Egyptian Informatics Journal 21 (2020) 73–81

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

# Extractive Arabic Text Summarization Using Modified PageRank Algorithm

Reda Elbarougy [a], Gamal Behery [a], Akram El Khatib [b,*]

[a] Department of Computer Science, Faculty of Computer and Information Sciences, Damietta University, New Damietta, Egypt
[b] Department of Mathematics-Computer Science, Faculty of Science, Damietta University, New Damietta, Egypt

ARTICLE INFO

*Article history:*
Received 15 June 2019
Revised 16 October 2019
Accepted 6 November 2019
Available online 5 December 2019

*Keywords:*
Extractive Arabic text summarization
PageRank
Morphological analyzer
Graph based

ABSTRACT

This paper proposed an approach for Arabic text summarization. Text summarization is one of the natural language processing's applications which is used for reducing the original text amount and retrieving only the important information from the original text. The Arabic language has a complex morphological structure which makes it very difficult to extract nouns to be used as a feature for summarization process. Therefore, Al-Khalil morphological analyzer is used to solve the problem of nouns extraction. The proposed approach is a graph-based system, which represents the document as a graph where the vertices of the graph are the sentences. A Modified PageRank algorithm is applied with an initial score for each node that is the number of nouns in this sentence. More nouns in the sentence mean more information, so nouns count used here as initial rank for the sentence. Edges between sentences are the cosine similarity between the sentences, to get a final summary that contains sentences with more information and well connected with each other. The process of text summarization consists of three major stages: preprocessing stage, features extraction and graph construction stage, and finally applying the Modified PageRank algorithm and summary extraction. The Modified PageRank algorithm used a different number of iterations to find the number returns the best summary results, and the extracted summary depends on compression ratio, taking into account removing redundancy depending on the overlapping between the sentences. To evaluate the performance of this approach EASC Corpus is used as a standard. LexRank and TextRank algorithms were used under the same circumstances, the proposed approach provides better results when compared with other Arabic text summarization techniques. The proposed approach performs efficiently with the number of iteration 10,000.



## 1. Introduction

Due to the huge amount of data written daily on the internet since its invention two decades, the desire for automatic text summarization to extract the most important information from the document intensified. A good text summarization system saves the user's effort and time to get the desired data from the text without needing to read the whole text [1].

Text summarization is the process of reducing the amount of text to get the most important parts from the original text and provide it to users. Automatic text summarization carries the summarization process automatically. Many text summarization researches were conducted for English language because it is a simple language in its structure and grammar, which is the opposite of Arabic language, as it is characterized with a complex structure and morphology. More than 350 million people speak Arabic around the world [2] so, Arabic text summarization is widely required.

Text summarization can be divided into multiple categories depending on the factor that is used for the comparison. According to the number of documents, text summarization can be divided into single document and multiple document summarization. In the other hand, according to the type of retrieved sentences, text

* Corresponding author.
*E-mail addresses:* elbarougy@du.edu.eg (R. Elbarougy), gbehery@du.edu.eg (G. Behery), akram_elkhatib@hotmail.com (A. El Khatib).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.eij.2019.11.001
1110-8665/

summarization can be categorized into extractive summary, which includes the selected sentences from the original text without any modification on the sentence structure, and abstractive summarization, which depends on retrieving the meaning of the sentences without adhering sentence structure. In addition, it can be divided into general or query-based summary. The general summary returns sentences without any consideration to any question or relation with the title. While, the query-based summary is returned due to the relation between the summary and the asked question or the relation between the sentence and the document title [1].

Arabic text summarization still suffers from the low performance and the lack of researches done in this application of natural language processing [3]. Moreover, these techniques do not include the power of using nouns [4] in their works, therefore, it is important to develop a new approach based on noun information.

Nouns directly affect the importance of the sentence [5]. More nouns in the sentence gives the sentence more importance. According to the complexity of Arabic language, Al-Khalil morphological analyzer is used to analyze the Arabic sentence and then extract the nouns from each sentence in the document [6].

This paper investigates a new approach that depends on combining graph theory and nouns count in sentences to extract Arabic text summarization. This approach starts by applying the following preprocessing techniques: normalization, tokenization, stemming, stop words removal and morphological analyzes techniques. Then extracting the needed features, building the graph, then applying the Modified PageRank algorithm (MPR) and finally extracting the summary and remove redundancy.

In the final stage the Modified PageRank algorithm [7] is applied, then sentences are sorted depending on its final score, and then the summary is extracted and redundant sentences are removed. The average performance of the summary got 67.98 in F-measure metrics when 10,000 iterations are used. Later in this paper related works will be discussed in Section 2. Motivation and problem of statement in Section 3. Graphs in Text summarization in Section 4. PageRank algorithm is discussed in Section 5. Modified PageRank algorithm is discussed in Section 6. In Section 7 proposed approach is discussed. Then results and conclusion are discussed in Sections 8, 9 respectively.

## 2. Related works

Many researches were conducted in the Arabic text summarization. These researchers used different techniques and algorithm as presented in the following subsections.

### 2.1. Symbolic and rhetorical-based approaches

In this category researchers build their approaches depending on rhetorical relations between different parts of the text [8]. The proposed approach depends on the rhetorical approach, which goes through several stages starting from defining textual units based on cue phrases. Then, generating rhetorical relationships depending on these phrases, building RS-trees, and finally, choosing the best RS-tree to produce the summary. This approach requires the user to identify the cue words manually to feed the algorithm, which is a hard task especially for a huge amount of texts.

