Feature Subset Selection for Arabic Document Categorization using BPSO-KNN

Hamouda K.Chantar
School of Mathematical and Computer Sciences
Heriot-Watt University
EDINBURGH, UK
hamoudak77@yahoo.com

David W. Corne
School of Mathematical and Computer Sciences
Heriot-Watt University
EDINBURGH, UK
dwcorne@gmail.com

***Abstract*—Document categorization is an important topic that is central to many applications that demand reasoning about and organisation of text documents, web pages, and so forth. Document classification is commonly achieved by choosing appropriate features (terms) and building a term-frequency inerse-document frequency (TFIDF) feature vector. In this process, feature selection is a key factor in the accuracy and effectiveness of resulting classifications. For a given task, the right choice of features means accurate classification with suitable levels of computational efficiency. Meanwhile, most document classification work is based on English language documents. In this paper we make three main contributions: (i) we demonstrate successful document classification in the context of Arabic documents (although previous work has demonstrated text classification in Arabic, the datasets used, and the experimental setup, have not been revealed); (ii) we offer our datasets to enable other researchers to compare directly with our results; (iii) we demonstrate a combination of Binary PSO and *K* nearest neighbour that performs well in selecting good sets of features for this task.**

Keywords-component; feature selection, text mining,Arabic language processing

I. INTRODUCTION

With rapid growth in the availability and use of natural language text documents in electronic form, automatic text classification becomes an important technique for understanding and organizing these data. Text categorization or 'topic spotting' is the task of classifying (largely unstructured) natural language documents into one or more pre-defined categories based on their content. The ability to do this supports an increasing number of applications, including more informative search engine interfaces, and replacing very time-consuming human effort in the manual organization of large collections of text documents.

The basis of document/text processing is to transform a document into a term-frequency vector [1], but this immediately brings up the issue of what terms, and how many terms, to use to represent a document. This general question of feature selection (FS) has a great impact in data mining in general and text mining in particular. FS has been an active research area since the 1970s. In text classification in particular, feature selection aims to improve the classification accuracy and computational efficiency by removing irrelevant and redundant terms (features), while retaining features that contain sufficient information to assist with the classification task at hand.

There are broadly two approaches to FS, the wrapper and filter approaches [2]. In the wrapper approach, typically a search is performed for an ideal subset of features, using the accuracy of classifiers (given those features) as a guide to evaluating an individual feature subset. In the filter approach, a subset of features is selected using *a priori* feature scoring metrics – e.g. in the text categorization field features are ranked and selected in this way using metrics such as document frequency, information gain, mutual information and so forth [1,2]. Generally, the wrapper approach is beneficial since it considers how well a group of features work together, and thus can implicitly detect and exploit nonlinear interactions among large subsets of features; however wrapper approaches are relatively slow. Meanwhile, filter approaches always have the danger of missing such interactions between two or more features, and may often discard features that may be highly relevant to the classification task. In this paper we choose a wrapper approach, since we are mostly interested in developing accurate classifiers (e.g. to support a tool that post-processes the results from an Arabic search engine), and in that context it is not critical that the time spent developing the tool be particularly fast.

Finally we note some basic differences between Arabic and English. Arabic has 28 letters and is written from right to left. In contrast with English, Arabic has a richer morphology that makes developing automatic processing systems for it a highly challenging task [3]. The basic nature of the language, in the context of text classification, is similar to English in that we can hope to rely on the frequency distributions of 'content terms' to underpin the development of automatic text categorisation. However, the large degree of inflections, word gender, and pluralities (Arabic has forms for singular, dual, and plural), means the pre-processing (e.g. stemming) stage is more complex than in the English case.

