JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 36, 347-363 (2020)
DOI: 10.6688/JISE.202003_36(2).0013

Graph-Based Extractive Arabic Text Summarization Using Multiple Morphological Analyzers

REDA ELBAROUGY[1], GAMAL BEHERY[1] AND AKRAM EL KHATIB[2]
[1]*Department of Computer Science*
[2]*Department of Mathematics*
Damietta University
New Damietta, 0020 Egypt
E-mail: {elbarougy; gbehery}@du.edu.eg; akram_elkhatib@hotmail.com

This paper investigates the effectiveness of using multi-morphological analysis for improving the performance of graph-based approach for extractive Arabic text summarization (ATS). This approach represents the text-document as a graph in which; sentences are the graph nodes and the relationships between the sentences are edges' weights of the graph. These weights measure the similarity between the relevant sentences which traditionally calculated using the cosine similarity on the basis of term frequency-inverse document frequency (TF-IDF). The performance of graph-based ATS is still low because calculating these weights are very challenging for Arabic language due to the following reasons: complex morphological structure of Arabic language, absence of capital letters and diacritics, and the change of the order of the words on the sentence. In this study, the summation of the cosine similarity and mutual nouns between the connected sentences is chosen as measure to represent the edges' weights. Nouns were chosen because, the more nouns in the sentence the more information is, thus we assume that using nouns lead to an improvement in the final summary. To overcome Arabic language limitations when calculating the proposed measure, it is required to investigate the impact of using different morphological analyzers for extracting nouns from each sentence on ATS accuracy. Three morphological analyzers algorithms are proposed to enhance the performance of graph-based ATS system. These algorithms are: BAMA, Safar Alkhalil and Stanford NLP. Firstly, graph-based ATS system was constructed the input of this system is text-document and the output are summary. Then redundant sentences were removed according to sentences overlapping criteria. To evaluate the impact of different morphological on the proposed summarization approach, EASC corpus is used as a standard dataset. The results show that Safar Alkhalil morphological analyzer gives the best performance among the three proposed analyzers.

Keywords: Arabic text summarization, morphological analyzer, natural language processing, graph based, minimum spanning tree

1. INTRODUCTION

Due to the huge amount of text documents and articles on the web, users need more time to obtain the important information included in that documents. To ensure saving their time and effort spent in dealing with the electronic documents, automatic text summarization is needed, which is one of most popular solutions proposed to save human time and efforts [1]. Automatic text summarization is the process of eliminating the non-informative sentences in the document and keeping only the informative sentences of the selected document. In general, text summarization process passes through three stages: analysis, transformation, and synthesis stage. In the analysis stage the inserted text is analyzed and salient

Received September 28, 2019; revised October 2&11, 2019; accepted October 17, 2019.
Communicated by Osamah Ibrahim Khalaf.

features are selected. Then the transformation stage transforms the output into a summary representation and prepares it for stage three. Finally, in the synthesis stage, the summary is built depending on the user's needs.

In the literature of text summarization, the categories of summaries depending on multiple factors such as the count of the input documents in summarization process, the structure of the extracted sentences in the generated summaries, and the number of sentences in the summary [2]. According to the number of documents summaries into two approaches [3]: single document approach, which inputs only one document to the summarization system, and multiple documents which apply summarization process into multiple documents to conclude one summary from them. Summaries also can be categorized depending on the output of the summary, the summarization could be indicative, which gives a brief idea of the original text, or it could be informative, which gives more detailed information [1]. Also depending on the goal and the output of the summary together, a summary could be query-focused summarization, where the content of the summary is driven by a user need, or generated using generic summarization, which gives relevant facts of the source text [4].

On the other hand, text summarization approaches can be divided into extractive and abstractive summarization depending on the type of sentences in the final summary. In extractive approach, the most important sentences are selected as the final summary without any changing on their structure. While, abstractive summarization applies some linguistic methods to present the text by its meaning rather than its structure. The abstractive summarization goal produces a generalized summary [1].

This study focused on the extractive summarization approach which has many techniques such as statistical technique by extracting statistical features for each sentence in the document such as term frequency and sentence position in the text as in [5]. Another technique in extractive text summarization is a hybrid technique, which combines between statistical and linguistic knowledge approaches as in [6], which done by using morphological analyzers to return the words in the sentences into their roots then applying some statistical features such as term frequency. A graph-based approach is another summarization approach, which represents the document as a graph with sentences in its vertices and uses some statistical features to calculate the initial weights of the edges between nodes, and then it applies some graph algorithm such as shortest path [7].

Arabic language is a very complex in its syntax, which makes it very difficult to deal with it. Arabic is written from right to left. Also, it lacks capital letters or small letters so we cannot determine whether a word is noun or not, and has letters that varies in shape according to their location in the word (beginning, middle, or end of the word), which adds some complexity to the language processing [8]. The complex morphology of Arabic is another feature that allows the writer to switch between the positions of words in the sentence while retaining the same meaning. In the written text, the absence of diacritics is another problem in Arabic, diacritics define the function of the word within the sentence [9, 10]. To overcome the complexity of Arabic language, morphological analyzers are used to improve the analysis of Arabic text leading to improve the performance of the summary. Three types of morphological analyzers BAMA, Safar Alkhalil and Stanford NLP were used in this paper to find the best results on the Arabic language summary.

