A Hybrid BSO-Chi2-SVM Approach to Arabic Text Categorization

Riadh Belkebir
USTHB, Computer Science department,
Laboratory of research in Artificial Intelligence,
BP 32 El-Alia Bab-Ezzouar, 16111 Algiers, Algeria
belkebir.riadh@gmail.com

Ahmed Guessoum
USTHB, Computer Science department,
Laboratory of research in Artificial Intelligence,
BP 32 El-Alia Bab-Ezzouar, 16111 Algiers, Algeria
aguessoum@usthb.dz

***Abstract*—Automatic categorization of documents has become an important task, especially with the rapid growth of the number of documents available online. Automatic categorization of documents consists in assigning a category to a text based on the information it contains. It aims to automate the association of a document with a category. Automatic categorization can allow solving several problems such as identifying the language of a document, the filtering and detection of spam (junk mail), the routing and forwarding of emails to their recipients, etc. In this paper, we present the results of Arabic text categorization based on three different approaches: artificial neural networks, support vector machines (SVMs) and a hybrid approach BSO-CHI-SVM. We explain the approach and present the results of the implementation and evaluation using two types of representations: root-based stemming and light stemming. The evaluation in each case was done on the Open Source Arabic Corpora (OSAC) using different performance measures.**

I. Introduction

Today, information production and exchange is exposing users to a glut of content. This challenging problem is being tackled with automatic text categorization. The latter consists in automatically assigning a category to a specific document. It can solve problems such as the identification of the language of a document, filtering and detection of spam (junk mail) [1], routing and forwarding of emails to recipients, categorization of multimedia documents, automatic indexing of texts [2], disambiguation of words [3], etc.

Automatic categorization of Arabic documents has become very important, especially that their number online is rapidly growing. A good number of research projects have been conducted on the categorization of English documents. There have also been a number of studies on European languages like French, German and Spanish, and Asian languages like Chinese and Japanese. By contrast, there is little research underway on the categorization of Arabic documents. A possible explanation for this state is the richness of the Arabic language morphology and the complexity of its spelling.

One way to approach the problem of automatic text categorization is by means of supervised machine learning techniques. Indeed, the idea is to get a computer program to be trained on and to learn from a set of documents to which category labels have previously been assigned by human experts. This is what we do in this paper, in which we present the results of using three different approaches to address the problem of automatic Arabic text categorization. The first approach is based on neural networks, the second on support vector machines (SVM), while the third is a hybrid approach that combines the SVM learning algorithm with the Bee Swarm Optimization algorithm (BSO) [4] and the statistical X^2 method (Chisquare). Each of these approaches has been tested using two modes of representation, root-based stemming and light stemming. The various combinations are compared.

The remainder of this paper is organized as follows. Section II presents some related work. Section III considers the problem of dimensionality reduction. The various techniques used in this paper are presented in Section IV. This is followed in Section V by the design of the proposed solutions. In Section VI, we present the results we have obtained with the various approaches and we perform a comparison between them. Section VII concludes the work.

II. Related Work

A. Literature on Text Categorization of non-Arabic Documents

In [3], Sebastiani gives an overview of text categorization, and presents the advantages of using machine learning techniques instead of knowledge engineering techniques. He also discusses in detail the three main steps of text categorization: document representation, classifier construction, and classifier evaluation. In [5], Harish et al. give an overview of different machine learning techniques used in text categorization. They also present various text representation schemes, the existing methods with regard to qualitative parameters, the adopted algorithms and time complexity. [6] is a survey on the state of the art of text categorization. Its authors focus mainly on the studies that have used the Reuters-21578 Text Categorization Test Collection. [7] is a book review by Joachims which presents the process of learning to classify text using Support Vector Machines. The author explains several aspects: theory, application and implementation of support vector machines.

