Journal of Engineering Science and Technology
Vol. 12, No. 6 (2017) 1475 - 1487
© School of Engineering, Taylor's University

ARABIC TEXT CLASSIFICATION USING NEW STEMMER FOR FEATURE SELECTION AND DECISION TREES

SAID BAHASSINE[1], ABDELLAH MADANI[2], MOHAMED KISSI[3,]*

[1]LIMA Laboratory, Department of Computer Science, Chouaib Doukkali University,
Faculty of Science, B.P. 20, 24000, El Jadida, Morocco
[2]LAROSERI Laboratory, Department of Computer Science, Chouaib Doukkali University,
Faculty of Science, B.P. 20, 24000, El Jadida, Morocco
[3]LIM Laboratory, Department of Computer Science, HASSAN II University Casablanca,
Faculty of Sciences and Technologies, B.P. 146, 20650, Mohammedia, Morocco
*Corresponding author: kissim@gmail.com

Abstract

Text classification is the process of assignment of unclassified text to appropriate classes based on their content. The most prevalent representation for text classification is the bag of words vector. In this representation, the words that appear in documents often have multiple morphological structures, grammatical forms. In most cases, this morphological variant of words belongs to the same category. In the first part of this paper, anew stemming algorithm was developed in which each term of a given document is represented by its root. In the second part, a comparative study is conducted of the impact of two stemming algorithms namely Khoja's stemmer and our new stemmer (referred to hereafter by origin-stemmer) on Arabic text classification. This investigation was carried out using chi-square as a feature of selection to reduce the dimensionality of the feature space and decision tree classifier. In order to evaluate the performance of the classifier, this study used a corpus that consists of 5070 documents independently classified into six categories: sport, entertainment, business, Middle East, switch and world on WEKA toolkit. The recall, f-measure and precision measures are used to compare the performance of the obtained models. The experimental results show that text classification using rout stemmer outperforms classification using Khoja's stemmer. The f-measure was 92.9% in sport category and 89.1% in business category.

Keywords: Arabic text classification, Stemming, Decision tree, Chi-square, Feature selection.

Nomenclatures

F	F-measure
Fn	False negative
P	precision
R	Recall
tn	True negative
fp	False positive
tp	True positive

Abbreviations

BOT	Bag-Of-Token
KNN	K-Nearest Neighbors
NB	Naïve Bayesian
OSAC	Open Source Arabic Corpora
SMO	Sequential Minimal Optimization
SVM	Support Vector Machines
TC	Text Categorisation
TFIDF	Term Frequency-Inverse Document Frequency
WEKA	Waikato Environment for Knowledge Analysis

1. Introduction

Nowadays, text categorization has gained an important interest in data mining applications. Selecting an appropriate category, from a predefined set, for an unclassified text is the goal of TC process. Considering the huge number of online Arabic documents, there is an increasing need for Arabic text classification. Such categorization can enhance the accuracy of decision making. The major applications of text classification include digital library systems, classification of email messages [1], document management systems, spam filtering, web pages classification, and sentiment analysis for marketing, as well as detecting spam in Arabic opinion reviews [2] .

Arabic is the 5th widely used language in the world. It contains 28 letters called Huruf Alhijaa, 25 consonants and three long vowels, which are written from right to left and shaped according to their position in the word. It is also a challenging language because it has a rich morphology, a complex syntax, and difficult semantics. This distinguishes it from other languages and makes its learning, analysis, and automatic processing a very hard and complex task.

An Arabic text classification system can be divided into three steps:

- Pre-processing step: where punctuation marks, stop words, diacritics and no letters are removed.
- Features selection: a set of features is selected from the text, which will represent the text in the next step.
- Learning step: many algorithms were used to learn systems how to classify Arabic text documents.

In this paper, the performance of two stemming methods, Khoja's stemmer and origin-stemmer, used as feature selection in the design of decision tree based text classifier are compared. In addition, this research also investigates the accuracy of these models while varying the number of selected features.

The remainder of the paper is organized as follows: section 2 is a state of the art in Arabic text classification. Section 3 discusses previous work in Arabic stem root extraction and present root-stemmer algorithm. Experimental results are given in section 4, and then some conclusions are drawn and provided suggestions for future research.