The remainder is set out as follows. In section II we briefly overview related work on Arabic text categorization. This essentially provides a list of indicative performance values (in terms of accuracy or F1-measure) for such work, and points towards the more promising approaches,

although we note here that each paper seems to have used a separate dataset, so it is not yet possible to draw clear conclusions. The lesson we drew from this review was that selection of good feature subsets (i.e. subsets that lead to good accuracy in text categorisation of Arabic documents) could be well-served by investigating a wrapper feature selection method. We therefore decided to explore BPSO (Binary Particle Swarm Optimisation) as the feature selection approach, which had been found to work excellently in [4]. In that work, the classifier used was a radial basis function network, however here we decide to use instead one of the simplest possible classification methods, *K*-nearest neighbour. In this way we reduce the number of parameters that need to be set, while we set baseline results against which future methods can be compared using the same data. Hence, in section III we respectively and briefly describe Particle Swarm Optimisation (PSO) and *K*-nearest neighbour (KNN), the two main elements of the text classification method proposed here, and then we describe our method more fully in section IV. In section V we introduce our dataset, which is available with full annotation (i.e. so that other researchers can use precisely the same training and test splits) here: http://is.gd/arabdata, and then we describe our experiments and results; we summarize and conclude in section VI.

II. Related Work

In comparison with the English language, limited work in the text categorization field has been done for Arabic, and here we survey a significant selection of the recent published work in this area. For instance, [5] evaluated the performance of two well known classification algorithms, C5.0 and Support Vector Machines, on classifying Arabic texts. The average accuracy of SVM using seven Arabic data sets was found to be 68.65%, while C5.0 outperformed this with average accuracy 78.42%. In [6], Naïve Bayes was used to classify Arabic web documents. The experiments showed that the categorization accuracy over all categories was 67.78%. In [7] a maximum entropy method was used to classify Arabic texts into pre-defined groups based on their content. The system was tested using only nouns and pronouns as key words while excluding all other words. The performance on test data reached 80.41%. It was argued in [8] that SVMs in combination with Chi-square based feature selection is an appropriate method to classify Arabic texts. The experimental results showed that the average of the accuracy over all categories, using the F1-measure accuracy estimation method, was 88.11%. In [9], *N*gram frequency statistics were employed to classify Arabic text documents into four pre-defined classes. The results showed that tri-gram text classification for Arabic texts using the Dice similarity measure outperforms classification using Manhattan distance; they found Precision and Recall values that varied between 0.5 and 1 across 4 distinct document categories.

Among the most promising methods in this area, in terms of accuracy, was investigated in [4], where wrapper based feature selection was done using Binary Particle Swarm Optimization (BPSO) for searching feature subsets, and Radial Basis Function (RBF) networks used for classifying Arabic documents. In terms of the F1-measure, the performance of the proposed method reached 93.9% over 10 distinct text categories; per category, individual precision and recall values were almost always above 90%. Finally, [10] investigated the performance of well-known ML algorithms: CBA, Naïve Bayes and SVM on classifying Arabic text documents. The results show that CBA outperformed NB and SVM and the average F1-measure was 0.804..

III. Particle Swarm Optimization and KNN

Particle swarm optimization (PSO) was developed by Eberhart and Kennedy in 1995 [11], motivated in part by the social behaviour of flocks of birds. We provide only a brief description here and refer the reader to the original paper [11], or recent surveys [12, 13] or their favoured search engine to discover more. PSO is a population based stochastic optimization algorithm in which potential solutions are called particles. As well as a position in the search space (which essentially defines the solution it represents), each particle also has a velocity in the search space, which is initially random., A population of particles is randomly initialised in terms of position and velocity, and then each is evaluated, and each particle updates its velocity according to its experience and the experience of other particles in the swarm [11, 12]. Essentially, a particle will first update its velocity by moving it partly in the direction of the position that has best fitness in its neighbourhood (this need not be defined geographically), and partly in the direction of the best fitness that particle has seen in its own experience so far. The velocity thus updated, the particle itself will then adjust its position with the new velocity.

The original version of PSO was defined for real-valued continuous search spaces; variants have since been developed that deal with discrete spaces. In binary PSO (BPSO) [13], a particle's position is simply a binary vector, which initially seems difficult to reconcile with the notion of having velocities associated with a particle. Kennedy and Eberhart's approach retains the equations used to manage velocities in PSO, with the key difference being that in BPSO a velocity vector (a real-valued vector in which each component is kept between 0 and 1) represents a set of probabilities, one for each component. Particle positions are realised by sampling from this vector. Meanwhile, BPSO is convenient and appropriate to use here (as in [4]), since binary encoding is natural for a feature selection task.