### 2.2. Statistical-based approaches

This approach depends on the statistical features of the sentences like: term frequency, sentence position and many other features. Alami et al. [9], is an extractive approach depends on statistical based. This approach uses three types of stemmers Khoja. Light10 and Alkhalil on the generated text summarization. One of the drawbacks of [9] is that they did not use a standard corpus to compare with them, they collected a 42 articles from the internet and asked a specialist in Arabic Language to create the compared summary, that is not accurate as we know the human generated summary differ from person to person. El-Harby et al. [10], is another one which proposed an automatic system that has the ability to restore diacritics (vowels) for non-diacritic Qur'an words, using a unigram base-line model and a bigram Hidden Markov Model (HMM). It was found that the HMMs are useful tools for the task of diacritic restoration in Arabic language. Alami et al. [11], is another one which proposed a single-document graph based extractive summary presented by using a mixed method to generate an extractive summary of Arabic documents, this approach using EASC corpus for evaluation.

### 2.3. Hybrid approaches

Is an approach using both rhetorical and statistical features to build their approaches. Ibrahim et al. [12], proposed an extractive Arabic text summarization system that depends on combining Rhetorical Structure Theory (RST) and Vector Space Model (VSM) as a hybrid system to summarize Arabic text. [13] Proposed a similar system using RST with frequency along with "position", "title key-words" and "numerical values" features. This system combined between RST and SVM, this combination proved that hybrid approach enhances the performance compared to RST only.

### 2.4. Machine learning approaches

Machine learning approaches depend on one or more algorithm from machine learning algorithms like genetic algorithm, or particle swarm and so on. Al-Abdallah and Al-Taani [14] proposed a single document extractive Arabic text summarization approach based on particle swarm optimization algorithm. This approach combined informative and semantic scoring to enhance the performance of summarization process.

### 2.5. Graph-based approaches

This approach uses graph theory principles. In this approach the document is represented as a graph and the sentences represented as the vertices of the graph. Belkebir and Guessoum [12] proposed an approach that uses multi-graph represented by different metrics depending on number theory and probabilities to calculate the importance of the text partition. Al-Taani and Al-Omour [15] is another approach for extracting Arabic text summarization based on graph theory, in this approach PageRank algorithm is used alongside multiple basic units like word, stem and n-gram to determine the final score for each sentence. Also in [15] they did not use nouns in the initial rank for the algorithm. On the other hand, they used EASC corpus as a standard data set and the following performance metrics: precision, recall and f-measure. Alami et al. [16] is another research in graph based Arabic text summarization, which build the document as a graph with sentences in vertices and the edges between two vertices is the cosine similarity between these sentences, if the similarity between two nodes is less than some threshold, it is assumed that these two sentence is not connected, Khoja stemmer as stemming algorithm was used. In the evaluation process, the authors collected 25 documents from the internet, the summary was manually produced by an Arabic expert, which makes it difficult and unfair to compare because there is no standard corpus. In [16] 1 was assigned as initial rank for all the sentences in the graph then, iteration starts on the graph until the difference between the new rank and old rank is 0.001 for

all nodes. [16] use TF-IDF, sentence position and indicative expressions as features and did not using part of speech to enhance the performance using the power of nouns. Malallah and Ali [17] proposed an approach for text summarization based on Linear Discriminant Analysis(LDA) and modified PageRank, this approach is for multilingual documents contains on the Arabic and English language, this approach done by applying the LDA classifier first to classify sentences to important and non-important according to a specified threshold, then the page rank algorithm is applied on the class of important sentences. TAC-2011 was used as a dataset in the training and testing process. Al-Abdallah and Altaani [18] proposed a single document graph-based approach using the Firefly algorithm for the extraction of summaries.

## 3. Motivation and problem of statement

The Arabic summarization system still has low performance, according to the complexity of the language and the lack of research conducted in this discipline. Nouns in the sentence increase its importance; more nouns means more information in this sentence. Many researchers conducted in Arabic text summarization use graph based with the PageRank algorithm. This research uses a Modified PageRank algorithm by including the weight of the edge as a part of the equation, also by using the number of nouns in the sentence to be the initial rank of the sentence. But in Arabic There is no simple way like English to check if a word is a noun because there are no uppercase or lowercase letters like those in English; in addition, modern writers do not include diacritics in their written text. Therefore, to extract nouns from text Morphological solution is required. Also, this research attempts to apply a different number of iterations with Modified PageRank to get the best performance.

## 4. Graphs in text summarization

A graph G(*V*,*E*) is a mathematical structure used to represent the pairwise relation between objects. The graph has two main items *V*: vertices and *E*: edges. Vertices represent the basic item of the represented system, and edges represent the nature of the relation between two vertices. To build any solution using graph model, you need to specify three main issues: (1) what is the basic units of your application, which in text summarization may be words, or phrases, or sentences or even paragraphs. (2) The type of relation between the nodes to calculate the weight of the edges in text summarization it may be cosine similarity or sentences overlapping, etc... (3) the ranking algorithm which used to rank vertices in the graph. In text summarization there are many approaches like LexRank [19], TextRank [20] or PageRank algorithm [7].

TextRank is a single document, graph-based ranking approach, derived from Google PageRank algorithm [7]. TextRank is a undirected connected graph represents sentences as nodes, and the similarity between them as an edge weight. TextRank is used to extract both sentences and key words. After applying TextRank sentences are sorted depending on its score and the highest ranked sentences are selected as a summary.

LexRank is a multi-document graph-based summarization system, all sentences are represented in a graph. Two sentences are connected if they have similarity above a predefined threshold. After building the graph, the most central sentences are selected as a summary.

## 5. PageRank algorithm

The internet is a very complex network. The relation between pages can be represented as a graph. The importance of any vertex is the in-degree (out-degree); which is the number of inbound (outbound) links to this vertex [21]. The inbound of a given page, is an indicator of a page's importance or quality. PageRank algorithm [7] uses this idea to rank the pages that appear in the search results. PageRank does not consider all the inbound links from the pages equal, the link will take an extra importance depending on the importance of the page that comes from it.