2.Related Works

Many algorithms have been implemented to solve the problem of text categorization. Most of the work in this area was performed for English, French, and Turkish texts [3, 4], but few ones have been introduced for the Arabic text.

Odeh et al. [5] have implemented a text classification system for Arabic. The implemented system calculates the weight of each word, and then chooses the two words which have the largest weight in the document. At the end; it finds categories that match their words. They evaluated the performance of this algorithm using eleven categories corpus containing 982 documents. The results showed that the accuracy was 98% in one category and 93% in another.

Al-Shalabi and Obeidat [6] implemented a text classification system for Arabic. This system compares the representation of document by N-grams (unigrams and bigrams) and single terms (bag of words) as a feature extraction method in the pre-processing step. Afterwards, he used Term Frequency and Inverse Document Frequency (TFIDF) to reduce dimensionality and K-Nearest Neighbors (KNN) as classifier for Arabic text classification. The experimental results showed that using unigrams and bigrams as representation of documents outperformed the use of bag of words in terms of accuracy.

Mesleh [7] implemented the support vector machines (SVM) classifier. He used chi-square as a feature selection method in the pre-processing step. He evaluated the performance of this classifier by an in-house corpus collected from online Arabic newspaper archives, including Aljazeera, al Nahar, al Hayat, al Ahram and alDostor as well as a few other specialized websites. The collected corpus contains 1445 documents that vary in length. These documents belong to nine categories. The author concluded that the SVM algorithm with the chi-square method outperformed NB and the KNN classifier in terms of F-measure.

Al-Harbi et al. [8]compared two classification algorithms on Arabic text namely C5.0 decision tree algorithm and SVM algorithm. They used seven Arabic corpora: Saudi News Agency, Saudi newspapers, website, writers, Discussions, Islamic topics and Arabic Poems. The authors implemented a tool for Arabic text classification to accomplish selection and feature extraction. The results indicate that the C5.0 algorithm outperformed SVM algorithm in terms of accuracy by about 10%.

Naïve Bayes algorithm has been used by El-Kourdi et al. [9] for Arabic text classification. They used a corpus of 1500 text documents collected from al Jazeera website belonging to 5 categories: sport, business, culture and art, science and health. Each category contained 300 text documents. They used stemmer algorithm to convert all words to their roots. The results showed that the average accuracy was about 68.78% in cross validation and 62% in evaluation set experiments. The best accuracy by category was 92.8%.

Abu-Errub [10] classified Arabic text documents using TFIDF measurement for categorization. First, the document was compared with pre-defined document categories based on its content, then the document was classified into the appropriate sub-category using chi-square measure.Finally, Abu-Errub evaluated the performance of his algorithm using 1090 testing documents categorized into ten main categories and 50 sub-categories. The results indicated that the best accuracy by category was 98.93%.

Majed et al. [11] tested the performance of three popular classification algorithms: Sequential Minimal Optimization (SMO), Naïve Bayesian (NB) and J48 (C4.5) on Arabic text using Weka. They used corpus of 2363 documents that vary in length and that belonged to six categories: Sport, Economic, medicine, politics, religion and science. In the pre-processing step, the authors reduced the number of features extracted from the documents by elimination of stop words and normalization approach. They used the recall, precision and error rate to compare the classifiers. The results showed that SMO classifier achieves the highest accuracy and the lowest error rate, followed by J48 (C4.5) and the NB classifier. WhereasJ48 classifier took a highest amount of time to get the results, followed by NB classifier then SMO classifier.

Most of the previous works viewed text as Bag-of-Token (BOT). In the Arabic language, words have multiple morphological structures. For example, "لعب"(to play), "ملعب" (stadium), "لعبة" (game), and "لاعب" (player). In most cases, these variants have similar semantic features and belong to the same category “Sport”, they use 4 attributes ("لعب", " ملعب", "لعبة" and "لاعب") instead of 1 ("لعب"). To overcome this shortcoming the stemming is used before classification, which is intended to reduce the number of attributes and increase the accuracy of classification [12].

Another problem of text classification is the high number of features in documents. This study uses chi-square which is one of the most common measurements used in feature selections. Most previous studies used their own corpus which hinders the comparison of the stemming algorithm and reduces the accuracy of the results. In our work, the corpus of open source Arabic corpora (OSAC) is used.

