International Journal on Islamic Applications in Computer Science And Technology, Vol. 3, Issue 3, September 2015, 1-12

Comparing Arabic NLP tools for Hadith Classification

Kaouther Faidi[1,a], Raja Ayed[3,b], Ibrahim Bounhas[1,2,c], Bilel Elayeb[3,4,d]

[1]LISI Laboratory of computer science for industrial systems, Carthage University, Tunisia

[2]Higher Institute of Documentation (ISD), Manouba University, 2010 Tunisia

[3]RIADI Laboratory, The National School of Computer Science (ENSI),Manouba University, 2010 Tunisia

[4]Emirates College of Technology, P.O. Box: 41009. Abu Dhabi, United Arab Emirates

[a]faidikaouther@gmail.com, [b]ayed.raja@gmail.com, [c]bounhas.ibrahim@yahoo.fr, [d]Bilel.Elayeb@riadi.rnu.tn

ABSTRACT

Text classification is the process of classifying documents into a predefined set of categories based on their content. As Arabic words may have more complicated forms than many other languages, it is challenging to choose the indexing unit and to get rid of affixes. In this paper we compare the performance of different techniques for classifying Al-Hadith Al-Shareef which was analyzed with six Arabic tools (Al-Stem Darwish, Al-Stem Alex, Khoja's stemmer, Quadrigrams, Trigrams and a disambiguation tool based on AraMorph). We also compare three classification techniques implemented on WEKA toolkit; namely decision trees (DT), Naïve Bayes algorithm (NB) and SVM algorithm (Support Vector Machines). We used the TF-IDF to compute the relative frequency of each word in a particular document and the cross validation to evaluate the result of the classifiers. Experimental results show that Khoja's stemmer outperformed the other tools and that the SVM classifier achieves the highest accuracy followed by the Naïve Bayes classifier, and decisions trees classifier respectively.

***Keywords*: Arabic text classification, Arabic stemming, Al-Hadith Al-Shareef, Indexing unit.**

1. INTRODUCTION

With the existence of a huge number of documents, it is necessary to be able to automatically organize information into predefined classes. Automatic text categorization attempts to replace and save human effort required in performing manual categorization. It consists of assigning and labeling documents using a set of predefined categories based on their content (Harrag et al., 2011). Many text classification techniques from data mining and machine learning exist such as Decision Trees, Support Vector Machine, Naïve Bayes, KNN, and Neural Network (Aggarwal & Zhai, 2012).

Text classification 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 is written from right to left. It has complex morphology than other languages, so it needs a set of preprocessing routines to be suitable for manipulation. Stemming is a preprocessing task which consists in removing affixes from words and extracting the root or the stem in order to choose the best indexing unit.

In this study, we compare the performance of different techniques (i.e. SMO SVM classifier, J48 DT classifier and NB) for classifying Al-Hadith Al-Shareef. We also evaluate six Arabic NLP tools, namely Al-Stem Darwish (Darwish et al. 2009), Al-Stem Alex (Fraser et al., 2002), Khoja's stemmer (Khoja, 1999), a morphological disambiguation tool based on Aramorph (Ayed et al., 2012), Quadrigrams, and Trigrams (Syiam et al., 2006). We aim to study the problem of "indexing unit" in Arabic document processing. In this context, the hadith corpus is a suitable choice, as its documents are vocalized, thus reducing ambiguity. Besides, its books are segmented into coherent chapters, which represent classes. Thus, this corpus may represent a gold standard for evaluating and comparing text classification approaches.

The remainder of the paper is organized as follows. Section 2 shows the related work in text categorization. Section 3 presents the proposed model for Al-Hadith text categorization. The achieved experimental results are discussed in section 4. Finally, the conclusion is presented in section 5.

2. RELATED WORK

Different studies addressed the problem of text classification using different techniques. Most of the work, in this area, was performed for English texts, while few researches have been applied on Arabic texts. However, the nature of Arabic text is different from other languages. This section presents a number of studies and experiments in Arabic text classification.

In his research, El-Kourdi et al. (2004) used the Naïve Bayes (NB) to classify non-vocalized Arabic web documents into five predefined categories, and the average accuracy over all categories was 68.78%. Al-Harbi et al. (2008) evaluated the performance of two classification algorithms (SVM and C5.0) on classifying Arabic texts using seven Arabic corpora, and the ATC Tool was implemented for feature extraction and selection. The results showed that C5.0 classifier gives better accuracy. The work of Al-Shalabi et al. (2006) used the key Nearest Neighbor (KNN) algorithm to Arabic text, along with the Support Vector Machines (SVMs) algorithm to extract keywords based on the Document Frequency threshold (DF) method (Soucy et al., 2005).

