Using Statistical and Semantic Analysis for Arabic Text Summarization

Nabil Alami(✉), Yassine El Adlouni,
Noureddine En-nahnahi, and Mohammed Meknassi

Laboratory of Informatics and Modelling, Faculty of Science Dhar EL Mahraz,
University Sidi Mohamed Ben Abdellah (USMBA), Fez, Morocco
nab.alami@gmail.com, yeladlouni@gmail.com,
nahnnourd@yahoo.fr, m.meknassi@gmail.com

Abstract. Automatic text summarization is an essential tool to overcome the problem of information overload. So far this field has not been studied enough for Arabic language and currently only few related works are available. Arabic text summarization is faced with two main issues: how to extract semantic relationships between textual units and deal with redundancy. To overcome these problems, we propose in this paper a hybrid method to generate an extractive summary of Arabic documents. Our approach is based on a two-dimensional undirected and weighted graph with sentences as nodes and each pair of sentences are connected by two edges representing the statistical and semantic similarity measure. The statistical similarity measure builds on the content overlap between two sentences, while the semantic one is based upon semantic information extracted from Arabic WordNet (AWN) ontology. Then, the score of each sentence is computed by performing the ranking algorithm PageRank on the generated graph. Thereafter, the score of each sentence is performed by adding other statistical features of the text such as TF.ISF and sentence position. The final summary is built by selecting the top-ranking sentences. Finally, we deal with redundancy and information diversity issues by using an adapted maximal marginal relevance (MMR) method. Experimental results on EASC dataset show that our proposed approach outperforms some of existing Arabic summarization systems.

Keywords: Arabic text summarization · Arabic NLP · Statistic approach · Sematic approach · AWN · Graph model

1 Introduction

Automatic Text summarization is the task of automatically generating a condensed version of an input text that provides useful information for the user with preserving the main ideas in the original document. Text summarization is one of the most challenging tasks in natural language processing area (NLP). Hovy [1] defined a summary as a text that is produced from one or more texts which contains a significant portion of the information in the original text(s), and that is no longer than half of the original text(s).

G. Noreddine and J. Kacprzyk (eds.), *International Conference on Information Technology and Communication Systems*, Advances in Intelligent Systems and Computing 640, DOI 10.1007/978-3-319-64719-7_4

As a text document, automatic summarization can be applied to any other media such as speech, multimedia documents, hypertext or even video.

Because of the rapid growth in online information, the huge amount of information available electronically becomes unmanageable, and users are unable to read and extract useful information from them. That is why the automatic summarization tool has become an essential task that allows users to quickly find and extract the data that they need to quickly make mission-critical business decisions. The first work on automatic text summarization was proposed by Luhn [2] in the late fifties and to date; automatic summarization is an active field in natural language processing area with a challenging issue.

Unlike English, which many outstanding achievements have been made in the field of automatic summarization, few and no excellent systems have been developed for automatic summarization of Arabic text. This particular field in Arabic natural language processing is not studied enough compared to the large number of studies carried out for English. Therefore, there is a considerable opportunity for further research in automatic summarization for Arabic documents specially when the existing systems are note mature enough and efficient as we need.

General speaking, Text summarization can be divided into two major classes: abstractive and extractive. In the abstractive summarization, the system has to regenerate either the extracted content or the text. It requires human knowledge and heavy machinery for language generation; however, in extractive class, the sentences have to be ranked based on the most salient information. The summary is built by the most ranked sentences arranged with the same order as presented in the original text. This is equally known as sentence ranking, for which the importance of each sentence is computed according to linguistic and statistical features.

It is worth noting that Arabic is the language spoken by more than 22 countries and the fifth most spoken language in the word. Recent studies indicate that Arabic language is fastest-growing language on the web in terms of the number of internet users. It is ranked the fourth language used in the web after English, Chinese and Spanish. However, research on Arabic NLP is still in its infancy since researchers become less interested in devoting the time and effort to Arabic than to other languages. Therefore, developing ANLP systems has become paramount for accessing a large number of Arabic documents available in the Internet. Automatic summarization is one of these systems.

Traditional summarization systems rely on the Bag of Words (BOW) approach. The BOW approach is based on the words existing in the text to be summarized. One of the obvious disadvantages of this approach is that it cannot accurately represent the meaning of documents, because it ignores the semantic relationship existing between different textual units. The system is always limited to the words explicitly mentioned within the input text document. For example, if the system is not able to find the relationships between terms like « بـــترول » (Petroleum) and « نفط » (Oil), it would handle these words separately as two different unrelated terms, and this may affect negatively their importance in the input document.

