EFFECTIVE ARABIC STEMMER BASED HYBRID APPROACH FOR ARABIC TEXT CATEGORIZATION

Meryeme Hadni[1], Said Alaoui Ouatik[1] and Abdelmonaime Lachkar[2]

[1]L.I.M, Faculty of Science Dhar EL Mahraz (FSDM), Fez, Morocco
meryemehadni@gmail.com, s_ouatik@yahoo.com
[2]L.S.I.S, E.N.S.A,University Sidi Mohamed Ben Abdellah (USMBA),Fez, Morocco
abdelmonaime_lachkar@yahoo.fr

ABSTRACT

Text pre-processing of Arabic Language is a challenge and crucial stage in Text Categorization (TC) particularly and Text Mining (TM) generally. Stemming algorithms can be employed in Arabic text pre-processing to reduces words to their stems/or root. Arabic stemming algorithms can be ranked, according to three category, as root-based approach (ex. Khoja); stem-based approach (ex. Larkey); and statistical approach (ex. N-Garm). However, no stemming of this language is perfect: The existing stemmers have a small efficiency.

In this paper, in order to improve the accuracy of stemming and therefore the accuracy of our proposed TC system, an efficient hybrid method is proposed for stemming Arabic text. The effectiveness of the aforementioned four methods was evaluated and compared in term of the F-measure of the Naïve Bayesian classifier and the Support Vector Machine classifier used in our TC system. The proposed stemming algorithm was found to supersede the other stemming ones: The obtained results illustrate that using the proposed stemmer enhances greatly the performance of Arabic Text Categorization.

KEYWORDS

Arabic Language, Stemming Approaches, Text Categorization.

1. INTRODUCTION

The text categorization try to find a relation between a set of texts and a set of categories (tags, classes). This functional link was used in classification, filtering, and retrieval purposes. Machine learning is the tool that allows decide whether a document belongs to a set of predefined classes [26].Several text categorization systems have been conducted for English and other European languages, yet very little researches have been done out for the Arabic Text Categorization. Arabic language is a highly inflected language and it requires a set of pre-processing to be manipulated, it is a Semitic language that has a very complex morphology compared with English. In the process of text categorization the document must pass through a series of steps (Figure1): transformation the different types of documents into brut text, removed the stop words which are considered irrelevant words (prepositions and particles) ; and finally all words must be stemmed. Stemming is the process consists to extract the root from the word by removing the affixes [3, 4, 5, 7, 13, 16, 17]. To represent the internal of each document, the document must

passed by the indexing process after pre-processing. Indexing process consists of three phases [27]:

a) All the terms appear in the documents corpus has been stocked in the super vector.
b) Term selection is a kind of dimensionality reduction, it aims at proposing a new set of terms in the super vector to some criteria [18, 25, 31];
c) Term weighting in which, for each term selected in phase (b) and for every document, a weight is calculated by TF-IDF which combine the definitions of term frequency and inverse document frequency[1] (Figure2).

Finally, the classifier is built by learning the characteristics of each category from a training set of documents. After building of classifier, its effectiveness is tested by applying it to the test set and verifies the degree of correspondence between the obtained results and those encoded in the corpus.

In our work, we believe that the pre-processing of Arabic text is challenge and crucial stage. It may impact positively or negatively on the accuracy of any Text Categorization system, and therefore the improvement of the pre-processing step will lead by necessity to the improvement of any Text Categorization system very greatly.

To demonstrate this, in figure 2, we present an opposite example using Khoja stemmer. It produces a root that is not associated to the original word. For example, the word (منظمات) which means (organizations) is stemmed to (ظمأ) which means (he was thirsty) instead of the correct root (نظم).

The main objective of this work is to propose and evaluate a new and efficient stemming method as pre-processing tools for Arabic language in TC.