In another study, Wahbeh et al. (2010) compared three classification techniques using Arabic text documents which lie into four classes (sports, economics, politics, Al-Hadith Al-Shareef). The comparison is based on two main aspects, namely accuracy and time. In terms of accuracy, the results showed that the NB (Naïve Bayes) classifier achieves the best rates, followed by the SMO (Support Vector Machine) classifier, and the J48 (decision trees) classifier. On the other hand, the results highlighted that the time taken to build the SMO model is the lowest one, followed by the NB model, and the J48 classifier.

In the following, we will mainly focus of works applied on hadith. In this context, Harrag et al. (2008; 2009; 2011) experimented document classification on a hadith corpus composed of 453 hadiths distributed over 14 domains extracted from the encyclopedia of the nine books (Harrag et al., 2008). They first proceeded to stop-word removal and rule-based morphological stemming. They segmented the corpus into training and testing sets. They performed series of experiments on decision trees based classification. First, they evaluated the impact of term filtering based on term frequency and document frequency. This impact is measured by F1-measure and is equal to 11% in the hadith corpus. On another scientific corpus, they obtained 28% of improvement. This shows that the hadith corpus is more ambiguous. They also varied the sizes of training and testing sets and showed that the improvement is better in the scientific corpus and the more the corpus is bigger, the better the results are. Finally, they showed that decision trees performed better than Bayesian, Entropy and Vector space models, with an F1-mesaure equal to 0.70. In a more recent paper (Harrag et al., 2011), the same authors evaluated, on the same dataset, Artificial Neural Network (ANN) and SVM (Support Vector Machines) classifiers. They also assessed three stemming techniques: (i) the rule-based morphological stemming (i.e. Dictionary-Lookup stemming); (ii) root-based stemming; and (iii) light stemming. The results showed that ANN performed better than SVM. The three stemming techniques enhanced the results of these two classifiers compared to the experiments with no stemming. The best results were obtained with the ANN classifier plus light stemming or Dictionary-Lookup stemming with an F-measure equal to 0.5.

Alkhatib (2010) proposed to classify hadiths of Sahih Al-Bukhari. She started by removing chains of narrators, stop words and affixes, without detailing the used stemming tool. Then she computed TF-IDF and compared four classifiers: Rocchio algorithm, K-NN algorithm (K- Nearest Neighbor), Naïve Bayes algorithm and SVM algorithm. In the experiments, she used 1500 Hadiths from 8 themes. 90% of the hadiths were used for training and 10% for testing. The authors claimed to reach 100% of recall in all the experiments. The average precision ranged from 63.36% (for SVM) to 67.11% (for Rocchio).

A similar work based on the bag-of-words approach for text representation has been presented by Al-kabi and Al-Sinjilawi (2007). They performed sanad and stop-word removal, stemming and indexing with TF-IDF. They proposed supervised text classification based on Vector Space Models including several similarity measures. Their work concerned 12 chapters from Sahih Al-Bukhari, but did not precise the exact size of the training set, while they used only 80 hadiths to assess the results. The F-measure ranged from 0.42 for the Dice Factor to 0.85 for the Naïve Bayesian similarity measure. The authors also confirmed that the results deteriorate without stemming.

Jbara (2010) continued the work of Al-Kabi and Al-Sinjilawi (2007) by adopting the same preprocessing and indexing techniques. However, they used only the cosine coefficient in the classification step. Nevertheless, the compared three methods for representing the features: (i) the stem-based method of Al-Kabi and Al- Sinjilawi (2007); (ii) the word-based method; and, (iii) a hybrid method using a vector of words expanded by their stems. In the experiments, the extended the training set to 13 domains and 1321 hadiths. The results show that the third method performed better than the second one (respectively the first one) with an average improvement of F-measure by 49% (respectively approximately 37%).

Table 1 compares the above cited hadith classification approaches, focusing on the key elements. We remark that all these approaches used TF-IDF to represent hadith texts. We should also add that these works focus, mainly, on Sahih Al-Bukhari (only Harrag et al. (2008; 2009; 2011) tested on the nine books). Also, these works considered a limited number of classes from these books i.e. 14 chapters for the largest dataset (Harrag et al., 2008; 2009; 2011).