K-NN is widely used in almost all other areas of machine learning, due to its combination of effectiveness and robust simplicity [14—16]. It is technically an 'instance based' learning method that simply stores the training instances. When a new instance (x) is to be classified, a set of the K most similar training instances is retrieved (using an appropriate distance metric) and used to predict the class of the new instance. The predicted class is the most frequent class among these K nearest neighbours to x. Standard Euclidian distance is often used [14], and is also what we use here. As we describe more clearly later, we use *K*-NN to estimate the accuracy with which a collection of features (selected by BPSO) can classify the category of Arabic text documents.

IV. FEATURE SELECTION WITH BPSO/KNN

The approach we propose and test in the next section is aimed at finding a good subset of features to support the task of Arabic text categorisation. Comparisons with other techniques are made with the help of the weka machine learning library [17—19]. Note that we clearly distinguish the task of feature selection from the question of classification. That is, we use a BPSO/KNN hybrid method, working on a *training set*, to output a specific subset of features. We then evaluate this subset of features on a test set, in which we can use any machine learning/classification method for the evaluation. In fact we evaluate feature subsets using each of Naïve Bayes, J48 (Weka's implementation of C4.5) and an SVM with a linear kernel.

In this section, we describe only our feature selection method, which is a BPSO/KNN hybrid. A step by step view is given below; this all follows a text pre-processing step, described in the next section, in which a total of *N* terms (features) are pre-determined from the document collection.

Step (1): Create a population of particles on *N*-dimensions in the feature space. Each particle is represented by three vectors: the particle's current position (*Xi*), the particle's best previous position (*Pi*) and its velocity (*Vi*). *Xi* is initialized with random binary values where 1 means the corresponding feature is selected and 0 means not-selected. *Pi* is initialized with a copy of *Xi*. (Following evaluation of each particle, the global *gbest* is initialized with the index of the particle with best fitness value).

Step (2): For each particle:

- Evaluate fitness using *K*-NN (see below).
- Update particle's personal best.

Step (3): Update global best *gbest*.

Step (4): Update velocity and position of all particles in the population according to standard approach in BPSO [13].

Step (5): Terminate if termination criterion satisfied, outputting the selected subset of features (represented by the current global best particle), else go to step (2).

The fitness of a particle is calculated using:

$$\text{Fitness} = (\boldsymbol{\alpha} \times \text{Acc}) + (ß \times ((N\text{-}T)/N))$$

Where

- *Acc* is the classification accuracy of the particle found using *K*-NN (see below).
- α and ß are two parameters used to balance between classification accuracy and feature subset size, where α is in the range [0,1] and $ß=1-\alpha$.
- *N* is the total number of features.
- *T* is length of the selected subset of features.

The classification accuracy of a particle (*P*) is calculated using the following procedure:

- Filter the subset of features selected by *P*.
- Set *C*=0.
- For each instance in the training set (recall, however, that is during training; all results presented in this paper are based on unseen test data).
- Calculate the Euclidean distance from the current instance to all instances in the training set.
- Classify the current instance according to its *K* nearest neighbours in the training set.
- If the predicted classification matches the known classification of the instance, increase C by 1.

Finally the Classification accuracy of *P* is recorded as *C* divided by the total number of instances in the training set.

V. EXPERIMENTAL STUDY

A. *Arabic Datasets:*

Three separate Arabic datasets have been used to test the proposed method. Each is taken from a different previous paper in this area, and we use all or part of it in our experiments. We note again here that direct comparison with previous work, despite the dataset availability in some cases, is compromised by the fact that in these cases it has been difficult to clarify the precise ways in which datasets were organized into training and test sets, and/or how the results in the published papers relate to training or test data. In our case, we provide full details below and the associated datasets, with clear partition into training and test data, here: http://is.gd/arabdata,

The Akhbar-Alkhaleej Arabic Dataset is a collection of 5690 Arabic news documents gathered evenly from the online newspaper "Akhbar-Alkhaleej" by. It is available from [20] and an example of research using it is [21]. It consists of five categories and each document in this collection has only one category label (single-labeled). In this work, we have selected 1708 documents randomly. Table I shows the distribution of the selected documents among the four categories.