In this research the process of summarization starts with reading a text from selected single document. After loading the text, a normalization process is applied to remove

punctuations, digits, diacritics and so on. Then, features extraction process takes place by extracting the needed features for weighting process, then the document is presented as a graph, the sentences are the vertices of the graph, then applying Minimum Spanning Tree (MST) algorithm to calculate the ranks of sentences by building the spanning tree that contains all sentences in hieratical structure, then the summary is extracted according to predefined compression ratio, and the redundant sentences are removed depending on the overlapping between the sentence and the extracted summary [11]. The summary yielded from this approach is an extractive text summarization approach. This paper, examine the impact of using three types of morphological analyzers on the generated summary. Morphological analyzers affect edges weight as presented on the graph by varying the number of the detected nouns according to the type applied of morphological analyzers [9].

The paper follows the following structure; motivation and problem of statement in Section 2, related works are reviewed in Section 3. In Section 4, Arabic morphological analysis. Section 5 discusses MST. The proposed approach is discussed in Section 6. The experimental results are shown in Section 7. Finally, the paper is concluded in Section 8.

2. MOTIVATION AND PROBLEM OF STATEMENT

There is a lack in the previous researches, which depends on the roles of words in the sentence. In general, researches done in Arabic text summarization still have low performance. Arabic is a very complex language in syntax, and there is no simple way like English to determine if a word is a noun or not because there are no uppercase letters like those in English. In Addition, the absence of diacritics in the written text is another problem in Arabic because nowadays most writers ignoring putting it. From the other hand, nouns make sentence more informative. More nouns in the sentence mean more information. Therefore, to determine whether the word is a noun or not the morphological analysis is the solution. Therefore, this research attempts to use a new technique to summarize the text by using minimum spanning tree algorithm for ranking the sentences and apply three different types of morphological transformers to find the effects of them on the summary and to improve the performance of the resulting summary.

3. RELATED WORK

Since the second half of the previous century, many researches have focused on text summarization. Summarization has different approaches, each of which has different categories depending on specific characteristics. Douzidia [5], suggested a form of text summarization applying various criteria to get the weight of sentences. A position, combination of frequency, and indicative expression was used to give a conclusion for the sentence [5]. Also, Mani & Maybury [12] describes the text summarization as “a process of finding the main source of information, the main important contents and presenting them as a concise text in the predefined template”.

Earlier researches depended on statistical methods to calculate the weight of sentences in the document. Among these is Lin and Hovy [2], who proposed a statistical model of text summarization using different criteria to calculate the weight of sentences. A formula that combines between frequency, position, and indicative expression as used to give the sen-

tence a score. Then according to the score and compression ratio the summary is extracted.

Extractive text summarization is another approach that makes use of linguistic and morphological methods as in Sawalha and Atwell [6], whose approach uses different types of Arabic morphological stemmers and analyzers. They found that Khoja stemmer has been more accurate than the other analyzers in the words having three characters' root which constitute 80-85% of Arabic words. While the rest of the words are formed from four, five and six letters roots [6]. Their result agrees with Alami *et al.* [9], Light [11] and Safar Alkhalil on the generated text summarization. Therefore, the current research uses Khoja stemmer because of the high accuracy rate for words with three letters roots and also performs well in four letters roots. Alami *et al.* [9], is an extractive approach depends on statistical based. One of the drawbacks of [9] is that they did not use a standard corpus to compare with them, they collected 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.

Another research is Lagrini *et al.* [13]. In this approach, they proposed to split original input text into non-overlapping elementary discourse units. Then to identify the rhetorical relations among these units. After that to build RST-tree by using two kinds of RS-tree-building strategies greedy strategies and non-greedy strategies. Finally, to calculate the sentences scores and generate the summary.

Later the hybrid approach that combines between statistical and morphological algorithms was used. One of hybrid approach researches is Hadni *et al.* [14], they proposed a combination between Latent Semantic Analysis (LSA) and Arabic Word Morphological model. The combined model is used as a hybrid approach for Arabic multiword term extraction. The research starts by selecting the sentences, then eliminating the repeated sentences. The eliminated sentences are identified depending on three techniques word, root and stem.

Another research is Jaradat and Taani [15], they built an extractive single document ATS approach based on genetic algorithm. This approach used term frequency (TF) and inverse document frequency (IDF) as features and then applied genetic algorithm for sorting and extracting the summary. They evaluated the proposed approach using EASC corpus evaluation metrics.

In addition, the graph-based approach was used, where the graph vertices are the document sentences. Many researches, such as Mihalcea [3], proposed an extractive graph-based algorithm for ranking sentences in the document for summarization process. He evaluated his application to extract unsupervised sentence in the context of a text summarization task. The results obtained from this new unsupervised method did not comply with the most advanced systems. The data set used for evaluating this approach is DUC 2002, where the F-measure is about 50.08%. Different research is Patil and Brazdil [16], they proposed a Sum-Graph technique which is a theoretic graph technique applied on a single document to extract the summary. The Document's sentences were represented as graph nodes. The weights of the edges between graph nodes were represented by calculating intra-sentence dissimilarity between sentences. Also, Alami *et al.* [17], 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, if the similarity between two nodes is less than some threshold he assumes that these two sentences are not connected. 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. [15] assigned 1 as initial rank for all the sentences in the graph then he iterates on the graph until the difference between the new rank and old rank is 0.001 for all nodes. [15] 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. However, [7] proposed an Arabic extractive graph-based text summarization. The researcher used several basic units such as stem and n-gram in the summarization process. The summary is extracted using shortest path algorithm, this approach used EASC corpus for the evaluation process and had an F-measure of 51%. Finally, Belkebir and Guessoum [18], proposed an approach that uses multi-graph represented by deferent metrics depending on number theory and probabilities to calculate the importance of the text partition. Malallah and Ali [19] 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 is 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 Al-Taani [20] proposed a single document graph-based approach using the Firefly algorithm for the extraction of summaries, EASC is used to measure the performance of the summary. Elbarougy *et al.*, [21] proposed an Extractive Arabic Text Summarization approach using the Modified PageRank Algorithm. The researchers represented the document as a graph then making the initial rank for each node is the number of nouns on it the weight of the edge is the cosine similarity between sentences. The PageRank algorithm was applied to about 10000 iterations, then the sentences were ordered according to its final rank. then finally the summary extracted according to the pre-defined compression ratio and the redundant sentences are removed from the summary. Elbarougy *et al.*, [22] discussed the effects of using natural language processing techniques on the performance of the Arabic language summarization process. In this research, they mainly discussed the effects of removing stop words from the text in the pre-processing stage. The document is represented as a graph, then the summary is extracted. Researchers found that removing stop words from the text improves the performance of the summary.

