Arabic Stemming Techniques as Feature Extraction applied in Arabic Text Classification

Samir Boukil, Fatiha El Adnani, Abd Elmajid El Moutaouakkil, Loubna Cherrat, Mostafa Ezziyyani*
Chouaib Doukkali University, Faculty of Sciences, Department of Computer
El Jadida, Morocco
*Abdelmalek Essaadi University, Faculty of Sciences and Technologies
Tangier, Morocco

boukilsamir@yahoo.fr, ftheladnani@gmail.com, elmsn@hotmail.com, cherrat-loubna2@gmail.com, *ezziyyani@gmail.com

Abstract. In this paper, we conduct a comparative study about the impact of stemming algorithms, as feature extraction systems, on the task of classification of Arabic text documents. Stemming is forceful and fierce as in reducing words to their three-letters roots. Which may influence the semantics, as various words with divers implications may share the same root. Light stemming, by examination, expels oftentimes utilized prefixes and suffixes in Arabic words. Light stemming doesn't extract the root and thus doesn't influence the semantics of words. However, the result of the light stemming is not necessarily a word. For the evaluation, we used corpus contains 5,070 records that fall into six classes. A several tests were done utilizing two separate illustrations of the same corpus. The K-Nearest Neighbors (KNN) classifier was utilized for the classification task. The recall measure is used to evaluate the performance of these methods.

Keywords: Text Mining, Automatic Language Processing, Classification, Feature Extraction, Arabic Language, Stemming, Light-Stemming, K-Nearest Neighbors.

1 Introduction

Arabic Language is one of the widest spread languages. It is the seventh most used language in the internet [1] and the fourth in the word. It is used by 6.6 % of the world's inhabitants [2] (more than 442 million Arabic speakers: 295 million as first language and 246 million as second language [3]).

The volume of information that is accessible on the Internet is expanding persistently. Thus, the complexity level of applications processing these immense amount of information is increasing. Organizing these information resources into classes are required to help the applications better do its tasks [4]. Text classification is a task of assigning one or more predefined classes to the analyzed document, based on its content. Several classification algorithms have been tested on Arabic text classification, for

instance, the Naive Bayes probabilistic classifiers [5], Decision Tree classifiers [6], Neural Networks [7]., K-Nearest Neighbor classifier (KNN) [8]. and Support Vector Machines [9]. The high dimensionality of the feature space is a notable issue for some machine-learning algorithms. Since the complexity of many learning algorithms increments with increased data dimension, algorithms that can enhance the classification efficiency, by decreasing the data into smaller dimensional space, are profoundly wanted and preferred. These algorithms make the learning task of classification and information retrieval systems faster, more proficient and spare more space [10].

While classifying text document, the features does not represent semantically the document in a similar way. Some of these features might be excess and add nothing to the significance of the document, others may be synonymous and consequently choosing one of them is sufficient to increase the semantic for classification purposes. Thus, the effective determination of feature words is a critical task in text classification, this operation is called: feature extraction. Feature extraction extracts a subset of features that contain solid and informative information about the original dataset, while expelling unessential or excess features [11].

This paper present a description and a comparison of two morphologic feature extraction techniques for Arabic text classification. To be specific it's about stemming and light-stemming techniques. The KNN classifier was tested on Arabic dataset. The dataset includes 5,070 Arabic documents. The documents are physically grouped into six classes: Sports, Business, Entertainment, Middle_East, Scitech and World. The efficiency of the previously mentioned methods was measured regarding vector sizes, time to run the classifier; and the recall of the classifier.

This paper is organized as follows: Introduction has been presented in Section 1. Section 2 describes a state of art about feature extraction techniques and classification of Arabic text document. Section 3 resumes the structure in which the feature extraction techniques were utilized. Section 4 presents the outcomes and Section 5 outlines the conclusions and future work.

2 Related Works

Feature extraction techniques uses numerous systems to locate and extract the optimal and ideal subset of features. Eliminating stop words from the records is one way; calculating document frequency, information gain and other statistical characteristics is another way [12]. Other methods are based on similarity and relations between the words, such as, stemming systems that extracts the word's root and light stemming that removes the suffixes and prefixes and keep the resulting word.

