ISSN 1840-4855
e-ISSN 2233-0046

Original scientific article
http://dx.doi.org/10.70102/afts.2026.1835.087

GRAPH-BASED METHODS FOR EXTRACTIVE ARABIC NEWS TEXT SUMMARIZATION

Rasha Almutairi[1*], Sahar Jambi[2], Tawfiq Hasanin[3]

[1]Department of Information Systems, King Abdulaziz University, Jeddah, Saudi Arabia.*
e-mail: rbrktallahalmutairi@stu.kau.edu.sa,
orcid: https://orcid.org/0000-0002-9434-6891
[2]Assistant Professor, Department of Information Systems, King Abdulaziz University, Jeddah, Saudi Arabia. e-mail: shjambi@kau.edu.sa,
orcid: https://orcid.org/0009-0006-5390-1656
[3]Associate Professor, Department of Information Systems, King Abdulaziz University, Jeddah, Saudi Arabia. e-mail: thasanin@kau.edu.sa,
orcid: https://orcid.org/0000-0003-1072-278X

Received: December 27, 2025; Revised: February 12, 2026; Accepted: March 30, 2026; Published: May 29, 2026

SUMMARY

The speed of the increasing digital content requires the creation of successful Automatic Text Summarization (ATS) systems. Although major improvements have been made in the summarization of high-resource languages, the summarization of Arabic texts has not been effectively studied, especially in terms of comparative studies of preprocessing methods of documents and word-embedding algorithms. This paper explores the effects of some of the most important variables on the work of graph-based extractive summarization of Arabic news articles, namely, preprocessing methods, word embeddings, ranking methods, and compression ratios. There were experiments using the Essex Arabic Summary Corpus (EASC) with four preprocessing methods (Khoja, Farasa, Qalsadi, and Stanza), two word embedding models (GloVe and AraBERT), two ranking algorithms (PageRank and HITS), and two compression ratios (30% and 40%). The quality of summarizing was measured by the ROUGE-1 F- score. The findings indicated a significant difference ($p < 0.001$) in all factors, and GloVe performs better than AraBERT (average ROUGE-1 F-score of 0.389 vs. 0.36), and a higher compression ratio (40% more) achieved better performance. To be more precise, such preprocessing techniques as Khoja and Farasa yielded the same ROUGE-1 F-scores of 0.381 and 0.379, respectively, and Stanza gave much lower ones (0.364). It was statistically significant that there have been interactions between preprocessing model and word embedding model, ranking algorithm and compression ratio. Future research will offer more extensive guidelines on how to choose the best preprocessing and representation strategies to use with Arabic ATS systems by including larger and more varied datasets, as well as human evaluation methods to offer a wider range of evaluation. More studies will also be done on the fusion of the supervised summarization technique and deep learning-based systems and multilingual summarization systems.

Key words: *extractive text summarization, arabic text summarization, automatic text summarization (ATS), graph-based, stemming, lemmatization, word embeddings (WE).*

INTRODUCTION

The volume of digital data from various sources, including user reviews, news, blogs, social media platforms, books, and scientific papers, is growing daily. As a result, accessing the required information has become increasingly time- and effort-consuming, making automated text summarization (ATS) systems essential [1]. In the current era of data overload, there is a significant need for ATS tools. ATS aims to generate concise summaries from large volumes of text, enabling users to access essential information without extensive browsing or searching [2][3].

According to El-Kassas et al. [1], a summary is a writing that is based on a single or several original documents, actually giving a brief description of the main points and being shorter than half of the original text. Overall, there are two broad methods of text summarization, which are extractive and abstractive [4]. In extractive summarization, important sentences are directly selected and copied from the original text, whereas abstractive summarization produces fresh sentences that summarize the content of the source text. The given work is devoted to the extractive method of summarizing single documents.

Arabic, spoken by approximately 250 million people worldwide, is characterized by a rich linguistic heritage in terms of vocabulary and morphology. Nevertheless, it is characterized by a complicated structure, which is characterized by ambiguity and a great morphological variation, and is unusual in terms of text summarization. Although ATS has already received considerable research focus on high-resource languages like English, no similar work has been done in Arabic [5][6]. The ATS methods may be categorized into three types: machine learning (ML) [7], graph-based [8], or hybrid [9]. This paper concentrates on graph-based sentence ranking algorithms that have been very beneficial in Arabic text summarization. The recent findings have shown good results when using graph-based approaches [10][11]. Existing research has several limitations, despite the fact that several studies have been conducted on Arabic text summarization [7][8][12]. A large number of studies are concerned with the evaluation of a single preprocessing method or word representation model individually [13][14], whereas others do not have statistically based comparisons across multiple system components [15].

Also, there has been little literature to examine interaction effects of preprocessing methods, word embedding models, ranking algorithms, and compression ratios, as a unified experimental study. Consequently, there are no obvious empirical guidelines on how to create effective Arabic extractive summarization systems that have not been studied properly.

To fill these gaps, this research paper develops a detailed experimental analysis of graph-based extractive Arabic text summarization by analyzing the effect of various important factors in a systematic way. In particular, the impact of various preprocessing methods, word embedding algorithms, sentence-ranking algorithms, and compression ratios is evaluated with the help of stringent statistical tests.

The paper explores the effects of preprocessing, namely stemming and lemmatization, and the results of different word embedding models in graph-based techniques of extractive, single-document Arabic summarization. The content of the rest of the paper will have the following form: Section II will be the review of related work, Section III will be the methodology, Section IV will discuss the data used and experiments, and finally, Section V will be the conclusion of the paper and future research directions.

RELATED WORKS

This section examines past literature on text summarization that was pertinent to the work. The discussion is organized into three categories: preprocessing techniques, feature extraction approaches, and text summarization methods. These studies provide important foundations for Arabic text summarization; however, they also reveal limitations that motivate the need for further comparative investigation.