All previous researches results had low performance. Therefore, the current paper tries to improve the performance of the summarization process by using MST with different types of morphological analysis techniques.

4. ARABIC MORPHOLOGICAL ANALYSIS

Arabic language has one of the most complex morphological systems [8]. For example, there is no capital and small letters in Arabic language so we cannot distinguish between nouns and non-noun words. Another problem in Arabic sentence is that it can start with a noun or a verb. Arabic language morphology is based on the basic pattern of forming words. Therefore, most of the native Arabic words are derived from basic roots or stems. Roots generation depends on a predefined list of patterns called morphological balances or patterns. Each Arabic word is formed by using its root or by adding suffixes or prefixes to its root. Each word in the sentence has its own Part of Speech (POS) [19].

Different approaches for Arabic morphological analyzers are implemented. This paper depends on three types of morphological analyzers.

4.1 Buckwalter Arabic Morphological Analyzer (BAMA) [20]

One of the earliest Arabic Morphological Analyzers. It depends on three Arabic-English lexicon files: prefixes, suffixes, and stems. It contains a truth table to indicate a correct combination of these three segments and offers morphological categories such as Nouns, Function word, and Verbs. In addition, it uses Buckwalter transliteration, which may be converted directly to Unicode Arabic with least amount of automatic processing. To use BAMA [20], you should translate Arabic words to ASCII then apply BAMA then after applying the analyzer you should reverse the process.

4.2 Safar Alkhalil [21]

Integrates the morphological Arabic analyzers and saves the output as xml file. Every morphological analyzer processes the input text then saves it into an xml file. The results retrieved as memory objects, measure the performance of a given morphological analyzer. This analyzer depends on finding all possible vowelized forms that belong to the current word then divides the word into about 5 parts enclitic, suffix, lemma, prefix, and proclitic [21].

4.3 Stanford NLP [22]

A project developed by Stanford University [22]. It contains libraries and algorithms for processing multiple human languages; the most important features provided by this group are: tokenization which tokenizes the text into a sequence of tokens, POS which labels tokens with their part of speech tag. Arabic parsers based on the Penn Arabic Treebank.

5. MINIMUM SPANNING TREE (MST)

Graph is a mathematical structure used to represent the pairwise relation between objects. The graph is construct of vertices or nodes representing the objects connected with each other by edges containing the value of the relation between these two nodes. There are two major types of graphs: directed and undirected graphs. In undirected graphs the direction of the edge does not affect the weight of that edge [11].

A spanning tree of an undirected graph is a sub-graph with no cycle which includes all of the vertices of the graph. In general, a connected graph may have several spanning trees, so the term MST [11], appear which is a spanning tree with the minimum possible total edges weight. Algorithm 1 shows how MST works. While (G) is a graph, (E) is the edges in the graph and (M) represents the MST. The algorithm starts by finding an edge that is not forming a cycle in the graph cycle means; that edge starts from node then goes through multiple nodes until finally it returns to the same node. Moreover, it continues doing this until forming the spanning tree. Algorithm 1 shows the algorithm for MST algorithm [11].

Algorithm 1: Basic algorithm for MST

Input: A weighted, undirected graph $G = (V, E, w)$.
Output: MST (M).

```
1    M equals {}
2    While M Does Not Form a Spanning Tree:
3        Find the Minimum Weighted Edge e(u, v) in E is safe for M
4        M equals M union {e(u, v)}
```

6. THE PROPOSED APPROACH

This paper investigates the effects of using multi-morphological analyzer in the process of extracting ATS by using MST.

Fig. 1 shows the steps of the proposed approach:

6.1 Input Single Document

In this step, the system reads a single document written in Arabic language and encoding in utf-8, by extracting only the text from it.

6.2 Normalization

This step can be considering as pre-processing step. In this step punctuations and digits along with any none alphabet characters are removed from the sentence. Also, the first ALEF in every word is replaced by "ا" and replace The Marbota in the end of each word by Heh "ـــة" >>> "ـــه".

6.3 Tokenization

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

6.4 Removing Stop Words

Stop words or functional words are the words that repeated in the text to make the sentences more readable and understandable. Removing stop words reduces text to contain the useful words. And the non-removal affects the efficiency of the process of weighting.

6.5 Stemming

In this process Khoja [19], stemmer is used to get the root of every words in the sentence; every word is returned to its tri-literal root. This process is used to reduce the number of distinct terms in the document to make better term frequency calculation.