All those methods of extracting information in Arabic could be categorized into two categories as in Fig. 1, based on linguistic dependencies:

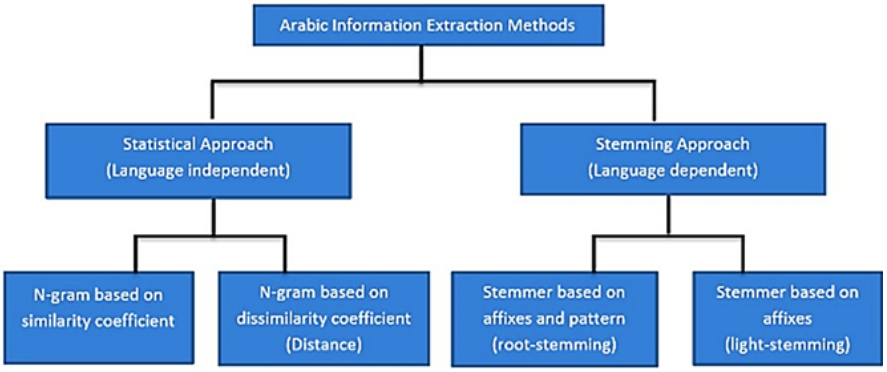

Fig. 1 Methods of extracting information from Arabic texts

- Language-independent approaches, called statistical approaches, can be categorized per similarity and dissimilarity coefficients.
- Language-dependent approaches, known as stemming/morphological approaches, are classified into two major approaches:
  - Root-based stemming approach.
  - Light stemming approach.

To understand the mechanism of stemming approaches, we must understand the structure of arabic word.

The structure of an arabic word is broken down into five components: the proclitic, the prefix, the stem, the suffix, and the enclitic.

In Table 1, the agglutinated word *لبيراقبوهم* is framed by the proclitic *ل* (for), the prefix *ي* and the suffix *و* (which indicate the third person of plural), the enclitic هم (them) and the stem *راقب* (watch). The stem (without prefix and suffix), presents the kernel vocabulary, possibly surrounded by extensions [13].

2.1 Light-Stemming

Light Stemming implies on pulling out the light-stem from the word. It is to expel prefixes, suffixes and extensions as illustrating in Table 2 and 3:

In Table 2, after dropping affixes and extensions from the verb *يراقبونهم* (they watch them), we get the stem *راقب* (watch), and in Table 3, the word *المواصلات* (the communicators) is decreased to *مواصل* (communicator).

Table 1. Word ليراقبوهم decortication

Proclitic	Prefix	Stem	Suffix	Enclitic
ل	ي	راقب	و	هم
For	(they)	Observ	(they)	Them

Table 2. Light stemming example 1

Word	Un-agglutinated	Stem
يراقبونهم	ي + راقب + ون + هم	راقب
They observe them	Them + (e) + observ + (they)	Observe

Table 3. Light stemming example 2

Word	Un-agglutinated	Stem
المواصلات	ال + مواصل + ات	مواصل
The communication	s + communicator + the	Communicator

2.2 Stemming

Stemming is the final procedure that comprises on reducing the word to its root as shown in Table 4

In Table 4, after light-stemming the word يراقبونهم (they observe them) to راقب (observe), the stemming gives us رقب (see) which is the root of the verb راقب.

2.3 Light-Stemming/Stemming Comparison

The procedure of stemming denies a word from its augmentations and form. By examining 'derived nouns' " الأسماء المشتقة " [14][15], we can say that the significance of a word is learned from its pattern "وزن". In this way, getting the word out to its root may rise error rate when the work is on the semantics. We give as ex-amples:

We give as examples:
مستحسن ⟶ حسن : (Well enough ⟶ Good)
حسن ⟶ حسن : (Good ⟶ Good)
أحسن ⟶ حسن : (Better ⟶ Good)
The light-stemming main goal is to drop only the affixes from the word as shown in the next example:
أحسن ⟶ أحسن : (Better ⟶ Better)
الأحسن ⟶ أحسن (The Best ⟶ Better) 2.4 Stemming Models

Table 4. Stemming example

Word	Stem	Root
يراقبونهم	راقب	رقب
They observe them	observe	See

2.4 Stemming Models

Main Stemming Models. Stemming algorithms can be organized in two categories: those that extract the stem by comparing the word's pattern with a predefined lexicon and those that extract the stem directly.