Preprocessing Techniques

The NLP applications that use preprocessing methods include text summarization, where useful preprocessing can enhance the effectiveness and quality of text summarization outputs. The typical preprocessing functions are lemmatization, stemming, removal of stop-words, and morphological

analysis. The inflected word forms are mapped to their base form by lemmatization [16], which allows semantically related words to be clustered. Stemming [17] decreases words to their roots such that words that have a semantic relationship can be handled in a similar manner. Stop words [18] are high-frequency words with little semantic value, which are often removed to reduce text size and noise. Morphological analysis [19] identifies the internal structure of words and supports tasks such as named entity extraction and sentence similarity computation.

Several studies have examined the effect of preprocessing techniques on Arabic extractive summarization. Al-Numai & Azmi [20] proposed a lemma-based evaluation method and showed that lemmatization improves text similarity measures. Alami et al. [19] made a comparison of three Arabic stemming methods, Khoja, Light, and Alkhalil, and tested them under the ROUGE metric, and stated that the Khoja stemmer performed better. Removal of stop-words has also been studied with reference to Arabic summarization. Elbarougy et al. [13] examined the effect of the removal of stop words on a graph-based summary system and reported that the removal of stop words resulted in better performance of the system.

On a similar note, Elbarougy et al. [8] studied the application of morphological analyzers (Safar Alkhalil, Stanford NLP, and BAMA) and came to the conclusion that Safar Alkhalil was the most effective in generating summaries. Despite the fact that these studies prove the significance of preprocessing in Arabic summarization, the majority of them concentrate on the evaluation of single techniques separately. Also, comparative studies on the use of various preprocessing techniques through coherent experimental conditions and statistical authentication have not been developed.

Feature Extraction

The purpose of feature extraction is to recognize and model the most significant data of a document so that the summarization process can be facilitated. Generally speaking, the methods of extraction of features can be divided into statistical and non-statistical (semantic) ones. Statistical features comprise sentence position, sentence location, and term frequency-inverse document frequency (TF-IDF), whereas semantic features are based on word embeddings and contextual representations. A number of papers have suggested models of feature extraction for Arabic text summarization. Qaroush et al. [12] have proposed a single document summarization method based on combining statistical and semantic features through scoring and machine learning techniques, and they have obtained better results with respect to benchmark data.

Abdulateef et al. [7] came up with a machine learning (ML) document summarization algorithm, which combines word2vec, clustering, and statistical algorithms. Their work dealt with some of the most frequent problems of text summarization, such as redundancy and information noise. The results of the evaluation conducted with the help of the ROUGE metric have shown that the offered approach performed better than the methods that were considered the state of the art. Over the last few years, there have been a few studies carried out to measure the performance of different word embedding approaches as well as text summarization systems. Word embeddings have several different algorithms, among them Word2Vec, AraBERT, and FastText. In [14], the authors suggested a single-document summarization extractive approach that used a combination of the AraBERT model and a clustering algorithm. The strategy was tested against the ROUGE metric as well as human evaluation and got an F-score of around 0.51 and a human evaluation score of 0.52.

Wazery et al. [21] developed an abstractive text summarization method based on a sequence-to-sequence architecture comprising an encoder and a decoder. Their experiment investigated how a variety of deep neural network structures affect the performance of summarization, as well as comparing the performance of different word embedding models. The findings of the experiment showed that the skip-gram Word2Vec model was superior to the continuous bag-of-words (CBOW) model. AarBERT deployment in the preprocessing step made the performance even better, and the proposed method gave good results in comparison with available methods. In the research, Burmani et al. [15] suggested a graphical technique for extractive Arabic text summarization. Their work covered the methods of sentence representation, methods of measuring similarity and ranking algorithms, and the system was tested using

the EASC Arabic corpus. The findings showed that using the TF–IDF representation with cosine similarity and the PageRank algorithm produced high-quality summaries.

More recently, Alselwi et al. [11] introduced a graph-based extractive summarization approach that applied PageRank in conjunction with word embeddings. The model consisted of three stages: preprocessing, feature extraction, and graph construction. Evaluation results demonstrated that this method achieved strong performance compared to other techniques.

Although these achievements have been made, the currently existing studies consider a small number of embedding methods or a specific summarization architecture. Therefore, prior studies have not adequately compared the relative performance of both the graph-based Arabic extractive summarization task using both the static and contextual word embeddings [29].

Text Summarization Techniques

Arabic text summarization has been addressed using a variety of approaches, including graph-based methods [8], machine learning techniques [7], and deep learning models [21]. These approaches can be classified as supervised or unsupervised, depending on whether annotated training data are required. Several studies [22] have confirmed the effectiveness of graph-based methods in improving performance in Arabic natural language processing tasks, particularly text summarization.

Several studies have confirmed the effectiveness of graph-based methods for Arabic summarization. Graph-based techniques are unsupervised algorithms, with PageRank [23] and Hyperlink-Induced Topic Search (HITS) [24] being the two most common for summarization.

Qaroush et al. [25] proposed a graph-based summarization method that uses stems, words, and n-grams as textual units. The text is transformed into a graph, and the PageRank algorithm is applied to rank sentences. Experiments on the EASC dataset demonstrated that the method outperformed other techniques.

Elbarougy et al. [10] developed a graph-based text extraction method comprising four main stages: preprocessing, feature extraction, graph construction, and sentence ranking, followed by summary generation. This method applied a slightly different version of the PageRank algorithm and had a high accuracy. Although such methods have delivered encouraging results, the majority of the previous studies have focused on optimizing each part of the summarization pipeline. General comparisons of preprocessing methods, word embedding strategies, ranking algorithms, and compression rates on statistically tested bases are relatively rare.

The table 1 will summarize the research that other studies have performed with similar work as the study provided in this paper. It describes the summarization techniques, preprocessing techniques, feature extraction techniques, datasets, and evaluation measurements in these research works. As an illustration, examples like [13][8] investigated the effects of various preprocessing methods, whereas examples like [12][7][21] integrated various feature extraction methods. More recent studies, including [11][15], analyzed and compared different aspects of summarization techniques. The primary distinction between our study and previous Arabic summarization systems is that we conduct a broader comparison, examining multiple techniques and presenting the results in a different manner.