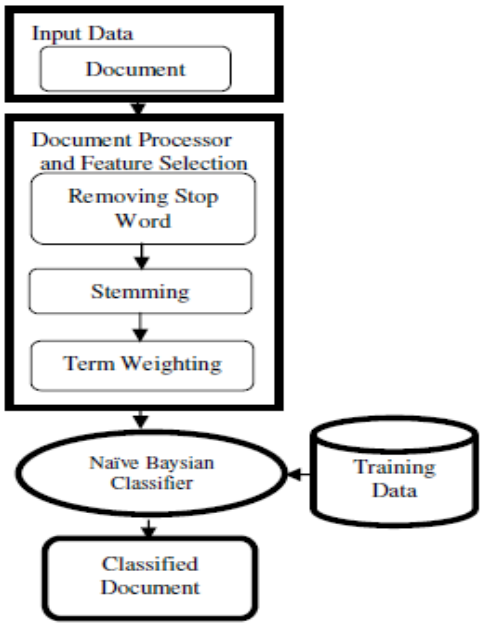

Figure1. Architecture of TC System

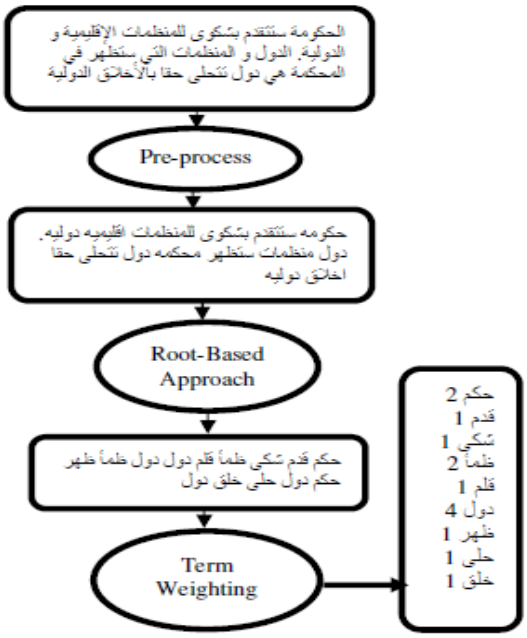

Figure2. Example of Pre-processing with Khoja Stemmer algorithm

Many Stemming algorithms have been developed for a wide range of languages including English [32], Malay [29], Latin [23], Indonesian [6], Swedish [8], Dutch [15], German and Italian [19], French [20], Slovene [24], and Turkish [12], Bangla [21], Chinese [2]. For Arabic Language, there are three different Stemming approaches: the root-based approach (Khoja [13]); the light stemmer approach (Larkey [17]); and the statistical stemmer approach (N-Gram [14, 22]). Yet no a complete stemmer for this language is available.

In this paper, in order to improve the performance of stemming process and therefore the accuracy of our TC system, an efficient hybrid method is proposed for stemming Arabic text. The proposed stemming method was found to supersede the other stemming ones.

The rest of this paper is organized as follow. A brief review on related work in Arabic Text pre-processing is presented in the next section. Section 3 describes the proposed Stemming algorithm for the Arabic Language. Section 4 presents two Approaches for Text Categorization: Naïve Bayesian classifier and SVM classifier used as machines algorithm learning for our Arabic TC. The achieved experimental results are presented and discussed in the section 5. Finally, section 6 concludes this work.

2. RELATED WORKS IN ARABIC TEXT PRE-PROCESSING

When categorizing text documents, not all features equally represent the document's semantics; in fact, some of these features may be redundant and add nothing to the meaning of the document; others might be synonymous and therefore capturing one of them is enough to enhance the semantic for categorization purposes. Consequently, the effective selection of feature words, which reflect the main topics of the text, is an important factor in text categorization.

Stemming algorithms can be employed in Arabic text pre-processing to reduces words to their stems/or root. Very little research has been carried out on Arabic text. The nature of Arabic text is different than English text, and pre-processing of Arabic text is more challenging stage in Text Categorization (TC) particularly and Text Mining (TM) generally. The effect of the pre-processing tools on Arabic TC is an area of research.

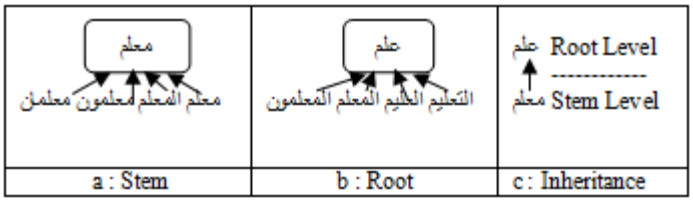

Figure3. An Example of Root/Stem Pre-processing with Stemming

Stemming is the process for reducing inflected words to their stem or root form (generally a written word form). The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. For example, in Figure 3, the root of the Arabic word (المعلمون, the teachers) is (علم, science). While a stem is simply defined as a word without a prefix or/and suffix. For example, the stem of the Arabic word (المعلمون, the teachers) is (معلم, teacher).

Arabic stemming algorithms are classified under three categories: *Root-Based Approach* (Khoja [13]); S*tem-Based Approach* (Larkey [17]); and *Statistical Approach* (N-Gram [14, 22]).In this section, a brief review on the three stemming approaches for stemming Arabic Text is presented.