Among the first category of stemming algorithms [14], the most famous is the *Khoja* stemmer [15]. Its standard is to eliminate the longest prefix and the longest suffix, and after that comparing the result with some predefined base roots to extract the correct root.

By processing the word " *المواصلات* ", *Khoja* stemmer begins by removing the prefix"*ال*" and the suffix "*ات*", the stem that results "*مواصل*" has the pattern "*مفاعل*". For this pattern, it drops the first and the third letters to get the stem "*وصل*" which exists in *Khoja* dictionary.

Second category of stemming algorithms doesn't use any dictionary; they reduce the word depending on statistical approach. The most famous is the *AI-Shalabi* extraction algorithm that give the root of a word from its letters, each letter is attributed to, and characterized by, a weight, and its rank relies on its position in the word [17] [18]. Hence, the letters of affixes (*سألتمونيها*) are instantly recognized by their weight (see Table 5).

In Table 5, letters "*ر*", "*ق*" and "*ب*" have the smallest products in the word "*يراقبونهم*". They constitute the root "*رقب*".

At the point when the root doesn't contain any letter from (*سألتمونيها*), this model gives great results as appeared in the case above. However, when the root incorporates these letters, it can create troubles like the word *المواصلات* (the communications) which is decreased to "*لصل*" despite "*وصل*".

Models Comparison. *Sawalha and Atwell* looked at the principle stemming algorithms from both categories on four measures of precision [19]. They credited to *Khoja* stemming model the most elevated precision on the extraction of three letters root, then the morphological analyzer of *Tim Buckwalter*, and the Tri-literal root algorithm and the Voting algorithm calculation (realizing that 80 – 85 % of arabic words are derived from roots of three letters).

Table 5. Example of al shalaby root extractor

Word	يراقبونهم								
Letters	م	ه	ن	و	ب	ق	ا	ر	ي
Weight	2	1	2	3	0	0	5	0	3.5
Rank	8.5	6.5	5.5	4.5	3.5	4	7	8	9
Prod-uct	17	6.5	11	13.5	0	0	35	0	31.5
Root	رقب								

In this paper, we have utilized *Khoja* stemming algorithm and light-stemmer10 algorithm [20] techniques as feature extraction methods. These methods were compared in a text classification test.

Feature extraction has several advantages for text classification such as:

1. It lessens the running time of the classification operation. It wipes out the repetitive and needless features, so that the classifier uses a feature vector relevant and smaller than the original ones, this will diminish the running time of the classification operation.
2. It yields more precise outcomes. Feature extraction enhances the outcomes' precision by eliminating the unimportant and pointless features that doesn't help in the classification, and keeping the important ones that helps exploring the semantic of the text documents.
3. It limits the obliged memory to deal with the documents' vectors by decreasing and lowering the quantity of the features described by the vectors.

2.5 Arabic Text Classification

A large number of studies have been done to find a solution of the text classification problem. Most of its concerns the English and French texts [21], yet couple of ones have been applied to Arabic content.

Mesleh has used the Support Vector Machines (SVM) algorithm with the chi-square as a feature selection technique [22]. He assessed the execution of his classifier by a corpus gathered from online Arabic newspaper archives, including Aljazeera, Al Nahar, Al Hayat, Al Ahram and AlDostor and a couple of other websites. The gathered corpus contains 1445 records with various length. These records are classified into nine classes. *Mesleh* has reasoned that the SVM algorithm combined the chi-square technique surpass Naïve Bayesian and the KNN classifier in terms of F-measure.

Al-Harbi et al. have measure the performance of two famous classification algorithms: C5.0 decision tree algorithm and SVM algorithm, in classification of Arabic text. They used seven Arabic corpora: Saudi News Agency, Saudi News Paper, website,

journalists, Discussions, Islamic subject and Arabic Poems [23]. The results demonstrated that the C5.0 algorithm has outrun SVM classifier as far as precision by around 10%, the SVM the average precision for SVM is 68.65%, whereas the average precision for the C5.0 is 78.42%.