Table 1. Summary of selected related works on arabic text summarization

Ref	Year	ATS Approach	Single/ Multi Document	ATS Method	Preprocessing Techniques	Feature Extraction	Dataset	Evaluation Method
[13]	2021	Extractive	Single	Graph-based method	Normalization, stop-word removal, tokenization, and stemming	TF/ISF and cosine similarity	Author's dataset	Precision, recall, and F-measure
[8]	2020	Extractive	Single	Graph-based method	Normalization, tokenization, stop-word removal, morphological analysis, and stemming	TF–IDF and cosine similarity	EASC	Precision, recall, and F-measure
[12]	2019	Extractive	Single	Machine learning and score-based	Normalization, tokenization, stop-word removal, and stemming	Key-phrases feature and similarity measures	EASC	ROUGE-N (ROUGE 1, ROUGE 2)
[7]	2020	Extractive	Multi Document	Machine-based Method	Tokenization, stop-word removal, and stemming	Word2vec and CBOW	EASC	ROUGE-N (ROUGE 1, ROUGE 2)
[21]	2022	Abstractive	Single	Deep learning model	Normalization, stop-word removal, and AraBERT preprocess	Word2vec (Skip-gram CBOW)	AHS and AMN	ROUGE and BLUE
[15]	2022	Extractive	Single	Graph-based method	Tokenization, stop-word removal, and lemmatization	FastText, Word2Vec, TF–IDF, overlap, Euclidean distance, and cosine similarity	EASC	ROUGE N (ROUGE 1, ROUGE 2)
[11]	2024	Extractive	Single	Graph-based method	Normalization, tokenization, stemming, and stop-word removal	Word2Vec, TF–IDF, and cosine similarity	EASC	ROUGE N (ROUGE 1, ROUGE 2)

METHODOLOGY

This section describes the experimental design and procedures for evaluating graph-based extractive summarization of Arabic documents. The results of the study are organized in a systematic analysis of the impact of different elements of the system, such as text preprocessing methods, word embedding models, feature extraction methods, sentence ranking algorithms, and compression ratio levels. The EASC dataset serves as the primary dataset for all experiments, and the ROUGE-1 F score is used to assess summarization performance. To ensure the reliability of the results, analysis of variance (ANOVA) and Tukey's test were used to assess significant differences in performance. The figure 1 provides an overview of the experimental workflow.

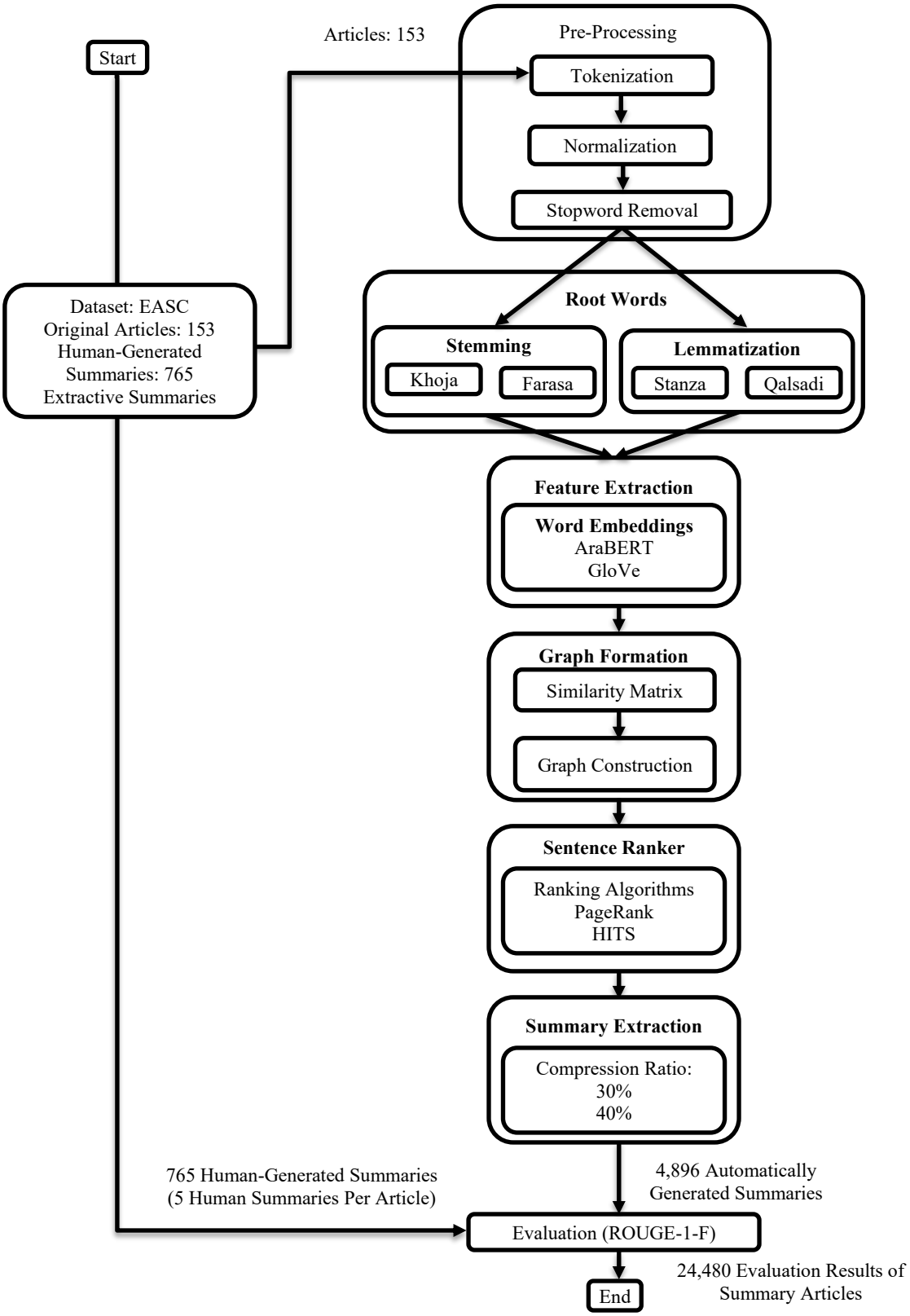

Figure 1. Study methodology

Text Preprocessing

Preprocessing is also an essential part of text summarization, and it is instrumental in enhancing the quality of the summaries that they produce, especially in Arabic writing, because of its good morphology and linguistic diversity [31]. The preprocessing also involved in this study involved tokenization,

normalization, elimination of stop-words, followed by stemming or lemmatization as a comparison method in NLP of shortening words to their root forms.

