ABHATH AL-YARMOUK: "*Basic Sci. & Eng.*" **Vol. 21, No. 1, 2012, pp. 15- 28**

Comparative Assessment of the Performance of Three WEKA Text Classifiers Applied to Arabic Text

Abdullah H. Wahbeh* and Mohammed Al-Kabi**

Received on Aug. 5, 2010 **Accepted for publication on March 2, 2012**

Abstract

This research is conducted in order to compare the performance of three known text classification techniques namely, Support Vector Machine (SVM) classifier, Naïve Bayes (NB) classifier, and C4.5 Classifier. Text classification aims to automatically assign the text to a predefined category based on linguistic features, and content. These three techniques are compared using a set of Arabic text documents that are collected from different websites. The document set falls into four major categories, namely, sports, economics, politics, and prophet Mohammed sayings (Al-Hadeeth Al-Shareef). The text documents pass through a set of preprocessing steps such as removing stop words, normalizing some characters, removing non Arabic text and symbols. These documents are then converted to the appropriate file format that can be used to run the three classification techniques on WEKA toolkit. After conducting the experiments the Naïve Bayes classifier achieves the highest accuracy followed by the SVM classifier, and C4.5 classifier respectively. The SVM requires the lowest amount of time to build the model needed to classify Arabic documents, followed by Naïve Bayes Classifier, and C4.5 classifier respectively.

Keywords: Arabic Text Classification, SVM, NB, C4.5, Information Retrieval, Text mining.

Introduction

Nowadays there is a rapid growth in the usage of internet for accessing information stored in web pages and databases, and there is a need to make it easy for the user to access this information and make useful use of them. So it is important to use data mining and its application to make better organization of web and documents.

Data mining is very important because it can handle the rapid growth of data that is collected and stored into a large and numerous databases, these databases exceeds the human ability for comprehension, classification, and organization without a powerful tool. Data mining is needed for turning data stored in these databases into useful information that may help decision makers making their decision [1]. One of the important applications of data mining is text classification. Text classification aims to automatically assign the text to a predefined category based on linguistic features and content [2-5].

* College of Business and Information System, Dakota State University, Madison, SD, USA

** Faculty of Sciences & IT, Zarqa University, Zarqa - Jordan

There are two main approaches for text classification: the knowledge engineering approach and the supervised learning approach. In the first one, the classification rules are manually constructed by domain experts, whereas in the second, the classifiers are automatically built from a set of labeled (already categorized) documents by applying machine learning techniques. Evidently, due to the high cost associated with the manual construction of classifiers, most text classification methods are based on supervised learning techniques[6].

Technically, for each input document *d* and category *c*, text classification involves two steps: (a) estimating the extent to which *d* shares semantics with *c*, and (b) based on the estimation, deciding whether *d* may be classified into *c*. For the first step, classifiers often need to estimate the similarity score of *d* with respect to *c*. For the second step, classifiers often need to associate a threshold to *c*. If the similarity score of *d* with respect to *c* is higher than or equal to the threshold of *c*, *d* is accepted by *c* (i.e., classified into c), otherwise it is rejected by *c* [7].

Developing text classification systems for Arabic documents is a challenging task due to the complex and rich nature of the Arabic language. The Arabic language consists of 28 letters, and written from right to left and it has complex morphology [8]. Arabic exhibits two genders: masculine and feminine, three number categories: singular, dual, and plural. Whereas singular and plural are familiar categories to most Western learners, the dual is less familiar. The dual in Arabic is used whenever the category of two applies, whether it be in nouns, adjectives, pronouns, or verbs. The Arabic plurals are divided into two categories: regular and broken. A noun has three cases, the nominative, accusative, and genitive [9].

Many text classification approaches from data mining and machine learning exist such as Decision Trees, Support Vector Machine (SVM), Rule Induction, Associative Classification, and Neural Network [10].This research assesses the performances of three text classification techniques, to rank them according to their accuracies to classify Arabic text documents. The rank process is based on the accuracy measure, which represents the percentage of correctly classified documents within the corpus.