**Algorithm 1. PageRank algorithm [6]**

```
    Input: Weighted Graph G.
    Output: Scored Graph.
1   Configure N = Number of Nodes in G.
2   Current_Rank ← Double[N]
3   Temp_Rank ← Double[N]
4     Foreach n = 1 to N
5     Current_Rank[n] = 1/N
6       For i = 1: Number_of_Iterations
7     Foreach nd: G.Nodes
8       Temp_Rank[nd.index] = Calc_Page_Rank(nd)
9     Current_Rank = Temp_Rank
```

The PageRank algorithm [7] of a Web page *u*, denoted by *PR* (*u*), as shown in Formula (1) where *d* is the damping factor with 0.85 and *N*: is the total number of nodes. Algorithm 1 shows how the PageRank algorithm works; it starts by initializing the rank of each node by 1/*N*, where *N* is the number of nodes in the graph. Then the algorithm iterates and calculates the new ranks of the nodes according to Formula (1). After calculating the new ranks for all nodes, the ranks are updated. This process is repeated depending on iterations count. The iteration here is used to update the rank for each sentence to get the best and most stable rank, since the order of the sentence is directly associated with the order of the associated sentences, which changes every time the algorithm is applied.

$$PR(P) = (1-d) + d * \sum_{i=1}^{N} \left( \frac{PR(v_i)}{N(v_i)} \right) \quad (1)$$

## 6. Modified PageRank algorithm

Nouns in Arabic language have a special importance; i.e. the more nouns a sentence has the more important it becomes [4]. So this research uses a new technique that uses the number of nouns in each sentence to modify the original PageRank algorithm to extract Arabic summaries. A Modified PageRank is based mainly on the aspects of PageRank algorithm with the following differences: (1) pages are replaced with the sentences of the document, (2) the weight of the edges between nodes calculated by the cosine similarity, while in the original PageRank there is no weight on the edges. (3) the initial rank of each sentence is the number of nouns in this sentence not like the original PageRank which gives the initial rank equally to all nodes which equals 1/*N*, Where *N* is the number of sentences in the document. (4) The $PR(v_i)$ is modified as in Formula (2). Formula (2) is used to calculate the new rank of node (g), Where $PR(v_i)$ is the current rank of sentence ($v_i$) and $E(g,v_i)$ is the weight of edge connect sentences ($g$) and ($v_i$) which is also the cosine similarity between these two sentences, finally the summation is divided by the number of the remaining sentences in the document *N*-1, which is the number of sentences in the document *D* with excluding the current sentence, to get the new rank of sentence ($g$).

$$MPR(g) = (1-d) + d * \sum_{i=1}^{N} \frac{PR(v_i) * E(g, v_i)}{N-1} \quad (2)$$

**Table 1**
Modified PageRank demo (Sentences initial rank).

| Sentence # | S1 | S2 | S3 | S4 |
|---|---|---|---|---|
| Initial Rank (# of Nouns) | 3 | 8 | 5 | 6 |

**Table 2**
Modified PageRank demo (Edges Weights).

| | S1 | S2 | S3 | S4 |
|---|---|---|---|---|
| S1 | - | 3 | 4 | 1 |
| S2 | 3 | - | 6 | 5 |
| S3 | 4 | 6 | - | 3 |
| S4 | 1 | 5 | 3 | - |

**Table 3**
Modified PageRank demo (ranks after iteration # 1).

| Sentence | S1 | S2 | S3 | S4 |
|---|---|---|---|---|
| New Ranks after Iteration #1 | 14.31 | 19.7 | 22.25 | 16.58 |

Assume we have a document $D$ that contains 4 sentences, and the number of nouns in each sentence presented in Table 1, the cosine similarity or the edge weights listed in Table 2. Then Table 3 shows the first iteration of calculating the new ranks of the sentence by taking into account that the value of the damping factor $d$ here is 0.85 as best practice [7]. Fig. 1 show an example of a document that contains 4 sentences, the nodes represent the sentences of the document and the weight of the edges is the cosine similarity between the connected nodes. Each nodes has two ranks: its initial rank, and the new rank after applying the first iteration.

So, the Modified PageRank for $(S1)$ calculated as in the following:

$$MPR(S1) = \frac{PR(S2) * E(S1,S2) + \mathrm{PR}(S3) * E(S1,S3) + PR(S4) * E(S1,S4)}{3}$$

$$MPR(S1) = (8 * 3 + 5^{*}4 + 6 * 1)/3 = (24 + 20 + 6)/3 = 50/3 = 16.67$$

$$New\ Ranks\ after\ Iteration\ \#1 = (1 - \mathrm{d}) + \mathrm{d} * \mathrm{MPR}(\mathrm{S1})$$

$$New\ Ranks\ after\ Iteration\ \#1 = (1 - 0.85) + 0.85 * 16.67$$

$$New\ Ranks\ after\ Iteration\ \#1 = 14.31$$

Table 3 shows the new ranks for the sentences after applying Modified PageRank for one iteration.

## 7. The proposed approach

In this section the proposed approach is discussed. Fig. 2 shows the flow charts of the proposed approach which contains three major stages. The first stage starts by extracting text from document, then doing the preprocessing tasks: normalization, tokenization, stop words removal, stemming and morphological analysis. In the second stage the desired features are extracted, then the document is modeled as a graph. Finally, in the third stage PageRank algorithm is modified, summary extracted, then the performance is evaluated.

*The stages of proposed approach*

*7.1. Stage 1: Pre-processing*

At this stage the document is entered, then analyzed to be ready for features extraction.

*7.1.1. Input single document.* Extracting text from a single document written in Arabic language and encoding in utf-8.