Tokenization

Tokenization [26] is the process of dividing text into meaningful units such as sentences and words. It represents the first step after loading the input documents and is essential for handling linguistic ambiguity and enabling subsequent processing stages.

Normalization

Normalization is used to guarantee the textual consistency of a text by eliminating noise and normalizing the various surface forms. It involves removing punctuation marks, numbers, and unnecessary white space and transforming some letters of the Arabic language into standard forms. As an example, the plural variations of the letter Alef (آ، إ، أ) are optimised into one variant (ا).

Stop-Word Removal

Stop-word removal is used to remove the high-frequency words with limited semantic meaning and, as such, reduces the data size and enhances the processing efficiency. The number of Arabic stop-word lists employed in this study was 750 words, which is the combination of the existing standard list and the one suggested by Khoja [27].

Stemming

Stemming simplifies words by removing affixes and enabling words that have common semantic roots to be treated in the same way. An example of this is the term in Arabic for a book, which is called كتاب. In this study, two stemming algorithms were compared: Khoja and Farasa.

Lemmatization

Lemmatization is used to recognize a canonical base form of the word in the linguistic background. Lemmatization is not like stemming, as it maintains grammatical correctness. Two lemmatization tools, which are Stanza and Qalsadi, have been used in this study in comparison.

Feature Extraction

After doing preprocessing, it then goes through a feature extraction process where textual information is processed into numerical form [28] since machine learning algorithms cannot process untouched textual data.

Word Embeddings

Word embeddings (WE) [34] are a view of words as continuous vectors in which semantically similar words are placed close to each other in the vector space. The study compared two varieties of word embedding models, including contextual ones with the use of AraBERT [30] and static ones with GloVe [33]. In the case of all sentences, word vectors were averaged to obtain a sentence-level representation suitable to be used in the computation of similarity in graph-based summarization.

Similarity Measure and Graph Construction

Sentence embeddings generated during the feature extraction stage were used to compute pairwise sentence similarity scores. Cosine similarity [32] was adopted as the similarity measure, as it effectively captures the angular distance between vector representations, such as a smaller angle indicating higher similarity between the sentences. The cosine similarity between two sentence vectors is computed as shown in equation (1).

$$Cosine\ Similarity = \frac{\sum_{k=1}^{N} S_{ik}\, S_{ik}}{\sqrt{\sum_{k=1}^{N} S_{ik}^2}\sqrt{\sum_{k=1}^{N} S_{ik}^2}} \quad (1)$$

where S_{ik} and S_{jk} refer to components of the vectors S_i and S_j, respectively.

Based on the similarity matrix, each document was modeled as a graph in which sentences correspond to nodes. An edge was established between two nodes if their corresponding sentences exhibited similarity, with edge weights proportional to the cosine similarity values.

Sentence Ranking Algorithm

Once the sentence graph had been constructed, the sentence importance was estimated using two graph-based ranking algorithms, namely PageRank [23] and Hyperlinked Induced Topic Search HITS [24]. The ranking algorithms evaluate the relevance of a vertex in a graph based on the data obtained during the graph construction. This algorithm assigns a score to each sentence based on the features extracted from it. The most important sentences are extracted from the original text. Important sentences are those that are closely related to other sentences and thus receive the highest score. Ultimately, the final summary is generated by selecting the highest-scoring sentences.

Summary Extraction

The last summary is created by picking the best-ranked sentences with regard to the score that is calculated. Sentences are ranked in order of decreasing importance; the most important sentences are listed in the summary based on a given compression ratio (CR). CR is a measure of sentences that have been chosen within the original text. CR values of 30% and 40% were considered in this research, where these ratios have been found by other studies to be concise and cover the content.

Evaluation Measure

In order to measure the output of the automated summarization system, we have used ROUGE, which is one of the most popular metrics used to measure machine-generated summaries. ROUGE which is short of Recall-Oriented Understudy of Gisting Evaluation has various variants including ROUGE-N, ROUGE-W, ROUGE-L and ROUGE-S. This measure quantifies the quality of the summary by comparing the resultant summary to a human reference summary, and counting the commonality of the two summaries in terms of n-grams, word sequences, or word pairs. ROUGE-N is a peculiar value that measures the recall of n-grams between an automated summary and a human-generated reference summary. Outsourced variants are ROUGE-1 with the unigrams and ROUGE-2 with the bigrams. ROUGE-N is computed as shown in equation (2).

$$ROUGE - N = \frac{\sum_{S \in (ReferenceSummaries)} gram_x \in S \sum Count_{match}(gram_n)}{\sum_{S \in (ReferenceSummaries)} gram_n \in S \sum Count(gram_n)} \quad (2)$$

where N denotes the size of the n-gram sequence and Count match ($gram_n$) is the maximum number of co-occurrences of the n-gram in both the reference summary and the generated summary.

ROUGE-1 is commonly adopted in Arabic summarization studies due to its robustness in handling morphological variation. The ROUGE-1 scores lie in the range of 0 to 1, with the highest score of 1 indicating the highest level of similarity between the generated text and the reference text, and the lowest score of 0 is used to indicate the lack of similarity. The difference in scores, therefore, implies a greater similarity in the two summaries. It evaluates three scores, namely, precision (P), recall (R), and the F score. In this work, we report the F score, as it incorporates both P and R in a single metric.

Comparison Factors

In this research, the researcher will provide a detailed analysis of a range of models that are used to extract text summaries in the Arabic language. The objective is to assess how several factors influence the quality

of automated summaries, including preprocessing techniques, word embedding models, compression ratio levels, and graph ranking algorithms. In order to determine the impact of these factors, ANOVA was used to compare group means and determine statistically significant differences due to the differences in and between the groups by using the F-test to analyze groups. Since each factor contains multiple methods, Tukey's test was employed to determine pairwise differences between individual group levels.