The three text classification techniques are the SVM algorithm, C5.0 algorithm, and the Naïve Bayes algorithm. These algorithms have been used in previous studies for Arabic text classification. Al-Harbi et al. [2] used the SVM and the C5.0 to classify more than 1500 Arabic text documents collected from seven sources into different classes. The results of their study indicate that's C5.0 algorithm is better than SVM, since C5.0 achieves an accuracy of 78.42% compared with accuracy of 68.65% for SVM. El-Kourdi et al. [11] used the Naïve Bayes algorithm to classify 300 Arabic text documents collected from five sources into different classes achieving 68.78% classification accuracy.

This paper is organized as follows: Section 2 presents an overview to the related work, while section 3 presents the proposed methodology, section 4 demonstrates the experimentation, section 5 presents an overview of the WEKA toolkit, section 6 presents the results and evaluation, and conclude our work and presents future works in section 7.

Related Work

Different studies address the problem of text classification using different techniques to classify text documents, and different metrics to evaluate the accuracies of these techniques. This section presents a number of studies and experiments in the text classification field for the Arabic language.

Al-Harbi et al. [2] addressed the issue of Arabic text classification. The general framework for their study consists of compiling the text document and label them, then selecting a set of features, and finally training and testing the classification algorithm. They used a corpus consists of 17,658 text documents with more than 11,500,000 words. The internet was used to compile the dataset, the dataset consist of different documents divided into seven corpuses, such as the Saudi Press Agency, Saudi News Papers, WEB sites, writers, discussion forums, Islamic topics, and Arabic poems. A tool is implemented for Arabic text classification (ATC tool) in order to accomplish feature extraction and selection and can automatically split the dataset into training and testing sets. The size of these two partitions is determined by the user. This tool can generate training and testing matrices. The experiment evaluates the performance of two popular classification algorithms, the SVM and C5.0. Two data mining software (RapidMiner and Clementine) were used. The Chi Square statistics were used to choose the top 30 terms of each class in the training set, Chi Square is applied on document frequency instead of term frequency. The dataset was divided as 70% for training and 30% for testing in each corpus. The results show an average accuracy of 68.65%, 78.42% for SVM and C5.0 respectively.

Kanaan et al. [12] conducted a comparison between three text classification techniques applied to Arabic text. These techniques are the *k*-Nearest Neighbor (*K*NN) technique, Rocchio technique, and Naïve Bayes technique. Kanaan et al. [12] used different term weighting methods in the experiments. For the first two techniques, namely *K*NN and Rocchio, the vector-space model is adopted, where each document is represented as a vector of term weights. Different weighting schemes are used including the raw term frequency (TF) where each term is assumed to have importance proportional to the number of times it occurs in a text, the term frequency-inverse document frequency (TF-IDF) which concerns term occurrence across a collection of text documents, and the Weighted IDF (WIDF) which is simply the term frequency normalized by the collection frequency (total term frequency over the whole corpus). They also used the Naïve Bayes classifier as a probabilistic model to calculate the probability that a document *d* belongs to a category *C*. The overall methodology consists of using *K*-fold cross validation approach for comparing the three techniques, a manually collected corpus consists of 1,445 documents from different newspaper websites is used, this corpus passes a set of preprocessing steps including normalizing some characters, Waw (و) removal, prefix removal, and suffix removal. Precision and recall measures are used for measuring the effectiveness of the classifiers under consideration. The results showed that the Naïve Bayes classifier performs the best followed by the *K*NN and Rocchio. For weighting methods, the WIDF scheme provides the best performance on *K*NN, while TF-IDF shows the best performance on Rocchio. Finally, the testing

experiments reveal that as the number of documents available in the training set increases and the number of categories decrease, the precision and recall approach a perfect value of 1 using all techniques.

El-Kourdi et al. study [11] used Naïve Bayes algorithm for categorizing Arabic text documents to one of five pre-defined categories. The Arabic dataset is collected from Aljazeera website; the collected documents were 1,500 documents, 300 documents for each of the following 5 categories: sports, business, culture and art, science, and health. Cross validation experiments are used to evaluate the Naïve Bayes categorizer. The results of cross validation in the leave-one-out experiment showed that, using 2,000 terms/roots, the categorization accuracy varies from one category to another with an average accuracy over all categories of 68.78%. Furthermore, the best categorization performance by category during cross validation experiments goes up to 92.8%. Further tests are carried out on a manually collected evaluation set which consists of 10 documents from each of the 5 categories, show that the overall classification accuracy achieved over all categories is 62%, and that the best result by category reaches 90%.

