International Journal of Sciences

Research Article Volume 3 - June 2014 (6)

Support Vector Machine versus k-Nearest Neighbor for Arabic Text Classification

Eman Al-Thwaib[1], Waseem Al-Romimah[2]

[1]University of Jordan, Amman, Jordan
[2]Ain Shams University, Cairo, Egypt. Email: *waseem.2020@yahoo.com*

ABSTRACT: Text Classification (TC) or text categorization can be described as the act of assigning text documents to predefined classes or categories. The need for automatic text classification came from the large amount of electronic documents on the web. The classification accuracy is affected by the documents content and the classification technique being used. In this research, an automatic Support Vector Machine (SVM) and k-Nearest Neighbor (kNN) classifiers will be developed and compared in classifying 800 Arabic documents into four categories (sport, politics, religion, and economy). The experimental results are presented in terms of F1-measure, precision, and recall.

Keywords: Text Classification, Machine Learning, Support Vector Machine, k-Nearest Neighbor

Introduction

Automatic TC came to help human deal with the enormous amounts of data on web, with the exponential increase of text files on Internet each day. TC, as the assignment of text files to one or more predefined classes based on the content of the text files, is an important component in information management tasks.

The goal of this paper is to present and compare results obtained against 800-Arabic document data set using SVM and kNN algorithms. The bases of our comparison of the SVM and kNN are the most popular text evaluation measures (F1, Recall, and Precision) [13]

The rest of this paper is organized as follows; related works are discussed in Section 2. TC problem is described in Section 3. In Section 4, experiment results are explained, and finally conclusion is given in Section 5.

1. Related Works:

Most of the TC research is designed and tested for English languages articles. However, there is a little TC work that carried out for Arabic articles because Arabic language has an extremely rich morphology and a complex orthography [2].

Many machine learning approaches have been proposed to classify Arabic documents such as NB[3] [7], kNN [3], SVM [1] [2], N-gram [4], Neural Networks [8], etc.

Reference [5] used a data set of 1500 Arabic web documents that are pre-classified into five classes (health, business, culture and art, science, and sport), 300 documents for each class. Documents were tokenized into words/terms, stop words were removed, then the remaining words were stemmed to their roots. NB classifier computes a posteriori probabilities of classes, using estimates obtained from a training set of labeled documents. When an unlabeled document is presented, the a posteriori probability is computed for each class using the Bayes theorem. Finally, the unlabeled document is assigned to the class which has the largest a posteriori probability.

Authors of [6] applied kNN algorithm on a corpus of 621 Arabic text documents. The documents were preprocessed, the stop words were removed, a light stemmer was applied on the remaining tokens, and keywords were extracted. Normalized TF×IDF weighting scheme have been used to give those keywords weights. Data set was transformed into the Vector Space Model (VSM), then vectors were split into two sets, training and testing sets. The system classifies a test document represented as a vector in the space model by comparing it to all training documents using the cosine similarity measure. k neighbors (of training documents) that have the highest similarity were taken into account in making decision for classifying the test document.

Three classification algorithms, SVM, KNN and NB were used in reference [9] to classify 1445 text documents taken from online Arabic newspaper archives. The compiled texts were classified into nine classes (Computer, Economics, Education, Engineering, Law, Medicine, Politics, Religion and

Eman Al-Thwaib (Correspondence)
thwaib1@hotmail.com
+

Sports). Chi Square statistics was used for feature selection.

Finally, the authors of [16] compared three classification techniques for Arabic text documents (SVM, NB, and J48) in terms of three aspects (the accuracy, error rate, and time taken to build the classification model). Another comparison was held before and after removing stop words. The results showed that SVM outperforms NB and J48 in accuracy and takes less time to be built. It also showed better classification performance (according to precision and recall) after removing stop words.

2. TC Problem:

3.1 Arabic Data Preprocessing:

The data set/corpus that we used consists of 800 Arabic text documents. It is a subset of 60913-document corpus collected from many newspapers and other web sites. The 800 documents were pre-classified to four different classes (Economy, Politics, Religion, and Sport), 200 documents for each class.

