.1-47.
3. Yang, Y., and X. Liu, 1999, A re-examination of text categorization methods," in 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), pp. 42-49.
4. Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features. In Proceedings of the 10th European Conference on Machine Learning, pages 137–142
5. Joachims, T., 2002, Learning to classify text using support vector machines, methods, theory and algorithms. Klumer academic publishers.
6. Schapire, R., and Y. Singer, 2000. BoosTexter: A boosting-based system for text categorization. Machine Learning, Vol.39, No.2/3.
7. Vladimir, N., Vapnik, 1998. Statistical learning theory, John Wiley & Sons, Inc., N.Y.
8. Benkhalifa, M., A. Mouradi, and H. Bouyakhf, 2001. Integrating WordNet knowledge to supplement training data in semi-supervised agglomerative hierarchical clustering for text categorization. Int. J. Intell. Syst. 16(8): 929-947.
9. Guo, G., H. Wang, D. Bell, Y. Bi, and K. Greer, 2004. "An kNN Model-based Approach and its Application in Text Categorization", Proc. of 5th International Conference on Intelligent Text Processing and Computational Linguistic, CICLing-2004, LNCS 2945, Springer-Verlag, pages 559-570.

10. El-Kourdi, M., A. Bensaid, and T. Rachidi, 2004. Automatic Arabic documents categorization based on the naive Bayes algorithm. Workshop on Computational Approaches to Arabic Script-Based Languages (COLING-2004), University of Geneva,Geneva, Switzerland.
11. Samir, A., W. Ata, and N. Darwish, 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.
12. Salton, G., A. Wong, and S. Yang, 1975. A Vector Space Model for Automatic Indexing. Communications of the ACM, 18(11), pp. 613-620.
13. Hofmann, T., 2003. Introduction to Machine Learning, Draft Version 1.1.5, November 10, 2003.
14. Yang Y., and J. Pedersen, 1997 A comparative study on feature selection in text categorization. In J. D. H. Fisher, editor, The Fourteenth International Conference on Machine Learning (ICML'97), pages 412-420. Morgan Kaufmann.
15. Schutze, H., D. Hull, and J. Pedersen, 1995. A comparison of classifiers and document representations for the routing problem. In International ACM SIGIR conference on research and development in information retrieval.
16. Yang Y., and J. Wilbur. 1996. Using corpus statistics to remove redundant words in text categorization. Journal of the American Society of Information Science, 47(5).
17. Hofmann, T., 2000. Learning the similarity of documents: An information geometric approach to document retrieval and categorization. In Advances in Neural Information Processing Systems, 12, pages 914–920.
18. Takamura, H., M.Yuji and H. Yamada, 2004, Modeling Category Structures with a Kernel Function, in Proc. of Computational Natural Language Learning (CoNLL), 2004.
19. Vladimir, N., Vapnik. 1995. The Nature of Statistical Learning Theory. Springer-Verlag Berlin.
20. Massimiliano, P., and A. Verri. 1998. Support vector machines for 3D object recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(6):637–646.
21. Cristianini, N., and J. Shawe-Taylor. 2000 An Introduction to Support Vector Machines (and other kernel-based learning methods). Cambridge University Press.
22. Mitchell, T., 1996, Machine Learning, New York, McGraw Hill.
23. Yang, Y., Ming. 1999. An evaluation of statistical approaches to text categorization. Inform Retrieval. 69–90.
24. Baeza- Yates, R., and B. Rieiro-Neto, 1999. Modern Information Retrieval. Addison-Wesley and ACM Press.
25. Larkey, L., L. Ballesteros, and M. Connell, 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.