- *Root-Based Approach* uses morphological analysis to extract the root of a given Arabic word. Several algorithms have been developed for this approach. Al-Fedaghi and Al-Anzi proposed an algorithm tries to find the root of the word by matching the word with different patterns with all possible affixes attached to it, and does not removed any prefixes or suffixes [3]. In this context and with other technical Al-Shalabi morphology system applies several algorithms to find the roots and patterns [4]. This algorithm searches the root in the first five letters of the word by removing the longest prefix. Khoja contribute with a very important algorithm that removes affixes, and verified for each time it does not removed part of the root. Finally, find the matching between patterns and the rest of word to extract the root [16].
- The objective of the *Stem-Based Approach* or Light Stemmer approach is removing the most frequent suffixes and prefixes but same time this algorithm change the form of word. Light stemmer approaches have been proposed by some authors [5, 19, 20, 8], but for the Arabic language there is no standard light stemmer algorithm, all studies in this field based of a set of suffixes and prefixes. However this list of affixes is not definite. Darwish in his work improve that Al-Stem light stemmer is less effective than light10 into TREC 2002 [10]. Chen and Gey [8] introduced a light stemmer similar to light10, but that removed more prefixes and suffixes. It provided the results more effective than Al-Stem, and was not directly compared to light10.
- In *Statistical Approach*, related words are grouped based on various string similarities measures. Such approaches often involve N-Gram [14, 22]. The main idea of N-Gram approach is that the similar words will have a proportion high of N-Grams. And extract the root after the comparing the similar characters. Specific values for n-Gram are bigrams or trigrams

Note that, among the best known Arabic Stemming algorithms for each approach, we can select the Khoja Stemmer as *Root-Based* [13]; Larkey as *Stem-Based* [17] and the N-Gram as *Statistical Based* [14]. These Stemming algorithms present some weakness where there are used alone. The following table summarizes some of these Weaknesses.

Table1. Some Weaknesses of Khoja, Larkey and N-Gram Stemmers

Algorithm	**Approach**	**Weakness**
Khoja	Root-Based	-The root dictionary requires an update to ensure that the new terms detected are correctly stemmed. - The Khoja Stemmer replaces a weak letter with (ـُو) which produces a incorrect root. For example, the word (منظمات) which mean (organizations) is stemmed to (ظمأ) which means (he was thirsty) instead of (نظم). - Third, the Khoja Stemmer fail to remove all of affixes .
Larkey	Stem-Based	Light stemming removes of affixes predefined in the list without checking if they remained is a stem. And in some cases, truncates it from the word and produces an erroneous stem.
N-Gram	Statistical Approach	The N-Gram algorithm returns a lot of documents that are not necessarily relevant. And the production of an index files size exorbitant.

Yet, no a complete stemmer for the Arabic language is available. To overcome this problem and therefore to enhance the performance of the stemming algorithm, in the next section, we propose a new efficient stemmer as a hybrid algorithm of the three existing approaches.

3. PROPOSED STEMMING ALGORITHM

In this section, we present a Hybrid method which incorporates three different techniques for Arabic Stemming to overcome the aforementioned weaknesses. The three techniques are: Khoja Stemmer, Light Stemmer and N-Gram. Each one of these techniques needs individually some adaptation to be appropriate for Arabic language usage.

The proposed algorithm starts with constructing the root file containing more than 9,000 valid Arabic roots taken from a dictionary of Arabic words, and constructing of the stop word file. The next step is the normalization of documents, after the removal of punctuation, diacritics and stop-word. The result of this step is used as input in the process of removing prefix / suffix, by checking if the word match on of the patterns extract the relevant word, otherwise to remove the suffix and prefix respectively, with verification of the length after each removal of affixes. Finally the valid root is found by using the bi-gram and the Dice measure similarity.

Our proposed Stemming algorithm can be presented as follow:

1. Remove punctuation, numbers, words written in other languages, and any Arabic word containing special characters.
2. Remove the diacritics of the words, because the diacritics are not used in extracting the Arabic roots.
3. Remove stop words.
4. Normalize the documents by doing the following:
    4.1. Replace the letter (“إ أ آ ا”) with (“ا”), and replace the letter (“ؤ ء”) with (“أ”).
    4.2. Replace the letter (“ى”) with (“ا”).
    4.3. Replace the letters (“أا اأ”) with (“أ”).
5. Remove the duplicate.
6. Remove (“ ال ”) and its components:
(“لل ال فبال لبال وبال فال تال وال بال كال”).
7. Consider three cases depending on length of the word:
    7.1. Length=3: Return stem-candidate if less than or equal to three. Attempting to shorten stems further results in ambiguous stems.
    7.2. If Length = 4 then:
        7.2.1. If the word matches one of the patterns (figure3, 4) extract the relevant stem-candidate and return.
        7.2.2. Else, attempt to remove length-one suffixes and prefixes from S1 and P1 in that order provided the word is not less than length three.
        7.2.3. Fin Si.
    7.3. If Length>4 then:
        7.3.1. If the word matches a pattern, extract stem-candidate of length three.
        7.3.2. Else, attempt to remove suffixes and prefixes in length four, three and two respectively.
        7.3.2.1. If the word is still superior four characters in length, remove one character suffix. If a suffix is remove and a resulting term of length four results, send the word back through step 7.2. Otherwise, attempt remove one character prefixes, and if successful send the resulting length-four term to step 7.2.
        7.3.2.2. Send the resulting term to step 7.2.
        7.3.3. Fin Si.
    8. Partition the stem-candidate into bi-gram.