*7.1.2. Normalization.* In this step punctuations and digits are eliminated from the sentence while no alphabet letters are removed from the sentence. Also, some characters like the first ALEF in every word is replaced by " ا " and replace the (Marbota " ـة ") at word end individually by (Heh " ـه ").

*7.1.3. Tokenization.* In this step the document is divided into paragraphs, then the paragraphs into sentences, and finally the sentences into words.

*7.1.4. Removing stop words.* Removing stop words reduces the text to more useful words. And the non-removal affects the efficiency of the process of weighting.

*7.1.5. Stemming.* In this process Khoja [22], stemmer is used to extract the root of every word in the sentence. This process is used to reduce the number of distinct words in the document to make a better term frequency calculation.

*7.1.6. Morphological analysis.* In this step Alkhalil morphological [6] is used as a morphological analyzer because its accuracy exceeded 98% [6]. In this process every word in the sentence takes a tag representing its Part of Speech (POS) position in the sentence [21]. The position of the words may be a noun, verb, preposition, stop word article, etc... This process is used to determine the number of nouns in each sentence. Table 4 shows an example of

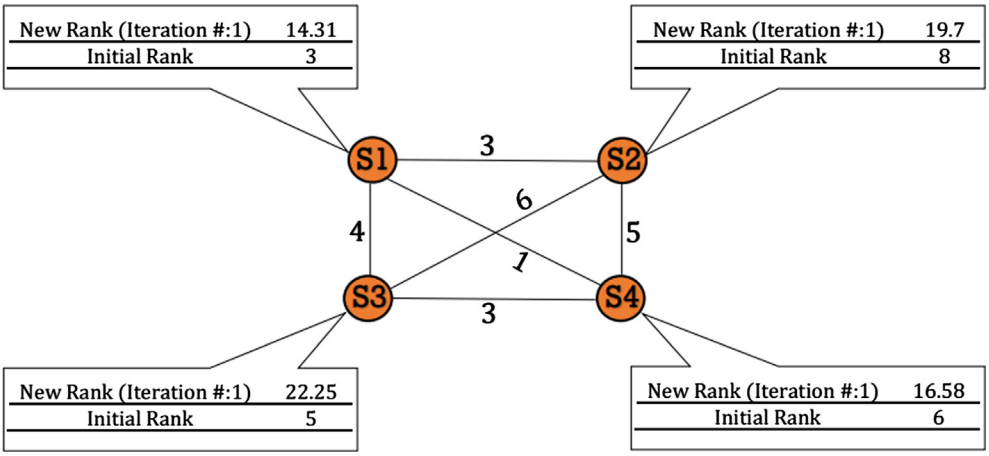


**Fig. 1.** Demo of 4 sentences document.

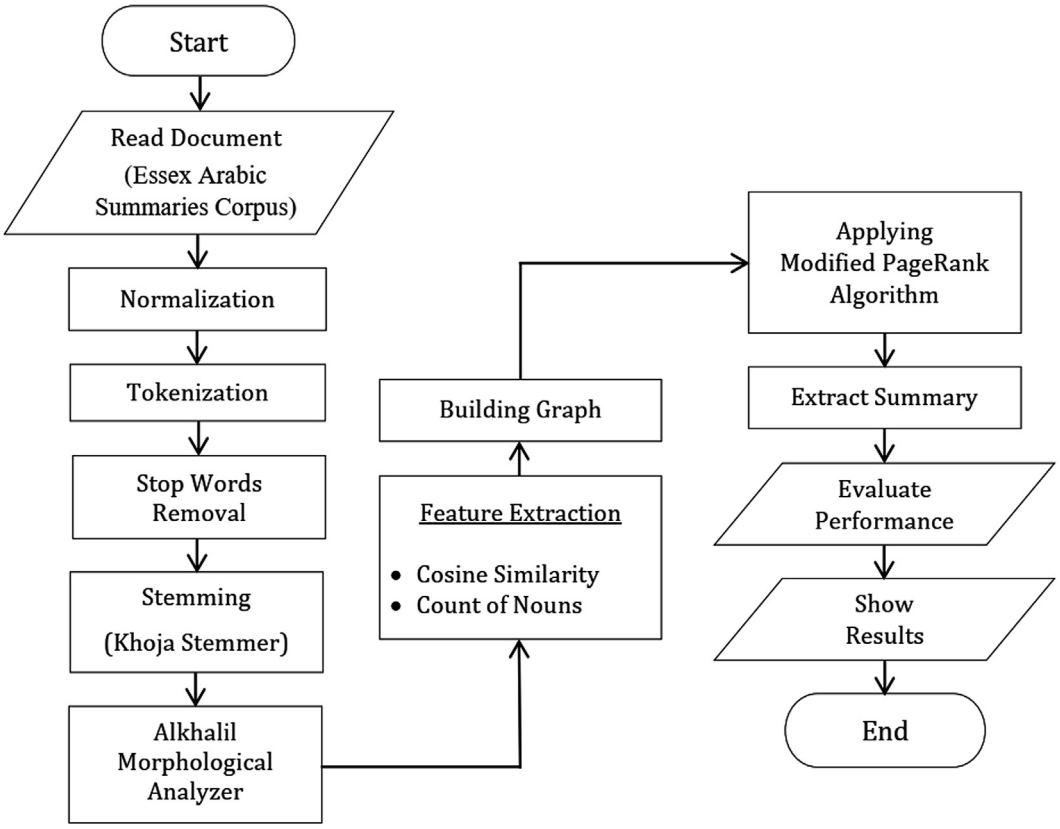


**Fig. 2.** Proposed approach.

| | |
|---|---|
| **S01** | نهر الأردن نهر يمر في بلاد الشام، يبلغ طوله 360 كلم ويتكون من ثلاثة روافد هي الحاصباني القادم من لبنان واللدان القادم من شمالي فلسطين وبانياس القادم من سوريا مخترقا سهل الحولة في الجولان حيث يصب في بحيرة طبرية التي تكونت جراء حدوث الشق السوري الافريقي وقد كون هذا الشق عدة بحار وبحيرات أخرى مهمة، يصب فيه روافد نهر اليرموك ونهر الزرقاء ووادي كفرنجة وجالوت، ويفصل النهر بين فلسطين والاردن إلى ان يصب في مياه البحر الميت المعروفة بملوحتها العالية. |
| **S02** | عرف هذا النهر منذ القدم باسم نهر الأردن ومعنى الأردن هو المياه السريعة التدفق بحسب اللغة اليونانية القديمة. |
| **S03** | وقد اعتقد بعض المؤرخون أن مجموعات من المقدونيين والحثيين تعايشت حول هذا النهر بعد أن نزحت من الجنوب الغربي من آسيا الصغرى ومن تساليا وذلك قبل العصر الهليني وأطلقت على النهر اسمه الخالد. |
| **S04** | وقد قال المؤرخ الروماني تاسيتوس أن جبل الشيخ هو والد نهر الأردن والذي يغذيه وفي القرن الرابع للميلاد كانت تقام في مغارة تقع على أحد جانبي جبل الشيخ عرفت باسم مغارة رأس النبع بالقرب من بانياس السورية تقام مراسم وثنية في يوم عيد معين كانت ترمى فيه ضحية مذبوحة وكانت هذه الضحية تختفي بطريقة عجيبة بقوة الشيطان. |
| **S05** | ويجتاز هذا النهر المستنقعات ويشكل بحيرة الحولة ثم بحيرة طبريا. |
| **S06** | وقد دارت رحى معارك كثيرة على ضفاف نهر الأردن مثل معركة اليرموك بين الروم والمسلمين والتي أنتصر فيها المسلمين نصراً عزيزاً. |
| **S07** | وحديثا في الثلث الأخير من القرن العشرين أحداث كمعركة الكرامة وحرب الاستنزاف وعديد من العمليات الفدائية للمقاومة الفلسطينية. |
| **S08** | حسب الإنجيل تعمد به السيد المسيح على يد يوحنا المعمدان. |
| **S09** | ونهر الأردن نهر مقدس عند المسيحيين لأن السيد المسيح قد تعمّد بمياهه في وادي الخرار على الضفة الشرقية. |

**Fig. 3.** Example of Arabic single document.

**Table 4**
Part of Speech Tagging example.