While ANOVA determines whether a factor has an overall effect, Tukey's test identifies which specific configurations differ significantly, using its letter grouping method, thus providing a more detailed understanding of the relationships between different comparison factors. The table 2 summarizes experimental settings with techniques/libraries used.

Table 2. Summary of experimental settings with techniques/libraries used

Category	Parameter/Settings	Description
Preprocessing	NLTK, spaCy, PyArabic	Text segmentation into sentences and words, removal of stop words, normalization of Arabic text, and removal of diacritics to prepare the data before summarization.
	Khoja, Farasa	Extraction of Arabic word root.
	Stanza, Qalsadi	Extraction of word lemmas.
Embedding Types	AraBERTv02, GloVe	Generating contextual and static word embeddings.
Evaluation	ROUGE-1-F	An automated metric for comparing system-generated summaries with human reference summaries.
Statistical Analysis	ANOVA, Tukey HSD	Statistical significance testing and pairwise comparison of group means.

EXPERIMENTS

This part explains the dataset utilized in the research, and then the description of the experiment setting and its execution is given. The objective of the experiments is to analyze the impact of different system components on summarization performance under controlled and statistically validated conditions. Furthermore, this section presents and discusses the details of the results obtained from conducting various experiments according to the proposed methodology for evaluating the graph-based abstraction of Arabic documents.

Dataset (Corpus)

To evaluate the proposed approach, the Essex Arabic Summaries Corpus (EASC) was used. EASC is a widely distributed dataset, created via the Amazon Mechanical Turk (Mturk) platform, containing 153 Arabic articles, each accompanied by five human-generated reference abstracts. The articles were collected from multiple sources, including Wikipedia, Al-Watan newspaper, and Al-Rai newspaper, and cover a variety of domains such as education, art and music, finance, environment, politics, health, tourism, sports, science and technology, and religion. In table 3 presents the details of the dataset.

Table 3. Details of the EASC dataset

Dataset	EASC data (2013)
Type	News
No. of Sentences	1957
No. of Words	50411
No. of Documents	153

Source: https://sourceforge.net/projects/easc-corpus/support

Experimental Setup

As described in the methodology, a series of experiments were conducted on the EASC dataset, using ROUGE-1 as a text summarization metric. Four factors were investigated: (1) two stemming algorithms and two lemmatization analysis algorithms, (2) two word embedding models, (3) two graph-based ranking algorithms, and (4) two determined CR values.

This resulted in 32 experiments (4 × 2 × 2 × 2 = 32 experiments), yielding 4869 summaries (32 × 153 = 4869 summaries). Each summary was compared to five human reference summaries, resulting in 24480 evaluation metric values (4896 × 5 = 24480 values). In figure 1 illustrates the experimental steps.

RESULTS AND DISCUSSION

Python 3.10.12 and a set of special libraries used to process, embed, and evaluate Arabic text were used to conduct the experiments. The data management and organization of experimental results were performed with the help of the Pandas library, and the efficient numeric computing was performed with the help of NumPy. The Arabic text processing was done using the NLTK, spaCy, and PyArabic libraries during the preprocessing stage. Khoja and Farasa libraries were applied in stemming and Stanza, and Qalsadi libraries were applied in lemmatization. Contextual representations of the sentences were created in the embedding phase, where the AraBERTv02 model was used to create the representations, and GloVe static embeddings. Lastly, the performance of the system was measured through the ROUGE-1 measure with the use of the rouge-score library.

The four factors of this study were statistically analyzed to examine their impact on the quality of automated summaries. The results of the analysis of variance (ANOVA) of each of the factors and their interaction were presented in table 4, and the ROUGE-1 F-score was used as a dependent variable. Statistical significance in ANOVA can be used to demonstrate whether the effect of a factor or an interaction between multiple factors is actually a real effect, and was not by chance.

Table 4. ANOVA test results for four factors in terms of ROUGE-1 F score

Cases	Sum of Squares	df	Mean Square	F	p
Preprocessing	1.083	3	0.361	10.197	<.001
Word embedding	5.326	1	5.326	150.387	<.001
Ranking algorithm	0.623	1	0.623	17.589	<.001
Compression ratio level	1.732	1	1.732	48.905	<.001
Preprocessing × Word embedding	0.656	3	0.219	6.174	<.001
Preprocessing × Ranking algorithm	0.848	3	0.283	7.986	<.001
Word embedding × Ranking algorithm	0.379	1	0.379	10.703	0.001
Preprocessing × Compression ratio level	1.010	3	0.337	9.503	<.001
Word embedding × Compression ratio level	0.128	1	0.128	3.606	0.058
Ranking algorithm × Compression ratio level	0.439	1	0.439	12.384	<.001
Preprocessing × Word embedding× Ranking algorithm	0.658	3	0.219	6.192	<.001
Preprocessing × Word embedding × Compression ratio level	0.738	3	0.246	6.950	<.001
Preprocessing × Ranking algorithm × Compression ratio level	0.855	3	0.285	8.046	<.001
Word embedding × Ranking algorithm × Compression ratio level	0.332	1	0.332	9.363	0.002
Preprocessing × Word embedding× Ranking algorithm × Compression ratio level	0.981	3	0.327	9.238	<.001
Residuals	862.608	24,358	0.035		

Note. Type III Sum of Squares

This significance is determined by the p-value whereby p value < .001 portray a statistically significant impact. The findings demonstrate that the four key variables, including preprocessing, model of word embedding, ranking algorithm, and compression ratio level, have statistically significant differences with the p-value of each being less than or equal to .001. This evidence shows that all factors have their independent effect on the work of the automated summarization system. In terms of interaction of the factors, two-way interaction has significant interaction of four factors and insignificant interaction of two-way interaction of Word embedding × Compression ratio level and Word embedding × Ranking algorithm, as they did not produce a significant interaction effect on the quality of summarization. The lack of significance for these two-factor interactions indicates that the effect of word representation models on summarization quality is not dependent on the type of ranking algorithm or compression ratio,

and vice versa; ranking algorithms and compression ratios maintain their performance regardless of the representation model used.