6.6 Morphological Analysis

In this step every word in the sentence takes a tag representing its part of speech position in the sentence. The position of the words may be a noun, verb, preposition, stop word article, *etc*. This research uses three types of morphological analyzers which discussed with more details in Section 4. These types are BAMA, Safar Alkhalil and Stanford NLP. In this process every word in the sentence takes a tag that represent its POS position in the sentence. Sentences that have the highest number of nouns takes rank higher than the others.

6.7 Features Extraction

In this step the features used in the process of weighting the graph are extracted. Here there are two features as follow:

6.7.1 Cosine similarity between two sentences

This characteristic is usually performed after stemming and stops words removal, the similarity is retrieved through getting (TF-IDF) [23] and the mutual words between two sentences. Eq. (1) illustrates the computation of (Term Frequency) in the term (t) here is the word after doing stemming process. Eq. (2) illustrates the computation of (Inverse Document Frequency) of the term (t). Eq. (3) shows TF-IDF calculation of the term (t) in the document by multiplying TF(t) with IDF(t). Eq. (4) illustrates the computation of (TF-IDF) of sentence (s) by calculating the summation of (TF-IDF) of each term (t) in this sentence according to Eq. (3). To compute the cosine similarity between sentence (S_i) and (S_j), Eq. (5) is used. 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 (S_i), TF-IDF(t_{jk}): is the TF-IDF of the term number "k" in the mutual list in (S_j). In another word to calculate the cosine similarity between sentence (S_i) and (S_j) we do the following steps: (1) calculate TF-IDF for every single term in the both sentences. (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 Eq. (5). For Eqs. (1)-(5), TF: is term frequency which is how many 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 the 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.

$$\mathrm{TF}(t) = \frac{\text{Number of occurrences of term } t \text{ in document}}{\text{Total number of all terms in the document}} \tag{1}$$

$$\mathrm{IDF}(t) = \log\left(\frac{\text{Number of all sentences in the document}}{\text{Number of sentences containing the term } t}\right) \tag{2}$$

$$\mathrm{TF-IDF}(t) = \mathrm{TF}(t) * \mathrm{IDF}(t) \tag{3}$$

$$\mathrm{TF-IDF}(s) = \Sigma_{t \in s} \mathrm{TF-IDF}(t) \tag{4}$$

$$\mathrm{Cosine\_Similarity}(S_i, S_j) = \frac{\sum_{k=1}^{m} \mathrm{TF-IDF}(t_{ik}) * \mathrm{TF-IDF}(t_{jk})}{\sqrt{\sum_{k=1}^{m} \mathrm{TF-IDF}(t_{ik})^2} * \sqrt{\sum_{k=1}^{m} \mathrm{TF-IDF}(t_{jk})^2}} \tag{5}$$

The count of nouns in each sentence that results from morphological analysis step:

Eq. (6) shows the calculation of nouns measure by getting the count of mutual nouns between two sentences then divided it by the total number of nouns in the two sentences.

$$\mathrm{Nouns\_Measure}(S_i, S_j) = \frac{\left|\mathrm{Nouns\_List}(S_i) \cap \mathrm{Nouns\_List}(S_j)\right|}{\left|\mathrm{Nouns\_List}(S_i) \cup \mathrm{Nouns\_List}(S_j)\right|} \tag{6}$$

Eq. (7) shows the calculation of the similarity between the sentences, which depends on Eqs. (5) and (6) to calculate cosine similarity and nouns measure respectively. The result of Eq. (7) will used as the edge weight which connect between these two sentences in the graph.

$$\text{Similarity}(S_i, S_j) = \text{Cosine\_Similarity}(S_i, S_j) + \text{Nouns\_Measure}(S_i, S_j) \quad (7)$$

6.8 Building Graph and Weighting

In this process the document is represented as a graph, where sentences represent the vertices of the graph with an edge connecting between every two nodes. The weight of the edge is the cosine similarity multiplied by the number of mutual names between these two sentences.

6.9 Apply (MST)

In this step MST is applied. A spanning tree is construct from the graph built in the previous step. The spanning tree contains the whole nodes in the graph which represented in hierarchical form. The spanning tree starts with the first sentence in the document as a root node.

6.10 Summary Extraction

In this step, the nodes are sorted descending according to the number of children nodes belongs to the node. Then the top sentences are extracted according to predefined compression ratio. If there is a high overlapping between any two sentences in the summary the last sentence is removed to prevent redundancy.

6.11 Choose the Pre-generated Summary Files and Compare with It

In this step, the pre-generated summaries are chosen from the corpus. There are five pre-generated summaries in the corpus. The resulting summary is compared with them.

Algorithm 2 shows the proposed approach in this paper. The input document is fed to the system then it goes through the process of summarization by doing NLP then extracting the needed features and building the graph, after graph is built MST is applied then finally the summary is extracted.