In this paper, we develop an Arabic text summarization system that combines statistical and semantic approaches to achieve the summarization task. In the first step, the proposed system proceeds to document preprocessing: sentence splitting, tokenization, stop words removal and root extraction. Second, it computes the similarity between each pair of sentences. For this, two kinds of similarity are used: (i) statistical similarity based on the content overlap between two sentences, this measure is used in TextRank [29]; (ii) semantic similarity measure based on the semantic information extracted from AWN for each pair of words of the given sentences using the first concept as a disambiguation strategy. Based on that, the proposed algorithm converts the text into a graph model with sentences as nodes and each pair of sentences is connected by two edges representing two types of relations: statistic similarity; and semantic similarity calculated in the previous step. Thereafter, PageRank [28], as a ranking algorithm, is performed on a generated graph in order to compute the score of each sentence. Then, the score of each sentence is improved by adding other statistical features such as TF.ISF and sentence position. The summary is formed by including the top-ranked sentences from the input document, and an adapted version of maximal marginal relevance (MMR) algorithm [35] is applied in order to remove information that is redundant and enhance the quality of the result summary.

The proposed summarization approach has several advantages. Firstly, the system is domain-independent and does not need any domain-specific knowledge or features. Secondly, this method does not require any training data or annotated corpus. Thirdly, and unlike the existing methods, we consider the semantic relationships among words and we introduce semantic information from Arabic WordNet (AWN) ontologies in order to accurately represent the meaning of documents. Finally, redundancy is a well-known issue in text summarization. Therefore, we deal with redundancy and information diversity issues by using the MMR method.

This paper is presented as follows. Section 2 summarizes existing works on automatic summarization, especially in Arabic and discusses the limitation of the current approaches. Section 3 describes in detail the proposed method. System evaluation and the experimental results are discussed in Sect. 4. A conclusion and future works are given in Sect. 5.

2 Related Works

Automatic text summarization can be classified into three approaches: numerical, symbolic and hybrid approaches. The first research in text summarization was published more than 50 years ago [2, 3]. Luhn [2] used word frequency to determine the importance of each sentence in a single document. The method proposed by Edmundson [3] used more than one feature to score sentences, word frequencies, sentence positions and cue words. These features are still used by many automatic summarization systems developed in recent years [4, 9].

Compared to English and other languages (Spanish, Chinese, French), a very little works has been made in Arabic document summarization field. According to our knowledge, the first system designed for Arabic text summarization was developed by Douzidia and Lapalme [4]. It uses a linear combination of many statistical features (tf, position of sentence, etc.).

AlSanie [5] developed a symbolic approach for Arabic based on Rhetorical Structure Theory (RST) where 11 relations were used in the summarization process. The system first generated all possible Rhetorical Structure Tree (RS-Tree) for the text based on relationships between textual units and then, the system generated the summary by selecting the best tree. The author used a set of documents with their summaries generated manually in order to evaluate the system performances. The system performed well the summarization task for small and medium sized documents.

Haboush et al. [6] proposed a single document summarization model based on the clustering technique. The words can be grouped into different clusters based on roots extracted from the document. The weight of the root is used, instead of the weight of the word itself, in order to compute the score of each sentence. The score is obtained by combining the weights of its words; the weight of words is computed according to its frequency and importance in the paragraph. The most ranked sentences are then selected to generate the summary.

El-Haj et al. [7] developed a multi-document Arabic summarization system based on a clustering approach. The authors investigated two kind of experiments. In the first experiment, the authors cluster sentences based on the k-mean clustering technique using the cosine similarity measure. Then, the summary is generated by selecting sentences using two selection methods. In the first method, sentences are selected from the largest cluster, while in the second method, the first sentence is selected from each cluster in order to eliminate redundancy within the produced summary. In the second experiment the difference is that the sentences are selected before applying the clustering. Thus, the first sentence is selected and then the most similar sentence to the first one. The authors evaluated and compared their system with other English summarizes using the English and Arabic version of DUC 2002 corpus. The Arabic version of the DUC 2002 datasets was generated using Google Translate.

Ibrahim and Elghazaly [8] followed a hybrid approach that uses two summarization techniques: Rhetorical Representation based on RST and Vector Representation based on Vector Space Model (VSM). The first technique used a rhetorical representation of the text using RST for generating the RS-Tree and extracting the most meaningful paragraphs to be included in the summary. The second method uses a cosine similarity metric to build a vector representation based on VSM. As shown in the evaluation results, the Rhetorical Representation technique produces better average in precision measure and better quality in the produced summaries than Vector Representation; however, the performance of the second was better with long articles.