Moreover, the three-way interactions showed statistically significant interaction effects, except for the Word embedding × Ranking algorithm × Compression ratio level interaction. This indicates that combining a specific word embedding model with a specific ranking algorithm and a specific compression ratio does not produce an additional effect on performance beyond the main effects of each factor, demonstrating that the performance of each factor remains relatively independent of the combination of the other factors.

Regarding the four-way interaction, it showed statistical significance, indicating that the effect of a single factor is not constant but may vary depending on the interactions with the other three factors.

After confirming the existence of statistically significant differences among the studied factors, as indicated by the ANOVA results, Tukey's HSD multiple comparison test was applied to examine pairwise differences between groups. The test was conducted separately for each of the four factors, in order to assess the main effects and identify statistically significant differences among the different methods of each factor independently of the other factors.

In addition, Tukey's HSD test was applied only to the four-way interaction among the four factors, as this configuration reflects the actual system performance when all factors are combined and represents the practical implementation of an automated text summarization system. This analysis facilitates the identification of the optimal overall combination of factors that yields the highest performance.

Tukey's test reveals statistically significant differences between pairs of groups within each factor, as presented in tables 5–8, as well as between pairs of different factor combinations across all factors, as shown in table 9. In this context, pairs denoted by different letters are considered statistically significantly different, whereas pairs sharing the same letter do not exhibit statistically significant differences.

Factor-1: *Preprocessing Techniques*

The results of Tukey's HSD post-hoc test, shown in table 5, indicate that Khoja, Farasa, and Qalsadi share the same grouping letter (A), indicating that no statistically significant differences exist among them. In contrast, Stanza was assigned a different grouping letter (B), reflecting a significant difference in performance compared to the other techniques.

In figure 2 shows the mean ROUGE-1 F-score for each preprocessing method. Stanza (0.364) scored lower, while Qalsadi (0.375), Farasa (0.379) and Khoja (0.381), performed similarly. Stanza sometimes fails to correctly convert words into their basic forms, especially in languages with complex morphological structures, which explains its lower performance compared to other preprocessing techniques in Arabic automatic summarization.

Table 2. Tukey test pairwise comparison based on preprocessing techniques

Preprocessing	**Grouping Letters**
Khoja	A
Farasa	A
Qalsadi	A
Stanza	B

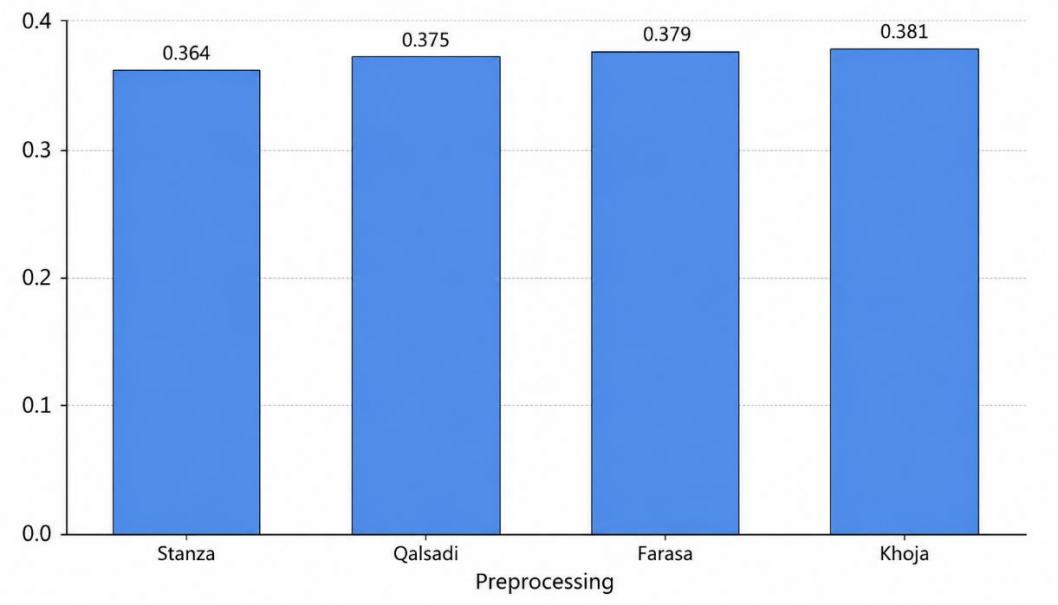

Figure 2. Mean ROUGE-1-F scores for preprocessing techniques

Factor-2: *Word Embeddings*

The second factor showed a clear impact on the performance of the two word embedding models: GloVe and AraBERT. As presented in table 6, Tukey HSD test showed that there is a statistically significant difference in the two models with two different letters assigned to each method. The average scores of the ROUGE-1-F technique are represented in figure 3, where the lowest mean score (0.36) is maintained by AraBERT, as opposed to the GloVe (0.389), which indicates the efficiency of the graph-based techniques in extractive summarization of Arabic with the aid of the static embeddings of GloVe. The lower performance of the AraBERT model can be attributed to the sensitivity of contextual embeddings to dataset size. contextual models of AraBERT require massive datasets to achieve accurate linguistic representation, which may be limited under these experimental conditions.