9. Compare the list of the stem bigrams with the bigrams in the root file that were created of the pre-processing steps and return all roots that contain any of the word's bigrams. Let L the list of all roots found.
10. for each element E to L:

10.1. Calculate the similarity between E and stem-candidate using the Dice measure and return the most similar root.

11. If distance>0 then return stem-valid.
12. Else, the original word should be returned.
13. Fin Si.

Table2. Suffix List

Suffix List	
List1	ن ا ت ة ك ه ي
List2	وه ين اة تم ته تن ني هن ون يه ان تك تة كم نك نه هم وك اه تن تي نا نة ها وا وة يك يا ية
List3	تين كهم نيه نهم ونه وها يهم ونا ونك وني وهم تكم تنا تها تني تهم كما كها ناه نكم هما تان
List4	كموه ناها وننني ونهم تكما تموه تماه كماه ناكم ناهم نيها وننا

Table 3. Prefix List

Préfix List	
List 1	ا ب ت س ف ك ل م ن ه و
List2	سن لي سي لن مت من نت وت يت ات ان سأ ست سن أت أن
List3	فست فسن فسي مست نست وتت يست أست است تست تتن
List4	أتست أنست اتست انست ستست سنست

After representing all documents by their weights vectors using Term Frequency Inverse Document Frequency weighting (TFIDF) [28], our system for Arabic TC has used two classifiers Naïve Bayesian (NB) and Support Vector Machine (SVM) to perform text classification.

4. APPROACH TO TEXT CATEGORIZATION

Many machine learning techniques have been applied to Automatic CT, Naïve Baysian (NB) [9] and Support Vector Machine (SVM) [11, 30].

This section covers briefly NB and SVM approach; it describes the general nature, process for training and document classification.

4.1. Naïve Bayesian Classifier

The Naïve Bayesian [9] is a simple probabilistic classifier based on Baye's theorem. When this classifier is applied to the categorization we use l'equation1:

$$\mathrm{p(class|document)} = \frac{\mathrm{p(class).\,p(document|class)}}{\mathrm{p(document)}} (1)$$

Where:

P (class| document): It's the probability that a given document D belongs to a given class C.

P (document): The probability of a document, it's a constant that can be ignored.

P (class): The probability of a class, it's calculated from the number of documents in the category divided by documents number in all categories.

P (document| class) : it's the probability of document given class, and documents can be represented by a sets of words:

$$p(document|class) = \prod_i p(wordi|class) (2)$$

So:

$$\mathrm{p(class|document)} = \mathrm{p(class).} \prod_{\mathrm{i}} \mathrm{p(wordi|class)} (3)$$

Where:

P($word_i$| class): The probability that a given word Occur in all documents of class C, and this can be computed as follows:

$$\mathrm{p(word_i|class)} = \frac{(\mathrm{Tct} + \lambda)}{(\mathrm{Nc} + \mathrm{V})} (4)$$

Where:

Tct: The number of times that the word occurs in that category C

Nc: The number of words in category C

V: The size of the vocabulary table

λ: The positive constant, usually 1, or 0.5 to avoid zero probability.

4.2. Support Vector Machine Classifier

Support Vector Machines (SVM) is a relatively new class of machine learning techniques first introduced by Vapnik [30] and has been introduced in TC by Joachims[11]. Based on the structural risk minimization principle from the computational learning theory, SVM seeks a decision surface to separate the training data points into two classes and makes decisions based on the support vectors that are selected as the only effective elements in the training set.