In [9], Yang et al. present a comparison of fourteen methods used for text categorization. This is further studied by Yang in [10] where he reports the results of a comparison between five statistical methods: Support Vector Machines (SVM), the k-Nearest Neighbor classifier, a Neural Network approach, the Linear Least squares Fit Mapping, and a Naive Bayes classifier. The results of this study show that SVMs outperform the other methods in cases where the number of positive training instances per category is less than ten. In [11], the effectiveness of five different machine learning algorithms

978-1-4799-0792-2/13/$31.00 ©2013 IEEE

for text categorization is studied in terms of learning speed, real time classification speed, and classification accuracy. Dumais et al. show that linear Support Vector machines outperform the other methods. In [8], a comparative study of five statistical feature selection methods was conducted using the k-nearest neighbor classifier. In [12] the use of Support Vector Machines for text categorization is presented along with arguments why SVMs are appropriate for this task. In [13], the use of inductive machine learning techniques to categorize documents is presented. The author compares the performance of a Bayesian classifier and a decision tree learning algorithm on two text categorization data sets. In [14], Leopold and Kindermann treat the kernel function issue for support vector machines and show the importance of the term frequency and claim that it has more impact on the performance of SVMs than the kernel itself. They also study the lemmatization and stemming and claim that the run time could be reduced even when dealing with a highly inflectional language like German. In [15], the focus is on dealing with problems related to textual errors (such as spelling and grammatical errors in emails) and character recognition errors in documents produced through OCR. Cavnar et al. design an approach for text categorization based on N-grams and achieve a 99.8% correct classification rate on Usenet newsgroup articles written in different languages (English, Portuguese, French, German, Italian, Dutch, Polish and Spanish) but the Arabic is not included in this study.

B. Relevant Literature on Arabic Text Categorization

Since several studies have shown that the performance of text categorization systems may be enhanced through the use of stemming techniques, we start in this section by presenting relevant work on word stemming.

The stemming process aims to remove all affixes from a word so as to extract its root. For the Arabic language we find two types of stemming: root-based stemming and light stemming. The purpose of root based stemming is to extract the root of a word by removing its suffix, prefix and infix. However, the aim of light stemming is to remove the suffix and prefix of a word. For example, Al-Shalabi's morphology system [16] uses different algorithms to find the roots and patterns for Arabic verb forms, noun forms, as well as forms of adjectives derived from verbs. In [17] an algorithm has been developed to remove prefixes and suffixes of Arabic words.

As to the impact of stemming on text categorization, one can cite [18] where Kourdi et al. have used a Naive Bayes classifier but have concluded that there is an indication that the performance of the Naive Bayes algorithm is not sensitive to the root extraction algorithm when classifying Arabic documents. On the other hand, in [19], Harrag et al. have compared the performance of Artificial Neural Networks and SVMs for Arabic text categorization using three different modes of text representation: root-based stemming, light stemming and a dictionary-based representation. They have concluded that light stemming improved the results better than the other two representation modes.

When it comes to using SVMs as a machine learning classifier for Arabic text categorization, one finds few publications in the literature that have addressed this task. In [20] SVMs have been used with Chi-square Feature Subset Selection to classify Arabic documents. The author reported that the SVM classifier outperforms the Naive Bayes and k-NN classifiers. [21] evaluated the performance of SVMs and C5.0 for the classification of Arabic documents. The authors presented the results of the classification of seven different Arabic corpora, and concluded that the decision tree algorithm C5.0 outperforms SVMs in terms of accuracy (but did not consider other performance measures such as recall and F-measure). [22] reported a comparative study of SVM and K-NN classifiers of Arabic documents.

An interesting challenge to Arabic text categorization is the dimensionality reduction. In the literature, one finds a lot of work which has addressed this problem using methods that are based on filters such as Chi-square, information gain, and entropy. Surprisingly, though methods which are based on wrappers are efficient, few studies have stressed the problem of dimensionality reduction using these methods. They claim that these methods have a high computational time. Nevertheless, in Arabic text categorization, we know of two studies which have addressed this problem using such techniques. In [20] an ACO-Based FSS Chi-square statistical method has been adapted as a heuristic using the SVM classifier. The author concluded that his approach achieves better efficiency than 6 state-of-the-art statistical methods. In a similar study [23], Zahran and Kanaan have used Particle Swarm Optimization (PSO) as a feature selection technique and have compared their approach to other statistical approaches such as document frequency, tfidf and Chi-square statistical algorithms. The simulation results on the Arabic dataset show the superiority of the algorithm they proposed.