El-Halees study [3] addressed text classification using maximum entropy. The maximum entropy model estimates probabilities based on the principle of making as few assumptions as possible other than the constrained imposed. In text classification, maximum entropy is a model which assigns a class *c* of each word *w* based on its document *d* in the training data *D*. He constructed a system called ArabCat to implement the proposed method that classifies Arabic documents. ArabCat consists of a corpus, preprocessor, trainer, and categorizer. The corpus for experiment consists of Arabic documents collected from the internet, it's mainly collected from Aljazeera Arabic news channel, where the documents are categorized into six domains: politics, sports, culture and arts, science and technology, economy and health. Precision, recall, and the F-measure are used as measures for evaluating performances and accuracies of the adopted algorithms. The results show an average accuracy of 80% for each measure. ArabCat outperforms the other systems such those developed by El-Halees, El-Kourdi et. al., Sakhr's Categorizer, and Sawaf et al.

Khreisat study [8] used the *N*-gram Frequency Statistics for classifying Arabic text, employing a dissimilarity measure called the Manhattan distance, and Dice's measure of similarity. The Dice measure was used for comparison purposes. A corpus of Arabic text documents was collected from online Arabic newspapers, 40% of the corpus was used as training classes and the remaining 60% of the corpus was used for testing the classification. For the training documents, the *N*-gram (*N*=3) frequency profile was generated for each document and saved in text files. Then for each document to be classified, the *N*-gram frequency profile was generated and compared against the *N*-gram frequency profiles of all the training classes. The Manhattan and Dice measures were computed. A corpus of Arabic text documents was built using Arabic news articles collected from online websites of several Arabic newspapers. The corpus consisted of text documents covering 4 categories: sports, economy, technology and weather. To compare the performance of the tri-gram technique using the Manhattan measure, and the Dice measure, the recall and precision values were computed. The results showed an

average accuracy of 56%, 66% for recall and precision respectively using the Manhattan measure, and 83%, 89% for recall and precision respectively using the Dice's measure.

Bawaneh et al. study [13] implemented the *K*NN and Naïve Bayes algorithms in order to make a practical comparison between them and previous studies. The basic terminology is based on the idea that a standardized text classification process passes several major phases. In the first phase (preprocessing) documents are prepared to make them adequate for further use, therefore stop words are removed. In the second phase (weighting assignment phase), it is defined as the assignment of real number that relies between 0 and 1 to each keyword and this number indicates the imperativeness of the keyword inside the document. Algorithms implementation is mainly developed for testing the effectiveness of *K*NN and Naïve Bayes algorithms when applied to Arabic text. A set of labeled text documents are supplied to the system, the labels indicate the class that the text document belongs to. All documents should be labeled in order to learn the system and then test it. The system classifies a test document comparing it to all the examples it has (i.e., the training set), the comparison is done using a two previous classifiers. Tests show that the Naïve Bayes and *K*NN achieves an accuracy of 73.6% and 84.2% respectively, these results indicated that the Naïve Bayes classifier outperform the results achieved with previous studies where the *K*NN has a poor performance when compared in these studies.

Methodology

This section demonstrates the overall methodology followed to conduct the classification accuracy comparison between the three selected algorithms.