Given a set of N linearly separable points $S = \{x_i \epsilon R^n \mid i = 1, 2\ldots N\}$, each point x_i belongs to one of the two classes, labeled as $y_i \epsilon \{-1, +1\}$. A separating hyper-plane divides S into 2 sides, each side containing points with the same class label only. The separating hyper-plane can be identified by the pair (w, b) that satisfies:

$$\mathrm{w.x} + \mathrm{b} = 0$$

And:

$$\begin{cases} w.x_i + b \geq +1 \; if \; y_i = +1 \quad (5) \\ \mathrm{w.x_i} + \mathrm{b} \leq -1 \; \mathrm{if} \; \mathrm{y_i} = -1 \end{cases}$$

For i = 1, 2… N; where the dot product operation (.) is defined by :

$$\mathrm{w.x} = \sum \mathrm{w_i.x_i}$$

For vectors w and x, Thus the goal of the SVM learning is to find the optimal separating hyper-plane (OSH) that has the maximal margin to both sides. This can be formularized as:

Minimize:

$$\frac{1}{2}\mathrm{w.w}$$

Subject to:

$$\begin{cases} w.x_i + b \geq +1 \; if \; y_i = +1 \;\; for \; i = 1,2,\ldots,N \quad (6) \\ \mathrm{w.x_i} + \mathrm{b} \leq -1 \; \mathrm{if} \; \mathrm{y_i} = -1 \end{cases}$$

Figure 4 shows the optimal separating hyper-plane.

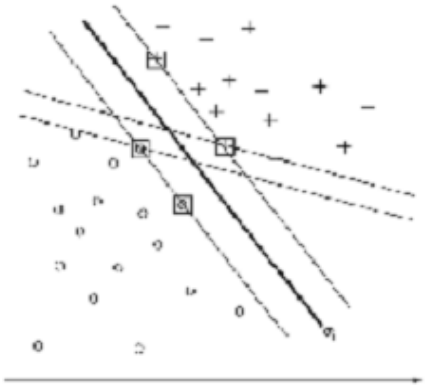

Figure4. Learning support vector classifier

The small crosses and circles represent positive and negative training examples, respectively, whereas lines represent decision surfaces. Decision surface σ_i (indicated by the thicker line) is, among those shown, the best possible one, as it is the middle element of the widest set of parallel decision surfaces (i.e., its minimum distance to any training example is maximum). Small boxes indicate the support vectors.

During classification, SVM makes decision based on the OSH instead of the whole training set. It simply finds out on which side of the OSH the test pattern is located. This property makes SVM highly competitive, compared with other traditional pattern recognition methods, in terms of computational efficiency and predictive accuracy [32].

The method described is applicable also to the case in which the positives and the negatives are not linearly separable. Yang and Liu [32] experimentally compared the linear

case (namely, when the assumption is made that the categories are linearly separable) with the nonlinear case on a standard benchmark, and obtained slightly better results in the former case.

5. EXPERIMENT RESULTS

To illustrate that our proposed Stemmer can improve the pre-processing step and therefore can enhance the performance of our Arabic Text Categorization system, in table 4 we present some results of Arabic words stemming using the three Stemmers: Khoja Stemmer, Light Stemmer and our Proposed Stemmer. In comparison with the other Stemmers, we observe that our Proposed Stemmer can produce better results:

For the Arabic words, the translation is presented in table 4. The using of N-Gram produce lower results in compared with the other stemmer, and an index files size exorbitant. Khoja Stemmer and Light cannot find root / stem for words (تستغرق,ركبتيه and فتستغفروا). However, our proposed Stemmer can extract all affixes /pattern and determines the correct roots (ركب, *غرق*, غفر).

For the words (افتتاح, خدمات, يبحثون), Khoja and Our Stemmer can provide de same root (فتح, خدم, بحث), which shows that we had a based-root approach. For the word ,بتوجيه Light cannot produce the correct stem, because of the used affixes (prefixes and suffixes); both Khoja and our Stemmer produce the correct root (وجه).

Table 4: Examples of words stemming results using Khoja Stemmer, Light Stemmer and our proposed Hybrid Stemmer

Terms	*translitération*	*translation*	*Khoja*	*Light*	*Our Stemmer*
بافتتاح	biftitah	opening	فتح	بافتتاح	فتح
منظمات	***monadamate***	***organization***	***ظما***	***منظم***	***نظم***
البنكية	al bankiya	banking	نكأ	بنك	بنك
لخدمات	likhadamate	services	خدم	لخدم	خدم
تستغرق	***tastaghrik***	***taking***	***تستغرق***	***تستغرق***	***غرق***
ركبتيه	***rokbatayhi***	***the knees***	***ركبتيه***	***ركبتي***	***ركب***
بتوجيه	bitawjih	orientation	وجه	بتوجي	وجه
فتستغفروا	fatastaghfiro u	you forgive	فتستغفروا	فتستغفروا	غفر
بستان	***bostane***	***garden***	***بسأ***	***بست***	***بستان***
ببشريتنا	bibachariyati na	for our humanity	بشر	ببشريتنا	بشر
بنقائصنا	binakaisina	with our imperfections	نقص	بنقائصنا	نقص
يتنافسها	yonafisoha	concurrence	نفس	يتنافسها	نافس
يبحثون	yabhatoune	you are searching	بحث	يبحث	بحث