3. Stemming Algorithm

Stemming is the process of reducing inflected words into their root form. It removes prefixes, suffixes and infixes. There are several types of stemming algorithms [2]: statistical, dictionary, morphological and light stemming.

3.1. Previous works

One of the morphological stemming algorithms is tri-literal root extraction algorithm also referred to as Al-Shalabi algorithm [13]. This algorithm doesn’t use any dictionary. It uses letter weights for a word‘s letters multiplied by the letter‘s position in the word. Consonants were assigned a weight of zero and different weights were assigned to the letters grouped in the word (سألتمونيها). All affixes letters are distinguished by their weight.

Khoja’s algorithm is one of the most used morphological stemming algorithms [14]. It removes the largest suffix and the largest prefix from the word, and then the algorithm compares the rest of the word with its 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, definite articles, and punctuation characters; lists of patterns and stop words.

Sawalha conducted a comparative study of three stemming algorithms: Khoja’s stemmer, Buckwalter’s morphological analyser and Al-Shalabi algorithm [15]. The results obtained showed that Khoja’s stemmer performs better in terms of accuracy. Therefore, we will further study this stemmer and compare it with origin-stemmer.

3.2. Proposed stemmer approach

Although, Khoja’s algorithm had the highest accuracy, it still suffers from several weaknesses [16]. For instance, the Khoja’s stemmer removes definite articles, conjunctions, prefixes or suffixes which can occasionally be part of the word’s root. The word « والدان » (parents), for example, is stemmed to « دون » instead of ولد (son).

Origin-stemmer tries to overcome this shortcoming. For this reason, the proposed algorithm differs from Khoja’s in the following terms:

- It verifies whether the affixes are part of the word before removing them.
- It uses more affixes as shown in Table 1.

It uses an enriched stop words file that can increase the accuracy in text categorization [17].

Table 1. Affixes used in our algorithm.

Affixes in Arabic	Examples
Length 1 prefixes	ل, ب, ف, س, و, ي, ت,ن, ا
Length 2 prefixes	ون,او,ول,فل,ال,سي,ست,ين,يت,ال,لي, با, وا,وب, وي, وت, وس, لل
Length 3 prefixes	كال, بال,للا,الت,الا,سيت,يست,تست, است, ولل, فسي,وال, للت
Length 4 prefixes	وفسي, وللت ,وكال, وبال,وللا,المت,والا,وسيت,وتست, واست
Length 1 suffixes	ة, ه, ي, ك,ت, ا, ن
Length 2 suffixes	ون, ات, ان, ين, تن,تي, ية, يه, كم, هن, نا, يا, ها,كه, وه, تم, كن,ته, ني, وا, ما, هم
Length 3 suffixes	تها, همل,تنا, تان, تين, هما, يها, كمل,وهم, وها, ونا, وني, يات, كما, يين, اته, اتك, اتي
Length 4 suffixes	تهما,اتنا, اتكن, اتهم ,اتها, اتهن, اتيه

Root Algorithm:

```
Pattern(i) = set of the Pattern the length i
P(i)= set of the prefixes the length i
S(i)= set of the suffixes the length i
V = set of the root of the verbs
M = set of the root of  the words
R = set of the root ( results)
F=input File
F  ← Cleanup(F)    // Remove diacritics, stop words,
                    //Punctuation and numbers
L← ToList(F)       // convert text to list using space as split
for Mot in L do
        If  Mot  ϵ  L  do
                n ← length(Mot)
                t ← n
                if Mot ϵ V or Mot ϵ M
                        return Mot
                while ( t-n<=2 and n >=2)
                        i← 0
                while(i<=t-n )
                if (start(Mot,i) ϵ P(i) and end(Mot,t-n-i) ϵ S(i))
                        if (extract(Mot,i,t-n-i) ϵ V or extract(Mot,i,t-n-i) ϵ M
                        return extract(Mot,i,t-n-i)
                        else if (n>3 and n<=6) then
                                H← searchpattern(extract(Mot,i,t-n-i) )
                                for root1 in H do
                        if root1 ϵ V  or root1 ϵ M then
                                        return root1
                                end if
                        end if
               end if
        end if
i←i+ 1
n← n-1
function  searchpattern( Mot )
n=length(Mot)
s=[ ] // set Null
for K in Pattern(n) do
        if Compare (K,Mot) = 1 then
                add(root(Mot)) in S
        end if
return S
```

4. Experimentation and Results Analysis

4.1. Dataset description

The dataset used is one of the corpus of open source Arabic corpora (OSAC) [18]. It is collected from cnnarabic.com. The dataset consists of 5070 Arabic text documents of different lengths that belong to six categories based on their content: sports, entertainment, business, the Middle East, switch and world. The distribution of the classes in the used portion of the dataset is represented in Table 2.