| و يجتاز هذا النهر المستنقعات ويشكل بحيرة الحولة ثم بحيرة طبريا | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| و | يجتاز | هذا | النهر | المستنقعات | و | يشكل | بحيرة | الحولة | ثم | بحيرة | طبريا |
| Stop word | verb | pronoun | noun | noun | Stop word | verb | noun | noun | Stop word | noun | noun |

morphological analysis and part of speech tagging. From Table 4 the sentence contains 6 nouns, 1 pronoun, 2 verbs and 3 stop words. This analysis is applied for all sentences in the document to calculate nouns in each sentence.

*7.2. Stage 2: Features extraction and building a graph*

In this stage the needed features are extracted, then the document is modeled as a graph.

*7.2.1. Features extraction.* In this stage, two types of features are extracted. In this step term is equal to the root of the word.

- *Cosine Similarity Between Two Sentences*

The cosine similarity is calculated as in Formula (7). TF-IDF for a term (word) in the sentence [23] is calculated as in Formula (3), (4) and (5). Where "$t$" is the specified word, TF is term frequency which is the number of times this term appears in the document divided by the number of all terms in the document; is used to define the importance of this term in the document, IDF is inverse document frequency. IDF used to define the amount of information this term provided. IDF equals the log of number of sentences where this term appears divided by the number of all sentences in the document. The TF-IDF for the sentence is the summation of TF-IDF for every word in this sentence as shown in Formula (6). Formula (7) is used to compute the cosine similarity between sentence ($Si$) and ($Sj$). where "$m$" is the count of mutual words between the two sentences and "$k$" is the offset of the word in the mutual list. TF-IDF($t_{ik}$): is the TF-IDF of the term number "$k$" in the mutual list in ($Si$), TF-IDF($t_{jk}$): is the TF-IDF of the term number "$k$" in the mutual list in ($Sj$). in another word to calculate the cosine similarity between sentence ($Si$) and ($Sj$) we do the following steps: (1) calculate TF-IDF for every single term in the both sentences according to Formula (5). (2) find the list of mutual words between the two sentences, the length of this list is "$m$". (3) we iterate on the mutual list and applying Formula (5).

$$\mathrm{TF}(t) = \frac{\text{Number of occurrences of term } t \text{ in document}}{\text{Total number of all terms in the document}} \tag{3}$$

$$\mathrm{IDF}(t) = log\left(\frac{\text{Number of all sentences in the document}}{\text{Number of sentences containing the term } t}\right) \tag{4}$$

$$\text{TF-IDF}(t) = \mathrm{TF}(t) * \mathrm{IDF}(t) \tag{5}$$

$$\text{TF-IDF}(s) = \sum_{t \in s} \text{TF-IDF}(t) \tag{6}$$

$$\text{Cosine\_Similarity}(Si, Sj) = \frac{\sum_{k=1}^{m} \text{TF-IDF}(t_{ik}) * \text{TF-IDF}(t_{jk})}{\sqrt{\sum_{k=1}^{m} \text{TF-IDF}(t_{ik})^2} * \sqrt{\sum_{k=1}^{m} \text{TF-IDF}(t_{jk})^2}} \tag{7}$$

- *Counting of nouns resulting from the morphological analysis step in each sentence*

This feature is used as an initial rank for each sentence. Table 4 shows an example in morphological analysis step i.e. how nouns are extracted from the sentences, which lead to the number of nouns in each sentence.