Khoja Stemmer removed a part of the root when it removed the affixes and then added a hamza (أ) at the end, which provide incorrect roots (ظمأ, نكأ) for two words (منظمات, بنكية). On the contrary our Stemmer can produce correctly the corresponding roots (نظم, بنك)

Light Stemmer fails to extract all affixes. It cannot determine a two-letter suffix (يه) and produces a stem that is not representative for some semantically similar words. For example, the term: بستان, we use advantage of corpus statistics to solve a case of ambiguity. The two letters (ان) do not represent a suffix for this word. Even if this is not a suffix, Light Stemmer removes it from the word and produces an erroneous stem (بست). In contrast, Our Stemmer, uses the dictionary to check the availability of words, and can yield correctly the corresponding stem (بستان.).

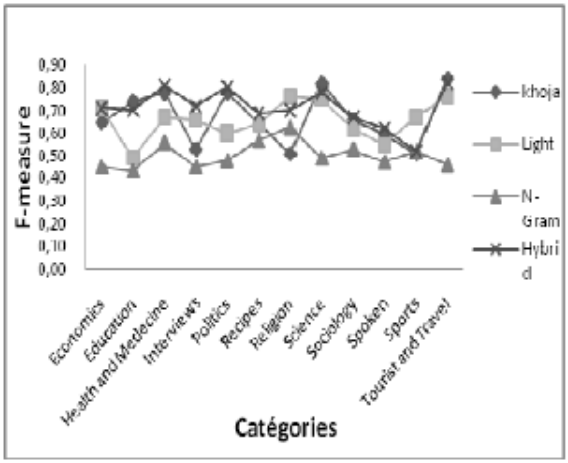

Figure 5: F-measure using Khoja Stemmer, Light Stemmer, N-Gram, and our proposed Hybrid Stemmer with NB classifier.

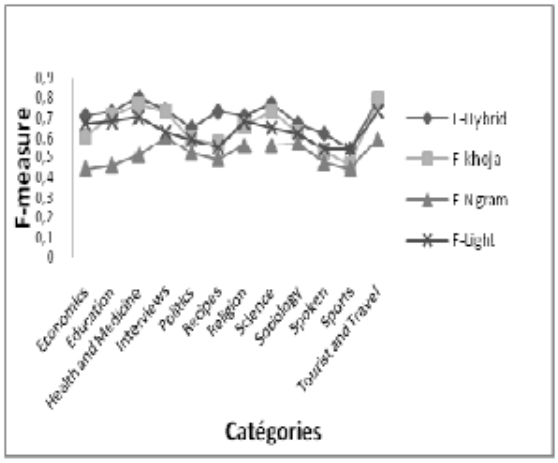

Figure 6.F-measure using Khoja Stemmer, Light Stemmer, N-Gram and our proposed Hybrid Stemmer with SVM classifier

To assess the performance of the proposed stemming algorithm, a series of experiments was conducted. The effectiveness of the aforementioned four Stemmers - Khoja Stemmer, Light Stemmer, N-Gram and our proposed Stemmer- was evaluated and compared in term of the F1-measure using the Naïve Bayesian [29] and the SVM classifiers [30] used in our TC system. The data set used in our experiments is extracted from a large Arabic corpus (Corpus of Contemporary Arabic (CCA)[26]. This corpus is classified into 12 categories with: economics, politics, education, science, health and medicine, interview, recipes, religion, sociology, spoken, sports, tourist and travel.

F1-measure can be calculated using Recall and Precision measures as follow:

$$F - measure = \frac{2 * (\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}} \quad (7)$$

Precision and Recall is defined as follows:

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (8)$$

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (9)$$

With:

- *True Positive (TP)* refers to the set of documents that are assigned correctly to the given category.
- *False Positive (FP)* refers to the set of documents incorrectly assigned to the category.
- *False Negative (FN)* refers to the set of documents incorrectly not assigned to the category

Figure 5 shows the obtained F1-measure results using Khoja Stemmer; Light Stemmer, N-Gram, and the proposed Hybrid Stemmer with NB classifier in our Arabic Text Categorization System. These results illustrate that using the proposed stemmer enhances greatly the performance of Arabic Text Categorization.