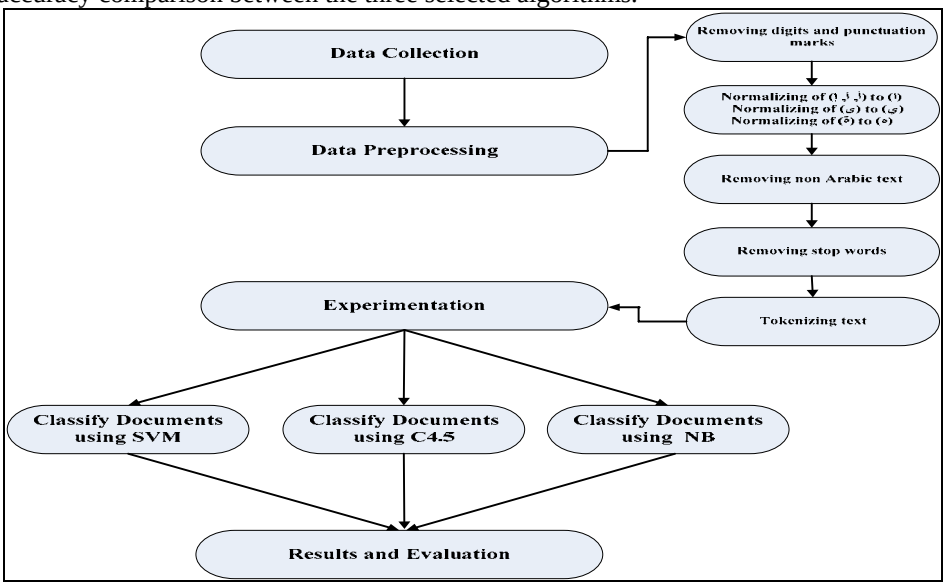

Figure 1: Overall Methodology

Figure 1 shows the entire steps that make up the methodology that is used in this research for comparing the performance of three classification techniques for Arabic text. The overall framework consists of four main steps which are, data collection, data preprocessing, experimentation, results and evaluation. Each of these steps is demonstrated in the next subsections.

The first step in the methodology is collecting the Arabic text needed for conducting the experiment and measuring the performance of the classifiers. Then these texts pass through a set of preprocessing steps that we are going to demonstrate in the next section. The data is collected from websites that already classify them into a number of categories, such as Sport, Economic, Religion, and Politic… etc. The text in the collection need some preprocessing, [3] and [14] propose some steps for normalizing and processing Arabic text. The collected data entered a set of preprocessing steps in order to achieve a standard representation for all text documents. A simple tool is created using C# to achieve this goal. The first step in data preprocessing is removing any occurrences of digits and punctuation marks, such as (1), (0.2), (100), and (!), (?), (,), (:) … etc. Since Arabic language has different morphology we are going to normalize a set of characters to a canonical form. The normalization steps include converting (أ، آ، إ) to (ا) and changing (ى) to (ي) and finally converting (ة) to (ه). This step aims to unify words typed differently. This step has its disadvantages since it leads to disambiguates of some terms, like اضعاف which means many folds or weaken, while writing the word with the appropriate Arabic Alif alphabet أضعاف means many folds, and writing it إضعاف means weaken.

Some of the documents in the dataset may contain non Arabic text such as an English terms and other characters such as @, #, $, &, ^ …etc are removed since these are insignificant for the classification process. Also, Stop words are removed from different Arabic documents. Since these words are used frequently in Arabic documents regardless of their topic, so they have no impact on classification process. Finally, we tokenize the Arabic documents and save it into a format suitable for our toolkit, to accomplish classification for each document. Several data formats are available to present data on WEKA, these formats include ARFF, C4.5, CSV, and XRRF data files and the ARFF file format. The most well known format for defining the data are as ARFF format, and CSV file. For the purpose of our research the ARFF format will be used. The ARFF file is constructed by using the core converters that are available in WEKA.

Experimentation

As a final step of the proposed methodology, we conduct the experiments. Three classification algorithms under test are, support vector machine (SVM) algorithm, C4.5 algorithm, and the Naïve Bayesian algorithm. The resulting dataset will be classified into four classes; it will be used to assess the performance and efficiency of the Sequential Minimal Optimization (SMO) which is The WEKA version of the support vector machine algorithm (SVM) [15].

SMO implements the sequential minimal optimization algorithm for training a support vector classifier, using polynomial or Gaussian kernels. Missing values are replaced globally, nominal attributes are transformed into binary ones, and attributes are normalized by default [16]. One advantage of using this implementation is that the amount of memory required by SMO is linear to the size of the data [17].

We employ SVM as a classification algorithm because it has been shown to perform well on classification problems [18]. We incorporate SVM for its classification power and robustness. SVM is able to handle hundreds and thousands of input values with great ease due to its ability to deal well with noisy data [19].

The dataset also is going to be tested using the C4.5 algorithm. This algorithm is implemented in WEKA under the name J48 algorithm.