TABLE I. Akhbar-Alkhaleej Arabic Dataset

Category	**Train**	**Test**	**Total**
International News	228	58	286
Local news	576	144	720
Sport	343	86	429
Economy	218	55	273
Total	**1365**	**343**	**1708**

The Alwatan Arabic Dataset is a collection of 20,291 Arabic news documents gathered evenly from the online newspaper "Alwatan" by [22]. It consists of six categories where each document in this collection has only one category label. In this work, we have selected 1173 documents from four categories randomly. Table II shows the distribution of the documents among these four categories. This corpus is available online at [20].

TABLE II. Alwatan Arabic Dataset

Category	**Train**	**Test**	**Total**
Culture	156	67	223
Religion	216	93	309

Sport	255	109	364
Economy	194	83	277
Total	**821**	**352**	**1173**

The Al-jazeera-News Arabic Dataset (Alj-News) is an Arabic dataset obtained from [16]. This dataset consists of 1500 documents. It includes five categories (Sport, Economy, Science, Politics and Art). The number of documents in each category is 300 documents. The size of the training set is 1200 documents (240 texts for each category), and the size of the test set is 300 documents (60 texts for each category). This dataset is available at [24].

B. Text Pre-processing

All Arabic text documents have been preprocessed according to the following steps:

- Conversion to UTF-8 encoding.
- Remove hyphens, punctuation marks, numbers, digits, non-Arabic letters and diacritics.
- Remove stop words.
- Eliminate rare words (words that occur less than five times in the dataset).
- We have not performed word stemming.
- We did not normalized some Arabic letters as in [4].
- The standard Vector Space Model (VSM) was used to represent Arabic texts [17] and TFIDF was used for the term weighting factors.

For completeness we briefly note here the basics of TFIDF weighting: the Term frequency (TF) of a term *t* is the number of times in which the term *t* appears in a specific document *d*. The Document frequency (DF) of a term t_i is the number of documents in the dataset that term *ti* occurs in at least once. The inverse document frequency (IDF) of the term *ti* is generally calculated as follows [18]:

$$\mathrm{IDF}(t_i) = \log \frac{D}{\mathrm{DF}(t_i)}$$

where *D* is the total number of documents in the dataset. The weight of term t_i in document d_i using TF.IDF is:

$$\mathrm{TF.IDF}(t_i, d_i) = \mathrm{TF}(t_i, d_i) \times \mathrm{IDF}(t_i)$$

Note that in our experiments all feature preprocessing and weighting was done separately for the training set and the test set. That is, for example, TFIDF weightings for the test set were not influenced at all by the test set.

C. BPSO_KNN parameter settings:

BPSO parameters were set as follows, after a modest amount of preliminary experimentation: The inertia weight *w* for BPSO was set at 1.02 (determined after trying different values in range between 0.4 and 1.2). In the basic PSO (or BPSO) velocity update step, inertia weight *w* dictates how much the new velocity is influenced by the current velocity. Swarm size is 30 particles (a fairly common setting), and C1 and C2 are set to 2.0 (standard values). The termination criterion is a maximum of 100 generations (determined after trying different values 50, 100 and 150)..*K* (in KNN) was set at 3 (determined after trying different values 1, 3 and 5)\ .and we used α = 0.85 and ß = 0.15 (following a range of initial tests).

All experiments made use of the Weka open source machine learning software [17]. We selected three classifiers to evaluate the selected subsets of features for each dataset. These classifiers are:

- SVM support vector machine (with linear kernel).
- Naïve Bayes classifier.
- J48 (weka implementation of C4.5).

In each case, 10-fold cross validation was used, yielding the results in the following tables.

In more detail, for each of the three datasets, the following was repeated ten times:

- BPSO/KNN was run on the training set to produce a feature subset (the final *gbest* particle)
- This feature subset was used to filter the test set. I.e. the test set was processed into TFIDF vectors with components only for the given set of terms.
- Each of SVM, Naïve Bayes, and J48 were run on this test set, using ten-fold cross-validation, leading to the accuracy values that we later report.