The text documents have been preprocessed before being used, each document have been tokenized, i.e. split it into tokens according to the white space position. Tokens that less than 3 letters were removed, then:
1. Punctuations (such as ! ؛ , . ؟), symbols (such as < > }]), and digits have been removed. The comma ” ,” has a special case, because it appears sometimes connected to a word (without a space in between). Our preprocessor searches the beginning and end of tokens for a comma and removes it.
2. Non-Arabic words have been removed.
3. Stop words (frequently occur in all corpus without any added value such as عن ,لكن ,في) have been removed.
4.Remaining terms have been normalized, i.e., Letters “ ء”, “ آ ” , “ أ ” , “ ؤ ”, “ ئ ”, and “ ئ” have been replaced with “ا”, letter “ى ” replaced with “ي”, and the letter “ة ” replaced with “ه ”

3.2. Feature Selection

In addition to the mentioned preprocessing steps, we have used Term Frequency (TF), which is the number of times a term occurs in a document, to give terms weight. Terms with TF less than 3 were eliminated.

Then we have used Vector Space Model (VSM) to represent text documents, where each vector represents one document and it has the weights of tokens. As mentioned, the weighting scheme used here is TF.

3.3. SVM Classifier

SVM is a class of supervised machine learning techniques. It is based on the principle of structural risk minimization. In linear classification, SVM creates a hyper plane that separates the data into two sets with the maximum margin. A hyper plane with the maximum-margin has the distances from the hyper plane to points when the two sides are equal [10]. Linear SVMs can be generalized for non-linear problems. To do so, the data is mapped into another space *H* [1].

SVM is one of the best existing Machine Learning (ML) approaches regarding the classification results accuracy, and it is better than many other ML techniques such as Naïve Bayes (NB), decision trees, and kNN [1] [11] [12].

3.4. kNN Classifier

kNN is a good example of instance-based classifiers. The idea of kNN can be explained as follows: given a test document to be classified, the algorithm searches for the *k* nearest neighbors among the pre-classified training documents based on some similarity measure, and ranks those *k* neighbors based on their similarity scores, the categories of the *k* nearest neighbors are used to predict the category of the test document by using the ranked scores of each as the weight of the candidate categories, if more than one neighbor belong to the same category then the sum of their scores is used as the weight of that category, the category with the highest score is assigned to the test document provided that it exceeds a predefined threshold, more than one category can be assigned to the test document [14].

In our study after representing documents in the VSM, we have used k=1, which means one neighbor is taken into account. The cosine similarity measure (as the cosine of the angle between vectors) is used to calculate the similarities between documents according to equation (1) :

$$\text{Sim}(d,x) = \frac{\sum_{i=1}^{t}(wdi \times wxi)}{\sqrt{\sum_{i=1}^{t} wdi^2 \times \sum_{i=1}^{t} wxi^2}} \quad (1)$$

Where *wdi* is the weight of term i in document d. We have tried many values for k (2,3,4,5,10,20, and 28) but the results were almost the same.

3. Experimental Results

As mentioned before, the performance of the SVM and kNN classifiers is measured with respect to the precision, recall, and F1-measure. Precision and recall are defined as follows [9]:

(+) a+c > 0 (2)
(+) a+b > 0 (3)

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.

F measure is defined by equation (4) according to reference [9]:

$$\text{F-measure} = \frac{2(\text{Precision x Recall})}{(\text{Precision} + \text{Recall})} \quad (4)$$

Table 1 gives the precision, recall, and F1 results generated by applying the two categorizers (SVM and kNN) on our data set when using the percentage split method.

Table 1 The Experimental Results using percentage split method

Category	**SVM**			**kNN**		
	Precision	Recall	F1	Precision	Recall	F1
Economy	0.897	0.7	0.787	0.086	0.030	0.044
Religion	0.969	0.950	0.960	0.897	0.960	0.928
Sport	0.756	0.990	0.857	0.534	0.940	0.681
Politic	0.731	0.680	0.705	0.152	0.125	0.137
Average	0.838	0.830	0.827	0.417	0.514	0.448

Figure 1 and figure 2 show that SVM outperforms kNN in terms of precision and recall using percentage split method.