III. DIMENSIONALITY REDUCTION

To overcome the problem of high dimensionality in the case of automatic text categorization, the notion of reduced dimensionality was introduced. Some reduction methods focus on the selection of features (filters); they aim at suggesting a new set of features with size $N_1 < N_0$, the original size of the features. Among these techniques one finds the X^2 method, the calculation of mutual information, information gain, and entropy. Another type of techniques is called feature extraction. Like the selection-based techniques, the aim of the extraction techniques is to propose a new subset N_1 with $N_1 < N_0$ but, unlike the selection-based techniques, the subset N_1 is a synthesis (linear combination of descriptors) which should maximize performance. Among these techniques, one finds Principal Component Analysis (PCA), Latent Semantic Analysis (LSA) which was originally introduced in the field of information retrieval, Latent Semantic Indexing (LSI), and grouping of terms (Term Clustering).

Theoretically, the problem of selecting the set of attributes has been shown to be NP-hard [24] . This important result has led researchers to thinking about automatically building the features set (wrapper). As a matter of fact, several algorithms such as genetic algorithms (GA), optimization by ant colonies (ACO), swarms of particles (PSO), etc., have proved their robustness in several optimization problems. These algorithms were used for different problems such as information retrieval, speech recognition, etc. For these reasons the community is interested in Machine Learning within which to consider the problem of feature selection as an optimization problem.

IV. BACKGROUND

A. Support Vector Machine Classifier

Support Vector Machines (SVMs) are a machine learning model proposed by V. N. Vapnik [25]. The purpose of this technique is to find a model from the observation of a number of input-output pairs. The problem amounts to finding a decision boundary, a hyperplane, which separates the space into two regions. As such, it correctly classifies the data and is as far as possible from all the examples. The margin is defined as the distance from the nearest point to the hyperplane.

An interesting property of SVMs is that the decision surface is determined solely by points called support vectors. These represent the data that will affect the learning process. Indeed, in the presence of only these training examples, the same function will be learned. This is different from instance-based learning algorithms like k-NN within which all the training examples are used during the learning process [26]. Even if SVMs seek the hyperplane that separates the vector space into two, their advantage is that they are easily adaptable to non-linearly separable problems. Prior to learning the best linear separation, the input vectors are transformed into feature vectors of higher dimension. In this way a linear separator found by an SVM in this new vector space becomes a non-linear separator in the original space. This vector transformation is done using the "kernel". A more detailed algorithm can be found in [27] .

In the case of text classification, the inputs are documents and the outputs are categories. Considering a binary classifier, the aim is to learn the hyperplane that separates the documents that belong to the category and those that do not. According to [28], SVMs are well suited for text classification for several reasons. First, they can handle high-dimensional data thereby avoiding an aggressive selection of attributes that would result in a loss of information. Hence we can afford to keep more attributes and the overfitting problem can thus be alleviated. Second, some attributes are completely useless to the task of classification by SVMs, taking zero values. Indeed a characteristic of text documents is that when they are represented by vectors, a majority of the entries are zero. SVMs are well suited for so-called sparse vectors.

B. General Bee Swarm Optimization Algorithm (BSO)

In [4], the meta-heuristic "Optimization by swarms of bees" is based on the behavior of artificial swarms of bees which cooperate to solve a problem. An initial bee, BeeInt, is assigned the task of looking for an initial solution, called reference solution (Sref). The search area is then subdivided among the various bees and each one is assigned the task of exploring its area neighborhood. Each bee receives all the solutions from its neighborhood, selects the best one, and the process is repeated iteratively until the problem-dependent stopping criteria are met. The general algorithm is as follows:

Algorithm 1 General Bee Swarm Optimization Algorithm (BSO)

```
1: let Sref be the solution found by BeeInit
2: while MaxIter not reached do
3:    insert Sref in a taboo list
4:    determine SearchArea from Sref
5:    assign a solution of SearchArea to each bee
6:    for each Bee k do do
7:       search starting with the assigned solution
8:       store the result in Dance
9:    end for
10: end while
```