Recently, a number of semantic-based approaches have been developed. WordNet [11], as a linguistic resources designed for English, is widely used in different NLP systems, and with its semantic relations of terms, it has been extensively used to improve the quality of automatic text summarization [12–15], text clustering [16–18], word sense disambiguation [19, 20] and other natural language processing tasks [21–23].

3 Our Arabic Text Summarization System

This paper introduces a new approach for automatic summarization of Arabic documents using statistical methods and semantic treatment to increase information diversity of the input Arabic document. The various steps followed by the proposed system are shown in Fig. 1. The input is a large Arabic document and the output is a small one which contains the most important sentences. The following section explains in detail each step used in the proposed system.

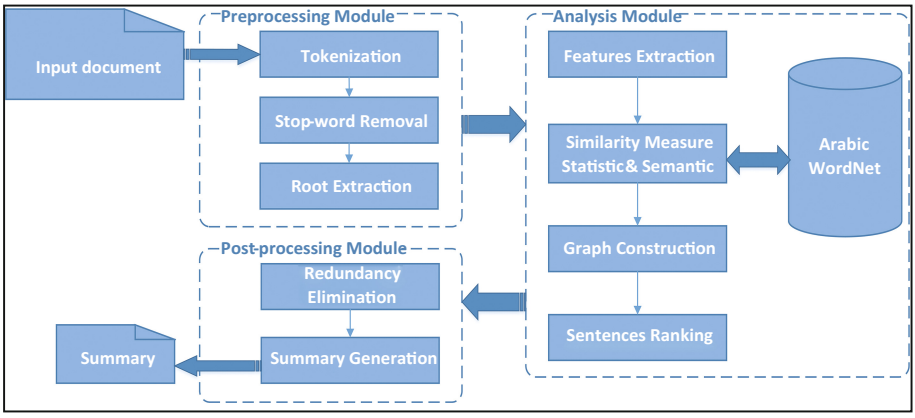

Fig. 1. The main process of our approach

3.1 Preprocessing

Step 1: Tokenization. Tokenization is an important preprocessing step to enable NLP tools which require clean and standardized data as input. First, the document is normalized in order to convert a text to a standard format (removing punctuation, non-letters and diacritics, replacing أ , آ, and إ with ا and final ى with ي and final ة with ه). Then, the document is segmented into words (tokens) and sentences. In our system, and depending on the datasets used, the white space character (" ") defines the boundaries of words and the dot character (".") is considered as a sentence separator.

Step 2: Stop words removal. The most frequent common words which the main function is structural and give a little value to the meaning of the text are removed from the text vocabulary. These words are called stop words. They are filtered out before the preprocessing step in any NLP task because they serve only a syntactic function, and do

not indicate subject matter. The performances of automatic summarization systems are significantly reduced when using stop-words. In Arabic, words like (هي,هو, هذا, الذي) are often used in sentences, and have a little meaning in the implication of a document. There is no standard list of stop words used by all Arabic NLP tools. In this work we used a list of 168 words extracted from Khoja [24] to remove stop words.

Step 3: Root extraction. In Arabic, words that share a related meaning are generally derived from a same root. As shown in Table 1, all the three Arabic words "كتب", "كتاب" and "مكتبة" are related to the same root "كتب".

Table 1. Different derivation of root "كتب"

Arabic word	English sense	Root
كتب	Write	كتب
كتاب	Book	كتب
مكتبة	Library	كتب

Word stemming is one the most challenging issues in Arabic. Several researches have been made to develop powerful stemming tools. Khoja's stemmer presented by [24] is one of the most popular one. It extracts the root using pattern matching after removing predefined suffixes, infixes, and prefixes. In this work, we have included the Khoja's stemmer to perform the stemming task.

3.2 Analysis

After preprocessing the input Arabic document, the analyzing stage begins scoring the sentences based on the computed set of features. Each sentence is given two kinds of scores, implying its significance in relation with the rest of the sentences: statistical score and semantic score. Statistical score is the result of summing all weights given to each extracted feature. The semantic score is determined by the importance of the sentence against other sentences using the semantic relationship between them.

Step 4: Features extraction. We used two features to compute the statistical score of each sentence: TF-ISF and sentence position.

Term frequency/Inverse sentence frequency (TF-ISF). Tf-IDF stands for term frequency-inverse document frequency. It is composed by two terms: the first (TF) represents the number of time a term appears in a document. The second term is the Inverse Document Frequency (IDF), which measures the importance of a word in a document collection. It is calculated by the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears. TF-IDF is widely used as a weighting factor in many information retrieval and text mining applications. In this paper, we have adopted the Term Frequency-Inverse Sentence Frequency (TF-ISF) measure which is the same of TF-IDF by replacing a set of documents by a set of sentences. The inverse sentence frequency (ISF) measures the importance of a term within the sentence collection [31].

$$ISF_{wi} = \log_2 \frac{N}{df_{wi}} \tag{1}$$

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

Sentence position. Usually, the most important sentences in a document are in a specific position. The sentence position feature is another statistical feature which can help in extracting salient information from the original text. In this paper, the position of sentence is considered as a scoring factor in the summarization task. We consider the first and the last sentences are most related to the topic so their weight is high.