El-Kourdi et al used the Naïve Bayes algorithm for Arabic text classification [24]. They employed a corpus of 1,500 records gathered from Al Jazeera website categorized in 5 classes: Sport, Business, Culture and Art, Science and Health, every class contains 300 records. All words in the records were transformed to their roots. The outcomes demonstrated that the average precision was around 68.78% in cross validation and 62% in evaluation set experiments. The best precision by class was 92.8%.

Houssien et al. used three classification algorithms on Arabic text documents: Naïve Bayesian (NB), Sequential Minimal Optimization (SMO) and J48(C4.5) employing Weka [24]. The gathered corpus contained 2,363 records that fall into six classes: Sport, Economic, medication, politic, religion and science. The authors removed the stop words and decrease the number of features extracted from the records by using normalization approach. The precision, recall and error rate were deployed for comparing the accuracy of those classifiers. The outcomes demonstrate that SMO classifier accomplishes the most astounding accuracy and the least error rate, trailed by J48 (C4.5), and the NB classifier; while J48 classifier took the longest time, trailed by NB classifier then SMO classifier.

3 KNN with Multiple Feature Selection Techniques

In this work, text classification was tested using the KNN classifier on the CNN-Arabic corpus. Considering that the aim of our work is to study the impact of using the morphological feature extraction methods on Arabic text classification, we tested this classification operation in two cases: the first case is using the stemmer approach and the second one uses the light stemming approach.

Fig. 2 presents the principle modules in our system; the next sections depict each of these modules.

Preprocessing: its goal is to present a modified copy of the documents that the classifier can understand and manipulate with ease. Basic functions for preprocessors include: converting the document, removing stop words, removing foreign characters, removing punctuation, removing articles and numbers.

Root Stemming: The *Khoja* algorithm was followed here as a feature extraction method. The *Khoja* algorithm finds the three-letter roots for Arabic words by following this procedure:

1. Remove diacritics
2. Remove stop words, punctuation and numbers
3. Remove definite article (ال, وال, فال, كال, بال)
4. Remove conjunction (و)
5. Remove suffixes

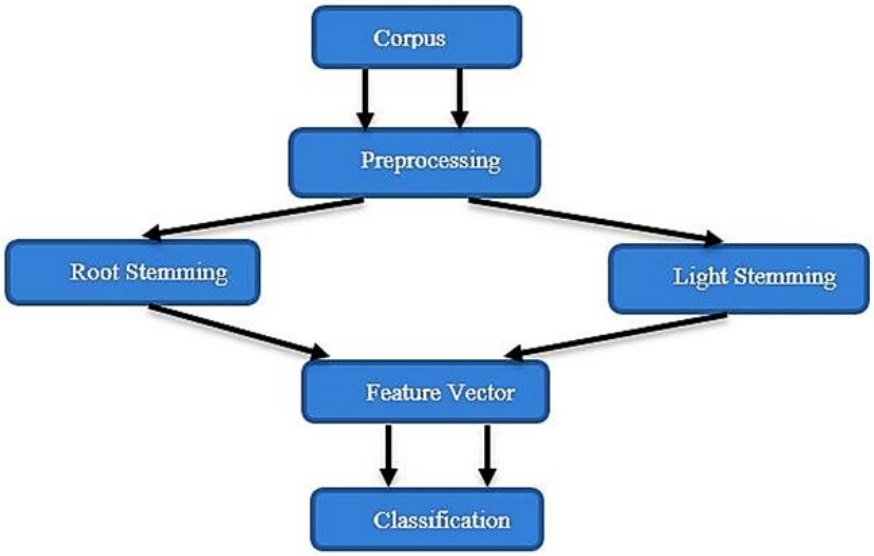

Fig. 2 System architecture

6. Remove prefixes
7. Compare result against a list of patterns. If a match is found extract the root.
8. Check the match with the predefined root based.
9. Replace: ا,و,ي by و
10. Replace: ئ,ؤ by أ
11. If the root contains only two letters, check if they should contain a double character

Fig. 3 describes an example of the *Khoja* algorithm. Note that several words such as *(اللاعب الملعب اللعبة)* which mean " The player ", " playground " and " the game" respectively are reduced to one stem (*لعب*).