V. HYBRID APPROACH BSO-CHI-SVM

A. Main idea

We have mentioned in Section III that the problem of selecting the set of attributes is NP-hard. On the other hand, meta-heuristics have given good results for several optimization problems. Among these meta-heuristics, we can mention Genetic Algorithms (GA), Particle Swarm Optimisation (PSO), Tabu Search (TS), etc. In the area of automatic classification, although the methods based on meta-heuristics are very powerful, little research has addressed the problem of feature selection using these approaches. Researchers avoid using them for reasons related to computation time which is extremely high when compared with methods based on filters (Chi-square, information gain, mutual information, etc.). Indeed, when dealing with this problem of feature selection using meta-heuristics, it is necessary to repeat the learning process after the generation of any solution and hence the learning time becomes very high. As a remedy, a meta-heuristic that exploits the parallelism on a large scale can be adopted, hence the choice of BSO. This is coupled with Chi-square (X^2) in a way to guide the search so as to avoid bad solutions. The set of features is fed into the SVM, which produces a learned model for the categorization problem using this subset of features.

B. Architecture and design

We start by generating the vocabulary, i.e. all the corpus words without redundancy. A feature selection process using Chi2-BSO takes this vocabulary and produces a subset of features. The latter is fed into the SVM classifier which produces a model of the categorization system. This in turn gets evaluated and the result of this evaluation is affected to this model. The process is repeated iteratively a predefined number of times. Note that the call to the SVM module is done from within the general BSO master process. Figure 1 shows the architecture of BSO-CHI-SVM.

C. BSO-CHI-SVM Algorithm: bee swarm optimization (BSO) hybridized with X^2 (chisquare) and the SVM classifier

Let us start by introducing the basic concepts relevant to our solution.

1) Size of the problem: The problem size is equal to the vocabulary size, i.e. the set of terms that appear at least once in the training corpus. The size of the search space is equal to 2^N, where N is the size of the vocabulary.

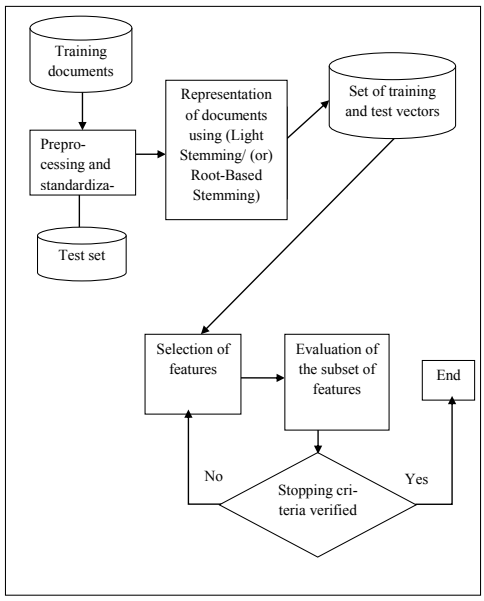

Fig. 1. Architecture of BSO-CHI-SVM

2) Coding of the solution : The solution is represented by a binary vector of size N, where:

N: Size of vocabulary,

0: Means that the attribute (feature) must be removed from the vocabulary, the training set and the test set,

1: Means that the attribute (feature) must be kept.

3) Fitness : The fitness represents the accuracy of the SVM model. The purpose is to maximize this accuracy.

4) Calculation of the neighborhood: This pseudo-code shows how the hybridization is performed. It receives as input the solution carried by the bee, the Chisquare vector, and the vocabulary size and returns a neighbor of the solution. In this case, we consider that Chisquare was already calculated and normalized and its values are between 0 and 1. It is computed only once and its value will be fixed throughout the process; it has the same size as the vocabulary. This method allows the generation of a neighbor so as to avoid assessing bad solutions, i.e. through chisquare we accept only solutions of a certain quality.