Step 5: Semantic similarity measure. Semantic similarity is becoming more and more popular, and plays a significant role in many NLP tasks such as text mining, information retrieval and extraction, text summarization, text categorization, text clustering and so on. The lack of common terms in two sentences (Or two documents) does not necessarily mean that the sentences are not related. Therefore, summarization by using only classical methods will fail to retrieve sentences (Or documents) with semantically similar terms. This is exactly the problem this work is addressing.

Several semantic similarities have been proposed to quantify the semantic similarity based on ontology hierarchy. Some utilize the taxonomies within WordNet [32] and the relations defined between its units. WordNet is the hierarchically-structured repository that was created by linguistic experts and is rich in its explicitly defined lexical relations. This kind of measures based on WordNet, has been widely used in NLP applications [25]. In this work, we used Arabic WordNet (AWN) which is a lexical resource for standard modern Arabic based on Princeton WordNet, and is built according to methods developed for EuroWordNet.

In this work, we have adopted the concepts based representation model to compute the semantic similarity between two terms. In this model, we replace each term by its associated concepts in AWN ontology, and we apply a disambiguation strategy in order to assign one concept to one term. In our approach, we have adopted the "First concept" strategy as a simple disambiguation method. This disambiguation strategy consists in taking only the first concept of the list as the most suitable concept [33]. Figure 2 shows the flowchart of the semantic similarity measure process between two Arabic words wi and wj. Moreover, we have used the Wu and Palmer [26] measure to compute the semantic similarity between any two concepts. This measure was found to be simple to calculate, and presents more performances while remaining competitive and expressive as others similarity measures [27].

Based on this measure, the semantic similarity between two sentences is computed using the formula proposed by [34]. This measure is based on the sum of maximum word similarity scores of words normalized by the sum of the sentence length. The sentence semantic similarity formulation is defined in (2).

$$Sim(S_i, S_j) = \frac{\sum_{w \in S_i} \max Sim_w(w, S_j) + \sum_{w \in S_j} \max Sim_w(w, S_i)}{|S_i| + |S_j|} \quad (2)$$

In this equation:

- S_i and S_j are the given sentences.
- $\max Sim_w(w, S_j)$ is the maximum similarity scores of the word w and the words in S_j.
- $|S_i|$ represent the length of the sentence S_i.

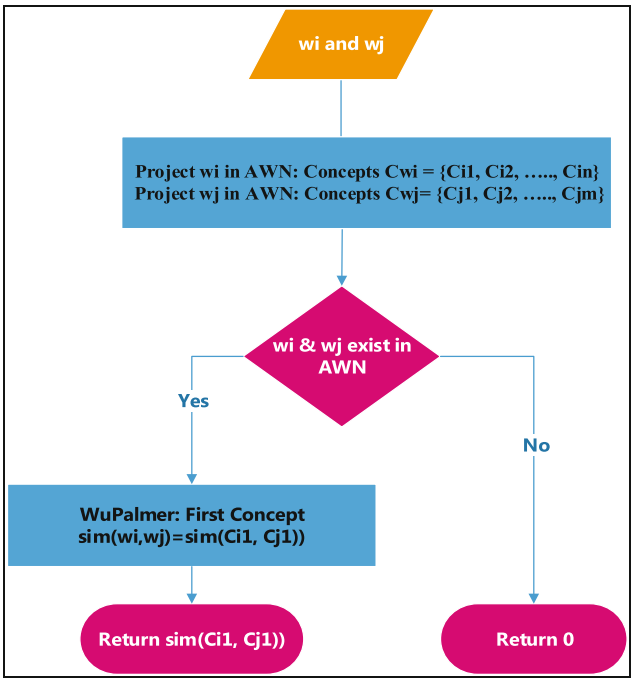

Fig. 2. Semantic similarity measure between two Arabic words wi and wj

Step 6: Graph construction. In this step we transform an Arabic text document into graph format. To draw the graph, we need to identify text units that best define the task of automatic summarization and consider them as vertices of the graph. Then, we need to identify relations that connect those units.