Light Stemming: it's main goal is to improve the efficiency of classification while holding the words' semantics. It maintains the word's signification untouchable. In this stage, we applied the light-stemmer10 algorithm as a feature extraction technique. The principal of this algorithm lies on many runs that try to find and take off the most recurrent prefixes and suffixes from the word. Fig. 4 represents an example of preprocessing with light-stemming algorithm. Here we mention that light stemming keeps up the distinction between *(اللاعبون اللعبة)* which means "players" and "the game"; their light stems are *(اللاعب اللعبة)* which means the player and the game.

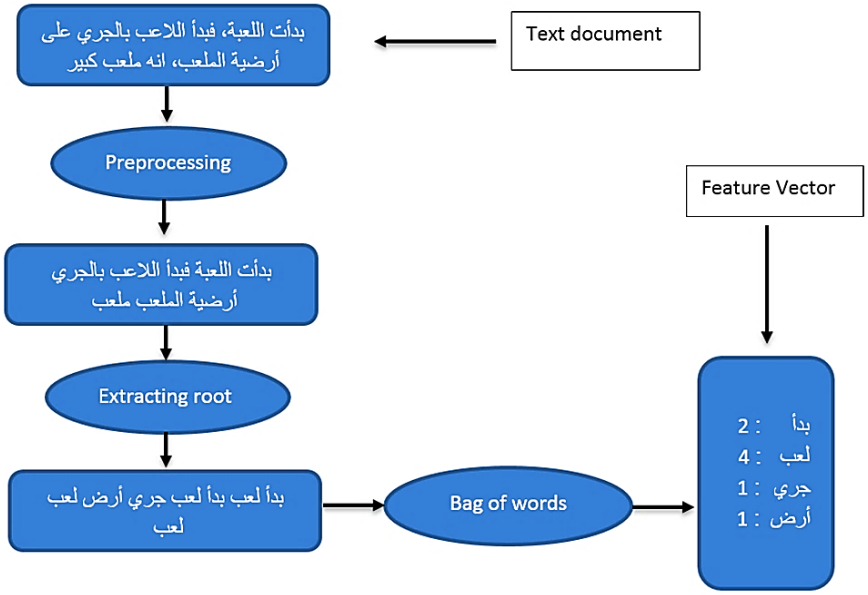

Fig. 3 An example of preprocessing with root-stemming algorithm (*Khoja* algorithm).

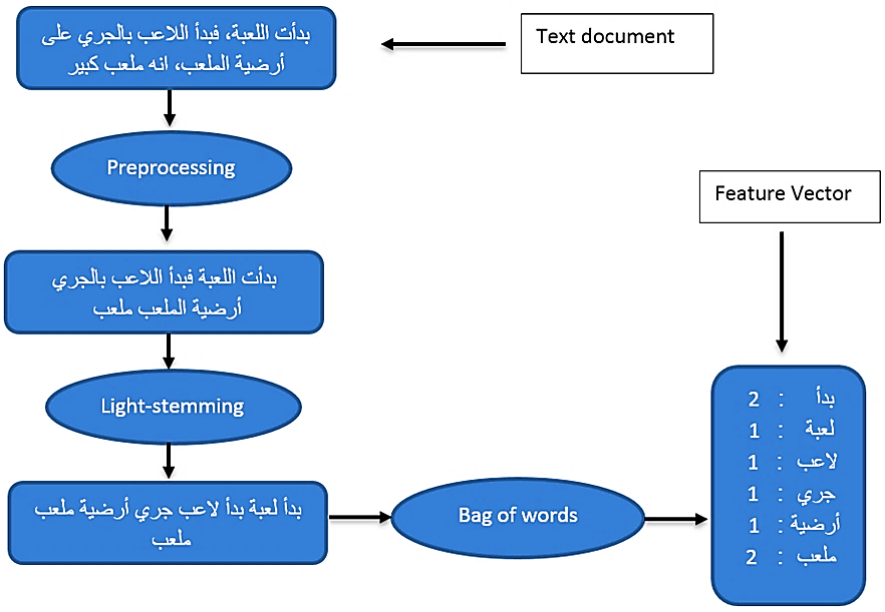

Fig. 4 An example of preprocessing with light-stemming algorithm.