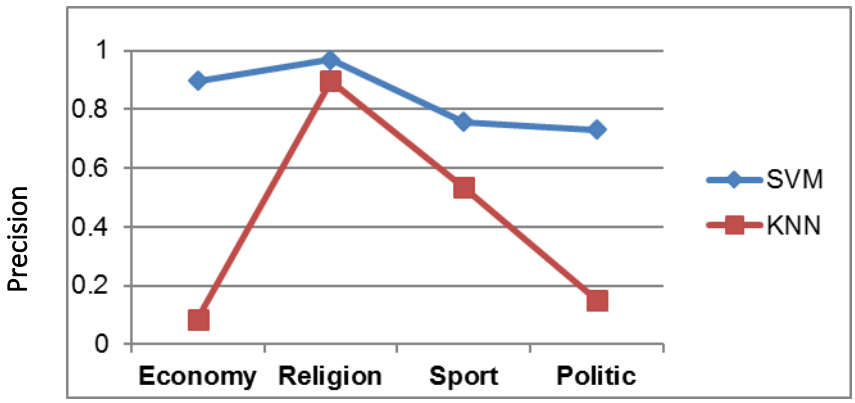

Figure 1 precision results for SVM and kNN using percentage split method

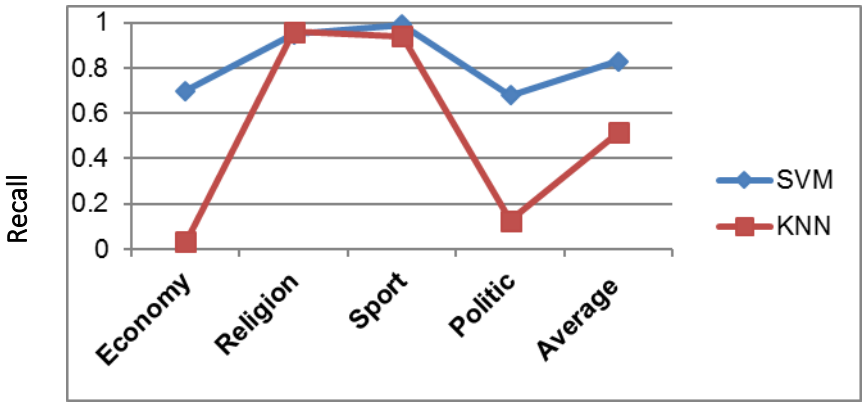

Figure 2 Recall results for SVM and kNN using percentage split method

Table 2 gives the precision, recall, and F1 results generated by applying the two categorizers (SVM and kNN) on our data set using 10-fold cross-validation method.

.

Table 2 The Experimental Results using 10-fold method

Category	**SVM**			**kNN**		
	Precision	Recall	F1	Precision	Recall	F1
Economy	0.886	0.62	0.73	1	0.035	0.052
Religion	0.965	0.96	0.962	0.95	0.96	0.955
Sport	0.724	0.995	0.838	0.562	0.91	0.695
Politic	0.71	0.66	0.684	0.279	0.285	0.282
Average	0.821	0.808	0.803	0.697	0.547	0.49

Figures 3 and 4 show that SVM outperforms kNN in terms of precision and recall using 10-fold cross-validation method..