Using NB classifier, Khoja Stemmer performed better results for the three classes: Education (0.74) Science (0.82) and Tourist (0.84), compared to our Hybrid Stemmer: Education (0.7) Science (0.78) and Tourist (0.82). But an average F-measure the 67% using Khoja Stemmer compared to 71.16% for our Hybrid Stemmer, 66% for Light Stemmer and 50.33 for N-Gram.

To validate our results, we used another classifier which is the SVM, and the Figure6 illustrate the result using Khoja Stemmer, Light Stemmer, N-Gram and our Hybrid Stemmer.

Figure 6, presents the obtained F-measure results with SVM classifier, using hybrid proposed method, Khoja Stemmer, Light Stemmer and N-Gram. Our Hybrid Stemmer product of good results in the majority of class, with an average F-measure of 70.5%, 63.2% to Light Stemmer, 51% for N-Gram and 63.2% for Khoja Stemmer.

Table5. Average F1-measure using Khoja Stemmer, Light Stemmer, N-Gram and our proposed Hybrid Stemmer with NB and SVM classifier.

	Khoja	Light	N-Gram	Hybrid
NB	67%	63,2%	51%	70,5%
SVM	64,75%	66%	50,33%	71,16%

For Our Stemmer, we found that the NB classifier outperformed SVM.

6. CONCLUSION

Arabic text pre-processing is challenge and crucial stage. It may impact positively or negatively on the performance of any Text categorization system, and therefore the improvement of the pre-processing step will lead by necessity to the improvement of any Text categorization system very greatly. Many Stemming algorithms can be employed in Arabic Text Pre-processing to reduces words to their root/ or stem. Arabic stemming algorithms can be ranked, according to three category, as root-based approach (ex. Khoja); stem-based approach (ex. Larkey); and statistical approach (ex. N-Garm). However, no stemming of this language is perfect: The existing stemmers not have a high performance.

In this paper, in order to improve the accuracy of stemming and therefore the accuracy of our TC system, a new and efficient algorithm for Arabic text stemming is proposed. This latter is a hybridization of three well known Stemmers.

Our system for Arabic TC has used Naïve Bayesian classifier and SVM classifier to perform text classification. The obtained results illustrate that using the proposed stemmer with NB enhances greatly the performance of Arabic Text Categorization.

REFERENCES

[1] Aas K. and L. Eikvil. "Text categorisation: A survey", Technical report, Norwegian Computing Center. 1999.

[2] A.-H.Tan and P.Yu, A Comparative Study on Chinese Text Categorization Methods, PRICAI 2000 Workshop on Text and Web Ming, Melbourne,pp.24-35,August 2000.

[3] Al-Fedaghi S. and F. Al-Anzi. "A new algorithm to generate Arabic root-pattern forms". In proceedings of the 11th national Computer Conference and Exhibition. PP 391-400. March 1989.

[4] Al-Shalabi R. and M. Evens. "A computational morphology system for Arabic". In Workshop on Computational Approaches to Semitic Languages, COLING-ACL98. August 1998.

[5] Aljlayl M. and O. Frieder. "On Arabic search: improving the retrieval effectiveness via a light stemming approach". Proceedings of ACM CIKM 2002 International Conference on Information and Knowledge Management, McLean, VA, USA, 2002, pp. 340-347.

[6] Berlian, V., Vega, S. N., and Bressan, S. Indexing the Indonesian web: Language identification and miscellaneous issues. Presented at Tenth International World Wide Web Conference, Hong Kong, 2001.

[7] Chen A. and F. Gey. "Building an Arabic Stemmer for Information Retrieval". In Proceedings of the 11th Text Retrieval Conference (TREC 2002), National Institute of Standards and Technology. 2002.

[8] Carlberger, J., Dalianis, H., Hassel, M., and Knutsson, O. Improving precision in information retrieval for Swedish using stemming. In Proceedings of NODALIDA '01 - 13th Nordic conference on computational linguistics. Uppsala, Sweden, 2001.

[9] El-Kourdi, M., Bensaid, A., and Rachidi, T. "Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm," 20th International Conference on Computational Linguistics, 2004, Geneva.

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

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

[12] Ekmekcioglu, F. C., Lynch, M. F., and Willett, P. Stemming and N-Gram matching for term conflation in Turkish texts. Information Research News, 7 (1), pp. 2-6, 1996.

[13] Khoja S.. "Stemming Arabic Text". Lancaster, U.K., Computing Department, Lancaster University. 1999.

[14] Khreisat, L. "Arabic text classification using N-Gram frequency statistics a comparative study". Proceedings of the 2006 International Conference on Data Mining (pp. 78–82). Las Vegas, NV: USCCM.