Besides, existent work showed the impact of linguistic processing, as varying the stemming/text representation techniques affected the classification results. Nevertheless, only Harrag et al. (2011) and Jbara (2010) tried to compare different indexing units. We lack a work which shows the relative accuracy of the most frequently used indexing units. Besides, the performance of Arabic stemmers and morphological disambiguation tools has not been deeply studied in this field. Thus, we feel a growing need for assessing the accuracy of Arabic NLP tools in text classification.

Indeed, existing works focused mainly on comparing classification algorithms. Despite the great efforts and the variety of the algorithms which have been tested, it is hard to select the best model without unifying the assessment framework. In addition, we cannot interpret the F-measure values and compare objectively these works as they did not use exactly the same datasets.

Table 1: comparative study of hadith classification approaches.

Reference	#domains	#hadiths	Linguistic tools/approaches	Classification algorithm	Results
Harrag et al. (2008; 2009)	14	453	stop-word removal and rule-based morphological stemming	Decision trees, Bayesian, Entropy and Vector space models	F1-mesaure = 0.70 with decision trees
Harrag et al. (2011)			Three stemming approaches: rule-based, root-based and light stemming	ANN vs. SVM	F-measure = 0.5 with ANN + light or rule-based stemming
Alkhatib (2010)	8	1500	Removing chains of narrators, stop words and affixes	Rocchio, K-NN, Naïve Bayes and SVM	Recall=100% Precision=63.36% (SVM) and 67.11% (Rocchio)
Al-Kabi and Al-Sinjilawi (2007)	12	80 (for testing)		Vector Space Models with several similarity measures	F-measure: from 0.42 (Dice Factor) to 0.85 (Naïve Bayesian)
Jbara (2010)	13	1321	Removing chains of narrators, stop words and affixes Stem-based, Word-	The cosine coefficient	49% and 37% of improvement in F-measure for the hybrid method

			based and hybrid representation.		compared to the word-based and the stem-based methods

3. THE PROPOSED TEXT CLASSIFICATION PROCESS

Based on our discussion on the previous section, our work in this paper stands by the following aspects:

1- We will vary as much as possible the indexing unit, thus assessing six different NLP tools.
2- We will enlarge the dataset, thus covering 23 classes.
3- As many classification models have been tested on the hadith corpus, we will assess only the most successful ones.

Fig. 1 illustrates the main phases of text classification process followed to conduct the accuracy of comparison between the three selected algorithms. These steps are detailed in the next subsections.

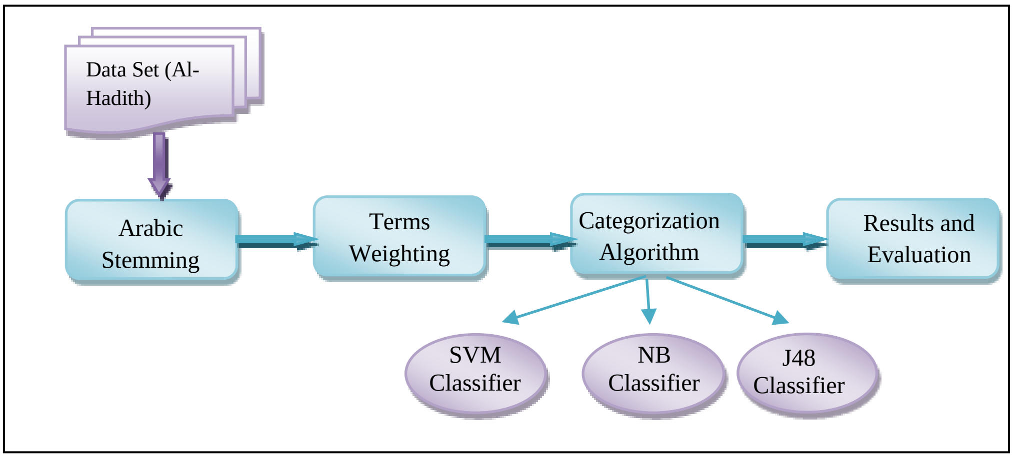

Fig.1. The general methodology

3.1 DataSet Description

Our dataset is composed by hadiths extracted from Sahih Al-Bukhari, which is a collection of the traditions of the Prophet of Islam Muhammad (PBUH). A hadith presents the reports of the Prophet's saying and deeds. It is composed of two branches: (i) the **Sanad** which refers to the chain of narrators and (ii) the **Metn** which refers to real content of the hadith (Al-Kabi, et al. 2007). Al-Bukhari uses the term "book" for the classification of the Al-Hadith's subject. The book means a chapter, a category and a class. Sahih Bukhari is divided into 7031 hadiths depending on their subjects. In our work, we select 795 Hadiths divided into 23 categories to be included in the experiment, as show in table 2.

