Impact of stemming on Arabic text summarization

Nabil Alami[1], Mohammed Meknassi[2], Said Alaoui Ouatik[3] and NourEddine Ennahnahi[4]
Laboratoire Informatique et Modélisation
Faculty of Science Dhar EL Mahraz
University Sidi Mohamed Ben Abdellah (USMBA), Fez, Morocco
[1]nab.alami@gmail.com, [2]m.meknassi@gmail.com, [3]s_ouatik@yahoo.com, [4]nahnnourd@yahoo.fr

***Abstract*—Stemming is a process of reducing inflected words to their stem or root from a generally written word form. This process is used in many text mining application as a feature selection technique. Moreover, Arabic text summarization has increasingly become an important task in natural language processing area (NLP). Therefore, the aim of this paper is to evaluate the impact of three different Arabic stemmers (i.e. Khoja, Larekey and Alkhalil's stemmer) on the text summarization performance for Arabic language. The evaluation of the proposed system, with the three different stemmers and without stemming, on the dataset used shows that the best performance was achieved by Khoja stemmer in term of recall, precision and F1-measure. The evaluation also shows that the performances of the proposed system are significantly improved by applying the stemming process in the pre-processing stage.**

Keywords—Arabic Text Summarization; Graph model; Stemming; Light Stemming

I. INTRODUCTION

With the rapid growth of the Internet, and the multiplicity of media mass storage, the amount of electronic documents and textual data became huge and unmanageable. The human, unable to manually handle large text volumes, must finds automatic analysis methods adapted to automatic processing of personal data. This situation has significantly increased the need for highly effective systems that reduce the length of those documents and find information quickly. Automatic text summarization systems offer a solution to this problem. With summaries, we can make effective decisions and get useful information quickly in less time without looking at the whole document by extracting brief information from a given text. Recently a great effort was spent on NLP of Arabic language and many research works have been developed. But, this work and efforts still not sufficient compared to what has been done in English language especially in the field of automatic text summarization. For Arabic language, automatic text summarization is a new research focus and the literature in this field is fairly limited and still in their beginning. However, it is regrettable that Arabic language has not received significant attention in this field.

We mention that Arabic is the official language of 22 countries. Arabic is the fifth most spoken language in the world and the fourth used in the internet after English, Chinese and Spanish. Arabic is also the language of the Holy Quran, the holy book of the Islamic world and is read and spoken by thousands of millions of Muslims across the globe [1].

In Arabic language, words are classified into three main categories: nouns (أسماء), verbs (أفعال), and particles (أدوات). These words are generally derived from a root, which mean that the root is the original form of the word before any transformation process [1], and with adding a suffix to the root we can build a set of derivations. However, a stem is a morpheme or a set of concatenated morphemes that can accept an affix [2]. Finding a root or stem of an Arabic word helps in mapping grammatical variations of a word to the instances of the same term. Table I shows a set of words derived from the same root "علم".

TABLE I. DIFFERENT DERIVATION OF ROOT "علم"

Arabic word	English sense	Stem	Root
التعليم	The education	تعليم	علم
المعلم	The teacher	معلم	علم
المعلمون	The teachers	معلم	علم
معلمان	Two teachers	معلم	علم
يتعلمون	They learn	تعلم	علم

Automatic processing of Arabic language has been considered as a challenging task for automatic text summarization and information retrieval due to different reasons. First, Arabic is highly inflectional and derivational, and words can have many different forms which makes the task of morphology very complex. Second, written character in different ways depends on the position of letter in the word, which can add a complexity to Arabic words analysis. Third, Arabic words are often ambiguous due to the tri-literal root system. Based on such specifications in Arabic language, it is a hard matter to determine the root/stem of any Arabic word since it requires a detailed morphological, syntactic and semantic analysis of the text.