We first note that after pre-processing the numbers of distinct features found in the separate datasets were as shown in Table III. These translate directly into the sizes of the BPSO vectors.

TABLE III. Number of distinct features in the training portions of the three datasets

Dataset	**Distinct features from the training set**
Alj-News	5329
Alwatan	12282
Akhbar-Alkhaleej	8913

Again, these are features of the training set only. To produce the experimental results we report next, a subset of the features for a particular dataset is used as the basis for learning a classifier on the test set.

Finally, before showing a summary of results, we remind the reader of the definitions of *precision*, *recall* and *F*1-*measure* in this context. Consider the documents in the test set that are category *A*. The classifier predict a category for each document, and these predictions will fall into four classes with respect to category *A*.

- TP (true positives) – the set of documents that are in category *A*, and were correctly predicted to be in category *A*..
- TN (negatives) – the set of documents that are not in category *A*, and were predicted to be in a different category than *A*.
- FP (false positives)—the set of documents that were predicted to be in category *A*, but in fact they are of a different category.

- FN (false negatives) – the set of documents that were predicted not to be in a category *A*, but are actually in category *A*.

Precision is the proportion of predicted category *A* documents that were correctly predicted, i.e. |TP|/(|TP|+|FP|). Recall is the proportion of actual category *A* documents that were correctly predicted, i.e. |TP|/(|TP|+|FN|). The F1 measure is the harmonic mean of precision (*p*) and recall (*r*), i.e. 2*pr*(*p*+*r*).

First, we show in Table IV the overall summarised results on the test set in terms of classification accuracy (recall this is an average of 10 complete trials of the training/test process).

We note that the sizes of the feature sets returned by BPSO (shown here rounded to the nearest unit) tended to be a little more than half of the total number of features for the dataset in question. Clearly, SVM was able to classify most accurately, with results that seem quite competitive given the results, ad discussed before, that tend to be achieved in this research area.

Tables V—XIII set out more detailed views of the results on each of the three datasets, showing mean values for precision, recall and F-measure for each category in the dataset in question, where the averages are weighted according to the numbers of documents in each category. Tables V, VI, and VII respectively show the results for SVM, Naïve Bayes, and J48 on the Ali-News dataset, while the sequence is repeated for the Akhbar-Alkhaleej in Tables VIII—X and the Alwatan dataset in Tables XI—XIII.

TABLE IV. Classification accuracy of SVM, Naïve Bayes and Decision tress on the three datasets

Dataset	**No. features selected by BPSO-KNN**	**J48**	**Naive Bayes**	**SVM**
Alj-News	2967	0.729	0.846	**0.931**
Alwatan	6578	0.769	0.887	**0.961**
Akhbar-Alkhaleej	4562	0.785	0.831	**0.887**

TABLE V. Accuracy by Class for SVM on Alj-News Dataset

Class	**Precision**	**Recall**	**F-Measure**
Sport	1	0.983	0.992
Art	0.934	0.95	0.942
Science	1	0.933	0.966
Politics	0.789	0.933	0.855
Economic	0.962	0.85	0.903
W. Avg.	0.937	0.93	**0.931**

TABLE VI. Accuracy by Class for Naive Bayes on Alj-News Dataset

Class	**Precision**	**Recall**	**F-Measure**
Sport	1	0.9	0.947
Art	0.86	0.717	0.782
Science	0.914	0.883	0.898
Politics	0.662	0.85	0.745
Economic	0.852	0.867	0.86
W. Avg.	0.858	0.843	**0.846**

TABLE VII. Accuracy by Class for J48 on Alj-News Dataset

Class	**Precision**	**Recall**	**F-Measure**
Sport	0.917	0.917	0.917
Art	0.711	0.533	0.61
Science	0.849	0.75	0.796
Politics	0.471	0.667	0.552
Economic	0.789	0.75	0.769
W. Avg.	0.747	0.723	**0.729**