Table 3. Tukey test pairwise comparison based on word embedding models

Word Embedding	Grouping Letters
GloVe	A
AraBERT	B

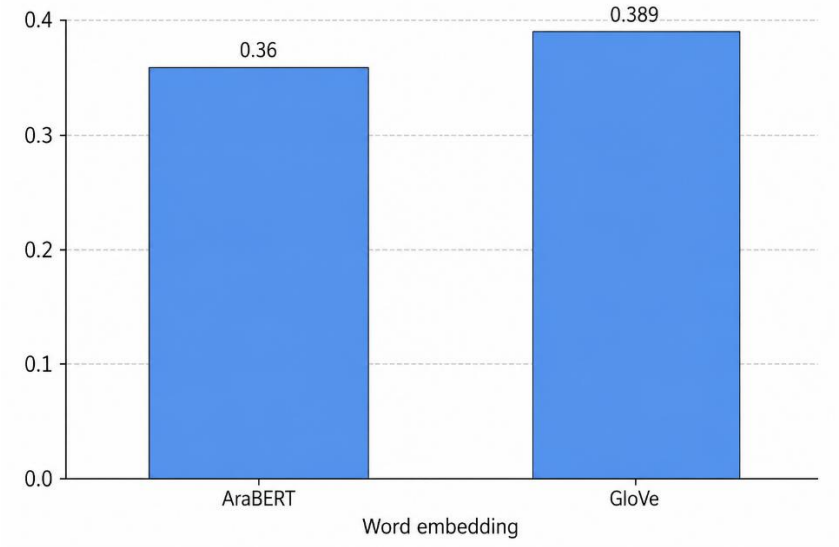

Figure 2. Mean ROUGE-1-F scores for different word embedding methods

Factor-3: Ranking Algorithms

The third factor revealed a difference in the performance of the two graph-based sentence-ranking algorithms, PageRank and HITS. According to Tukey's HSD test, shown in table 7, a statistically significant difference was found between the two algorithms.

The figure 4 shows that PageRank achieved the lowest ROUGE-1-F mean score (0.37) compared to HITS (0.38). This difference is attributed to the fact that the performance of ranking algorithms is highly dependent on experimental settings, particularly the word representation used to establish sentence similarity relationships. In this context, the HITS mechanism could be more compatible with the characteristics of the word representation used, which contributed to highlighting the most important sentences matching the human summaries, rather than PageRank.

Table 4. Tukey test pairwise comparison based on ranking algorithm methods

Ranking algorithm	Grouping Letters
HITS	A
PageRank	B

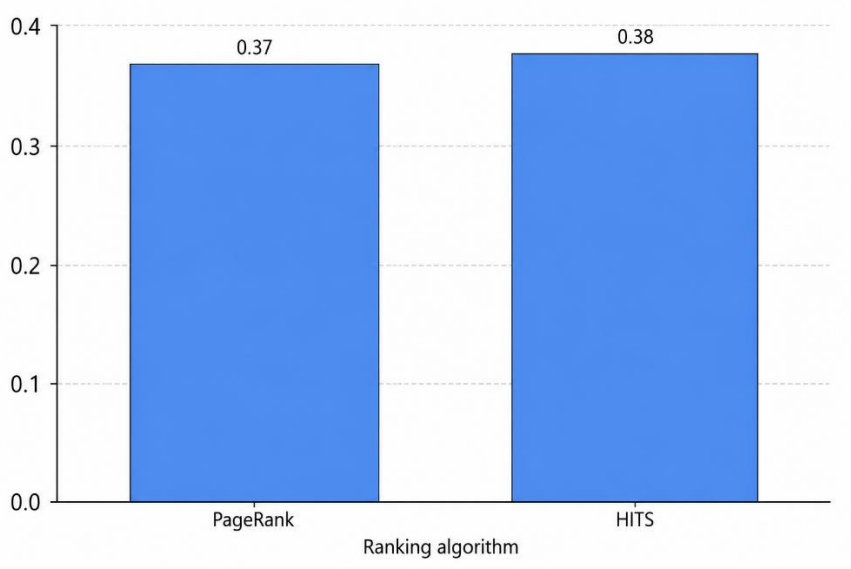

Figure 3. Mean ROUGE-1-F scores for different ranking algorithms

Factor-4: Compression Ratio

The fourth factor, the compression ratio (CR), showed a significant effect on the performance of the summarization system. According to Tukey's HSD test, shown in table 8, a statistically significant difference was found between the 30% and 40% compression ratios.

In figure 5 shows that the 30% compression ratio achieved the lowest ROUGE-1-F mean score (0.366) compared to 40% (0.384). This difference may indicate that lower compression ratios may result in more concise summaries, but at the expense of content comprehensiveness, which is reflected in the automated evaluation scores. On the other hand, an increased compression ratio can be used to provide a more comprehensive view of the semantic content of the text, and therefore, less significant information is lost during the summarization process. These results confirm that selecting the appropriate compression ratio is a crucial element in improving the quality of Arabic extractive summaries.

Table 5. Tukey test pairwise comparison based on compression ratio levels

Compression ratio level	Grouping Letters
40	A
30	B

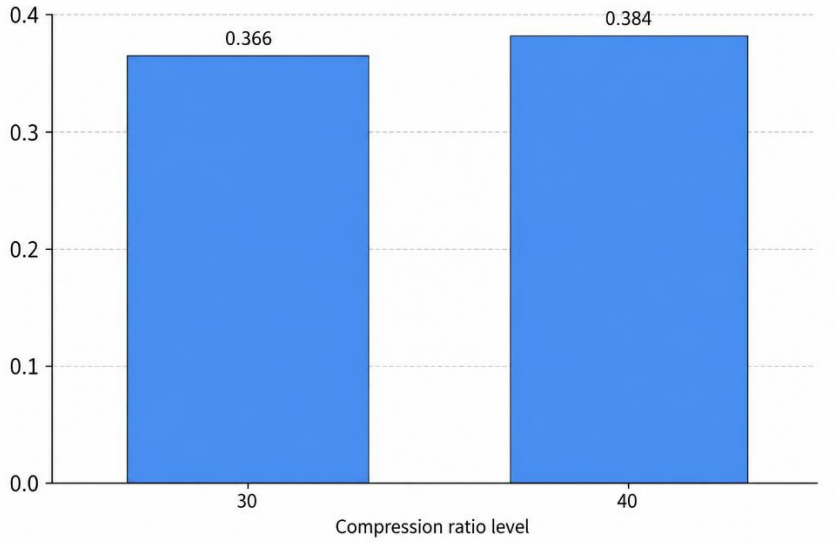

Figure 4. Mean ROUGE-1-F scores for different compression ratios (CR) levels

Factor Combination

In table 9 presents the results of Tukey's HSD post-hoc comparison test, applied to the four-way interaction among preprocessing techniques, word embedding models, ranking algorithms, and compression ratio levels. There were 32 combinations that were considered, and a row in the table reflects a specific combination of text summarization systems, and they were evaluated based on the quality of summarization. The statistical relationships among these combinations are represented by letters, whereby combinations that have at least one letter represent no statistically significant differences, and combinations that have no letters represent statistically significant differences. The outcomes are categorized into eight different statistical categories (A, AB, ABC, ABCD, BCD, CD, D, and E). The letters will denote the separate levels of statistical performance, with the highest level of performance (A) indicating a strict level of performance, and the lowest level of performance (E) is illustrated. It is noted that the combinations with the highest performance in the initial three rows (Group A) had the highest level of performance. All these combinations share the use of GloVe word embedding and a 40% compression ratio, demonstrating the importance of these two factors in improving summary quality.