In this work, we consider the sentences of the input Arabic document as a text unit and the similarity between those sentences as a relation between sentences. We build an undirected weighted graph G = (V, E). In this graph, sentences are represented by a set of vertices V and the relation between each sentence is represented by the edge that connects the two correspondent vertices. Two different kinds of edges are used: Semantic similarity and statistical similarity.

Semantic similarity: An edge is drawn between sentences that are similar to each other. The edge represents the semantic similarity between the sentences and edge weight represents the power of the relationship between the sentences in the text. In this work, we used the semantic similarity calculated in the previous section.

Statistical similarity: Another edge is considered that represents the content overlap between two sentences. The weight of the edge represents the number of common tokens between the two sentences divided by the length of each sentence. Formally, the statistical similarity between two sentences S_i and S_j is defined as [29]:

$$Similarity(S_i, S_j) = \frac{\left|\{w_k | w_k \in S_i \cap S_j\}\right|}{\log\left(|S_i|\right) + \log(|S_j|)} \tag{3}$$

Where $|S_i|$ represents the length of the sentence S_i.

Step 7: Sentence ranker. The input of this process is the undirected weighted graph resulted from the previous step. PageRank algorithm [28] was used to compute a salient score for each vertex of the graph. PageRank is one of the most popular link analysis algorithms and was proposed as a method for Web link analysis. The information existing in the graph structure reflects the importance of a vertex against each other. In our case, the key intuition is that a sentence should be highly ranked if it is recommended by many other highly ranked sentences. PageRank is well adapted to undirected graph. In this way, the input edges and the output edges for a vertex are the same. In this work, since the graph is undirected $In(V_i)$ is equal to $Out(V_i)$.

Equation (4) computes the score of a vertex V_i, where $adj(V_i)$ is the set of vertices adjacent to V_i, w_{ij} is the weight of the edge between vertex V_i and vertex 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 [29].

$$PR(V_i) = (1 - d) + d * \sum_{V_j \in adj(V_i)} w_{ij} \frac{PR(V_j)}{\sum_{V_k \in adj(V_j)} w_{jk}} \tag{4}$$

We apply (4) iteratively on a weighed graph G to calculate PR. First, all nodes in the generated graph are initialized by a score of 1. Then Eq. (4) is applied until the difference in scores between iterations becomes less than a predefined threshold for all nodes. in our experiments we chose the value of the threshold equal to 0.001. The score of each sentence is given by the weight of its associated vertex. The importance of a sentence that is salient to the text and has a strong relationship with others sentences is defined by the weight degree of its vertex.

It is to be noted that (4) is applied on both statistical and semantic edges. We obtain two scores for each vertex $PR_{static}(V_i)$ and $PR_{semantic}(V_i)$. The final score of each vertex in the graph is obtained by the following formula:

$$PR(V_i) = PR_{static}(V_i) + PR_{semantic}(V_i) \tag{5}$$

In the final step of the ranking process, $PR(V_i)$ is improved by other statistical features like *TF.ISF* of the root and position of the sentence. The final score assigned to each sentence is given by the following formula:

$$score(S_i) = PR(S_i) + \frac{\sum_{wj \in S_i} TF.ISF(w_j)}{rootCount(S_i)} + Position(S_i) \tag{6}$$

Where:

- $PR(S_i)$ is the rank of the sentence S_i in the graph given by (5).
- *TF.ISF* is the term frequency inverse sentence frequency of the root.
- $Position(S_i) = 1$ for sentences in the first and last position, 0 otherwise.

3.3 Post-processing

Post-treatment is the final step of our system. It consists of eliminating redundancy from the best scored sentences by the formula (6). In this way, we are sure that our final generated summary covers a diversity of most information contained in the original input document.

Step 8: Redundancy elimination and summary generation. In this step, each sentence has its salient score $Score(S_i)$ obtained after the ranking process. Simply and as other graph based summarization systems, we can choose the sentences with the highest score to be included in the final summary. However, several information is redundant in the summary, because many similar sentences representing the same idea in the document have similar score, so they can be included together in the summary. Moreover, some important sentences may not be included in the final summary and other ideas of the text can be ignored. To overcome this problem, the final summary is build using an adapted version of MMR [35] algorithm. MMR re-rank all sentences and select the most relevant between them without redundancy.

The main idea of MMR is that the summary is constructed by a high ranked sentences which are not similar to any existing sentence in the summary. The summary S is initialized by the most ranked sentence which is removed from the ranked list R. Then, the sentence with the highest score from (7) is added to the summary and removed from the ranked list. This process is repeated until the length of the summary is reached. The MMR method works according to the flowing equation:

$$MMR = argmax_{s_i \in R \setminus S}\left[\lambda * score(s_i) - (1-\lambda) * \max_{s_j \in S} * sim(s_i, s_j)\right] \tag{7}$$

In this equation, R is the set of sentences, S is the set of summary sentences, $score(s_i)$ is the score of sentence S_i computed by Eq. (6) and $sim(s_i, s_j)$ is the measure of the semantic similarity between sentences s_i and s_j; λ is a tuning factor between the importance of the sentence and its relevance to previously selected sentences.