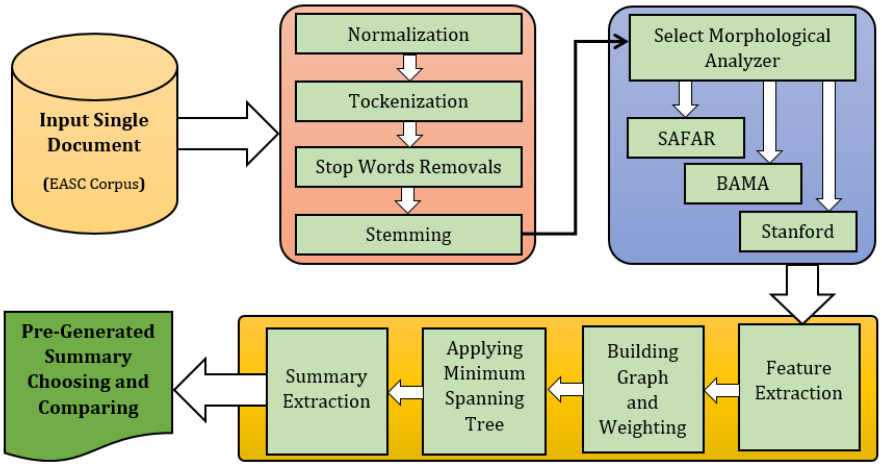

Fig. 1. Proposed approach.

Algorithm 2: Proposed Approach for Text Summarization

Input: Entire Single Document.
Output: Output Document.

```
Collection the Max Sentences in the Summary ← Total Sentences in Document.
Document_Category ← Detect Document Category.
Stop_Words_List ← Choose Stop Words List (Document_category)
Choose Morphological Analyzer (BAMA, Safar Alkhalil, and Stanford NLP)
Graph ← New Graph ()
Foreach Sentence: Document.Sentences
    Normalization ()
    Tokenization ()
    StopWordsRemoval (Stop_Words_List)
    Stemming ()
    S_TFIDF ← Calculate Sentence TF-IDF ()
    S_Noun_List ← Applying Morphological Analyzer & Get Nouns List ()
    New_Node ← CreateGraphNode (S_TFIDF, S_Noun_List )
    Graph.add (New_Node)
    Foreach Node: Graph.Nodes
        If (Node <> New_Node)
            Cosine_Similarity ← Cosine_Similarity (Node, New_Node)
            Nouns_Measure ← Noun_Calc (Node, New_Node)
            Edge_Weight = Cosine_Similarity + Nouns_Measure
            Graph.CreateEdge (Node, New_Node, Similarity, Edge_Weight)
Apply_MST ()
OUTPUT ← Extract_Summary (Compresion_Ration)
```

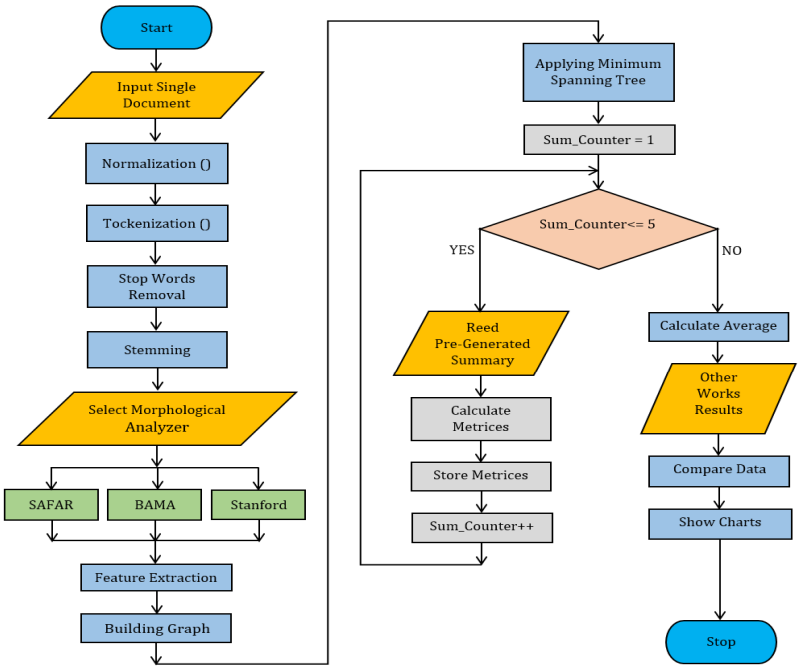

Fig. 2. Proposed approach flow chart.

Fig. 2 presents the flow chart of the proposed approach which starts with inputting the proposed document, then normalizing the text by removing digits, punctuations and characters. In the next stage removing stop words, and stemming take place. Moreover, morphological processing takes place by doing three types of morphological analyzers BAMA, Safar Alkhalil and Stanford NLP. Then, the document displayed as a graph (G) (v, u) with G: graph, v: a set of graph nodes representing the document sentences, u: is the edges that connects between graph nodes. In next step MST is applied and summary is generated. Finally, the pre-generated summaries are selected, calculating the measurement metrics and storing it for comparison processes.

7. EXPERIMENTATION AND RESULTS

7.1 Dataset (Corpus)

To evaluate the proposed approach, the EASC is used as a standard corpus, the corpus contains 153 documents, with 5 summaries for each document, and with total of 765 Arabic human-made summaries [24]. EASC includes 10 subjects: art and music, environment, politics, sports, health, finance, science and technology, tourism, religion, and education. The system extracts three summaries for each document according to the selected morphological analyzer.

7.2 Evaluation Metrics

The evaluation calculates with respect to Precision, Recall and F-measure. The value of Precision recall and F-measure will be calculated as in Eqs. (8)-(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 of the coverage of the system. It reflects the ratio of relevant sentences that the system extracted.

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

• **F-measure:** Works a balance relation among recall metric and precision metric.

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

7.3 Results Discussion and Analysis

Table 1 shows the results of the system depending on document category and the type of morphological analyzer. According to average results, Safar Alkhalil morphological achieved the best precision; Stanford NLP was the best results in the Recall metrics.