We should point out that in automatic natural language processing area and especially automatic text summarization field, a preprocessing step is indispensable to transform the unstructured data in textual documents into structured format in order to apply data mining techniques. This transformation aims to make a representation of an Arabic document and depending on the quality of this representation, the accuracy of any text mining tasks may be impacted positively or negatively. There are several methods used in text mining for preprocessing text documents, such as tokenization, stop-word removal, stemming, and term weighting [28].

978-1-5090-6751-6/16/$31.00 ©2016 IEEE

Most of the researches in NLP and text mining domains are focused on texts in English. Despite the lack of research works for Arabic, there are a number of notable studies that examine the stemming impact and effectiveness on many fields in Arabic NLP such as information retrieval, document clustering, text categorization and similarity measure. Basically, they have examined three different levels including word-based, stem-based, and root-based. However, to the best of our knowledge there is no study that systematically examines the impact and effectiveness of stemming process on automatic summarization of Arabic texts. Motivated by this, we empirically analyze the effect of using stemming process in the preprocessing step in Arabic text summarization.

This research presents result of an experimental study of comparison of the three stemmers for Arabic language in automatic text summarization using graph-based approach and redundancy elimination as described in section 3. Hence, the idea of comparing the three different Arabic stemmers is to shed light on its effect in increasing the summarization system effectiveness for Arabic documents.

The remainder of this paper is organized as follows. Section 2 reviews related work on stemming and it impact on some text mining tasks. Section 3, which is divided into three subsections, describes the method proposed for Arabic document summarization. The results of experiments on the dataset of Arabic are discussed in Section 4. We conclude the paper in Section 5 with pointers to future works.

II. Literature Review

One of the most challenging issues in Arabic language is the word stemming. A wide body of research has been carried out in this particular of stemming domain. The two most effective Arabic stemmers are Khoja [3] based on root-extraction stemmer and Larkey's light stemmer [4, 5].

Based to the desired level of analysis, Arabic stemmers algorithms are classified as either root-based [7] or stem-based [8], [9], [10]. Root-Based approach uses morphological analysis to extract the root of a given Arabic word. Many algorithms have been developed for this approach. A superior root-based Arabic stemmer is Khoja's stemmer [3]. The Khoja algorithm removes suffixes, infixes, and prefixes and uses pattern matching to extract the roots. However, the algorithm suffers from problems especially with names and nouns. While the stem-based approach or light stemmer approach aims only to remove the most frequent suffixes and prefixes of a given Arabic word. Light stemmer is mentioned by some authors [4, 11, 12, 13]. Larkey's stemmer or light10, developed by [4, 5], is the most widely used Arabic light stemmer. Light stemming does not deal with patterns or infixes; it is simply a process of stripping off prefixes and/or suffixes. Although light stemmers produce fewer errors than aggressive root-based stemmers, aggressive stemmers reduce the size of the corpus significantly. Both Arabic root-based and stem-based algorithms suffer from stemming errors. The main cause of this problem is the stemmer's lack of knowledge of the word's lexical category (e.g., noun, verb, and preposition) [6].