Algorithm 2 Pseudo-code that calculates the neighborhood

```
1: Input: Solution, numberOfFeatures
2: Output: NeighborSolution
3: NeighborSolution:=solution
4: for i = 1 to numberOfFeatures do
5:    generate a random value between 0 and 1 (randomVal)
6:    if solution [i] = 1 and randomVal >Chisquare [i] then
7:       NeighborSolution[i]:= 0
8:    end if
9:    if solution [i] = 0 and randomVal <Chisquare [i] then
10:      NeighborSolution[i]:=1
11:   end if
12: end for
13: Return NeighborSolution
```

5) Criteria for the termination of the algorithm: The algorithm stops if one of two conditions is satisfied:

- After a number of iterations is reached.
- A solution of a good quality has been found (as defined by the user)

VI. Implementation and Tests

A. Presentation of the Corpus

We have used the OSAC[1] corpus (Open Source Arabic Corpora) [29]. This corpus is collected from several web sites (BBC Arabic, CNN Arabic, etc.). It includes 22,429 textual records. Each text document belongs to one of ten different categories (Economics, History, Religion, Health, Education and Family, Sports, Astronomy, Law, Stories, and Cooking Recipes).

For our study, we have randomly selected 1000 texts distributed as follows: 100 hundred textual records from each category, 70 for the training and 30 for the test.

B. Pre-treatment of the Arabic Language

An important phase before the learning process consists in cleaning up the texts so as to improve the results. It includes the following:

- Removal of digits (numbers): we have removed all sequences of digits.
- Removal of Latin alphabet: we have eliminated the characters "A... Z, a ... z".
- Removal of isolated letters: we have removed all isolated letters such as ب (with), و (and),ف (then), ل (so, to), ك (such as), since they do not add any relevant information to the categorization process.
- Removal of punctuation marks: we have removed any sequence of punctuation letters or spaces such as comma and semicolon, etc.
- Removal of stopwords: like other languages, Arabic contains function words (words or tools) that do not convey any particular meaning in the text. Therefore, it is necessary to eliminate these words before the learning phase. They are also called "Stop Words". Examples: the words لأن (because), كأن (as), تحت (below) are considered as empty words.

C. Standardization

This step is specific to the Arabic language; it comes as part of the pre-treatments dealing with the morphological normalization of certain Arabic characters. We have applied the following:

- Removal of diacritics: we have removed the various vowels Fatha, Damma, Kasra, Sukun, Shadda, FATHATAN (double Fatha), DAMMATAN (double Damma), and KASRATAN (double Kasra).
  Example: The word اَلْعَرَبِيّةُ becomes العربية

[1] http://sourceforge.net/projects/ar-text-mining/files/Arabic-Corpora/

- Removal of Elongation : (purely aesthetic letters)
  Example: The word العـــــربية becomes العربية
- Normalization of "HAMZA" : the following letters are converted to ALEF by systematically removing the Hamza: ALEF_MADDA, ALEF_HAMZA_ ABOVE, BELOW ALEF_HAMZA, HAMZA_ABOVE, and BELOW_HAMZA
  Example: أهؤلاء becomes اهؤلاء

D. Representation of Documents

The document representation is an important step in automated text categorization. In the sequel, we will present the representation methods and the coding mode that we have used. These modes are the same for all approaches presented in this paper (neural network, support vector machine and our hybrid approach BSO-CHI-SVM).

1) Stemming: For the representation of the documents we have used two different stemmers, a light and a root-based stemmer. The light stemmer removes only prefixes and suffixes while the root-based stemmer removes prefixes, suffixes and infixes.

2) Coding (TFIDF): The TF/IDF representation is widely known in the field of Information Retrieval. The formula used is:

$$TF = \frac{freq}{freq + 0.5 + 1.5 * \frac{length\_doc}{avr\_length\_doc}} \quad (1)$$

$$IDF = log(\frac{N}{n_i}) \quad (2)$$

Where:

TF: term frequency
IDF: Inverse Document Frequency
freq: the frequency of the word in the document
length_doc: the document length
avr_length_doc: the average length of training documents
N: the number of training documents
n_i: the number of documents containing the term

Table I shows the number of features for the two representation modes (light and root-based stemmings).

Representation method	Root-Based Stemming	Light Stemming
# Features	12490	15460

TABLE I. NUMBER OF FEATURES

E. Classification

To test and compare the effectiveness and performance of the proposed approaches, we will perform a series of tests with different parameters that are summarized in this section. First, we will test the approach based on SVMs, then that based on neural networks and, finally, we will test the hybrid BSO-CHI-SVM approach. Individual tests on each approach will be carried out to determine the best parameters for each algorithm. The performance of these approaches are measured and compared.