The dataset was used in two ways. In the first one, data is divided into partitions where the first partition contains 3382 texts (66.66% from the dataset), and it is used for the training phase. The second partition contains 1688 texts (33.33% of the dataset) and it is used for the testing phase.

Table 2. Datasets description.

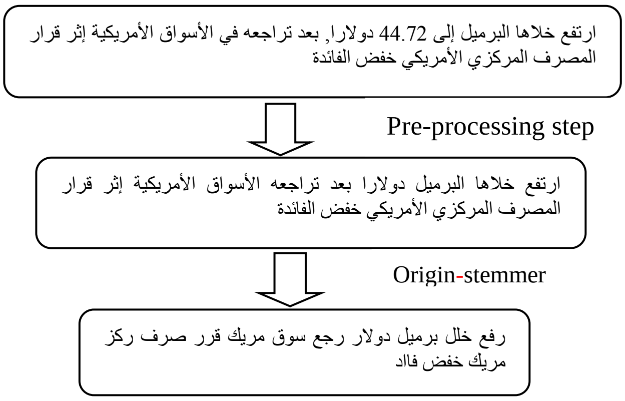
Categories	Number of text	Number of training set	Number of test set
Business	836	558	278
Entertainment	474	316	158
Middle east	1462	975	487
Switch	526	351	175
Sport	762	508	254
World	1010	674	336
All	5070	3382	1688

4.2. Preprocessing step

WEKA (Waikato Environment for Knowledge Analysis) is a well-known suite of machine learning software written in Java and developed at the University of Waikato. It is a free software available under the GNU general public license [19].

Text pre-processing is an important step carried out before the documents categorization. The purpose is to get important information from massive data and reduce operation processing time. This way, the data from non meaningful words like stop words will be cleaned up. First, the data is converted into attribute-relation file format (ARFF) using Weka TextDirectorytoArff converter. Then, Weka StringtoWordVector tool is used with different options: term frequency-inverse document frequency (TFIDF) [10], and normalization to equalize the length of the document vectors.

Figure 1 shows an example of a text document after the pre-processing and the stemming steps. Table 3 describes the characteristics of the documents vectors in addition to the original version (Null Stemmer) obtained after the pre-processing step.



ارتفع خلاها البرميل إلى 44.72 دولارا, بعد تراجعه في الأسواق الأمريكية إثر قرار
المصرف المركزي الأمريكي خفض الفائدة

ارتفع خلاها البرميل دولارا بعد تراجعه الأسواق الأمريكية إثر قرار
المصرف المركزي الأمريكي خفض الفائدة

رفع خلل برميل دولار رجع سوق مريك قرر صرف ركز
مريك خفض فااد

Fig. 1. Example of the pre-processing and stemming step.

As it can be seen from Table 3, even after stemming, the number of features is still very high. To address this issue, several methods were developed to reduce the input dimensions by choosing a subset of features that might lead to better classification. In the suggested approach, chi-square statistics (X^2) [8, 10, 20] which is one of the most used metrics for feature selection is adopted.

Table 3. Number of features after pre-processing step.

Stemmer	Number of distinct features
Khoja's stemmer	22372
Our Stemmer	16995
Null Stemmer	95409

The chi-square statistics formula is related to information-theoretic feature selection functions which try to capture the intuition that the best terms for the class *c* are the ones distributed most differently in the sets of positive and negative examples of *c* [7, 21].

The calculation procedure of X^2 for a term *t* and a particular class *c* is displayed in the contingency table (see Table 4) [8].