C4.5 builds decision trees from a set of training data, using the concept of information entropy. C4.5 uses the fact that each attribute of the data can be used to make a decision that splits the data into smaller subsets [20]. Decision tree learning is a way of learning that is used by placing the knowledge in the form of a decision tree. It is used to categorize the types of examples which may come in negative or positive forms. In addition, we can insert more than two types of examples, that is, instead of just positive and negative examples, we can have many other types of examples as well [21]. Decision tree models are widely used in machine learning and data mining, since they can be easily converted into a set of humanly readable if-then rules [22].

The final algorithm that is going to be tested using the dataset is the Naïve Bayes. A Naive Bayes classifier is a well-known and practical probabilistic classifier and has been employed in many Applications [23]. This algorithm is based on Bayes' rule of conditional probability, the rule says that if you have a hypothesis *H* and evidence *E* that bears on the hypothesis the conditional probability of *H* given *E* is given by [16]:

$$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)} \tag{1}$$

where:

$P(H)$ denotes the priori probability, the probability of the hypothesis before the presentation of any evidence.

$P(E|H)$ denotes the conditional probability that *H* is true given evidence *E*.

$P(E)$ denotes the probability of the evidence associated with the hypothesis.

Naïve Bayesian Classifier is one of the Bayesian Classifier techniques which also known as the state-of-the-art of the Bayesian Classifiers. In many works it has been proven that Naïve Bayesian classifiers are one of the most computationally efficient, effective and simple algorithms for DM applications [24]. Naïve Bayes works very well when tested on a dataset, particularly when combined with some attribute features techniques [16].

The basic idea in Naïve Bayes methods is to use the joint probabilities of words and categories to estimate the probabilities of categories given a document. The naïve aspect of the method has to do with the fact that the dependencies between words are ignored,

i.e. the conditional probability of a word given a category is assumed to be independent from the conditional probabilities of other words given that category. This assumption is necessary for efficiency reasons [25].

The WEKA Toolkit

WEKA (Weikato Environment for Knowledge Analysis) [26] machine learning platform that provide a workbench which consists of collection of implemented popular learning schemes that can be used for practical data mining and machine learning works [24].

WEKA is a popular suite of machine learning software written in Java, developed at the University of Waikato [26]. WEKA supports several standard data mining tasks such as clustering, classification and visualization. All of WEKA's techniques are predicated on the assumption that the data is available as a single flat file or relation, where each data point is described by a fixed number of attributes (normally, numeric or nominal attributes, but some other attribute types are also supported.

Results and Evaluation

Data is collected mainly from three sources, the first one is Kooora (http://www.kooora.com/) Website, news-all website (http://news-all.com/), and from Saheeh Al-Bukhari book and other websites. It consist of four categories, which are sports, politics, economics, and prophet Mohammad sayings (Al-Hadeeth Al-shareef), these categories consist of 250 text documents for each, where the total size of the corpus is 1,000 documents. The text collection is preprocessed using the simple C# program developed by the second author. The data is converted into a sparse ARFF file format using WEKA TextDirectoryToArrf converter and StringToWordVector converter. Once the data are ready for experimentation, we conduct the experiment using the WEKA toolkit, the results are collected for each algorithm, in order to measure the accuracy for each classifier. An overall comparison between these classifiers is performed using the accuracy measure to determine the best of them. As shown in table 1, in literature the size of the dataset and number of classes is not agreed upon, also, there is no agreement on specific sizes for them in order to have a fair comparison.

Table 1. Corpus Size and Number of Classes in Selected Literature

Article	No. of Documents	No. of Classes
Duwairi [27]	1000	10
Al-Shargabi et al. [28]	2363	6
Gharib et al. [29]	1132	6
Al-Harbi et al. [2]	17658	7
Zaghloul et al. [30]	728	9
Mesleh and Kanaan [31]	1445	9
Kanaan et al. [12]	1445	9

The data is used into two ways, in the first one data is divided into two partitions, where the first partition is 60% from the dataset, and it is used for training phase, the second partition is 40% in size and is used for testing phase, there is no standards for selecting the size of training and testing sets in the percentage split method, however more data in the training set is required than the testing one. In literature Al-Sharghabi et al. [28] use the PS methods for evaluating the classifiers performance. In PS mode, 40% of the data set is used for testing the classifiers and the remaining 60% are used for training the classifiers. Al-Harbi et al. [2] and Mesleh and Kanaan [31] adopt the PS method for evaluating the classifiers performance, where one third of the data set is used for testing and the remaining two thirds are used for training the classification algorithms. Finally, Duwairi [27] use the PS method for evaluating the classifiers performance, where 50% of the data set is used for testing and the remaining 50% is used for training the classification algorithms.