BAMA, represented the best precision in science, technology and sports and Stanford NLP got the best precision in tourism and religion.

Table 1. Metrics results.

Document Category	Morphological Type	Precision	Recall	F-measure
Science and Technology	BAMA	61.86	66.67	62.37
	Safar Alkhalil	60.57	64.01	61.56
	Stanford NLP	60.85	66.82	62.11
Education	BAMA	71.35	79.08	73.12
	Safar Alkhalil	73.80	82.67	76.37
	Stanford NLP	69.03	87.28	76.18
Health	BAMA	66.25	72.26	67.49
	Safar Alkhalil	64.87	70.05	65.54
	Stanford NLP	66.46	71.89	67.57
Sport	BAMA	68.39	62.73	64.05
	Safar Alkhalil	67.22	61.94	63.05
	Stanford NLP	65.12	67.18	64.48
Art and Music	BAMA	64.61	72.92	67.14
	Safar Alkhalil	65.42	73.54	67.91
	Stanford NLP	65.13	73.96	67.82
Environment	BAMA	60.31	63.68	60.60
	Safar Alkhalil	61.58	63.92	61.49
	Stanford NLP	60.41	65.21	61.45
Finance	BAMA	72.90	78.02	73.46
	Safar Alkhalil	73.09	80.97	74.98
	Stanford NLP	70.31	79.97	73.01
Politics	BAMA	63.81	74.60	67.17
	Safar Alkhalil	64.32	78.30	69.17
	Stanford NLP	62.87	75.35	67.15
Religion	BAMA	66.63	71.00	67.59
	Safar Alkhalil	67.59	71.92	68.22
	Stanford NLP	68.38	70.85	68.10
Tourisms	BAMA	66.26	73.23	68.02
	Safar Alkhalil	65.49	73.20	67.71
	Stanford NLP	67.13	71.89	67.83

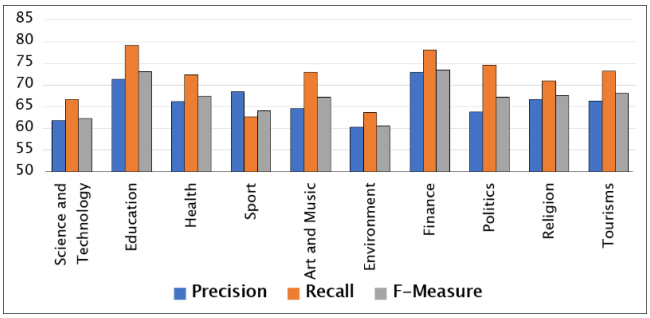

Fig. 3. BAMA effects on different document types.

Fig. 3 shows the values of metrics depending on file type when using BAMA Morphological Analyzer Applied. According to the figure BAMA got the best Results in education, finance and art respectively.

Applied: According to the figure BAMA got the best Results in education, finance and art respectively.

Fig. 4 shows the values of metrics depending on file type when using Safar Al-Khalil Morphological Analyzer. It shows the least values in religion tourism and sports.

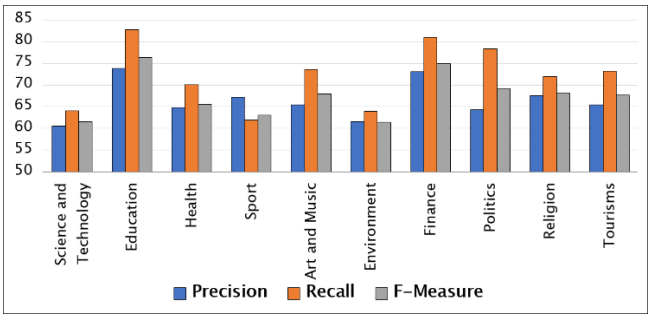

Fig. 4. Safar Al-Khalil effects on different document types.

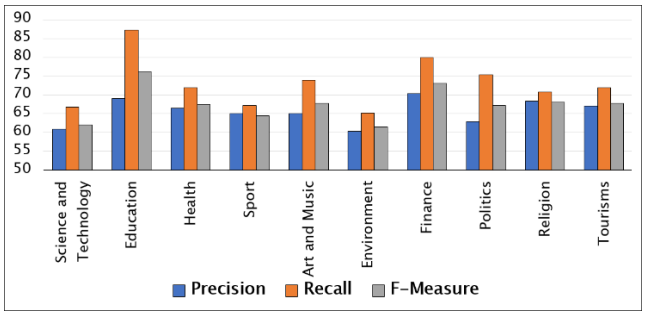

Fig. 5. Stanford NLP effects on different document types.

Fig. 5 shows the values of metrics depending on file type when using Stanford NLP as a Morphological Analyzer.

Table 2 shows a comparison between the evaluation metrics for the three types of morphological analyzers, from the table Safar Al-Khalil returns the best results among the other analyzers. Because Safar Al-Khalil analyzer depends on finding all possible vowelized forms that belong to the current word then divides the word into 5 parts enclitic, suffix, lemma, prefix, and proclitic. Also, the predefined xml tables for affixes is more complete than the other analyzers.

Table 2. Comparison of evaluation metrics between the three analyzers.

Morphological Analyzers	Precision	Recall	F-measure
BAMA	66.24	71.42	67.10
Safar Alkhalil	66.40	72.05	67.60
Stanford NLP	65.57	73.04	67.57

Fig. 6 shows the values of the average metrics for the different types of morphological analyzer into the extracted summary. This figure shows that Stanford NLP has the best value in recalling metrics, but the least value in precision.