Despite stemming errors, it has been empirically demonstrated that stemming improves retrieval in many languages, including Arabic [11, 12]. Based on the investigation made by [6], it is shown that the light10 stemmer increase the information retrieval effectiveness for Arabic documents. The evaluation of the three different stemmers shown that light10 stemmer achieved the best result in term of mean average precision. In [14], the authors evaluated the impact of five similarity/distance measures on document clustering using two stemming algorithms, morphology- and syntax-based Arabic lemmatization algorithm; and morphology-based Information Science Research Institute (ISRI) stemming and compare the results to raw data clustering 'without stemming'. Based on the experimental results, it can be concluded that similarity/distance measures are more effective in the lemmatization stemming of morphological and syntactically structured words than ISRI and raw data that is expected where ISRI has over-stemming, and Raw Data "without stemming" has under-stemming. The result obtained by [15] shown that the Light Stemming outperformed the stemming approach because Stemming affects the words meanings. In [16], the authors evaluated the impact of the stemming on the Arabic Text Document Clustering with five similarity/distance measures. The experiments show that the use of the stemming will not yield good results, but makes the representation of the document smaller and the clustering faster. The work made by [17] evaluated stemming techniques in clustering of Arabic language documents and determined the most efficient in preprocessing of Arabic language. The evaluation used three stemming techniques: root-based Stemming, light Stemming and without stemming. From experiments, results show that light stemming achieved best results in terms of recall, precision and F-measure when compared with other stemming techniques. The experiments depicted that Light Stemming is the best technique for feature selection in Arabic language document clustering, but root based stemming get deteriorated results for Arabic language document clustering; because Arabic language has a complex morphology, and it is a highly inflected language. The study carried out by [18], compares and analyzes the effectiveness of Latent semantic analysis model with a wide variety of distance functions and similarity measures to measure the similarity between Arabic words in two cases: with and without stemming, for two testing data. The obtained results show that the use of the Stemming gives more accuracy in some cases and the opposite in the others. A study of the effect of stemming on Arabic text categorization was performed in [19]. The literature shows that stemming is not the optimal choice for feature reduction. Moreover, the authors find that the stemming process is good in some cases and poor in others, as well as a result of using both the light stemming and the root-based stemming.

In this research, we evaluate the impact of three different Arabic stemmers on document summarization compared to the result obtained without stemming.

III. Arabic Text summarization

In this section we discuss our method used for summarizing Arabic documents. Recently, graph-based algorithms have been applied successfully to different Natural Language Processing tasks. Using the graph for displaying the structure of the text will help us to better understand the connection between different parts. Graph-based algorithms use a ranking algorithm to rank different sections of a text where each section is considered as a node. The use of graph-based ranking algorithms has been also shown to be effective in Text Summarization. A graph is constructed by adding a vertex for each sentence in the text, and edges between vertices are established using sentence inter-connections. These connections are defined using a similarity relation, where similarity is measured as a function of content overlap. In this work we present the results obtained by applying graph theory in Arabic Text summarization. We enhance our system by using Maximal Marginal Relevance (MMR) method to eliminate redundancy. And best scoring sentences that are less similar are selected to form the final summary.

Our proposed summarization method has several benefits. Firstly, because this method is an unsupervised learning approach, it requires no training data. Secondly, this method is domain-independent, so we do not need to consider domain-specific knowledge. Finally, this method does not require more linguistic resources which are still limited in Arabic language. In addition, redundancy is a problem in automatic summarization due to the fact that sentences with similar meaning can be included in the summary because they have a high score. Therefore, instead of typically selecting top ranked sentences, we use the MMR algorithm to form the final extractive summary.

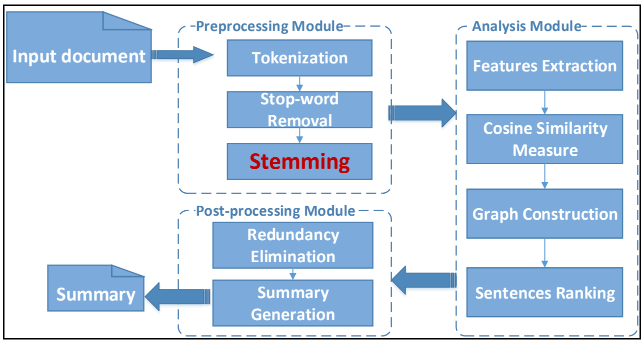

Fig. 1. Process of the proposed Arabic summarization system

Firstly, the set of documents is pre-processed. The undirected weighted graph is constructed for each document with sentences as nodes and similarities as edges. Thereafter, the weighted ranking algorithm PageRank is performed on the graph to generate salient score for each sentence in the document. The sentences are ranked according to their salient scores. The top-ranking sentences are selected to form the summary for the input document and MMR also is used to filter out redundant information. Figure 1 shows the overview of our summarization model. The input to the system is a large Arabic document and the output is a small Arabic document which contains the main ideas of the original text.