TABLE VIII. Accuracy by Class for SVM on Akhbar-Alkhaleej Dataset

Class	**Precision**	**Recall**	**F-Measure**
Economy	0.821	0.836	0.829
Int. News	0.98	0.845	0.907
Local News	0.835	0.917	0.874
Sport	0.975	0.895	0.933
W. Avg.	0.893	0.886	**0.887**

TABLE IX. Accuracies for Naive Bayes on Akhbar-Alkhaleej Dataset

Class	**Precision**	**Recall**	**F-Measure**
Economy	0.643	0.818	0.72
Int. News	0.929	0.897	0.912
Local News	0.825	0.785	0.804
Sport	0.925	0.86	0.892
W. Avg.	0.838	0.828	**0.831**

TABLE X. Accuracy by Class for J48 on Akhbar-Alkhaleej Dataset

Class	**Precision**	**Recall**	**F-Measure**
Economy	0.698	0.545	0.612
Int. News	0.754	0.793	0.773
Local News	0.753	0.847	0.797
Sport	0.935	0.837	0.883
W. Avg.	0.79	0.787	**0.785**

TABLE XI. Accuracy by Class for SVM on Alwatan Dataset

Class	**Precision**	**Recall**	**F-Measure**
Culture	0.838	1	0.912
Economy	0.892	0.943	0.946
Religion	1	0.978	0.989
Sport	0.991	0.972	0.981
W.Avg.	0.966	0.96	**0.961**

TABLE XII. Accuracy by Class for Naive Bayes on Alwatan Dataset

Class	**Precision**	**Recall**	**F-Measure**
Culture	0.714	0.746	0.73
Economy	0.949	0.892	0.919
Religion	0.88	0.946	0.912
Sport	0.962	0.917	0.939
W. Avg.	0.89	0.886	**0.887**

TABLE XIII. Accuracy by Class for J48 on Alwatan Dataset

Class	**Precision**	**Recall**	**F-Measure**
Culture	0.571	0.597	0.584
Economy	0.667	0.699	0.682
Religion	0.857	0.903	0.88
Sport	0.907	0.807	0.854
W. Avg.	0.773	0.767	**0.769**

A full set of confusion matrices and associated summary statistics are available from the authors for researchers who wish to compare in more detail (space constraints present is from showing them here). We nevertheless make some comments here about what is revealed by the confusion matrices. The confusions between topics are generally

understandable given the levels of similarity between the topics in different cases. For example, the dataset that is associated with the lowest accuracy values, *Akhbar-Alkhaleej*, requires the classifier to distinguish between international news, local news, economy and sport. Since either local news or international news can both often be about sport and/or the economy, it is not surprising that any automated method could be quite challenged in predicting the labelled categories. Similar potential for confusions exists in each dataset, but to a lesser extent, and we see each of these observations reflected in the confusion matrices, as well as the overall results.

VI. Conclusion

In this paper, BPSO-KNN is proposed as a feature selection method for Arabic text classification. Three Arabic datasets were used to test this method, and three well-known machine learning algorithms – SVM , Naïve Bayes and C4.5 decision tree learning (in its Weka implementation as J48) – were used to classify Arabic documents using features selected by this method. Our results suggest that the proposed method is effective. It led to values for classification accuracy and F1-measure that compare well with those reported in related work. However direct lie-for-like comparison with related work in this area is not currently possible, since either the datasets used in an associated publication are not available, or, where they are available, we have been unable to discover the way the data was split into training and/or validation and/or test data in the comparative results. Therefore another contribution of this work is to make our datasets available, with the latter issues clarified, to support continuing work on this topic.

Meanwhile, it seems clear that the results achieved by SVM, as well as Naïve Bayes, overall suggest that BPSO/KNN performs well as a feature selection technique for this task (well enough, for example, to underpin the selection of features for associated applications).

Acknowledgement

We are grateful to Kareem Shaker for assistance with the processing of Arabic text files. We also thank the Libyan Embassy in the UK for their support of the first author, especially during the current difficult times.

References

[1] Salton, G., Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management* **24** (5): 513–523 .