Table 2: The selected classes.

The Book of Prayer Hall	باب سترة المصلي
The Book of the Eclipse Prayer	باب الكسوف
The Book of Oppressions	باب المظالم

the Book of Bathing	باب الغسل
The Book of Menstrual Periods	باب الحيض
The Book of The Two Festivals	باب العيدين
The Book of Manumission of Slaves	باب العتق
The Book of Distribution of Water	باب المساقاة
The Book of Agriculture	باب المزارعة
The Book of Wills and Testaments	باب الوصايا
The Book of Patients	باب المرضى
The Book of Al-Adha Festival Sacrifice	باب الأضاحي
The Book of Virtues of Madinah	باب فضائل المدينة
The Book of Penalty of Hunting while on Pilgrimage	باب جزاءالصيد
The Book of Minor Pilgrimage	باب العمرة
The Book of Actions while Praying	باب العمل فى الصلاة
The Book of Invoking Allah for Rain	باب الاستسقاء
The Book of Shortening the Prayers	باب تقصيرالصَّلاة
The Book of Hiring	باب الإجارة
The Book of Loans, Payment of Loans, Freezing of Property, Bankruptcy	باب الاستقراض وأداء الديون والحجر والتفليس
The Book of Divine Will	باب القدر
The Book of Tricks	باب الحيل
The Book of Supporting the Family	باب النفقات

3.2 Arabic Stemming

Stemming is a very essential technique for processing strong morphological languages such as Arabic. Therefore, many stemming techniques were introduced for Arabic language among them we use Al-Stem Darwish (Darwish et al., 2009), Al-Stem Alex (Fraser et al., 2002), Khoja's stemmer (Khoja, 1999), Aramorph's analyzer (Ayed et al., 2012), Quadrigrams, and Trigrams (Syiam et al., 2006). We introduce, in the following paragraphs each stemmer.

3.2.1 Khoja's stemmer

The root-Based approach uses morphological analysis to find the root of a given Arabic word. Khoja stemmer (Khoja, 1999) is an example of root-based stemmer; it is designed at late 1990s. It has developed an algorithm that removes the longest suffix and the longest prefix. It, then, matches the remaining word with verbal and noun patterns, to extract the root. The stemmer makes use of several linguistic data files such as a list of all diacritic characters, punctuation characters, definite articles, and 168 stop words (Khoja, 1999).

3.2.2 Light stemming

The light stemming refers to the process of stripping off a small set of prefixes and/or suffixes without trying to deal with infixes or recognize patterns and find roots (Syiam et al., 2006). Al-Stem of Darwish is an example of light stemming which was modified by Leah Larkey from University of Massachusetts and further modified later by David Graff form LDC (Darwish, et al., 2009). The stemmer removes 24 frequently encountered prefixes (والـ،

فالـ ، بالـ ، بت، يت، لت، مت، وت، ست،نت، بمـ، لمـ ، ومـ ، كمـ ، فمـ ،الـ ، للـ ، وي، لي، في، وا، فا، لا، با), and 22 commonly occurring suffixes (ـة، يه، ين، نا، تك، ية، ها، هن، هم، كم، تم، ته، تي، ـان ، وه ، ون، وا، ـات ـه، ي، ا). Another approach of light stemming has been defined by Alexander Farser (Farser et al., 2002), which follows the same steps as Al-Stem Darwish except that two kinds of spelling variations were considered. The first is the confusing of the letter (ي) and the letter (ى) at the end of a word, and the second is to write (آ،أ،إ) as (ا).