For the implementation we have used lib-SVM[2] to develop solutions that are based on SVMs and we used Matlab nnet[3] (Neural Network) toolbox. The following SVM parameters are set experimentally

Type of Kernel	linear
Cost	4(default)
Degree	3(default)
Gamma	1 (default)
Coef0	0 (default)
Compute probability estimates	1.000
Use shrinking heuristics	1.000

TABLE II. PARAMETERS OF SVM

For the neural network, we have used a Feedforward with Backprobagation one. After several tests, the best results have been reached with the following parameters.

Number of hidden layers	1
Transfer functions	logsig for the hidden layer and softmax for the output layer
The learning function	trainscg
The learning rate	0.01
Momentum	0.95

TABLE III. ANN PARAMETERS

For BSO, the following parameter settings have given the best results.

Number of bees	10
Flip	5
MaxChances	1
Number of neighbors	3
Number of iterations	5

TABLE IV. PARAMETERS OF BSO-CHI-SVM

Table V below presents a comparison between all the approaches[4] presented in this paper in terms of Recall (R), Precision (P), and F-Measure(F1).

Table VI gives the performance measures in terms of accuracy.

Approach	Accuracy
SVM with root-based stemmer	93,33%
SVM with light stemmer	94,66%
ANN with root-based stemmer	92,66%
ANN with light stemmer	94%
BSO-CHI-SVM with root-based stemmer	95,33%
BSO-CHI-SVM with light stemmer	95,66%

TABLE VI. ACCURACY COMPARAISON

In this study we have reached the following results:

- The methods of representation with the light stemmer have given better results of performance than those based on the root-based stemmer. Indeed, we can see in Table I that the size of the vocabulary with the light stemmer is higher than with the root-based stemmer. This can be explained by the fact that the representation with the root-based stemmer leads to an aggressive selection, which gives rise to more ambiguity.

[2]http://www.csie.ntu.edu.tw/ cjlin/libsvm/
[3]http://www.mathworks.com/help/toolbox/nnet/
[4]LS: light stemming, RBS: root-based stemming

Approach	ANN LS			ANN RBS		
measures	R	P	F1	R	P	F1
Economics	0.94	1.0	0.96	0.96	0.93	0.95
History	1.0	1.0	1.0	1.0	0.96	0.98
Education and Family	0.8	0.85	0.82	0.53	1.0	0.69
Religion	0.76	0.96	0.85	0.86	0.81	0.83
Sports	0.96	1.0	0.98	1.0	0.96	0.98
Heath	0.96	0.88	0.92	1.0	0.88	0.93
Astronomy	1.0	0.96	0.98	1.0	1.0	1.0
Law	0.93	1.0	0.96	0.93	1.0	0.96
Stories	0.96	0.83	0.89	0.96	0.78	0.86
Cooking Recipes	1.0	1.0	1.0	1.0	1.0	1.0

Approach	SVM LS			SVM RBS		
measures	R	P	F1	R	P	F1
Economics	1.0	0.93	0.96	1.0	0.88	0.94
History	1.0	1.0	1.0	1.0	1.0	1.0
Education and Family	0.9	0.77	0.83	0.76	0.79	0.78
Religion	0.8	0.96	0.87	0.83	0.92	0.87
Sports	0.96	1.0	0.98	0.93	1.0	0.96
Heath	0.93	0.90	0.91	1.0	0.91	0.95
Astronomy	1.0	1.0	1.0	1.0	1.0	1.0
Law	0.93	1.0	0.96	0.93	1.0	0.96
Stories	0.93	0.93	0.93	0.9	0.87	0.88
Cooking Recipes	1.0	1.0	1.0	1.0	1.0	1.0