Feature Vector: In this step, the bag of words (BOW) method was used:

1. We collect all the words contained in all the documents, then we eliminate the doubling, we call the result " global document ".
2. For each document, its stems (roots) compared with the global document and their frequencies are recorded. The vector containing the stems with their frequencies is the characteristic vector.

Classification: we applied the KNN classifier. It is one of the simplest classification algorithm. Even with such simplicity, it can give highly interesting results. Moreover, KNN presents the following advantages:

1. Easy to interpret output
2. Low Calculation time
3. Hight Predictive Power

4 Experimentation and Result Analysis

4.1 Dataset Description

The dataset is assembled from the CNN-Arabic site and contains 5,070 textual records belonging to six classes: Middle East News 1,462, World News 1,010, Economics 836, Sport 762, Entertainment 474 and SciTech 526. It contains 2,241,348 words. This dataset was split into two sections: training and testing. The training dataset represents 70% of each class, while the testing dataset represents the remaining 30%. Records in the training dataset were stored as vectors, where every vector comprises of its unique words and their frequencies. Those Vectors were set up in two adaptations: Stem-level vectors and Light stem-level vectors.

Table. 6 presents the characteristics of the two adaptations of the document vectors in addition to the original version of dataset. The significance of this Table is to demonstrate that the feature extraction techniques decrease the size of the dataset in addition to minimizing the required memory to deal with the dataset. As it's described in the Table 6, the stem-level form presented the most reduced space (11 MB). This is normal, as stemming decreases many words to one stem. On the other hand, the light stem-level vectors devoured (18 MB). This is kind of higher than the stem-level vectors.

4.2 Experiments

Light-Stem Level. Applying text classification after using the light-stemming technique as a feature extraction technique. First of all, the preprocessing step was applied for every document, and the light-stems of the words was extracted. From that point

Table 6. The properties of the dataset

Dataset version	Size in MB
Original version	25 MB
Stemmed version	11 MB
Light-stemmed version	18 MB

forward, it is represented as a vector of light-stems with their relating weights. At last, these vectors are passed to the KNN classifier.

Stem Level. Applying text classification after using the stemming technique as a feature extraction technique. For this situation, the preprocessing step was applied for every document, and the stems of the words was extracted. Lastly, stem-level

vector was created by storing the stemmed words with their relating weights and passed to the KNN classifier.

The experiments have been done on an Intel Core i7 CPU 2.20 GHz, with a RAM of size 8 GB. Table 7 presents the time passed by pre-handling and classification to all testing datasets for both stem and light-stem techniques.

Table 7 demonstrates that the most minimal preprocessing time was accomplished in relation to stemming. And the raison of that is the smaller size of the vectors compared with light-stemming. Additionally, the used stemming algorithm needs to scan every given word just once to derive its stem.

Classification time mentions the time needed to classify the testing dataset by the KNN classifier. This time, stemming overrule light stemming. Note that the classification time is specifically corresponding to vector sizes. To whole up, stemming needs less preprocessing and classification times by comparison with light stemming.

4.3 Result Analysis

Fig. 5 shows the recall of the Arabic text classification using the two stemming approaches on the testing dataset.

The results were obtained from the testing dataset. Those results demonstrate that the lightstemmer10 algorithm surpass the *Khoja* algorithm in almost all categories.

Besides comparing the two algorithms, shifting the number of the chosen features helped us analyzing the behavior of the two algorithms. The observational outcomes demonstrated that the recall measure decreases when the number of features is high or low, which can be translated by the way that some chosen features are unimportant for the primary case or they aren't sufficiently representative for the second case.