A. Preprocessing

Pre-processing is the first phase in each summarization system. It involves cleaning the source documents, normalization and tokenization, stop-words removal, and stemming. This phase is very important for a good representation of the input document.

1) Normalization and Tokenization

Normalization is the process of replacing the different variants of characters by a single one in order to make one uniform character representing those characters. In the pre-processing of Arabic text, the process of normalization is advantageous to be conducted before carrying out the stemming task. First we normalize all Arabic words by removing punctuation, non-letters and diacritics. Tokenization is the process of splitting text into tokens. In this research after normalization and before stemming, the full word is considered a token and the white space its boundaries. The text document is also split into sentences and each sentence is segmented into words.

2) Stop-words removal

Stop-words are words whose main function is structural, they appear very frequently in a text and carry a little meaning, it serves only a syntactic function but doesn't give any hint value to the content of their documents. In Arabic language words like ((هو, هذا, الذي, هي)) are used frequently in sentences which have little significance in the implication of a document. These words can simply be removed for summarization process. There is no standard stop-words list to use for Arabic language. In this research we use a list included in Khoja [3].

3) Stemming

In order to test the effect of the stemming on our Arabic summarization system, we selected three famous stemming algorithms for which we had ready access to the implementation: The Morphological Analyzer from Khoja [3], the Light Stemmer developed by Larkey [4] and Alkhalil morphological system developed by [20].

Khoja's root-extraction stemmer: A superior root-based Arabic stemmer is Khoja's stemmer presented by [3]. The Khoja algorithm removes suffixes, infixes, and prefixes and matches the remaining word with verbal and noun patterns, to extract the root. The stemmer uses several linguistic data files such as a list of all diacritic characters, punctuation characters, definite articles, and 168 stop words [12].

Larkey's light stemmer: is a stem-based approach or Light Stemmer approach presented by [4]. It produces a stem instead of a root of a given Arabic word. The aim of the Stem-Based approach is to eliminate the most frequent prefixes and suffixes. Light stemming does not deal with patterns or infixes; it is simply a process of stripping off prefixes and/or suffixes. Although light stemmers produce fewer errors than aggressive root-based stemmers, root-based stemmers reduce the size of the corpus significantly.

Alkhalil Morpho Sys: is a morphological syntactic parser of Standard Arabic words presented by [20]. The system can process non vocalized texts as well as partially or totally vocalized ones. For a given word, it identifies all possible solutions with their morph syntactic features: vowelizations, proclitics and enclitics, nature of the word, voweled patterns, stems, roots and Syntactic form. The approach used is based on modelling a very large set of Arabic morphological rules, and also on integrating linguistic resources that are useful to the analysis, such as the root database, vocalized patterns associated with roots, and proclitic and enclitic tables.

However, the number of lexical items and stems makes the lexicon voluminous and as a result the process of analyzing an Arabic text becomes long. Alkhalil analyzer give for each word several possible stems and roots. We use the Viterbi algorithm to select the most appropriate root of the given word.

B. *Analyze*

1) Feature Extraction (TF/ISF)

Term frequency: Term frequency of a term or “TF” is the number of times the term appears in the document. Inverse document frequency of a term or “*IDF*“ represents the importance of a word or a term in a document corpus. In our situation we use Inverse Sentence Frequency which is the same as IDF by replacing a set of documents by a set of sentences:

$$ISF_{wi} = \log_2 \frac{N}{df_{w_i}} \qquad (1)$$

Where N is the total number of sentences in the document and df_{wi} is the number of sentences with the term wi appear.