Approach	BSO-CHI-SVM LS			BSO-CHI-SVM RBS		
measures	R	P	F1	R	P	F1
Economics	1.0	0.96	0.98	0.96	0.96	0.96
History	1.0	1.0	1.0	1.0	1.0	1.0
Education and Family	0.9	0.82	0.86	0.86	0.84	0.85
Religion	0.83	0.96	0.89	0.83	0.89	0.86
Sports	1.0	1.0	1.0	1.0	1.0	1.0
Heath	0.96	0.91	0.94	0.96	0.91	0.94
Astronomy	1.0	1.0	1.0	1.0	1.0	1.0
Law	0.93	1.0	0.96	0.93	1.0	0.96
Stories	0.93	0.93	0.93	0.96	0.93	0.95
Cooking Recipes	1.0	1.0	1.0	1.0	1.0	1.0

TABLE V. PERFORMANCE EVALUATION

- Approaches based on SVMs slightly outperform the approaches based on neural networks.
- We can also see in Table VI that the BSO-CHI-SVM approach we have introduced outperforms the other approaches.
- The execution time for the approach based on SVMs is less than that based on neural networks.
- The BSO-CHI-SVM approach is the most effective, though it requires more learning time than the other approaches. We point out however that in the case of the automatic classification of documents, the learning time is not very important. Indeed, since the learning takes place offline, we just need to save the learned model afterwards. This model will later be used to predict the class membership of new documents. This operation is very fast.
- We have analysed the results produced by the various approaches presented in this paper and we can conclude that the system makes a confusion between some categories. More precisely, the two categories "religion" and "education and family" have not given as good results as the other categories. One explanation for this is that the bag of words representation we have adopted in these approaches leads the system to confuse the two categories since they have many key words in common. As a solution to this problem, a more sophisticated representation such as n-gram or concept representation should be chosen instead of the bag of words representation.

VII. CONCLUSION

We have presented in this paper the results of using three approaches for the automatic categorization of Arabic texts: neural networks, support vector machines and a hybrid approach BSO-ChiSquare-SVM. We have explained the approaches and presented the results of the implementation using two modes of representation: root-based stemming and light stemming. The evaluation has been done using different performance measures on the Open Source Arabic Corpora (OSAC). We have shown that the approaches based on the representation with a light stemmer slightly outperform those based on a Root-Based stemmer. Moreover, the approaches based on SVMs outperform those based on neural networks. In particular, the hybrid BSO-CHI-SVM approach has proven to be the most efficient. Indeed, we have achieved with this approach a degree of accuracy of 95.67%.

We envisage the further development of this work in several directions. First, we intend to evaluate the approaches studied here on other corpora. We also want to use other modes of text representation such as n-grams or representation by concepts. It will also be interesting to use other learning algorithms such as k-Nearest Neighbors, Decision Trees, and Hidden Markov Models, and compare them with our BSO-CHI-SVM approach. Last, we can consider the use of other meta-heuristics or even hybridizations of the above techniques.

REFERENCES

[1] R. Kessler, J. Manuel, T. Marc, and M. EL-BEZE, "Classification thématique de courriels avec apprentissage supervisé , semi supervisé et non supervisé," VSST 2004, vol. B, Toulouse, pp. 493-504, 2004.

[2] K. Tzeras, S. Hartmann, T. H. Darmstadt, F. Informatik, and W.-Darmstadt, "Automatic Indexing Based on Bayesian Inference Networks ," in 16th ACM International Conference on Research and Development in Information Retrieval, pp. 22-34, 1993.

[3] F. Sebastiani, "Machine Learning in Automated Text Categorization."ACM computing surveys (CSUR), vol. 34, no 1, pp. 1-47, 2002.

[4] H. Drias, S. Sadeg, and S. Yahi "Cooperative bees swarm for solving the maximum weighted satisfiability problem," in the 8th International Workshop on Artificial Neural Networks, IWANN, Barcelona, Spain, pp. 318-325, 2005.

[5] B. S. Harish, D. S. Guru, and S. Manjunath, "Representation and Classification of Text Documents: A Brief Review," IJCA Special Issue on Recent Trends in Image Processing and Pattern Recognition, RTIPPR, pp. 110-119, 2010.

[6] K. Aas and L. Eikvil, "Text Categorisation: A Survey," Technical report, Norwegian Computing Center, P.B. 114 Blindern, N-0314, Oslo, Norway. Technical Report 941, 1999.