4 Experimentation and Results

4.1 Datasets (Corpus)

To better assess the quality of any automatic summarization method, we need to compare the obtained results with manually extracted summary. In Arabic language, several studies have been made to overcome the shortage of the Arabic language in corpora. El-Haj et al. [10] used Amazon's Mechanical Turk to build Essex Arabic Summaries Corpus (EASC). The dataset consists of 153 Arabic articles collected from two Arabic newspapers and the Arabic version of Wikipedia. The dataset contains 10 main topics: art and music, education, environment, finance, health, politics, religion, science and technology, sports and tourism. For each document, five model extractive summaries are available. These model summaries were created by native Arabic speakers using Mechanical Turk. The model summaries size does not exceed 50% of the source document's size. The dataset is available in two encodings: UTF-8 and ISO-Arabic. Nonetheless, nowadays and to the best of our knowledge, no standard dataset has been designed to evaluate the performances Arabic text summarization systems. In this work, we used the EASC corpus to evaluate the performance of our system over reference systems.

4.2 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 information returned by the system is correct.

$$P = \frac{|S_{manual} \cap S_{auto}|}{|S_{auto}|} \tag{8}$$

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.

$$R = \frac{|S_{manual} \cap S_{auto}|}{|S_{manual}|} \tag{9}$$

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:

$$F = \frac{2 * P * R}{P + R} \tag{10}$$

4.3 Experiment Setup

We developed all the system described below in java language. We evaluated our system against four other systems. The first system is the Arabic text summarization based on graph theory (ATSG) [36]. It uses a cosine similarity to compute the similarity between sentences. It makes a graph representation for an input Arabic document and applies the PageRank algorithm in order to rank each sentence in the graph. The system is then performed by removing redundancy from the final summary. The second system is TextRank [29]. TextRank is a graph-based ranking algorithm which is designed for keywords extraction and text summarization systems. It performs a Google PageRank algorithm [28] on a graph generated from the original text. Thus, all sentences in the document to be summarized is represented by a node in the graph and the edge between two note represents the similarity relation computed as a number of similar words existing between two sentences. The weight of each edge indicates the degree of similarity between related sentences. Sentences are ranked according to the score given by PageRank. The final summary is built by selecting sentences with the highest score. The third system is LexRank [30] which is the same of TextRank. The main difference is that LexRank is adapted to multi-document summarization and it uses the cosine similarity metric instead of the number of similar terms between two sentences. We developed TextRank and LexRank and adapted these two systems in order to support Arabic Text. The fourth system is a clustering based Arabic text summarization system described in [6]. We ran our algorithm to generate summaries for these sample texts in five different sizes: 20%, 25%, 30%, 35% and 40%.

4.4 Results and Discussion

We have calculated Precision, Recall and F1 score to evaluate the quality of the summary. Table 2 summarizes the results of running our algorithm on the ESCAS corpus with different sizes. As can be seen in Table 2, when the summary size decreases the recall also decreases and so will the F1-measure because the co-occurrence between candidate summary and gold summary increases. This should be clear from Eq. (9).

Table 2. Evaluation results of our system

Precision	Recall	F1-measure	Summary size
62.25	47.76	54.05	20%
59.98	54.11	56.89	25%
57.62	58.80	58.20	30%
53.24	65.17	58.60	35%
51.22	70.05	59.52	40%

The comparison between average Recall, precision and F1-measure of our system with other baseline systems is given in Table 3. The summary size taken into account in this comparison is 30% of the original document. We can see that our system has the highest value of average *F1* score as compared to other systems. *ATSG* system has a good *F1* score compared to other baseline systems, which shows that our method enhances the performance of the Graph-based summarization system.

Table 3. Comparison against other systems with 30% of size

System	Precision	Recall	F1-measure
Our system	57.62	58.80	58.20
ATSG [36]	46.22	47.31	46.76
TextRank [29]	44.26	36.24	39.85
LexRank [30]	31.03	25.71	28.12
Clustering technique [6]	47.45	38.89	42.74

With summary size 30%, the best F1-measure score of the other systems is reported by the *ATSG* system with 46.76% for the tested dataset, whereas in our experiment, the average value of F1-measure is 58.20% for the same dataset.