The second form for using the dataset for training and testing is using cross validation, in *k*-fold cross-validation technique, the dataset are randomly partitioned into *k* mutually exclusive subsets or folds D_1, D_2, …, D_k , each of approximately equal size. Training and testing is performed *k* times. In iteration *i*, partition D_i is reserved as the test set, and the remaining partitions are collectively used to train the model [1].

Table 2 shows the accuracy results for the percentage split and the cross validation method obtained over the three selected classifiers. The Naïve Bayes classifier achieves the highest accuracy (85.25%) using percentage split, on the other hand the SMO classifier achieve accuracy that is almost the same as the NB classifier, the J48 classifier achieves the lowest accuracy (76%) when we compare it with the other two classifiers. Using the percentage split method, the Naïve Bayes classifier and the SMO classifiers achieve the same accuracy (83.7%) using 10 folds cross validation. On the other hand the J48 classifier achieves the lowest accuracy (79.7%).

Table 2. Accuracy measure

	Accuracy		
	SMO	NB	J48
Percentage Split	82.50%	85.25%	76%
10 Folds CV	83.70%	83.70%	79.70%

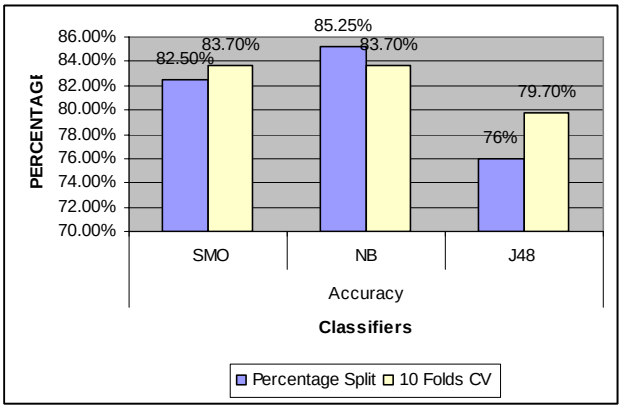

Figure 2 : Percentage split against10 folds CV

Figure 2 aims to show whether we have achieved any accuracy improvements when applying the 10 folds cross validation method instead of the percentage split one. This figure shows the accuracy of the SMO classifier increases by (1.2%), where the accuracy of the NB classifiers decreases by (1.55%). The J48 classifier achieves the best improvements when moving from the percentage split method to the *k*-fold cross validation method, where the accuracy increases by (3.7%).

Another measure that is obtained from the experiments is the amount of time taken for building the models which are used for testing the accuracy of the classifiers; this measure is illustrated in the next table

Table 2. Time taken to build the models in seconds

	SMO	NB	J48
Percentage Split	5.86	67.03	936.36
10 folds CV	8.66	68.94	1044.95

Table 2 shows that SMO classifier requires a small amount of time to build the needed model that is used for testing the accuracy of the classifier. Also the NB classifier requires a small amount of time to accomplish building the model, on the other hand the J48 requires a huge amount of time to build the needed model, and this is true. As we know J48 uses the main memory to build the models, which requires a huge amount of time to build a classification model for a large dataset.

Conclusion and Future Work

This study aims to compare three known classification techniques using Arabic text documents which lie into four classes. The comparison is based on two main aspects for the selected classifiers, accuracy and time. In terms of accuracy, results show that the Naive Bayes (NB) classifier achieves the highest accuracy, followed by the Sequential

Minimal Optimization (SMO) classifier, followed by the J48 (C4.5) classifier. On the other hand, results show that the time taken to build the SMO model is the lowest time, followed by the time taken to build the NB model, followed by the J48 classifier which takes a highest amount of time to build the needed model. These results show that there are differences between the classifiers from the two aspects, the accuracy in classifying the documents and time taken to build the classification models.