3.2.3 N-gram-based indexing

In statistical stemmer, a n-gram is a set of n consecutive characters extracted from a word. The main idea behind this approach is that, similar words will have a high proportion of n-grams in common (Syiam et al., 2006). In our work we use trigrams and quadrigrams for Al-Hadith classification referring to many research works, in the field which showed that n-grams character of lengths 3 or 4 were the most fruitful methods (e.g. Darwish & Oard, 2002; Mayfield et al., 2002). The trigrams of a token are a set of continuous 3 letter slices of the string. For example, the trigrams for the word الرسول are: الر، لرس، رسو، سول. The Quadrigrams of a token are a set of 4 letter slices for the same example: الرس، لرسو، رسول.

3.2.4 Ayed's morphological disambiguation tool

Morphological analyzers attempt to find stems or any number of possible stems for each word automatically using a software program. In 2002, Tim Buckwalter designed Aramorph system which is downloadable from the Linguistic Data Consortium (LDC). It is one of the most well-known Arabic morphological analyzer and part-Of-Speech tagger system. The text to be analyzed in AraMorph should be transliterated into ASCII before any processing. The lexicons are supplemented by three morphological compatibility tables used for controlling prefix-stem combinations, stem suffix combinations, and prefix-suffix combinations (Buckwalter, 2002). As Aramorph provides all the possible solutions for a given word, Ayed et al. (2012) developed a context-based disambiguation tool allowing to select the right solution and to recognize the morphological features (POS, gender, number, voice, etc.) of vocalized and/or non-vocalized Arabic text words.

3.3 Term Weighting

After stemming, we tokenize the analyzed hadith and we save it into a suitable format for the Weka toolkit (Hall et al., 2009), which uses ARF (Attribute Relation File) format using the converter "StringToWordVector". There are different approaches for text indexing among of them TF*IDF, which is the most commonly used weighting approach to describe documents in the vector space model. TF*IDF determine the relative term frequency (TF) in a specific document compared to the inverse proportion of that word over the entire document corpus (IDF). In our implementation, we use the normalized TF*IDF to overcome the problem of variant documents' lengths represented by the following formula:

$$w_{ij} = \mathrm{TF} * \mathrm{IDF}(\mathrm{t}_i, \mathrm{d}_j) = \frac{f_{ij}}{\sqrt{\sum_{k=1}^{M} f_{kj}^2}} * \log(\frac{\mathrm{N}}{\mathrm{n}_i}) \quad (1)$$

Where w_{ij} represents the weight of the word i in the document (hadith) j. N is the number of hadiths in the data set, M is the number of words used in the feature space, f_{ij} is the frequency of a word i in hadith j, and n_i denotes the number of hadiths that word i occurs in at least once.

3.4 Categorization algorithm

As a final step of the proposed methodology, we transform Al-Hadith into a vector model space (Harrag et al., 2008), each vector can be represented by the weights of words in a document with respect to the space dimension. The number of dimensions equals the number of terms or keywords used. For example:

$$\overrightarrow{H_{ij}} = (\overrightarrow{w_{1j}}, \overrightarrow{w_{2j}}, \dots, \overrightarrow{w_{ij}}) \quad (2)$$

Where $\overrightarrow{w_{ij}}$ is the weight vector of word i in Hadith j.

- Once the data are ready for experimentation, we conduct the experiment using the Weka toolkit. The resulting dataset will be classified into twenty four classes. It will be used to assess the performance and efficiency of (i) the Sequential Minimal Optimization (SMO), (ii) the C4.5 algorithm which is implemented in Weka under the name J48 algorithm, and (iii) the Naïve Bayes algorithms. We used k (10) fold cross-validation techniques where the datasets are randomly partitioned into 10 mutually exclusive subsets or folds D_1, $D_2 \dots$, D_k. In iteration i, the partition D_i is reserved as the test set, which is used to test the classifier effectiveness and the remaining partitions are collectively used to train the model.

4. RESULTS AND DISCUSSION

After classify data, the results are collected for each algorithm, in order to measure the accuracy of each classifier. Table 3 illustrates an overall comparison between these classifiers performed using the accuracy measure to determine the best of them.

Table 3: The results of accuracy measure.