Table 7. The elapsed classification time (seconds)

Experiment	Preprocessing Time	Classification Time	Total Time
Stem Level	418	3,258	3,676
Light-Stem Level	470	3,689	4,159

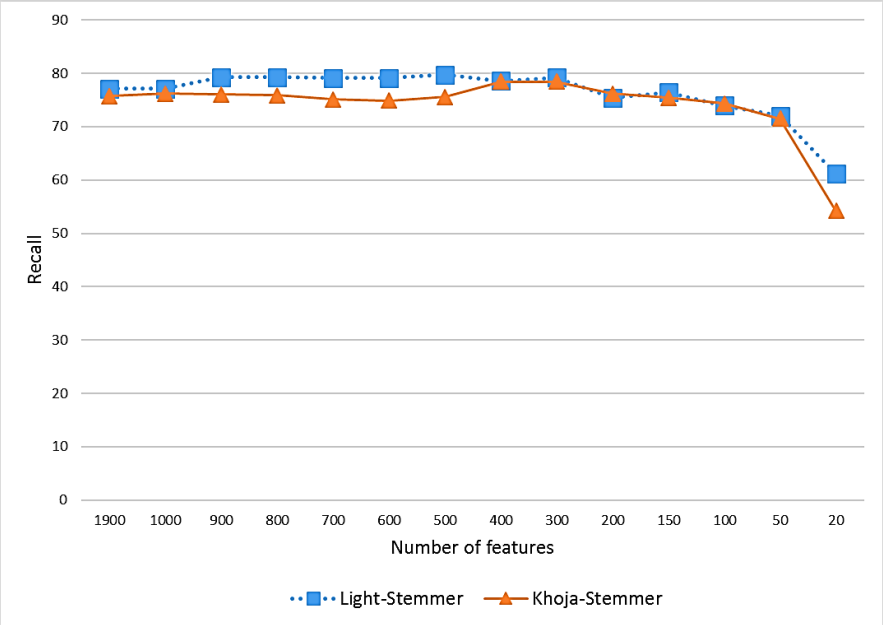

Fig. 5 Recall measure of classified testing dataset with different number of features

5 Conclusions and Future Work

In this work, we have tested feature extraction techniques for Arabic text classification. The dataset was grouped physically into six classes to be specific Middle East News, World News, Economics, Sport, Entertainment, Science and Technology and Arts And culture. The dataset was split into two sections: training dataset and testing dataset. The testing dataset present 30% of every class, while the training dataset represent the remaining 70%. The chosen approaches for feature extraction are the stemming [15]. and light stemming [16]. The Stemming approach finds the three-letter roots for Arabic words, while light stemming drops the prefixes and suffixes from the Arabic words and maintains the resulting word. The K-NN was utilized for the classification task. The

practices were as follows, the KNN classifier was run twice on the dataset: once with stemming (called stem-level), and once with light-stemming (called light stem-level).

The experiments showed that the best values of recall were achieved when light-stemming is used as a feature extraction method. Light stemming outperformed the stemming approach, and the main reason of that is that the stemming affects the words meanings.

However, the both algorithms suffer from several problems, such as; the results of light-stemmer10 algorithm does not always represent significant words, and that's because the algorithm is not based on any dictionary or linguistic rules. And the *Khoja* algorithm have difficulties when it concerns roots of four or five letters and irregular plurals. For those raisons, our future work will concentrate on improving those algorithms.