Firstly, English text summarization systems rely on some existing evaluation workshops organized to encourage research to evaluate their systems on a large test collection. These workshops like Document Understanding Conferences (DUC) or Text Analysis Conference (TAC) do not support Arabic language. Thus no approved benchmark dataset available to evaluate Arabic text summarization systems and each work uses its own dataset. This situation makes the performance comparison between the proposed approaches and the existing ones more difficult.

Secondly, and due to the complexity of the evaluation task, a new evaluation metrics are under development. Thus, various works use different evaluation measures in order to better asses the quality of the summarization task. Moreover, the scientific community interested in Arabic NLP and especially in automatic summarization field, is small and not always enough. Furthermore, Arabic is complex in terms of its morphology, vocabulary and spelling, which make lexical, syntactic and semantic analysis even difficult.

5 Conclusion and Future Work

This paper presents a novel Arabic text summarization system that the main goal is to take into account the semantic relationships existing between textual units and to deal with redundancy and information diversity issues. This system integrates the power of a graph model in order to score sentences according to the Google PageRank algorithm. The graph is used to represent the Arabic document using two different relations among sentences: (i) statistical similarity; and (ii) semantic similarity. Other statistical features extracted from the document are used to improve the summary quality. Our proposed system is knowledge-rich because it integrates an external knowledge database developed by human.

The main contribution of the proposed method is the creation of an Arabic summarization system that combines statistical and semantic treatment of the input text, besides using an adapted version of MMR technique to eliminate redundancy from the final summary.

A comparison of performance measures indicates the advantage of our system as compared to some other summarization systems. Benchmarking the proposed algorithm using the dataset described above showed that our system outperformed all other systems and enhances the performance of the Graph-based summarization system. In addition, the system is domain-dependent and does not use any structural features and was, therefore, successfully used in Arabic summarization task.

In the future work, we are going to investigate several directions in order to improve the performance of the proposed system. The first direction is to extend the dataset used in this paper by additional documents with their manual summaries. This will give more value to our method. A second direction is to consider more features of the text such as part-of speech and co-reference resolution. Another direction is to adapt our system to Arabic multiple document summarization.

References

1. Hovy, E.H.: Automated text summarization. In: Mitkov, R. (ed.) The Oxford Handbook of Computational Linguistics, pp. 583–598. Oxford University Press, Oxford (2005)
2. Luhn, H.P.: The automatic creation of literature abstracts. IBM J. Res. Dev. **2**, 159–165 (1958)
3. Edmundson, H.P.: New methods in automatic extracting. J. ACM **16**(2), 264–285 (1969)
4. Douzidia, F.S., Lapalme, G.: Lakhas, an Arabic summarization system. In: Proceedings of 2004 Document Understanding Conference (DUC 2004), Boston, MA (2004)
5. AlSanie, W.: Towards an infrastructure for Arabic text summarization using rhetorical structure theory. M.Sc. Thesis. King Saud University, Riyadh, Saudi Arabia (2005)
6. Haboush, A., Momani, A., Al-Zoubi, M., Tarazi, M.: Arabic text summarization model using clustering techniques. World Comput. Sci. Inf. Technol. J. (WCSIT) **2**(3), 62–67 (2012). ISSN: 2221-0741
7. El-Haj, M., Kruschwitz, U., Fox, C.: Exploring clustering for multi-document arabic summarisation. In: Salem, M., Shaalan, K., Oroumchian, F., Shakery, A., Khelalfa, H. (eds.) Information Retrieval Technology. Lecture Notes in Computer Science, vol. 7097, pp. 550–561. Springer, Berlin (2011)