2) Cosine similarity measure

Cosine similarity is one of the most popular similarity measure applied to text documents, such as in numerous information retrieval [21] applications and clustering too [22]. Given two Sentences S_x and S_y, their cosine similarity based on *TF.IDF* feature is:

$$sim(S_x, S_y) = \frac{\sum_{w \in (S_x, S_y)} tf_{w,S_x} tf_{w,S_y} (idf_w)^2}{\sqrt{\sum_{w \in S_x} \left(tf_{w,S_x} idf_w\right)^2 \cdot \sum_{w \in S_y} \left(tf_{w,S_y} idf_w\right)^2}} \quad (2)$$

Where $tf_{w,\,Sx}$ represent the term frequency of the word w in the sentence S_x and idf_w represent the inverse sentence frequency in the whole document.

3) Graph construction

In this step we transform Arabic text document into graph format. The undirected weighted graph $G=(V, E)$ with the set of vertices V and set of edges E representing the document is constructed. The sentences become the nodes of the graph. Sentences that are similar to each other have an edge between them. The edges of the graph represent similarity between the sentences and the weight of the edge represents the degree of this similarity. In this work we use the cosine similarity measure. The result of this step is a highly connected graph. This undirected weighted graph is the input of the process in the next section to calculate salient score for each sentence. An example of this graph is illustrated in Fig.2. In this figure, edges with higher weight are illustrated with stronger line [23].

4) Sentence ranker

Once document graph is built, the sentences in a document will be ranked through random walk on G. We compute a salience score for each node using PageRank algorithm [24]. PageRank is one of the most popular link analysis algorithms and was designed as a method for Web link analysis. It determines the importance of a node within a graph, based on information drawn from the graph structure.

Equation (3) gives the score of a vertex V_i, where $In(V_i)$ is the set of nodes that point to V_i, $Out(V_j)$ is the set of nodes to which node V_j points, in our situation, since the graph is undirected $In(V_i)$ is equal to $Out(V_i)$, w_{ij} as the weight of the edge directing from node V_i to node V_j, and d is a damping factor that can be set between 0 and 1, which has the role of integrating into the model the probability of jumping from a given vertex to another random vertex in the graph. The factor d is usually set to 0.85[25].

$$PR(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} w_{ij} \frac{PR(V_j)}{\sum_{V_k \in Out(V_j)} w_{jk}} \quad (3)$$

To calculate *PR*, an initial score of 1 is assigned to all nodes, and (3) is applied on a weighted graph G iteratively until the difference in scores between iterations falls below a threshold of 0.001 for all nodes. Sentences corresponding to nodes with higher scores are important, salient to the document, and have strong relationship with others sentences.

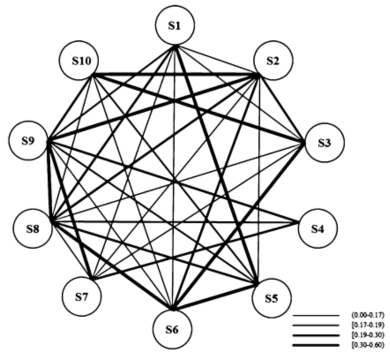

Fig. 2. Sample graph build for text representation

The final score associated to each sentence is represented by the flowing formula:

$$score(S_i) = PR(S_i) + \frac{\sum_{wj \in S_i} TF.ISF(w_j)}{termsCount(S_i)} \quad (4)$$

Where $PR(S_i)$ is the rank of the sentence S_i obtained by (3), *TF.ISF* is the term frequency inverse sentence frequency of the term and $termsCount(S_i)$ represents the terms count in the sentence S_i. The term can be root, stem or word.