	Accuracy		
	J48	**NB**	**SMO**
Khoja	**44.22 %**	48.34%	**57.50 %**
Aramorph	43.01 %	48.05 %	54.84 %
Trigrams	39.11 %	45.66 %	55.47 %
Quadrigrams	42.38 %	45.78 %	48.42 %
Al-Stem Darwish	38.86 %	48.42 %	50.94 %
Al-Stem Alex	38.11 %	**48.55%**	52.45 %

The SMO (SVM) classifier achieves the highest accuracy using Khoja's stemmer. On the other hand the results in NB classifier are less accurate than in the SMO classifier. The J48 classifier achieves the lowest accuracy compared with the other two classifiers using 10-cross validation. Another measure that is obtained from the experiments is the performance of Stemming algorithms applying to our Dataset, so we noticed that Khoja's stemmer outperformed the other stemming algorithms, followed by AraMorph analyzer. The statistical stemmer (Trigrams and Quadrigrams) is classified in the third place, followed by Al-Stem Darwish, and the worst was Al-Stem Alex.

The accuracy of the classifiers is expressed in terms of recall, precision averages and the F-measure, as described in (Lewis, 1995). The results are respectively shown in Fig. 1, Fig. 2 and Fig. 3.

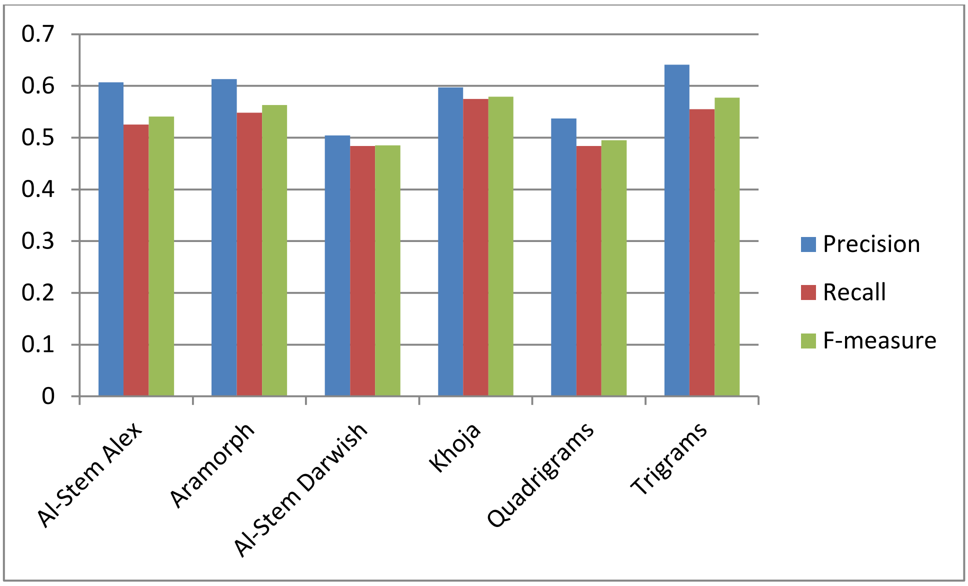

Fig 2. Comparison of stemming algorithms according to SMO classifier

The best result using the SVM classifier (SMO) was achieved for Khoja's stemmer with a recall average value of 0.575, and the worst result was for Al-Stem Darwish and quadrigrams with a recall average value of 0.484.