*7.2.2. Building graph and weighting.* The document is modeled as a graph as in Fig. 1. Sentences represent the vertices. Each two vertices are connected with edge, which has a weight equal to cosine similarity as in Formula (7).

*7.3. Stage 3: Applying Modified PageRank and summary extraction*

In this stage the Modified PageRank algorithm is applied, then the summary is extracted.

*7.3.1. Applying Modified PageRank.* In this step Modified PageRank algorithm is applied with initial rank for every sentence equals its own count of nouns. The PageRank is applied with different number of iterations 10, 100, 1000, 10,000, 100,000, 1000,000. These different numbers of iterations are used to get the best number of iterations to reach the best performance.

*7.3.2. Summary extraction.* In this step nodes are sorted depending on its final rank. Sentences are extracted one by one and added to the summary until the compression ratio is reached, if the overlapping between the selected sentence and any other sentence in the summary is very high then, this sentence is neglected to prevent redundancy.

*7.3.3. Removing redundancy.* In this step after summary is extracted, redundant sentences are removed from the summary. The redundant sentences is recognized depending on sentence overlapping. when the overlapping between sentences is greater than 90% the last sentence is removed from the summary.

*7.3.4. Choosing and comparing the pre-generated summary files.* In this step a pre-generated summary is chosen from the corpus to evaluate the performance of the summary. There are five pre-generated summaries in the corpus. The resulting summary is compared with them. Algorithm 2 shows the pseudo code for the proposed approach that starts with reading document, then pre-processing, features extraction and graph building, the PageRank, summary extraction, and removing redundancy stages are applied.

**Algorithm 2. Proposed approach algorithm**

```
    Input: Entire Single Document.
    Output: Output Document.
1   Configure/Set the Maximum Sentences in the
    Summary ← Total Sentences in Document.
2   Morphological Analyzer (Alkhalil)
3   Graph ← New Graph ()
4   Foreach Sentence: Document.Sentences
5     Normalization ()
6     Tokenization ()
7     StopWordsRemoval ()
8     Stemming ()
9     S_TF-IDF←Calculate Sentence TF-IDF ()
10    S_Noun_List ← Applying Morphological Analyzer & Get
      Nouns List ()
11    New_Node ← CreateGraphNode(S_TF-IDF, S_Noun_List)
12    Graph.add (New_Node)
13  Foreach Node: Graph.Nodes
14      If (Node <> New_Node)
15        Cosin_Similarity← Cosine_Smilarity(Node, New_Node)
16        Nouns_Measure ← Noun_Calc (Node)
17        Graph.CreateEdge (Node, New_Node, Cosine_Similarity)
18  Foreach_sentence_set_the_number_of_its_nouns_as_initial_
    rank()
19  Apply_Page_Rank ()
    Summary←Extract_Summary(Compression_Ratio)
    Summary←Removing_Reduandancy(Summary)
20  OUTPUT ← Summary
```

## 8. Experimentation and results

### 8.1. Dataset (Corpus)

To evaluate the proposed approach, the Essex Arabic Summaries Corpus (EASC) is used as a standard corpus, the corpus contains 153 documents, and each document has 5 summaries, with a total of 765 Arabic human-made summaries generated using Mechanical Turk (Mturk) [24]. EASC includes 10 subjects: art, music, environment, politics, sports, health, finance, science and technology, tourism, religion, and education. The system extracts six summaries for each document according to the number of iterations, which include 10, $10^2$, $10^3$, $10^4$, $10^5$ and $10^6$ iterations.

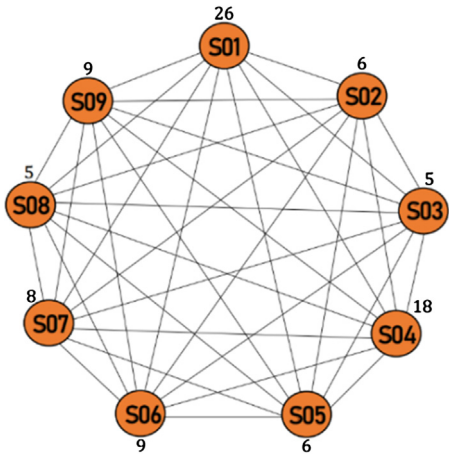


**Fig. 4.** Graph with initial ranks.

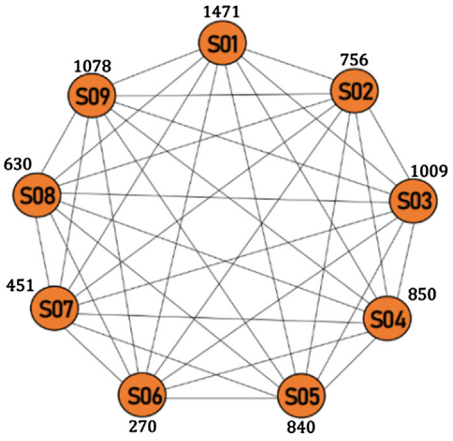


**Fig. 5.** The graph after applying Modified PageRank algorithm.

### 8.2. Evaluation metrics

The evaluation is calculated with respect to precision, recall and F-measure. The value of Precision recall and F-measure will be calculated as in Formula (8), Formula (9) and Formula (10) respectively.