C. *Summary generation*

After the ranking process is done, each sentence Si has its salient score $PR(S_i)$. In a graph-based automatic summarization such as TextRank [25] and LexRank [26], the summary is simply constructed by selecting only top ranked-sentences. In this situation, there may be much redundancy among the top ranking sentences, since similar sentences tend to get similar ranking scores during the ranking process and these similar sentences tend to be selected together and appear in the summary. This will cause the redundancy in the summary because too much similar sentences represent the same idea. This is why the modified version of MMR [27] is applied to re-rank and select sentences to add into summary. A sentence is added if it is high ranked and not too similar to any sentence existing in the summary. First, the sentence with the highest rank is removed from the ranked list and added to the summary. Then, the next sentence, which has the highest reranked score from (3), is chosen from the ranked list. This sentence is removed from the ranked list and added to the summary. This process is iterated until the summary reaches the predefined length.

$$MMR = argmax_{s_i \epsilon R \setminus S}\left[\lambda * PR(s_i) - (1-\lambda) * max_{s_j \epsilon S} * sim(s_i, s_j)\right] \quad (5)$$

In this equation, R is the set of all sentences, S is the set of summary sentences, $PR(s)$ is the ranking score for sentences computed in previous section and $sim(s_i, s_j)$ is the similarity measure between sentences s_i and s_j; λ is a tuning factor between a sentence's importance and its relevance to previously selected sentences. We choose the value $\lambda = 0.7$ for the best performance in the experiments.

IV. Experiments and Results

A. Dataset

Usually, the performance of a summarization technique is calculated by comparing the results with manually (intrinsic) extracted summary; but to our knowledge, there exists no standard dataset for Arabic used in the tests of Arabic text summarization. This is why, to test the accuracy of our summarization system, we have built a corpus of Arabic articles. These articles have been collected from different Arabic websites that cover various subjects. The total number of documents is 42. The summaries were manually produced by an expert in Arabic language.

B. Evaluation metrics

The generated summary is considered relevant or not relevant based on the comparison between the manual generated summary. Three important measures are commonly used, precision, recall and F-measure.

Precision (P): The measure of how much of the information that the system returned is correct.

$$P = \frac{|S_{manual} \cap S_{auto}|}{|S_{auto}|} \quad (6)$$

Where S_{manual} represents the set of sentences in the manual summary and S_{auto} represents the set of sentences in the auto-generated summary.

Recall (R): The measure of the coverage of the system. It reflects the ratio of relevant sentences that the system extracted.

$$R = \frac{|S_{manual} \cap S_{auto}|}{|S_{manual}|} \quad (7)$$

F-measure (F): Makes a balance between recall and precision using a parameter β. The (F-Measure/summary size) ratio is important when comparing systems. We obtain the *F1* score by setting the value of β to one:

$$F = \frac{2*P*R}{P+R} \quad (6)$$

C. Results and discussion

The main goal of this evaluation is to study the effectiveness and impact of the stemming process with three stemmer algorithms on the Arabic text summarization system. This is why we ran our system with four configurations: using Khaja's stemmer, Larkey's stemmer, Alkhalil's stemmer and without the use of stemming process.

Table II shows the average of recall, precision and F1-measure obtained with each configuration. The size of the summary is 30%. The above results show that the result obtained with Khoja's stemmer (with 0.51 of F1-measeur) outperformed those obtained using both Larkey's stemmer and Alkhalil's stemmer in term of precision, recall and F-measure. On the other hand, Larkey's stemmer and Alkhalil's stemmer get the same performances to our system with 0.48 of F1-measeur. This result was confirmed when the size of the summary is configured to 20% (table III)

TABLE II. PERFOMANCE EVALUATION WITH THE THREE STEMMERS AND 30% SUMMARY SIZE

	Precision	**Recall**	**F1-measure**	**Size**
Khoja's stemmer	**0.55**	**0.48**	**0.51**	**30%**
Larkey's stemmer	0.52	0.45	0.48	30%
Alkhalil's stemmer	0.52	0.44	0.48	30%

TABLE III. PERFOMANCE EVALUATION WITH THE THREE STEMMERS AND 20% SUMMARY SIZE

	Precision	**Recall**	**F1-measure**	**Size**
Khoja's stemmer	**0.57**	**0.35**	**0.44**	**20%**
Larkey's stemmer	0.56	0.34	0.42	20%
Alkhalil's stemmer	0.51	0.31	0.39	20%