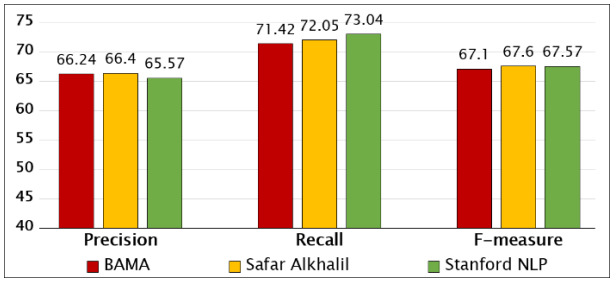

Fig. 6. Average metrics for all documents.

Table 3 shows the comparison between these paper results with others results. The results compared with two researches that uses the same data set used in this research which is EASC. The compared research is with the following titles: research (1) Applying semantic and Analysis for ATS [17], and research (2) studying the different types of stemmers on Arabic text [9], and research 3 using Graph-based ATS Approach with the shortest path algorithm [7]. All the previous researches examined in the current research apply EASC corpus.

The results show that this research has the best performance among the others, depending on the comparison. This improvement in performance came from using nouns in the relation between the graph nodes, also using the minimum spanning tree and its advantages of building an optimum tree contains all nodes.

Table 3. Comparison with others works.

Methods	Precision	Recall	F-measure
Statistical and Semantic Analysis [17]	57.62	58.80	58.20
Different Types of Stemmers [9]	55	48	51
Shortest Path Algorithm [7]	54	47	51
This Proposed Method	66.40	72.05	67.6

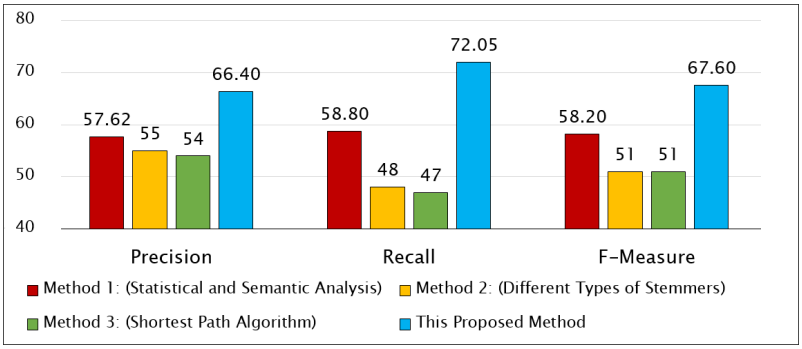

Fig. 7. Performance evaluation compared with other researches.

Fig. 7 describes the performance evaluation of this research results compared with other Researches. Safar Al-Khalil morphological analyzer is used to compare with the other researches. This research has a better performance in all metrics of comparison used

i.e. the results show that this research achieved 66.4% in precision, 72% in recall and 67.6% in F-measure which is higher than all the other researches.

From the above results, Safar Al-Khalil Morphological analyzer in general gives the best results. BAMA morphological analyzer gives better results than the others in tourisms and science categories. Results of Health and sports categories got the best performance, when using Stanford NLP morphological algorithm.

8. CONCLUSION

Text Summarization is becoming more famous in web and electronic libraries. Graph based approached focus on the relation between sentences not on the sentence itself. There is a lack in researches of Arabic text summarization depending on graph-based approaches. This research tries to enhance the performance of the generated summaries by applying MST algorithm with multi-morphological analyzers on the process of extracting ATS. MST build an optimum tree with less cost and no recycling, which help in the ranking of document's sentences to get the best summary. Arabic language suffers from the problem of finding the noun in the sentences due to the absence capital letter and small letters, in addition to the absence of diacritics in the written text. Therefore, the morphological analyzers are used to determine the position of every word in the sentence and extract only words that has tag of "noun" which will be used later in features extraction process. There are three morphological analyzers used in this research, to pick one that has the best effect on summarization process. 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. This corpus contains 153 documents divided into 10 subjects, each document has 5 pre-generated summaries. The metrics used here are Precision, Recall and F-measure. In general, Safar Al-Khalil morphological analyzer gives better results than the others after applying it with Minimum Spanning Tree algorithm.

REFERENCES

1. E. Lloret, "Text summarisation based on human language technologies and its applications," Department of Computer Languages and Systems, Universidad de Alicante, 2011.
2. C. Y. Lin and E. Hovy, "Manual and automatic evaluation of summaries," in *Proceedings of ACL Workshop on Automatic Summarization*, Vol. 4, 2002, pp. 45-51.
3. R. Mihalcea, "Graph-based ranking algorithms for sentence extraction, applied to text summarization," in *Proceedings of ACL Interactive Poster and Demonstration Sessions*, 2004, pp. 170–173.
4. E. Lloret and M. Palomar, "Text summarization in progress: a literature review," *Artificial Intelligence Review*, Vol. 37, 2012, pp. 1-41.
5. F. S. Douzidia, "Automatic summarization of Arabic text," Memory presented at the Faculty of Graduate Studies, University of Montreal, 2004.
6. M. Sawalha and E. Atwell, "Comparative evaluation of Arabic language morphological analyzers and stemmers," in *Proceedings of Coling Companion Volume:* Posters,