As a future work, we aim to compare the results obtained from these classifiers with other classifiers. Also, using another data mining tools such as TANAGRA in order to compare the performance of the selected classifiers. Finally, we plan to apply the text classification techniques to the dataset after applying the preprocessing steps mentioned in this research with one of the known Stemming algorithms for Arabic language.

مقارنة تقييميه لأداء ثلاث مصنفات نصية للنصوص العربية بإستخدام أداة الويكا (WEKA)

عبدالله وهبه و محمد الكعبي

ملخص

يهدف هذا البحث إلى مقارنة أداء ثلاث تقنيات معروفة في تصنيف النصوص وهي فضاء المتجهات Support Vector Machine (SVM)، نظرية بيز المبسطة (Naïve Bayes)، وشجرة القرار المعروفة بإسم (C4.5.) في الويكا (WEKA). والهدف من وراء عملية التصنيف هو وضع النص بشكل تلقائي ضمن فئة محددة سلفا على أساس الخصائص اللغوية والمحتوى. تقوم هذه الدراسة على مقارنة التقنيات الثلاث آنفة الذكر مستخدما مجموعة من الوثائق والنصوص العربية التي تم جمعها من مواقع مختلفة من شبكة الإنترنت. وتقع مجموعة النصوص هذه في أربع فئات رئيسية هي الرياضة والاقتصاد والسياسة والأحاديث النبوية الشريفة. تم القيام بمجموعة من العمليات الأولية بهدف إستخلاص المعلومات المهمة، وتتضمن هذه العمليات إزالة الكلمات عديمة الفائدة والتي تعرف بـ (Stop words). إضافة إلى توحيد الأحرف، وإزالة النص الأجنبي والرموز الخاصة. ويعقب ذلك تنسيق النصوص بشكل يساعد على إستخدامها من قبل تقنيات التصنيف الخاصة ببرمجية الويكا (WEKA). وتمخضت نتائج التجارب على مجموعة الملفات النصية العربية عن الإستنتاج بأن نظرية بيز المبسطة (Naïve Bayes) كانت الأفضل في عملية التصنيف تتبعها تقنية فضاء المتجهات (SVM) وشجرة القرار (C4.5.) على التوالي. أما بخصوص تقنية فضاء المتجهات (SVM) فقد تميزت بقصر فترة بناء خوارزميتها و تبعتها نظرية بيز المبسطة (Naïve Bayes) وشجرة القرار (C4.5.) على التوالي.

References

[1] Han, J., 'Data Mining: Concepts and Techniques', second edition, Morgan Kaufmann Publishers, San Francisco, (2006).

[2] Al-Harbi, S., Almuhareb, A., Al-Thubaity, A., Khorsheed, S., and Al-Rajeh, A., *Automatic Arabic Text Classification*. Ninth International Journal of Statistical Analysis of Textual Data, (2008) 77-83.

[3] El-Halees, A., M., Arabic Text Classification Using Maximum Entropy, *The Islamic University Journal (Series of Natural Studies and Engineering)*, 15(1) (2007) 157-167.

[4] Yang, Y., *An Evaluation of Statistical Approaches to Text Categorization*, Information Retrieval, 1(1) (1999) 69-90.

[5] Al-Kabi, M., Al- Sinjilawi, S., *A Comparative Study of the Efficiency of Different Measures to Classify Arabic Text*, University of Sharjah, Journal of Pure & Applied Sciences, 4(2) (2007) 13 – 24.

[6] Guzmán-Cabrera, R., Montes-Y-Gómez , M., Rosso, P., Villaseñor-Pineda, L., *Using the Web as Corpus for Self-training Text Categorization*, Information Retrieval, 12(3) (2009) 400-415.

[7] Liu, R., *Context Recognition for Hierarchical Text Classification*, Journal of the American society for information science and technology, 60(4) (2009) 803–813.

[8] Khreisat, L., *Arabic Text Classification Using N-Gram Frequency Statistics a Comparative Study*, In proceedings of the 2006 International Conference on Data Mining, Las Vegas, USA, (2006) 78-82.

[9] Ryding, C., *A Reference Grammar of Modern Standard Arabic*, Cambridge University Press, Cambridge, (2005).