The forth run aims to evaluate the effect of the stemming process on Arabic text summarization effectiveness and how sensitive is Arabic summarization to the use of stemmer. Table IV shows that the use of stemming performed well the Arabic text summarization,

We conclude that whether with the use of Khoja's stemmer (root-based), Larkey's stemmer (stem-based) or Alkhalil's stemmer, the system performances are improved. The worst result of our proposed system is obtained when our system was run without stemming process.

TABLE IV. PERFOMANCE EVALUATION WITH AND WITHOUT STEMMING

Size	**Stemmer**	**Precision**	**Recall**	**F1-measure**
30%	Without stemming	0.49	0.42	0.45
	Khoja's stemmer	**0.55**	**0.48**	**0.51**
20%	Without stemming	0.52	0.32	0.39
	Khoja's stemmer	**0.57**	**0.35**	**0.44**

As we have mentioned previously, Arabic Text summarization is not studied enough in the literature. Moreover, we can state that it is very difficult to provide an improved evaluation of the proposed Arabic summarization system, due to the lack of gold standard corpora, for Arabic language, there is not any approved benchmark to evaluate our approach in Arabic text summarization. By contrast, in English there are DUC human generated summaries that can be used as a benchmark.

V. Conclusion & Future Works

In this research, we investigated the effect of three stemmers (Light 10, Khoja and Alkhalil) on improving Arabic text summarization. Our system is graph-based and uses the cosine similarity measure to compute the similarity between each two sentences in order to draw the graph representation of the input document. Moreover, the cosine similarity is used in the MMR method in order to eliminate redundancy from the final summary. Based on our experiments and results we conclude that the Khoja stemmer got best stemmer for Arabic text summarization using benchmark dataset, in general, our experiments shows superior significant improvement by Khoja compared to light 10 (Larkey stemmer), because it gets the highest performance result, but comparing the raw data without stemmer got the worst performance of the system, but without significant improvement between light10 and Alkhalil stemmers. We also concluded based on our experiments, that the summarization of an Arabic text is more effective when using the stemming process. The performances of our system are significantly improved by applying the stemming process in the pre-processing stage.

Several suggestions for future research were made. First, we propose to use other similarity measures to make a graph representation of the input document. In the future, we are planning to evaluate the effectiveness of stemming process on our Arabic summarization system by using in addition to the cosine similarity, other similarity measures used in the literature like Pearson's correlation coefficient, Jaccard coefficient and the Euclidean distance. Second, we suggest evaluating our system on an extended corpus that contains more documents. Third, we plan to study the impact of stemming process on other Arabic summarization approaches like machine learning or Latent Semantic Analysis.

References

[1] Farghaly, A., Shaalan, K.: Arabic Natural Language Processing: Challenges and Solutions. ACM Transactions on Asian Language Information Processing (TALIP) 8(4), 14 (2009)

[2] Alqudsi, A., Omar, N., Shaker, K.: Arabic Machine Translation: A Survey. Artificial Intelligence Review, pp. 1-24 (2012)

[3] Khoja, Shereen, 1999. Stemming Arabic Text. http://zeus.cs.pacificu.edu/shereen/research.htm

[4] Larkey, S., Ballesteros, L., Connell, E.: Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Aanalysis. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, Finland (2002)

[5] Larkey, L., Ballesteros, L., Connell, M.: Light Stemming for Arabic Information Retrieval. Arabic Computational Morphology, 221–243 (2007)

[6] Atwan, J., Mohd, M., Kanaan, G.: Impact of stemmer on arabic text retrieval. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 8870, 2014, p. 314-326. Springer International Publishing Switzerland (2014)

[7] Salton, G., Buckley, C.: Improving Retrieval Performance by Relevance Feedback. In: Information Retrieval, pp. 355–364 (1997)