References

1. Al-Arabizi: Why do Google fight it more than Arabic ?, http://www.bbc.com/arabic/mobile/scienceandtech/2012/12/121220_arabic_language_internet_arab_days.shtml. Accessed 20 March 2017
2. Sorting the languages of the world in terms of proliferation. https://arabic.rt.com/news/786982-ترتيب-لغات-العالم-الانتشار/. Accessed 20 March 2017
3. List of languages by total number of speakers. https://ar.wikipedia.org/wiki/قائمة_اللغات_حسب_العدد_الكلي_للمتحدثين . Accessed 20th March, 2017
4. Correa, R.F., Ludermir, T.B.: Automatic Text Categorization: Case Study. In: the 7th Brazilian Symposium on Neural Networks, Pernambuco, Brazil, Nov 2002
5. Eyheramendy, S., Lewis, D., Madiagn, D.: On the naive Bayes model for text categorization. In: the Artificial Intelligence and Statistics Conference, Key West, Florida, Jan 2003
6. Peter, B.: Active Learning of SVM and Decision Tree Classifiers for Text Categorization. In: the 4th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence, Herlany, Slovakia, Jan 2006
7. Wang, P et al.: Semantic Clustering and Convolutional Neural Network for Short Text Categorization. In: 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp 352--357, Beijing, China, 26-31 July 2015
8. Gongde, G et al.: An kNN Model-Based Approach and Its Application in Text Categorization. In: the International Conference on Computational Linguistics and Intelligent Text Processing (CICLing), Seoul, Korea, Feb 2004
9. Basu, A., Walters, C., Shepherd, M.: Support Vector Machines for Text Categorization. In: 36th Annual Hawaii International Conference, Los Alamitos, California, USA, Jan 2003
10. Jun, Y et al.: OCFS: Optimal Orthogonal Centroid Feature Selection for Text Categorization. In: 28th Annual Intl. ACM SIGIR Conference, Salvador, Brazil, Aug 2005.
11. Seo, Y., Ankolekar, A., Sycara, K.: Feature Selection for Extracting Semantically Rich Words. Technical Report CMU-RI-TR-04-18, Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, March, 2004
12. Yang, Y., Pedersen, J.: A Comparative Study on Feature Selection in Text Categorization. In: 40th International Conference on Machine Learning, Nashville, Tennessee, USA, July 1997
13. Tuerlinckx, L.: La lemmatisation de l'arabe non classique. In: 7th international days of textual data statistical analysis (JADT), Louvainla- Neuve, Belgium, 2004.

14. Anjali, G.J. et al.: A Comparative Study of Stemming Algorithms. International Journal of Computer Technology and Applications (IJCTA), Vol. 2, No. 6, pp. 1930-1938. (2011)
15. Khoja, S., Garside, R.: Stemming arabic text. Computer Science Department, Lancaster University, Lancaster, UK, 1999
16. Al-Shalabi, R., Kanaan, G., Al-Serhan, H.: New Approach for Extracting Arabic Roots. In: the International Arab Conference on Information Technology, pp. 42-59, Alexandria, Egypt, 2003.
17. Evens, M.: A computational morphology system for Arabic. In: the Workshop on Computational Approaches to Semitic Languages, Montreal, Quebec, Canada, Aug 1998
18. Sawalha, M., Atwell, E.: Comparative Evaluation of Arabic Language Morphological Analysers and Stemmers. COLING : companion volume – Posters and Demonstrations pp. 107-110, Manchester, UK, 2008
19. Connell, M.: Improving Stemming for Arabic Information Retrieval: Light Stemming and Cooccurrence Analysis. In: 25th Annual International Conference on Research and Development in Information Retrieval, pp. 275-282, Tampere, Finland, Aug 2002
20. Joachims, T.: A statistical learning learning model of text classification for support vector machines. In: 24th annual international ACM SIGIR conference on Research and development in information retrieval, New Orleans, Louisiana, USA, 2001
21. Moh'd, A., Mesleh, A.: Chi Square Feature Extraction Based Svms Arabic Language Text Categorization System. Journal of Computer Science, vol. 3, p. 6 (2007)
22. Al-Harbi, S., Almuhareb, A., Al-Thubaity, A., Khorsheed, M.S., Al-Rajeh, A.: Automatic Arabic Text Classification. In: 9th International Conference on the Statistical Analysis of Textual Data, Lyon, France, 12 – 14 March 2008
23. Kourdi, M.E., Bensaid, A., Rachidi, T.: Automatic Arabic document categorization based on the Naive Bayes algorithm. In: the Workshop on Computational Approaches to Arabic Script-based Languages, Geneva, Switzerland, 2004
24. Majed, F.O., Hussien, I., AL-dwan, M., Shamsan, A.: Arabic Text Classification Using Smo,Naïve Bayesian, J48 Algorithms. International Journal of Research and Reviews in Applied Sciences, vol. 9, p. 10, November (2011)