[10] Thabtah, F., Eljininim, M., Zamzeer, M., and Hadi, W., *Naïve Bayesian Based on Chi Square to Categorize Arabic Data*, In proceedings of the 11th IBIMA conference, Cairo, Egypt, 10 (2009) 158-163.

[11] El-Kourdi, M., Bensaid, A., and Rachidi, T., *Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm*, In proceedings of COLING 20th Workshop on Computational Approaches to Arabic Script-based Languages, University of Geneva, Geneva, Switzerland, August 23rd—27th, (2004).

[12] Kanaan, G., Al-Shalabi, R., Ghawanmeh, S., and Al-Ma'adeed, H., *A Comparison of Text-Classification Techniques Applied to Arabic Text*, Journal of the American society for information science and technology, 60 (9) (2009) 1836–1844.

[13] Bawaneh, J., Alkoffash, S. and Al Rabea, I., *Arabic Text Classification using K-NN and Naïve Bayes*, Journal of Computer Science, 4(7) (2008) 600-605.

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

[15] http://www.cs.waikato.ac.nz/ml/weka/ (Visited October 8th, 2011).

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

[17] Naughton, M., Stokes, N., Carthy, J., *Sentence-Level Event Classification in Unstructured texts*, Information Retrieval, 13(2) (2010) 132 – 156.

[18] Wu, Y., Chang, C., and Lee, Y., *A General and Multi-lingual Phrase Chunking Model Based on Masking Method*, In proceedings of the 7th International Conference on Intelligent Text Processing and Computational Linguistics CICLing 2006, Mexico City, Mexico, (2006) 144 – 155.

[19] Abbasi, A., and Chen, H., *Applying Authorship Analysis to Arabic Web Content*. In Proceedings of IEEE international conference on intelligence and security informatics, IEEE ISI. Atlanta GA , ETATS-UNIS, 3495 (2005) 183-197.

[20] http://wikipedia.org/wiki/C4.5_algorithm, C4.5 algorithm/ (Visited November 11th, 2008).

[21] Wongpun, S., and Srivihok, A., Comparison of Attribute Selection Techniques and Algorithms in Classifying Bad Behaviors of Vocational Education Students, In proceedings of 2nd IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST), Australia, (2008) 526-531.

[22] Last, M., Markov, A., and Kandel, A., *Multi-lingual Detection of Web Terrorist Content*, In: Chen, H. (Ed.), WISI, Lecture Notes in Computer Science, Springer - Verlag, 3917 (2008) 16-30.

[23] Kim, S., Han, K., Rim, H., Myaeng, S., *Some Effective Techniques for Naive Bayes Text Classification*, IEEE Transactions on Knowledge and Data Engineering, 18(11) (2006) 1457-1466.

[24] Cufoglu, A., Lohi, M., and Madani, K., *A Comparative Study of Selected Classifiers with Classification Accuracy in User Profiling*, In proceedings of the 7th international conference on Machine Learning and Application icmla, San Diego, California, USA, (2008) 787-791.

[25] Fang, Y., Parthasarathy, S., and Schwartz, F., *Using Clustering to Boost Text Classification*, In proceedings of the IEEE International Conference on Data Mining, California, USA, (2001) 123-127.

[26] Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., and Witten I. H., *The WEKA Data Mining Software: An Update*, SIGKDD Explorations, 11(1) (2009).

[27] Duwairi, R., *Arabic Text Categorization*, International Arab Journal on Information Technology, 4(2) (2007) 125-131.

[28] Al-Shargabi, B., Al-Romimah, W., & Olayah, F. *A Comparative Study for Arabic Text Classification Algorithms Based on Stop Words Elimination*. Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications, Amman, Jordan, (2011).

[29] Gharib, T. F., Habib, M. B., & Fayed, Z. T. *Arabic Text Classification Using Support Vector Machines*, International Journal of Computers and Their Applications, 16(4) (2009) 192-199.

[30] Zaghloul, W., Lee, S. M., & Trimi, S. *Text Classification: Neural networks vs. Support Vector Machine*, Industrial Management & Data Systems, 109(5) (2009) 708-717.

[31] Mesleh, A. M., & Kanaan, G, *Support Vector Machine Text Classification System: Using Ant Colony Optimization Based Feature Subset Selection*. International Conference on Computer Engineering & Systems ICCES 2008, (2008).