[7] T. Joachims, "Learning to classify text using support vector machines: Methods, theory and algorithms,". Norwell, MA, USA: : Kluwer Academic Publishers, 2002.

[8] J. O. Yang, Yimig , Pedersen, "A Comparative Study on Feature Selection in Text Categorization." In The Fourteenth International Conference on Machine Learning (ICML97), pp. 412-420, 1997.

[9] Y. Yang, "An Evaluation of Statistical Approaches to Text Categorization," Information retrieval, vol. 1, no 1, pp. 69-90, 1999.

[10] Y. Yang and X. Liu, "A re-examination of text categorization methods," In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 42-49. 1999.

[11] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive Learning Algorithms and Representations for Text Categorization," In Proceedings of the seventh international conference on Information and knowledge management pp. 148-155. ACM, 1998.

[12] T. Joachims, "Text Categorization with Support Vector Machines:Learning with Many Relevant Features," Machine learning: ECML-98, pp. 137-142, 1998.

[13] D. D. Lewis, "A Comparison of Two Learning Algorithms for Text Categorization," In Proceedings of the seventh international conference on Information and knowledge management. ACM, pp. 148-155, 1998.

[14] E. Leopold and J. Kindermann, "Text Categorization with Support Vector Machines . How to Represent Texts in Input Space?," Machine Learning, vol. 46, no 1, pp. 423-444, 2002.

[15] W. B. Cavnar, J. M. Trenkle, and A. A. Mi, "N-Gram-Based Text Categorization."Ann Arbor MI, vol. 48113, no 2, pp. 161-175, 1994.

[16] R. AL-shalabi and M. Evens, "A computational morphology system for Arabic," In Workshop on Computational Approaches to Semitic Languages, COLING-ACL98. pp. 66-72, 1998.

[17] S. Khoja and R. Garside, "Stemming Arabic Text. Lancaster, UK, Computing Department, Lancaster University." 1999.

[18] M. E. L. Kourdi, M. Elkourdi, A. Bensaid, and T. Rachidi, "Automatic Arabic Document Categorization Based on the Naive Bayes Algorithm," in Proceedings of COLING 20th Workshop on Computational Approaches to Arabic Script-based Language, pp. 51-58, 2004.

[19] F. Harrag, E. El-Qawasmah, and A. M. S. Al-Salman, "Stemming as a feature reduction technique for Arabic Text Categorization," In 10th International Symposium on Programming and Systems (ISPS), pp. 128-133, 2011.

[20] A. M. Mesleh, "Support Vector Machine Text Classifier for Arabic Articles: Ant Colony Optimization Based Feature Subset Selection," the Arab Academy for Banking and Financial sciences, Ph. D. Thesis, 2008.

[21] S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, M. S. Khorsheed, and A. Al-Rajeh, "Automatic Arabic Text Classification," 9th International journal of statistical analysis of textual data, pp. 77-83, 2008.

[22] E.-Q. E. Hmeidi I., Hawashin B., "Performance of KNN and SVM classifiers on full word Arabic articles," Advanced Engineering Informatics, vol. 22, no. 1, pp. 106-111, 2008.

[23] B. M. Zahran and G. Kanaan, "Text Feature Selection using Particle Swarm Optimization Algorithm," Information Systems, vol. 7, pp. 69-74, 2009.

[24] A. L. Blum and R. L. Rivest, "Training a 3-node neural network is NP-complete," Neural Networks, vol. 5, no. 1, pp. 117-127, 1992.

[25] V. N. Vapnik, The nature of statistical learning theory. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[26] Y. Yang, "An Evaluation of Statistical Approaches to Text Categorization," Information Retrieval, vol. 1, pp. 69-90, 1999.

[27] J. C. Platt, "Fast Training of Support Vector Machines," Advances in Kernel Methods - Support Vector Learning, MIT Press, pp. 185-208, 1999.

[28] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," in ECML-98, 10th European Conference on Machine Learning, pp. 137-142, 1998.

[29] M. K. Saad and W. Ashour, "OSAC: Open Source Arabic Corpora," International Symposium on Electrical and Electronics Engineering and Computer Science, European University of Lefke, Cyprus, pp. 118-123, 2010.