Table 4. The contingency table of *t* and *c*.

	c	Not *c*	Total
t	*A*	*B*	*A*+*B*
Not *t*	*C*	*D*	*C*+*D*
Total	*A*+*C*	*B*+*D*	*N*

$$X^2(t,c) = \frac{N(AD - CB)^2}{(\mathrm{A}+\mathrm{C})(B+D)(A+B)(C+D)} \quad (1)$$

This equation illustrates chi-square statistics using Table 4, where:

N=Total number of documents in the corpus.
A = Number of documents in class *c* that contain the term *t*.
B = Number of documents that contain the term *t* in other classes.
C = Number of documents in class *c* that does not contain the term *t*.
D = Number of documents that does not contain the term *t* in other classes.

4.3. Results

In order to compare the performance of the previously mentioned Khoja's stemmer and origin-stemmer according to their effect on Arabic document classification, the number of selected features from 50 to 1900 was varied. The results obtained using origin-stemmer algorithm and Khoja's algorithm were compared in terms of recall, precision and F-measure.

The Recall measure is the ratio of the relevant data among the retrieved data. It is defined as follows:

$$r = \frac{tf}{tf + np} \quad (2)$$

The Precision is ratio of the accurate data among the retrieved data. Its formula is given as follow:

$$p = \frac{tp}{tp + tf} \tag{3}$$

The F-measure of the system is defined as the weighted harmonic mean of its precision and recall. It is defined as follows:

$$F = \frac{2rp}{r + p} \tag{4}$$

where *r* is recall which is given by Eq. (2).

The first set of experimentations, Table 5, was done on training data. The results obtained show that origin-stemmer algorithm outperforms Khoja's algorithm for all numbers of features except when the number of features is 300 and 1900.

Table 5. Correctly classified instances training.

Numbers of attributs	1900	900	700	500	300	150	50
Origin	80.11	81.32	82.06	82.44	82.56	83.27	78.15
Khoja	81.01	80.24	80.22	79.87	83.09	81.41	78.12

The second set of experimentations was done on test data, the results obtained show that origin-stemmer algorithm outperforms Khoja's algorithm, Fig. 2, the best average of recall obtained for our model is 79.7% when the number of features is 500, while the best recall obtained by Khoja's algorithm is 78.4%. Figures 3 and 4 show that the best average of precision and f-measure obtained for our model is 79.5% when the number of features is 500 while the best precision, f-measure obtained by Khoja's algorithm was 78.2% and 77.9% respectively when the number of feature is 300.

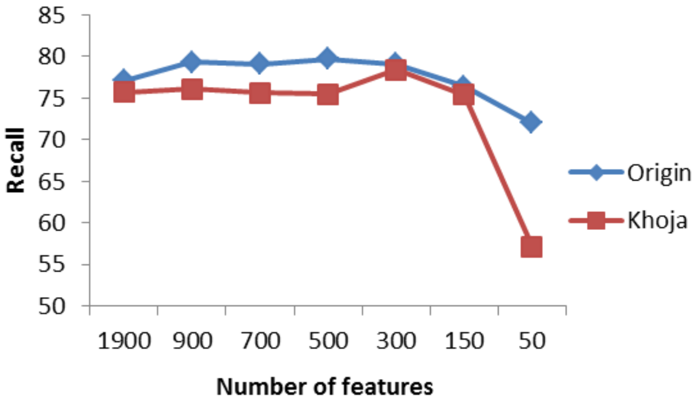


Fig. 2. Recall measure of classification using origin-stemmer algorithm and Khoja's algorithm.

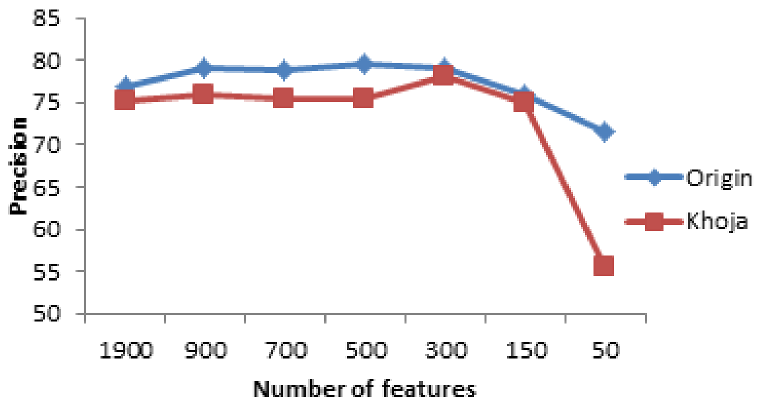


Fig. 3. Precision measure of classification using origin-stemmer algorithm and Khoja algorithm.