[8] Manning, C.D., Raghavan, P., Schutze, H.: Introduction to Information Retrieval, vol. 1. Cambridge University Press, Cambridge (2008)

[9] Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval, vol. 463. ACM Press, New York (1999)

[10] Llopis, F., Vicedo, J., Ferrández, A.: Passage Selection to Improve Question Answering. In: Proceedings of the 2002 Conference on Multilingual Summarization and Question Answering, vol. 19, pp. 1–6. Association for Computational Linguistics (2002)

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

[12] Larkey, L., Connell, M.: Arabic Information Retrieval at UMass. In: TREC-10. NIST SPECIAL PUBLICATION SP, pp. 562-570 (2002)

[13] 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.

[14] Bsoul, Qusay; Al-Shamari, Eiman; Mohd, Masnizah; Atwan, Jaffar.: Distance measures and stemming impact on arabic document clustering. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 8870, 2014, p. 327-339. Springer International Publishing Switzerland (2014)

[15] H. Froud, A. Lachkar and S. A. Ouatik, "Stemming versus Light Stemming for measuring the simitilarity between Arabic Words with Latent Semantic Analysis model," 2012 Colloquium in Information Science and Technology, Fez, 2012, pp. 69-73.

[16] H. Froud, R. Benslimane, A. Lachkar, A. Lachkar and S. A. Ouatik, "Stemming and similarity measures for Arabic Documents Clustering," I/V Communications and Mobile Network (ISVC), 2010 5th International Symposium on, Rabat, 2010, pp. 1-4.

[17] Osama A Ghanem and Wesam M Ashour. Article: Stemming Effectiveness in Clustering of Arabic Documents. International Journal of Computer Applications 49(5):1-6, July 2012

[18] H. Froud, A. Lachkar and S. A. Ouatik, "Stemming for Arabic words similarity measures based on Latent Semantic Analysis model," Multimedia Computing and Systems (ICMCS), 2012 International Conference on, Tangier, 2012, pp. 779-783.

[19] F. S. Al-Anzi and D. AbuZeina, "Stemming impact on Arabic text categorization performance: A survey," 2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA), Marrakech, 2015, pp. 1-7.

[20] Bebah, Mohamed Ould Abdallahi Ould, Abdelouafi Meziane, Azzeddine Mazroui, and Abdelhak Lakhouaja. (2011) Alkhalil morpho sys. In 7th Inter national Computing Conference in Arabic, May 31- June 2, 2011, Riyadh, Saudi Arabia.

[21] Baeza-Yates, R., And Ribeiro-Neto, B.: Modern Information Retrieval, vol. 463. ACM, New York (1999)

[22] Larsen, B., Aone, C.: Fast and Effective Text Mining Using Linear-Time Document Clustering. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD, pp. 16–22 (1999)

[23] T. Nguyen-Hoang, K. Nguyen, and Q. Tran, "TSGVi: a graph-based summarization system for Vietnamese documents", ;presented at J. Ambient Intelligence and Humanized Computing, 2012, pp.305-313.

[24] Brin S, Page L (1998).The anatomy of a large-scale hypertextual web search engine. Comput Netw 30(1-7):107–117

[25] Mihalcea R, Tarau P (2004) Textrank: bringing order into text. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain, 2004

[26] Erkan G, Radev DR (2004) Lexrank: graph-based lexical centrality as salience in text summarization. J Artif Intell Res 22:457–479

[27] Carbonell JG, Goldstein J (1998) The use of MMR, diversity-based re-ranking for reordering documents and producing summaries. In: SIGIR, pp 335–336

[28] V. Tunali and T. T. Bilgin, "Examining the impact of stemming on clustering Turkish texts," Innovations in Intelligent Systems and Applications (INISTA), 2012 International Symposium on, Trabzon, 2012, pp. 1-4.