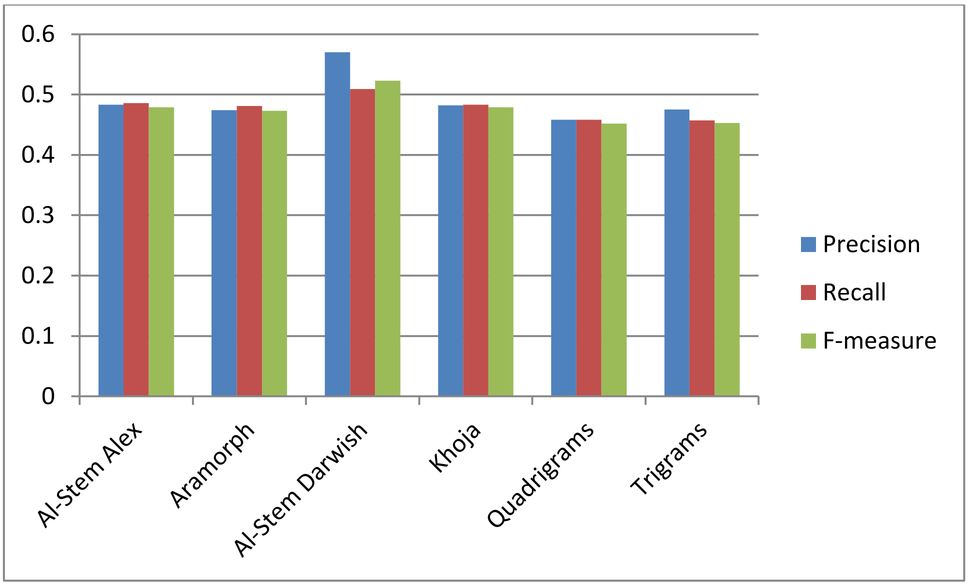

Fig 3. Comparison between stemming algorithms according to NB classifier

In Fig. 3, Al-Stem Darwish performs much better than the other stemmers at the recall level. It provides the highest value of recall (0.509) followed by Al-Stem Alex (0.486).

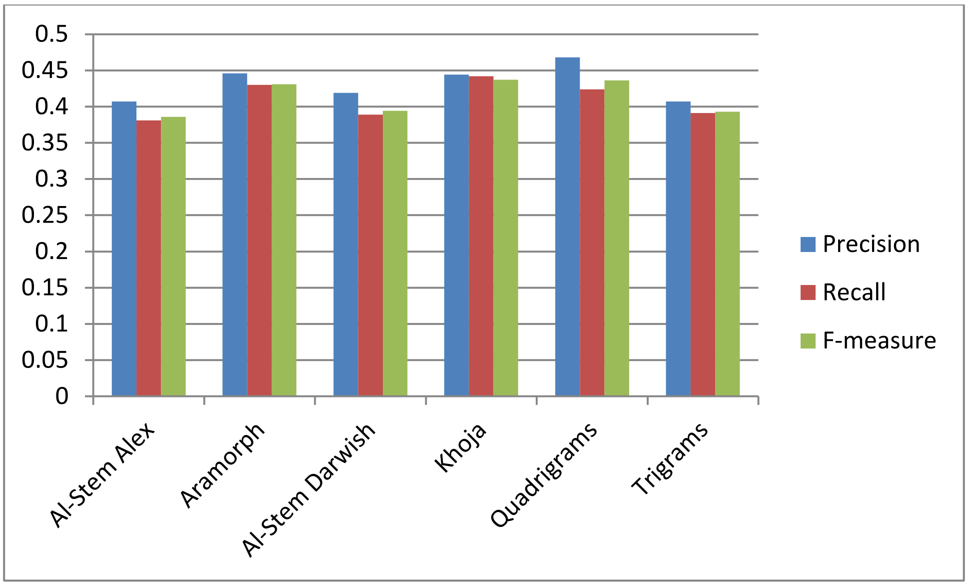

Fig 4. Comparison of stemming algorithms according to J48 classifier

The best value of recall using the J48 classifier is 0.442 given by Khoja stemmer. We note that Khoja gives, commonly, better results when it is matched with SMO and J48 classifiers. The Khoja stemmer presents a hybrid technique that defines a list of rules to determine the right stems (Khoja, 2001). This stemmer is considered as a statistical and rule-based tool. This hybrid characteristic corresponds to the data type of hadith texts. These texts need contextual knowledge with statistical measures extracted from other corpora to determine the accurate stem as the hadiths' words match, at the same time, the classical and the modern lexicon. The Khoja stemmer gives better results when it is coordinated with SMO classifier which is based on SVM approach. This combination gives the highest F-measure (0.579). The SVM classifier supports high dimensional spaces (Raghavan et al., 2007). This particularity corresponds to our datasets where each hadith may be described by a high number of terms or keywords (dimensions). We can conclude that the combination of the rule-based stemmers and the statistical classifiers performed better to give enhanced results of classification.

5. CONCLUSION

Several algorithms have been implemented to solve the problem of text categorization. Our study aimed to compare three known classification techniques using Arabic text documents which lie into twenty three classes. The comparison was based on two main aspects for the selected classifiers, accuracy and time. In terms of accuracy, results show that the Sequential Minimal Optimization (SMO) classifier achieves the highest accuracy, followed by the Naive Bayes (NB) classifier, followed by the J48 (C4.5) classifier. On the other hand, results show that Khoja's stemmer outperformed the other tools.

As a future work, we are looking to extend this work by applying some preprocessing steps to the data set such as removing stop words. Also, we aim to increase the number of classes and take more numbers of hadiths.

6. REFERENCES

Aggarwal, C.C, Zhai, C.X, (2012). A survey of text classification algorithms. pp 163-222, 2012.