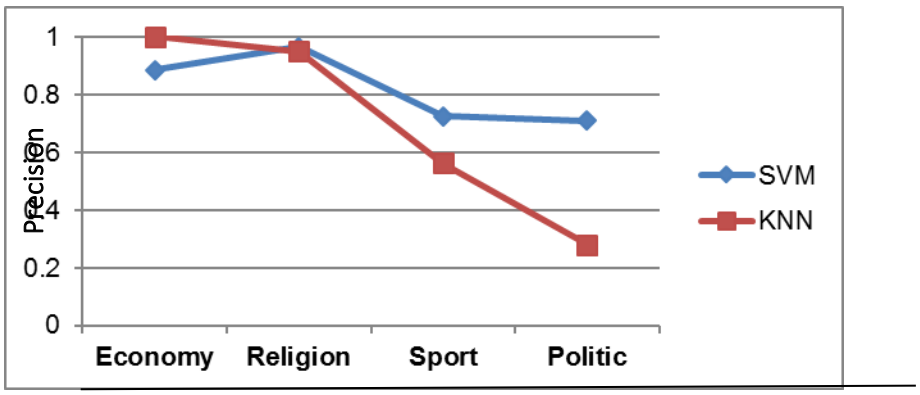

Figure 3 precision results for SVM and kNN using 10-fold method

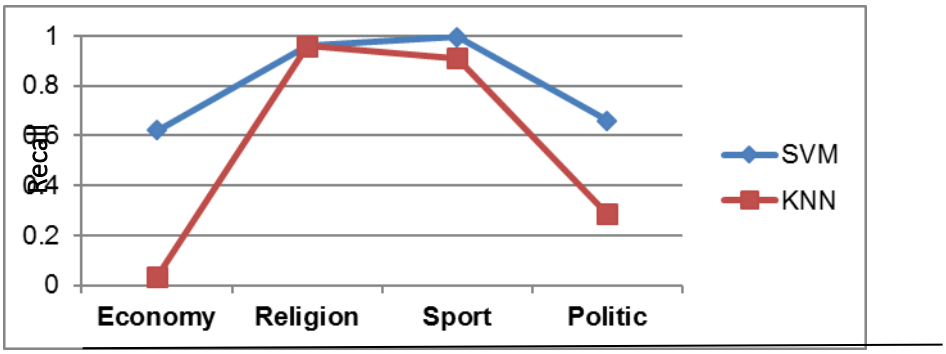

Figure 4 Recall results for SVM and kNN using 10-fold method

All the experiments were conducted using Weakaito Environment for Knowledge Acquisition (WEKA) [15], where SVM and kNN are already implemented in Java.

The data set was tested using percentage split method, where 70% of the data was used for training and the remaining 30% was used for testing. k-fold cross-validation method was used with k=10, where data is divided into 10 equal parts. One part is used for testing and the remaining nine parts are used for training the classifier.

4. CONCLUSION

In this paper, we have discussed the problem of automatic Arabic text classification. We have compared two classification algorithms, SVM and kNN according to Precision, Recall, and F1 performance measures. The experimental results indicated that the SVM algorithm outperform kNN algorithm in all used measures.

References:

[1] S. Al-Saleem, "Automated Arabic Text Categorization Using SVM and NB", International Arab Journal of e-Technology, Vol. 2, No. 2, June 2011.

[2] M. Abdelwadood, "Support Vector Machines based Arabic Language Text Classification System: Feature Selection Comparative Study", 12th WSEAS Int. Conf. on APPLIED MATHEMATICS, Cairo, Egypt, p.p 29-31, 2007.

[3] K. Al-Hindi, E. Al-Thwaib, "A Comparative Study of Machine Learning Techniques in Classifying Full-Text Arabic Documents versus Summarized Documents", World of Computer Science and Information Technology Journal (WCSIT) ISSN: 2221-0741, Vol. 2, No. 7, p.p 126-129, 2013.

[4] L. Khreisat, "Arabic text classification using N-Gram frequency statistics, a comparative study", Proceedings of the international conference on data mining (DMIN2006), Las Vegas, USA, p.p 78-82, 2006.

[5] M. El-Kourdi, A. Bensaid, and T. Rachidi "Automatic Arabic documents categorization based on the Naïve Bayes algorithm", In proceedings of the workshop on computational approaches to Arabic script-based languages (COLING-2004), University of Geneva, Geneva, Switzerland, p.p 51-58, 2004

[6] R. Al-Shalabi, G. Kanaan, and M. Gharaibeh, "Arabic text categorization using kNN algorithm", Proceedings of the 4th international multiconference on computer science and information technology (CSIT 2006), volume 4, Amman, Jordan, 2006.

[7] H. Zhang, D. Li, "Naïve Bayes text classifier", IEEE international conference on granular computing, p.p 708-711, 2007.

[8] G. Dayal, "Knowledge based Neural Network for text classification", IEEE international conference on granular computing. d'Analyse statistique des Donnees Textuelles, p.p 542-547, 2007.

[9] A. Mesleh, "Chi Square Feature Extraction Based SVMs Arabic Language Text Categorization System", Journal of Computer Science (3:6), pp. 430-435, 2007.

[10] V. Springer, V. Vapnik, "The Nature of Statistical Learning Theory", chapter 5, New York, 1995.

[11] T. Joachims, "Transductive Inference for Text Classification using Support Vector Machines", proceedings of the International Conference on Machine Learning (ICML), pp. 200-209, 1999.

[12] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features", In Proceedings of the European Conference on Machine Learning (ECML), pp.173-142, Berlin, 1998.

[13] C. Van, Rijsbergan, "Information Retrieval", Buttersmiths, 2nd Edition, 1979.

[14] R. Al-Shalabi, G. Kanaan, and M. Gharaibeh "Arabic Text Categorization Using KNN Algorithm", The 4th International Multiconference on Computer and Information Technology, CSIT 2006, Amman, Jordan, 2006.

[15] WEKA. Data Mining Software in Java: http://www.cs.waikato.ac.nz/ml/weka. last visit on May, 2014.

[16] B. Al-Shargabi, W. Al-Romimah, and F. Olayah, "A Comparative Study for Arabic Text Classification Algorithms Based on Stop Words Elimination", In proceedings of the International Conference on Intelligent Semantic Web-Services and Applications, 2011.