- *Precision*: To metric the correct text size that is returned by the system.

$$\text{Precision} = \frac{\text{Extracted Summary} \cap \text{Provided Summary}}{\text{Extracted Summary}} \tag{8}$$

- *Recall*: The metric coverage system reflects the ratio of the extracted relevant sentences.

$$\text{Recall} = \frac{\text{Extracted Summary} \cap \text{Provided Summary}}{\text{Provided Summary}} \tag{9}$$

- *F-measure*: Makes a balanced relation among recall metric and precision metric.

$$\text{F-measure} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

### 8.3. Experiment setup

This sub-section shows an example of how modifying PageRank works. Fig. 4 shows the document modeled as a graph, each node in the graph takes its initial rank. Fig. 3 shows an Example of Arabic single document sentences that are numbered and listed in the table. Fig. 5 shows the graph after applying Modified PageRank algorithm with 10,000 iterations, each node in the graph has its new weight.

After applying Modified PageRank algorithm. Fig. 6 shows the final ranks of graph nodes, according to the figure nodes that represent sentence 01, 03, 09 return the highest rank so it's included in the summary.

### 8.4. Results discussion and analysis

This subsection discusses the system results and compares between this approach and other approaches results. Table 5

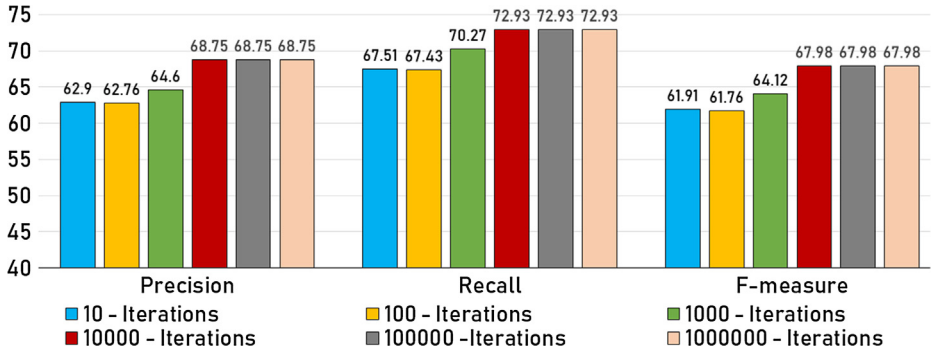


**Fig. 6.** Performance evaluation compared with other research.

shows the algorithm results with multiple number of iterations. According to the results listed in the table algorithm performance is increased when the number of iterations is increased until it reaches 10,000 iterations, after reaching 10,000 iterations or higher, the performance stabilizes. In PageRank algorithm every iteration, each vertices (sentence) has a new calculated weight that depends on the initial rank of the node and the its connected node, alongside with the weight of the edge connecting the nodes. In every iteration, the most important node weight increases faster than the other nodes, so, on the step of extracting the summary these nodes come first.

Fig. 6 shows the performance metrics of different number of iterations, according to the figure when the number of iterations is 10,000 or higher, the performance reaches its maximum and stabilizes.

Table 6 and Fig. 7 show the comparison between the current research results with other researches results. this table contains three results from our works applied using three different algorithms Modified PageRank, LexRank [19], and TextRank [20]. The results were compared with two other researches that uses the same data set used in this research which is EASC. The compared researches are with the following titles: research (1) applying semantic and analysis for ATS [11] this research uses EASC as data set. And research 3 used graph-based Arabic text summarization approach with PageRank [15] which also uses EASC as data set.

**Table 5**
Evaluation results of the algorithm with multiple number of iterations.

| # of Iterations | Precision | Recall | F-measure |
|---|---|---|---|
| 10 | 62.90 | 67.51 | 61.91 |
| 100 | 62.76 | 67.43 | 61.76 |
| 1000 | 64.60 | 70.27 | 64.12 |
| 10,000 | 68.75 | 72.93 | 67.98 |
| 100,000 | 68.75 | 72.93 | 67.98 |
| 1,000,000 | 68.75 | 72.93 | 67.98 |

**Table 6**
Comparison with others works.

| Methods | Precision | Recall | F-measure |
|---|---|---|---|
| LexRank | 51.03 | 56.5 | 50.6 |
| TextRank | 50.88 | 56.22 | 49.81 |
| Statistical and Semantic Analysis [11] | 57.62 | 58.80 | 58.20 |
| PageRank Algorithm [15] | 54 | 47 | 51 |
| This Proposed Method | 68.75 | 72.94 | 67.99 |

The comparison shows that the results returned from the Modified PageRank is better than PageRank, LexRank and TextRank algorithms. When the number of iterations is changed the performance is enhanced until the number of iterations reaches 10,000 after that the performance stabilizes. Also the performance is enhanced because of using Alkhalil morphological analyzer which returns the best analysis of nouns.

## 9. Conclusion

Arabic language suffers from the problem of finding the noun in the sentences because of the absence of capital and small letters in Arabic in addition to the absence of diacritics in the written text. So morphological analyzers are used. This research tries to enhance the performance of the generated summaries by applying Modified PageRank algorithm with multiple number of iterations. Al-Khalil morphological analyzer is used to overcome the problems of Arabic structure complexity and to extract nouns to use it in the process of building the graph as initial rank, and cosine similarity was used to weigh the edges between sentences. In summary extraction to prevent redundancy, if the overlapping between the selected sentence and any other sentence in the summary is very high then, this sentence is neglected. A Modified PageRank algorithm was used to extract the summary, this algorithm was used by making the initial rank of the sentence as the number of nouns it has, and the weight of the edge is the cosine similarity between the connected nodes. LexRank, TextRank and Modified PageRank were used under the same circumstances to extract the summary and evaluate the performance of the system.