8. Ibrahim, A., Elghazaly, T.: Rhetorical representation and vector representation in summarizing arabic text. In: Métais, E., Meziane, F., Saraee, M., Sugumaran, V., Vadera, S. (eds.) Natural Language Processing and Information Systems. Lecture Notes in Computer Science, vol. 7934, pp. 421–424. Springer, Berlin (2013)
9. Oufaida, H., Nouali, O., Blache, P.: Minimum redundancy and maximum relevance for single and multidocument Arabic text summarization. J King Saud Univ. Comput. Inf. Sci. **26**(4), 450–461 (2014). Special Issue on Arabic NLP
10. El-Haj, M., Kruschwitz, U., Fox, C.: Using mechanical turk to create a corpus of arabic summaries. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC), Valletta, Malta. In the Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages Workshop Held in Conjunction with the 7th International Language Resources and Evaluation Conference, pp. 36–39 (2010)
11. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
12. Ferreira, R., de Souza Cabral, L., Freitas, F., Lins, R.D., de Frana Silva, G., Simske, S.J., Favaro, L.: A multi-document summarization system based on statistics and linguistic treatment. Expert Syst. Appl. **41**(13), 5780–5787 (2014)
13. Pal, A.R., Saha, D.: An approach to automatic text summarization using WordNet. In: Advance Computing Conference (IACC), 2014 IEEE International, pp. 1169–1173 (2014). doi:10.1109/IAdCC.2014.6779492
14. Estiri, A., Kahani, M., Ghaemi, H., Abasi, M.: Improvement of an abstractive summarization evaluation tool using lexical-semantic relations and weighted syntax tags in Farsi language. In: 2014 Iranian Conference on Intelligent Systems (ICIS), pp. 1–6 (2014). doi:10.1109/IranianCIS.2014.6802594
15. Patil, A.P., Dalmia, S., Abu Ayub Ansari, S., Aul, T., Bhatnagar, V.: Automatic text summarizer. In: 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1530–1534 (2014). doi:10.1109/ICACCI.2014.6968629
16. Wei, T., Lu, Y., Chang, H., Zhou, Q., Bao, X.: A semantic approach for text clustering using WordNet and lexical chains. Expert Syst. Appl. **42**(4), 2264–2275 (2015). doi:10.1016/j.eswa.2014.10.023. ISSN: 0957-4174
17. Bouras, C., Tsogkas, V.: A clustering technique for news articles using WordNet. Knowl Based Syst. **36**, 115–128 (2012)
18. Chen, C.L., Tseng, F.S., Liang, T.: An integration of WordNet and fuzzy association rule mining for multi-label document clustering. Data Knowl. Eng. **69**(11), 1208–1226 (2010)
19. Sachdeva, P., Verma, S., Singh, S.K.: An improved approach to word sense disambiguation. In: 2014 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 000235–000240 (2014)
20. Dhungana, U.R., Shakya, S., Baral, K., Sharma, B.: Word Sense Disambiguation using WSD specific WordNet of polysemy words. In: 2015 IEEE International Conference on Semantic Computing (ICSC), pp. 148–152 (2015)
21. Gao, J.B., Zhang, B.W., Chen, X.H.: A WordNet-based semantic similarity measurement combining edge-counting and information content theory. Eng. Appl. Artif. Intell. **39**, 80–88 (2015)
22. Li, Y., Li, H., Cai, Q., Han, D.: A novel semantic similarity measure within sentences. In: 2012 2nd International Conference on Computer Science and Network Technology (ICCSNT), pp. 1176–1179 (2012). doi:10.1109/ICCSNT.2012.6526134
23. Varelas, G., Voutsakis, E., Raftopoulou, P., Petrakis, E.G.M., Milios, E.E.: Semantic similarity methods in wordnet and their application to information retrieval on the web. In: Proceedings of the Seventh Annual ACM International Workshop on Web Information and Data Management, WIDM 2005, pp. 10–16. ACM, New York (2005)

24. Khoja, S.: APT: Arabic part-of-speech tagger. In: Proceedings of the Student Workshop at NAACL, pp. 20–25 (2001)
25. Pedersen, T.: Information content measures of semantic similarity perform better without sense-tagged text. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT 2010, pp. 329–332. Association for Computational Linguistics, Stroudsburg (2010)
26. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. Paper Presented at the Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics (1994)
27. Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the 15th International Conference on Machine Learning, Madison, WI, pp. 296–304 (1998)
28. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Comput. Netw. **30**(1–7), 107–117 (1998)
29. Mihalcea, R., Tarau, P.: TextRank: bringing order into texts. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain (2004)
30. Erkan, G., Radev, D.R.: Lexrank: graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. **22**, 457–479 (2004)
31. Alguliyev, R.M., Aliguliyev, R.M., Isazade, N.R.: An unsupervised approach to generating generic summaries of documents. Appl. Soft Comput. **34**, 236–250 (2015). ISSN: 1568-4946
32. The Global Wordnet Association. http://www.globalwordnet.org/
33. Elberrichi, Z., Abidi, K.: Arabic text categorization: a comparative study of different representation modes. Int. Arab J. Inf. Technol. **9**, 465–470 (2012)
34. Malik, R., Subramaniam, V., Kaushik, S.: Automatically selecting answer templates to respond to customer emails. In: Proceedings of IJCAI 2007, Hyderabad, India, pp. 1659–1664 (2007)
35. Carbonell, J.G., Goldstein, J.: The use of MMR, diversity-based re-ranking for reordering documents and producing summaries. In: SIGIR, pp. 335–336 (1998)
36. Alami, N., Meknassi, M., Alaoui Ouatik, S., Ennahnahi, N.: Arabic text summarization based on graph theory. In: IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), Marrakech, pp. 1–8 (2015)