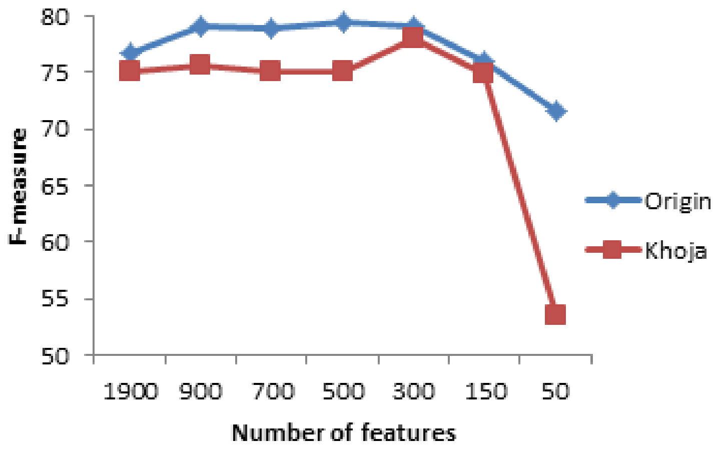


Fig. 4. F-measure of classification using origin-stemmer algorithm and Khoja algorithm.

In the comparison process, varying the number of the selected features helped us analyze the performance of origin-stemmer algorithm. The empirical results showed that the recall measure, precision measure and f-measure decrease when the number of features is high or low, which can be interpreted by the fact that some selected features are irrelevant for the first case or they aren’t representative enough for the second case.

Table 6 shows the recall measure for the sport, entertainment, business, Middle-East, switch, and world categories. Recall was calculated twice: once for origin-stemmer and once for Khoja’s stemmer case. For the business and entertainment texts, Khoja’s stemmer is slightly better than origin-stemmer.

Tables 7 and 8 show the precision and f-measure for the six categories. The results show that origin-stemmer outperformed Khoja’s stemmer for all categories and for most of the number of features.

Table 6. Recall results per class for the classification algorithms using origin-stemmer and Khoja's stemmer.

Class	1900		900		700		500		300		150		50	
	origin	Khoja	origin	Khoja	origin	Khoja	origin	Khoja	origin	Khoja	origin	Khoja	origin	Khoja
Business	**94.2**	94.2	**94.2**	94.2	**93.9**	94.2	**93.9**	94.2	**94.2**	94.2	**92.8**	95.00	**91.4**	93.2
Entertainment	**52.5**	53.2	**57.6**	60.8	**58.9**	60.1	**60.1**	65.2	**55.7**	50.6	**40.5**	44.3	**38.6**	7.00
Middle_East	**78.6**	73.1	**77.2**	73.1	**73.1**	72.5	**77.6**	73.5	**73.9**	77.6	**82.8**	73.3	**73.7**	63.9
Switch	**66.3**	50.3	**65.7**	50.3	**65.1**	46.3	**67.4**	48.00	**68.0**	57.7	**64.6**	60.6	**62.9**	5.7
Sport	**92.5**	94.5	**94.1**	91.7	**94.1**	94.5	**92.5**	94.5	**93.3**	90.9	**89.8**	90.9	**90.6**	94.1
World	**66.4**	73.8	**75.9**	74.1	**81.0**	72.9	**77.1**	67.9	**80.4**	81.00	**66.4**	72.9	**59.8**	40.2
Avg	**77.1**	75.7	**79.3**	76.1	**79.1**	75.6	**79.7**	75.5	**79.1**	78.4	**76.4**	75.4	**72.0**	57.2

Table 7. Precision results per class for the classification algorithms using origin-stemmer and Khoja's stemmer.