[15] Kraaij, W. and Pohlmann, R. Viewing stemming as recall enhancement. In Proceedings of ACM SIGIR96. pp. 40-48, 1996.

[16] Larkey L. and M. E. Connell. "Arabic information retrieval at UMass in TREC-10". Proceedings of TREC 2001, Gaithersburg: NIST. 2001.

[17] Larkey L., L. Ballesteros, and M. E. Connell. "Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis". Proceedings of SIGIR'02. PP 275–282. 2002.

[18] Liu T., S. Liu, Z. Chen and Wei-Ying Ma. "An Evaluation on Feature Selection for Text Clustering". Proceedings of the 12th International Conference (ICML 2003), Washington, DC, USA. PP 488-495. 2003.

[19] Monz, C. and de Rijke, M. Shallow morphological analysis in monolingual information retrieval for German and Italian. In Cross-language information retrieval and evaluation: Proceedings of the CLEF 2001 workshop, C. Peters, Ed.: Springer Verlag, 2001.

[20] Moulinier, I., McCulloh, A., and Lund, E. West group at CLEF 2000: Non-English monolingual retrieval. In Crosslanguage information retrieval and evaluation: Proceedings of the CLEF 2000 workshop, C. Peters, Ed.: Springer Verlag, pp. 176-187, 2001.

[21] Munirul Mansur, Naushad UzZaman and Mumit Khan, Analysis of N-Gram Based Text Categorization for Bangla in a Newspaper Corpus

[22] Mustafa S. H. and Q. A. Al-Radaideh. "Using N-Grams for Arabic text searching". Journal of the American Society for Information Science and Technology Volume 55, Issue 11. PP 1002–1007. 2004.

[23] Greengrass, M., Robertson, A. M., Robyn, S., and Willett, P. Processing morphological variants in searches of Latin text. Information research news, 6 (4), pp. 2-5, 1996.

[24] Popovic, M. and Willett, P. The effectiveness of stemming for natural-language access to Slovene textual data. JASIS, 43 (5), pp. 384-390, 1992.

[25] Rogati M. and Y. Yang. "High-Performing Feature Selection for Text classification". CIKM'02, ACM. 2002.

[26] Sebastiani F. "Machine learning in automated text categorization". ACM Computing Surveys, volume 34 number 1. PP 1-47. 2002.

[27] Sebastiani F. "A Tutorial on Automated Text Categorisation". Proceedings of ASAI-99, 1st Argentinian Symposium on Artificial Intelligence. PP 7-35. 1999.

[28] Syiam M., Z. T. Fayed & M. B. Habib, "an intelligent system for Arabic text categorization", IJICIS, Vol.6, No. 1, JANUARY 2006.

[29] Tai, S. Y., Ong, C. S., and Abdullah, N. A."On designing an automated Malaysian stemmer for the Malay language". In Proceedings of the fifth international workshop on information retrieval with Asian languages, Hong Kong, pp. 207-208, 2000.

[30] Vapnik V. (1995). "The Nature of Statistical Learning Theory", chapter 5. Springer-Verlag, New York.

[31] Yang Y., and J. O. Pedersen. "A comparative study on feature selection in text categorization". Proceedings of ICML-97. PP 412-420. 1997.

[32] Yang Y. and X. Liu, "A re-examination of text categorization methods," in 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), pp. 42–49, 1999.

AUTHORS

Miss. Meryeme Hadni Phd Student in Laboratory of computer and Modelization, Faculty of Sciences, University Sidi Mohamed Ben Abdellah (USMBA), Fez, Morocco. She has also presented different papers at different National and International conferences.

Pr. Abdelmonaime LACHKAR r eceived his PhD degree from the USMBA, Morocco in 2004 in computer science; He is working as a Professor and Head of Computer Science and Engineering (E.N.S.A), in University Sidi Mohamed Ben Abdellah (USMBA), Fez,Morocco. His current research interests include Arabic Text Mining Applications: Arabic Web Document Clustering and Categorization. Arabic Information and Retrieval Systems, Arabic Text Summarization, etc …, Image Indexing and Retrieval, 3D Shape Indexing and Retrieval in large 3D Objects Data bases, Color Image Segmentation, Unsupervised clustering, Cluster Validity Index, on-line and off-line Arabic and Latin handwritten recognition, and Medical Image Applications.

Pr. Said Alaoui Ouatik i s working as a Professor in Department of Computer Science, Faculty of Science Dhar EL Mahraz (FSDM), Fez, Morocco. His research interests include high-dimensional indexing and content-based retrieval, Arabic Document Categorization. 2D/3D Shapes Indexing and Retrieval in large 3D Objects Database.