The process of summarization starts by reading the documents, then normalizing data, removing stop words, stemming, morphological analyzer then finally applying the graph and getting the summary. EASC is used as a standard corpus in the testing stage. According to the results, the Modified PageRank returns better results when the number of iterations used is equal to 10000. According to previous research, this research returns better results than the other the final performance of this research is 67.98 in F-measure.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

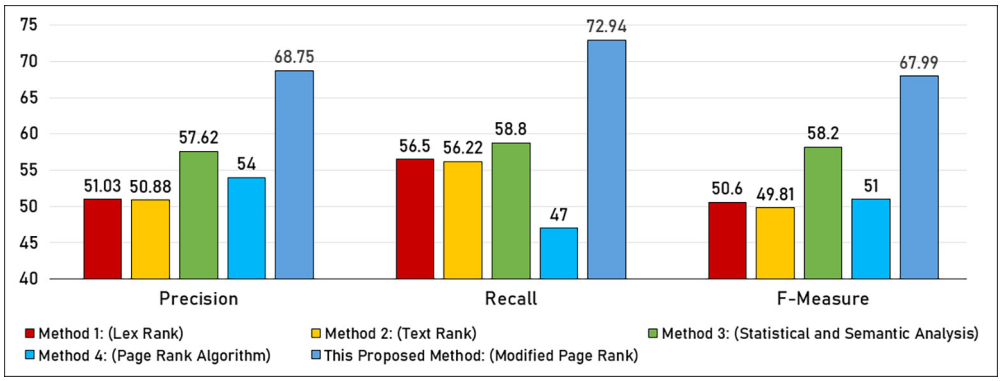


**Fig. 7.** Performance evaluation compared with other research.

## References


[1] Lloret E. Text summarization based on human language technologies and its applications. J Nat Lang Process 2011;48:119–22.
[2] Al-Saleh AB, Menai MEB. Automatic Arabic text summarization: a survey. Artif Intell Rev 2016;45(2):203–34.
[3] Al Qassem LM, Wang D, Al Mahmoud Z, Barada H, Al-Rubaie A, Almoosa NI. Automatic Arabic summarization: a survey of methodologies and systems. Proc Comput Sci 2017;117:10–8.
[4] Al-Shalabi R, Kanaan G, Al-Sarayreh B, Khanfar K, Al-Ghonmein A, Talhouni H, et al. Proper noun extracting algorithm for arabic language. In: International Conference on IT to Celebrate S. Charmonman's 72nd Birthday. p. 1–28.
[5] Al-Radaideh Q, Afif M. Arabic text summarization using aggregate similarity. International Arab conference on information technology (ACIT2009), 2009.
[6] Boudlal A, Lakhouaja A, Mazroui A, Meziane A, Bebah M. O. A. O, Shoul M. Alkhalil morpho sys1: A morphosyntactic analysis system for arabic texts. In International Arab conference on information technology. Benghazi Libya, 2010; p. 1–6.
[7] Page L, Brin S, Motwani R, Winograd T. The PageRank citation ranking: Bringing order to the web. Stanford Info Lab; 1999.
[8] AlSanie W, Touir A, Mathkour H. Towards an infrastructure for Arabic text summarization using rhetorical structure theory Master's thesis. Riyadh: King Saud University; 2005.
[9] Alami N, Meknassi M, Ouatik SA, Ennahnahi N. Impact of stemming on Arabic text summarization. In Information Science and Technology (CiSt). In: 2016 4th IEEE International Colloquium on. p. 338–43.
[10] El-Harby AA, El-Shehawey MA, El-Barogy R. A statistical approach for Qur'an vowel restoration. ICGST-AIML J 2008;8(3):9–16.
[11] Alami N, El Adlouni Y, En-nahnahi N, Meknassi M. Using statistical and semantic analysis for Arabic text summarization. In: International Conference on Information Technology and Communication Systems. Cham: Springer; 2017. p. 35–50.
[12] Ibrahim A, Elghazaly T, Gheith M. A novel arabic text summarization model based on rhetorical structure theory and vector space model. Int J Comput Linguist Nat Lang Proces 2013;2(8):480–5.
[13] Azmi AM, Al-Thanyyan S. A text summarizer for Arabic. Comput Speech Lang 2012;26(4):260–73.
[14] Al-Abdallah RZ, Al-Taani AT. Arabic single-document text summarization using particle swarm optimization algorithm. Proc Comput Sci 2017;117:30–7.
[15] Al-Taani AT, Al-Omour M. An extractive graph-based arabic text summarization approach. The International Arab Conference on Information Technology, 2014.
[16] Alami N, Meknassi M, Ouatik SA, Ennahnahi N. Arabic text summarization based on graph theory. In: 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA). IEEE; 2015. p. 1–8.
[17] Malallah S, Ali ZH. Multilingual Text Summarization based on LDA and Modified PageRank. Iraqi Journal of Information Technology 2019;9 (3):139–60.
[18] Al-Abdallah RZ, Al-Taani AT. Arabic Text Summarization using Firefly Algorithm. In 2019 Amity International Conference on Artificial Intelligence (AICAI) - IEEE 2019:61–5.
[19] Erkan G, Radev DR. Lexrank: Graph-based lexical centrality as salience in text summarization. J Artificial Intell Res 2004;22:457–79.
[20] Mihalcea R, Tarau P. Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing. p. 404–11.
[21] Cohen R, Havlin S, Ben-Avraham D. Structural properties of scale free networks. Handbook of graphs and networks, 2003.
[22] Khoja S, Garside R. Stemming Arabic text. Lancaster, UK: Computing Department, Lancaster University; 1999.
[23] Abu-Errub A. Arabic text classification algorithm using TF.IDF and Chi square measurements. Int J Comput App IJCA 2014;93(6):40–5.
[24] El-Haj M, Kruschwitz U, Fox C. Using mechanical turk to create a corpus of Arabic summaries. In: Language Resources and Evaluation conference (LREC), May 17–23. p. 36–9.