2008, pp. 107-110.
7. A. T. Al-Taani and M. M. Al-Omour, "An extractive graph-based Arabic text summarization approach," in *Proceedings of International Arab Conference on Information Technology*, 2014, pp. 158-163.
8. K. Kirchhoff, D. Vergyri, J. Bilmes, K. Duh, and A. Stolcke, "Morphology-based language modeling for conversational Arabic speech recognition," *Computer Speech & Language*, Vol. 20, 2006, pp. 589-608.
9. N. Alami, M. Meknassi, S. A. Ouatik, and N. Ennahnahi, "Impact of stemming on Arabic text summarization," in *Proceedings of the 4th IEEE International Colloquium on Information Science and Technology*, 2016, pp. 338-343.
10. A. A. El-Harby, M. A. El-Shehawey, and R. El-Barogy, "A statistical approach for Qur'an vowel restoration," *ICGST-AIML Journal*, Vol. 8, 2008, pp. 9-16.
11. R. L. Graham and P. Hell, "On the history of the minimum spanning tree problem," *Annals of the History of Computing*, Vol. 7, 1985, pp. 43-57.
12. I. Mani and M. T. Maybury, "Automatic summarization," *John Benjamin's Publishing Co*., 2001.
13. S. Lagrini, M. Redjimi, and N. Azizi, "Automatic Arabic text summarization approaches," *International Journal of Computer Applications*, Vol. 164, 2017, pp. 31-37.
14. M. Hadni, A. Lachkar, and S. A. Ouatik, "Multi-word term extraction based on new hybrid approach for Arabic language," in *Proceedings of the 2nd International Conference on Computational Science and Engineering*, 2014, pp. 109-120.
15. Y. A. Jaradat and A. T. Al-Taani, "Hybrid-based Arabic single-document text summarization approach using genetic algorithm," in *Proceedings of the 7th IEEE International Conference on Information and Communication Systems*, 2016, pp. 85-91.
16. K. Patil and P. Brazdil, "Text summarization: Using centrality in the pathfinder network," *International Journal of Computer Science and Information Technologies*, Vol. 2, 2007, pp. 18-32.
17. N. Alami, Y. el Adlouni, N. En-nahnahi, and M. Meknassi, "Using statistical and semantic analysis for Arabic text summarization," in *Proceedings of International Conference on Information Technology and Communication Systems*, 2017, pp. 35-50.
18. R. Belkebir and A. Guessoum, "TALAA-ATSF: A global operation-based Arabic text summarization framework," in *Intelligent Natural Language Processing: Trends and Applications*, Springer, Cham, 2018, pp. 435-459.
19. S. Khoja and R. Garside, "Stemming Arabic text," Computing Department, Lancaster University, UK, 1999.
20. T. Buckwalter, "Issues in Arabic orthography and morphology analysis," in *Proceedings of ACL Workshop on Computational Approaches to Arabic Script-based Languages*, 2004, pp. 31-34.
21. Y. Jaafar and K. Bouzoubaa, "Benchmark of Arabic morphological analyzers challenges and solutions," in *Proceedings of the 9th IEEE International Conference on Intelligent Systems: Theories and Applications*, 2014, pp. 1-6.
22. C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proceedings of the 52nd ACL Annual Meeting on System Demonstrations*, 2014, pp. 55-60.
23. A. Abu-Errub, "Arabic text classification algorithm using tfidf and chi square measurements," *International Journal of Computer Applications*, Vol. 93, 2014, pp. 40-45.

24. M. El-Haj, U. Kruschwitz, and C. Fox, "Using mechanical turk to create a corpus of Arabic summaries," in *Proceedings of the 7th International Language Resources and Evaluation Conference*, 2010, pp. 36-39.

Reda Elbarougy received his B.Sc. and M.Sc. degrees from Mansoura University, Egypt, in May 1997, and February 2006, respectively. Both were in Computer Science. He was with the Faculty of Science, Mansoura University from 1999 to 2009. In July 2009, he joined the Japan Advanced Institute of Science and Technology (JAIST), Japan, as a Ph.D. student. From September 2014 to August 2019, he was with Mathematics Department, Faculty of Science, Damietta University as an Assistant Professor. In 2017 he was a post-doctor researcher funded from JSPS to conduct a research in Japan Advanced Institute of Science and Technology (JAIST) from June 2017 till April 2019. Currently he is an Assistant Professor in Department of Computer Science, Faculty of Computer and Information Sciences, Damietta University, New Damietta, Egypt, from August 2019 till now. His current research interests include machine learning, artificial indulgence, speech analysis, speech emotion recognition, and synthesis.

Gamal Behery received the B. Sc. degree in Computer Science from the Faculty of Science, Suez Canal University, Egypt, in 1984, and the M.Sc. degree in Computer Science from Mansoura University, Egypt, in 1989 and the Ph.D. degree in Computer Science from Mansoura University (Egypt) / Friedrich-Alexander University (Erlangen-Nürnberg – Germany), in 1993. From 1993 to 2008, he was an Assistant Professor with Mansoura University. In 2008, he has been an Associate Professor in Computer Science with Mansoura University. From 2016 to August 2019, he has been a Professor in Computer Science with Damietta University. Currently he is a full Professor and the Dean of Faculty of Computer and Information Sciences, Damietta University, New Damietta, Egypt, from August 2019 till now. His current research interests include image processing, pattern recognition, animal recognition, AI, neural networks and computer language design.

Akram El Khatib was born in Gaza, Palestine, received his B.Sc., degree in Computer Science from Al-Quds Open University, Palestine, in 2002, and the M.Sc. degree in Computer Science from Suez Canal University, Egypt, in 2015. His current research interests include natural language processing, text summarization, data mining, graph based, machine learning, wireless network, network security and information systems.