[2] George H. John, Ron Kohavi, Karl PPeger, Irrelevant features and the subset selection problem, Proc.11th Int. Conf. Mach. Learning, 1994..

[3] Said D., Wanas N., Darwish N, Hegazy N,.(2009). A Study of Text Preprocessing Tools for Arabic Text Categorization, Proc. 2nd Int'l Conf. on Arabic Language Resources and Tools, pp 230-236. 21

[4] Zahran B., and Kanaan G., (2009).Text Feature Selection using Particle Swarm Optimization Algorithm. World Applied Sciences Journal 7 (Special Issue of Computer & IT): 69-74.

[5] Al-Harbi S., Almuhareb A., Al-Thubaity A., Khorsheed A., Al-Rajeh A., (2008). Automatic Arabic Text Classification, JADT: 9es Journées internationales d'Analyse statistique des Données Textuelles, pp 77-83.

[6] El-Kourdi M., Bensaid A., & Rachidi T., (2004). Automatic Arabic document categorization based on the Naive-Bayes Algorithm. Workshop on computational approaches to Arabic script-based languages, COLING, University of Geneva, Geneva, Switzerland.

[7] El-Halees A., (2007). Arabic Text Classification Using Maximum Entropy. The Islamic University Journal (Series of Natural Studies and Engineering) Vol. 15, No.1, pp 157-167.

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

[9] Khreisat L., (2009). A machine learning approach for Arabic text classification using N-gram frequency statistics. Journal of Informatics 3, 72-77.

[10] Al-Saleem S., (2010). Associative Classification to Categorize Arabic Data Sets. Int. J. ACM Jordan (ISSN 2078-7952), **1**(3):118-127.

[11] Kennedy J., and Eberhart R. C., (1995). Particle Swarm Optimization, Proc IEEE ICNN (Perth, Australia), IEEE Press, pp. 1942-1948.

[12] Bo Yang, Yunping Chen and Zunlian Zhao, Survey on Applications of Particle Swarm Optimization in Electric Power Systems, in *IEEE Int'l Conf. on Control and Automation*, pp. 481—486, 2007.

[13] Kennedy J., Eberhart R.C., (1997). "A discrete binary version of the particle swarm algorithm", Systems, Man, and Cybernetics, Computational Cybernetics and Simulation'. IEEE International Conference on Volume 5, Page(s):4104 - 4108 vol.5.

[14] Mitchell T., (1997). Machine Learning. McGraw-Hill.

[15] Jiang L., Cai Z., Wang D., Jiang S., (2007). "Survey of Improving K Nearest Neighbor for Classification", Fuzzy Systems and Knowledge Discovery, FSKD, pp. 679-683.

[16] Murad A., Smaili K., Berkani D., (2009). Comparing TRClassifier and kNN by using Reduced Sizes of Vocabularies. The 3rd Int'l Conf. on Arabic Language Processing, Mohammadia School of Engineers, Rabat, Morroco.

[17] WEKA. Software: http://www.cs.waikato.ac.nz/ml/weka, 2001.

[18] Witten H. and Frank E., (2005). Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, San Francisco.

[19] EL-Manzalawy Y., and Honavar V., WLSVM : Integrating LibSVM into Weka Environment, (2005). Software available at http://www.cs.iastate.edu/~yasser/wlsvm.

[20] http://sites.google.com/site/mouradabbas9/corpora.

[21] Murad A., Smaili K. (2005). Comparison of Topic Identification Methods for Arabic Language, International conference RANLP05 : Recent Advances in Natural Language Processing, Bulgaria.

[22] Syiam M., Fayed Z. and Habib M., (2006). An Intelligent System for Arabic Text Categorization. International Journal of Intelligent Computing and Information Sciences, 6 (1): 1-19.

[23] http://filebox.vt.edu/users/dsaid/Alj-News.tar.gz.

[29] Syiam M., Fayed Z. and Habib M., (2006). An Intelligent System for Arabic Text Categorization. International Journal of Intelligent Computing and Information Sciences, 6 (1): 1-19.

[25] Salton G., Wong A. and Yang C. S., (1975). A VSM for Automatic Indexing, Communications of the ACM, 18, 613- 620