Class	1900		900		700		500		300		150		50	
	origin	Khoja	origin	Khoja	origin	Khoja	origin	Khoja	origin	Khoja	origin	Khoja	origin	Khoja
Business	**81.9**	81.4	**81.9**	81.4	**83.1**	80.1	**83.1**	81.4	**84.5**	81.1	**85.7**	78.3	**81.7**	84.4
Entertainment	**68.0**	56.0	**63.2**	57.8	**60.8**	54.3	**66.0**	59.2	**61.1**	68.4	**52.9**	56.5	**43.0**	25.6
Middle_East	**70.7**	73.3	**77.0**	74.5	**78.8**	75.1	**77.9**	71.6	**82.2**	82.2	**73.5**	77.1	**70.9**	41.8
Switch	**74.4**	74.6	**76.7**	72.1	**77.6**	75.7	**74.2**	71.2	**67.6**	70.6	**66.1**	74.6	**57.6**	45.5
Sport	**91.8**	84.2	**90.5**	86.3	**90.5**	84.2	**91.8**	84.2	**92.6**	87.5	**92.3**	88.2	**90.9**	91.2
World	**76.4**	75.8	**79.2**	75.5	**76.0**	75.6	**78.5**	78.9	**74.2**	71.4	**74.3**	68.1	**70.5**	43.5
Avg	**77.0**	75.3	**79.0**	75.8	**78.9**	75.5	**79.5**	75.4	**79.1**	78.2	**75.8**	75.00	**71.6**	55.5

Table 8. F-measure results per class for the classification algorithms using origin-stemmer and Khoja's stemmer.

Class	1900		900		700		500		300		150		50	
	origin	Khoja	origin	Khoja	origin	Khoja	origin	Khoja	origin	Khoja	origin	Khoja	origin	Khoja
Business	**87.6**	87.3	**87.6**	87.3	**88.2**	86.6	**88.2**	87.3	**89.1**	87.2	**89.1**	85.9	**86.2**	88.5
Entertainment	**59.3**	54.5	**60.3**	59.3	**59.8**	57.1	**62.9**	62.0	**58.3**	58.2	**45.9**	49.6	**40.7**	10.9
Middle_East	**74.4**	73.2	**77.1**	73.8	**75.8**	73.8	**77.8**	72.5	**77.8**	79.8	**77.9**	75.2	**72.3**	50.5
Switch	**70.1**	60.1	**70.8**	59.3	**70.8**	57.4	**70.7**	57.3	**67.8**	63.5	**65.3**	66.9	**60.1**	10.2
Sport	**92.2**	89.1	**92.3**	88.9	**92.3**	89.1	**92.2**	89.1	**92.9**	89.2	**91.0**	89.5	**90.7**	92.6
World	**71.0**	74.8	**77.5**	74.8	**78.4**	74.2	**77.8**	73.0	**77.1**	75.9	**70.1**	70.4	**64.7**	41.8
Avg	**76.7**	75.1	**79.0**	75.6	**78.8**	75.0	**79.5**	75.0	**79.0**	77.9	**75.9**	74.9	**71.6**	53.5

Another notable result that was also reported is that all values vary among categories and for all numbers of features. The sport category achieved the highest precision, recall, and f-measure values compared with other categories because the attributes in this class are distinctive compared to other classes. The entertainment category has a noticeably poor precision, recall, and f-measure. These poor measures indicate that the entertainment category is highly overlapped with other categories.

The results also show that the average of recall, precision and f-measure of classified test instance was not more than 79.74%. This is due to the nature of the

Arabic text categories (sport, entertainment, business, Middle East, switch and world) in our corpus because there are some words that are used in both world category and Middle East category as there are similarities between these categories.

5. Conclusion

In this paper, the effect of origin-stemmer and Khoja's stemmer on Arabic document classification was compared. Chi-square statistics was used to reduce the number of features selected. The results show that text classification using origin-stemmer outperforms classification using Khoja's stemmer. It is worth noting that the feature selection step needs further empirical analysis which is beyond the scope of this paper. The researchers of the present study will try to overcome this shortcoming by developing a new algorithm. Therefore, work is currently being done on the feature selection step and on how to classify documents belonging to similar categories.