Al-Harbi, S., Almuhareb, A., Al-Thubaity, A. (2008). Automatic Arabic Text Classification, les Journées internationales d'Analyse statistique des Données Textuelles.

Al-Kabi, M., Kanaan, G., Al-Shalabi, R. (2005). Al-Hadith text classifier. Journal of Applied sciences 5(3):584-587, 2005.

Al-Kabi, M., Al-sinjilawi, S. I. (2007). A comparative study of the efficiency of different measures to classify arabic text, Journal of Pure & Applied Sciences Volume 4, No. 2 June 2007.

Al-Khatib, M. (2010). Classification of Al-Hadith Al-Shareef Using Data Mining Algorithm, the European Mediterranean & Middle Eastern Conference on Information systems (EMCIS'2010) Abu-Dhabi United Arab Emirates.

Al-Shalabi, R., Kanaan, G., Gharaibeh, M. H. (2006). Arabic Text Categorization Using kNN Algorithm.

Ayed R., Bounhas I., Elayeb B., Evard F., Bellamine B. S. N. (2012). A Possibilistic Approach for the Automatic Morphological Disambiguation of Arabic Texts, Proceedings of 13th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), August 08-10, 2012, Kyoto, Japan, IEEE Computer Society, pp. 187-194.

Buckwalter, T. (2002). Arabic Morphological Analyzer Version 1.0, Linguistic Data Consortium LDC 2002 L49.

Darwish, K., Arafa, W., Eldesouki, M. I. (2009). Stemming techniques of Arabic Language: Comparative Study from the Information Retrieval Perspective. ISSR Cairo University, 2009.

Darwish, K., Oard, D. W. (2002). Term Selection for Searching Printed Arabic. In proceeding of the 25th ACM SIGIR conference on research and development in information retrieval. pp. 261-268.

El-Kourdi, M., Bensaid, A., Rachidi, T. (2004). Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm.

Fraser, A., Xu, J. and Weischedel, R. (2002).TREC 2002 Cross-lingual Retrieval at BBN.

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

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

Harrag, F., El-Qawasmah, E., Pichappan, P. (2009). Improving Arabic text categorization using decision trees. The First International Conference on Networked Digital Technologies NDT '09, Ostrava, Czech Republic, 28-31 July, 2009.

Harrag, F., Hamdi-Chrif, A. (2008). Classification des Textes Arabes Basée sur l'Algorithme des Arbres de Décision, International Conference on Web and Information Technologies ICWIT'08, Sidi Bel Abbes, Algeria, 29-30 June, 2008.

Jbara, K. (2010). Knowledge discovery in Al-Hadith using text classification algorithm. Journal of American Science, 6(11): 409-419.

Khoja, S. (2001). APT: Arabic part-of-speech tagger. In: Proceedings of Student Workshop at the Second Meeting of the North American Association for Computational Linguistics, Carnegie Mellon University, Pennsylvania, USA.

Khoja, S., Garside, R. (1999). Stemming Arabic text. Computing Department, Lancaster University, Lancaster, 1999.

Lewis D. (1995). Evaluating and optimizing Autonomous Text classification systems. In proceeding of the 18th international ACM SIGHIR Conference on research and development in information retrieval, pp 246-254, 1995.

Mayfield, J., McNamee, P., Costello, C., Piatko C., and Banerjee, A. (2002). JHU/APL at TREC 2001: Experiments in Filtering and in Arabic, Video, and Web Retrieval. The Tenth Text REtrieval Conference, NIST Special Publication (500-250, pp. 322-330), Gaithersburg, Maryland: National Institute of Standards and Technology.

Raghavan, H., Allan, J. (2007). An interactive algorithm for asking and incorporating feature feedback into support vector machines. In: ACM SIGIR Conference.

Soucy, P., Mineau, G. W. (2005). Beyond TF-IDF weighting for Text categorization in the vector space model. IJCAI'05 Proceedings of the 19th international joint conference on Artificial intelligence, 2005.

Syiam, M., Fayed, Z. T., Habib, M. B. (2006). An intelligent system for Arabic text categorization, IJICIS 6 (1), January 2006.

Wahbeh, A. H, Al-Kabi M. (2010). Comparative Assessment of the Performance of Three WEKA Text Classifiers Applied to Arabic Text, 21(1):15- 28, 2010.