References

1. Carvalho, V.R.; and Cohen, W.W. (2005). On the collective classification of email "speech acts", *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM: Salvador, Brazil, 345-352.
2. Abu Hammad, A.; and El-Halees, A. (2015).An approach for detecting spam in Arabic opinion reviews. *The International Arab Journal of Information Technology*,12(1), 9-16.
3. Yildirim, S. (2014). A knowledge-poor approach to Turkish text categorization, *Computational Linguistics and Intelligent Text Processing*, vol. 8404, A. Gelbukh, ed: Springer Berlin Heidelberg, 428-440.
4. Joachims, T. (2001). A statistical learning learning model of text classification for support vector machines, *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval*, New Orleans, Louisiana, USA, 128-136.
5. Odeh, A.; Abu-Errub, A. ; Shambour,Q.; and Turab, N. (2014). Arabic text categorization algorithm using vector evaluation method. *International Journal of Computer Science & Information Technology*, 6(6), 83-92.
6. Al-Shalabi, R.; and Obeidat, R. (2008). Improving KNN Arabic text classification with n-grams based document indexing, *Proceedings of the Sixth International Conference on Informatics and Systems, Cairo, Egypt*, 108-112.
7. Mesleh, A.M. (2007). Chi square feature extraction based svms Arabic language text categorization system. *Journal of Computer Science*, 3(6), 430-435.
8. Al-Harbi, S.; Almuhareb, A.; Al-Thubaity, A.; Khorsheed, M. S.; and Al-Rajeh, A. (2008). Automatic Arabic text classification, *The 9th International Conference on the Statistical Analysis of Textual Data*, 77-83.
9. El Kourdi, M.; Bensaid, A.; and Rachidi, T. (2004). Automatic Arabic document categorization based on the naive bayes algorithm, *The Workshop on Computational Approaches to Arabic Script-based Languages, Association for Computational Linguistics,Geneva, Switzerland*, 51-58.

10. Abu-errub, A. (2014). Arabic text classification algorithm using TFIDF and chi square measurements. *International Journal of Computer Applications*, 93(6), 40-45.
11. Majed, I.H.; Fekry, O.; AL-dwan, M.;and Shamsan, A. (2011). Arabic text classification using SMO,naïve bayesian, J48 algorithms. *International Journal of Research and Reviews in Applied Sciences*, 9(2), 306-316.
12. Saad, M.; and Ashour, W. (2010). Arabic text classification using decision trees, *Proceedings of the 12th international workshop on computer scienceand information technologies CSIT*, 75-79.
13. Al-Shalabi, R.; and Evens, M. (1998). A computational morphology system for Arabic, *Proceedings of the Workshop on Computational Approaches to Semitic Languages, Association for Computational Linguistics: Montreal, Quebec, Canada*, 66-72.
14. Khoja, S.; and Garside, R. (1999). *Stemming Arabic Text.* Ph.D. Thesis. Lancaster University, Computing Department, Lancaster, UK.
15. Sawalha, M.; and Atwell, E.S. (2008). Comparative evaluation of Arabic language morphological analysers and stemmers, Coling 2008 Organizing Committee, Manchester, 107 - 110.
16. Cherif, W.; Madani, A.; and Kissi, M.. (*2014*). Building a syntactic rules-based stemmer to improve search effectiveness for Arabic language, *Intelligent Systems: Theories and Applications, 9th International Conference*, 1-6.
17. Al-Shargabi, B.; Al-Romimah, W.; and Olayah, F. (2011). A comparative study for Arabic text classification algorithms based on stop words elimination, *Proceedings of the International Conference on Intelligent Semantic Web-Services and Applications, ACM: Amman, Jordan*, 1-5.
18. Saad, M.; and Ashour, W. (2010). OSAC: Open Source Arabic Corpora, *EEECS'10 the 6th International Symposium on Electrical and Electronics Engineering and Computer Science*, *European University of Lefke*, Cyprus, 118-123.
19. Mark, H.; Eibe, F.; Geoffrey, H.; Bernhard, P.; Peter, R.; and Witten, H. (2009). The WEKA data mining software: an update. *SIGKDD Exploration Newsletter*, 11(1), 10-18.
20. Yang, Y.; and Pedersen, J.O. (1997). A comparative study on feature selection in text categorization, *Proceedings of the Fourteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc*, 412-420.
21. Sebastiani, F. (2002). Machine learning in automated text categorization. A*CM Computing Surveys*, 34(1), 1-47.