The following five combinations (group AB) demonstrated relatively high performance, all using GloVe and a 40% compression ratio, with some variations in preprocessing methods and ranking algorithms. This means a steady performance in case the underlying factors are held constant. The average-performance group is made of an approximation of 15 combinations (groups ABC, ABCD, and BCD). Variations in performance are also noted in these combinations because of differences in the factors, which include the type of representation or the compression ratio, which produce similar though less predictable results as compared to those of the higher performing combinations. These combinations in different groups signify that performance has been statistically overlapped. The findings further suggest that the application of large contextual models like AraBERT is not always associated with better performance in graph-based summarization because the performance is evidently affected by other factors, including the ranking algorithm used and the compression ratio. Furthermore, the table shows that reducing the compression ratio to 30% (group CD) resulted in a relative decrease in summary quality. The majority of combinations at this ratio were concentrated within the average-performing groups, indicating that reducing the number of selected sentences limits the system's ability to maintain coverage and summary quality.

Next, the least influential combinations, which include eight systems, exhibited a significant decline in summary quality. These results indicate that using AraBERT with low compression ratios negatively affects the quality of the resulting summaries.

Finally, the lowest-performing combination (group E) achieved significantly lower summary quality. This result indicates that using a low-performance preprocessing technique like Stanza with a contextual

model like AraBERT can significantly reduce summary quality. This proves that in order to realize the best performance in the extractive summarization using graphs, close integration and balance of all the factors is needed because none of them is capable of guaranteeing high-quality results.

Table 6. Tukey test pairwise comparison based on all experiments

Preprocessing_Word embedding_Ranking algorithm_Compression ratio level	Grouping Letters
Khoja_GloVe_HITS_40%	A
Khoja_GloVe_PageRank_40%	A
Farasa_GloVe_HITS_40%	A
Farasa_GloVe_PageRank_40%	AB
Qalsadi_GloVe_HITS_40%	AB
Stanza_GloVe_HITS_40%	AB
Qalsadi_GloVe_PageRank_40%	AB
Stanza_GloVe_PageRank_40%	AB
Khoja_AraBERT_HITS_40%	ABC
Qalsadi_AraBERT_HITS_40%	ABCD
Farasa_AraBERT_PageRank_40%	ABCD
Stanza_AraBERT_HITS_40%	ABCD
Khoja_GloVe_PageRank_30%	ABCD
Stanza_GloVe_HITS_30%	ABCD
Farasa_GloVe_HITS_30%	ABCD
Farasa_AraBERT_HITS_40%	ABCD
Khoja_GloVe_HITS_30%	ABCD
Farasa_GloVe_PageRank_30%	ABCD
Khoja_AraBERT_PageRank_40%	ABCD
Qalsadi_GloVe_PageRank_30%	ABCD
Qalsadi_GloVe_HITS_30%	ABCD
Stanza_GloVe_PageRank_30%	ABCD
Qalsadi_AraBERT_PageRank_40%	BCD
Khoja_AraBERT_HITS_30%	CD
Stanza_AraBERT_PageRank_30%	CD
Farasa_AraBERT_HITS_30%	CD
Qalsadi_AraBERT_HITS_30%	CD
Khoja_AraBERT_PageRank_30%	CD
Stanza_AraBERT_HITS_30%	CD
Farasa_AraBERT_PageRank_30%	CD
Qalsadi_AraBERT_PageRank_30%	D
Stanza_AraBERT_PageRank_40%	E

CONCLUSION AND FUTURE WORK

The increasing demand for efficient automatic text summarization systems highlights the need for systematic evaluations of design choices, particularly for Arabic language processing. This study presented a comprehensive experimental analysis of graph-based extractive Arabic text summarization, examining the effects of preprocessing techniques, word embedding models, sentence-ranking algorithms, and compression ratio levels.

The experimental results of a total of 32 experiments demonstrated that all examined factors significantly influence summarization performance. Stemming- and lemmatization-based preprocessing techniques, including Khoja, Farasa, and Qalsadi, exhibited comparable effectiveness, while the lemmatization-based Stanza method yielded lower performance. The experiment also revealed that the graph-based extractive setting methods were always better with the use of the static word embedding model, namely, GloVe, compared to the contextual AraBERT model. Moreover, HITS rating system was superior to PageRank and greater compression ratios (40%) were observed to be more informative in terms of summaries through improved ROUGE-1 F scores. Tukey's HSD four-way interaction test showed that the best-performing combination in this study was Khoja's preprocessing with GloVe embeddings using the HITS algorithm at 40% compression, followed by Khoja with GloVe and PageRank at 40% compression, and Farasa with GloVe and HITS at 40% compression. On the other hand, the poorest performing one

was Stanza preprocessing using AraBERT embeddings and a PageRank algorithm, with a compression ratio of 40%, as it showed the lowest quality of the summary. These findings indicate that the understanding of optimal performance is the painstaking combination of preprocessing methods, word embedding models, ranking schemes, and compression ratios since the selection of one of the factors has a critical effect on the quality of the acquired summary. Furthermore, the statistically-based evaluation techniques allowed the reliable comparison of each factor separately and the interaction effects, which led to a better perception of how the system behaves.

Nevertheless, a number of limitations are to be taken into account. To begin with, the dataset employed, the Essex Arabic Summary Corpus (EASC), is relatively small, and this might interfere with the generalizability of the results. Future research ought to embrace the use of bigger and more heterogeneous datasets with the view to determining the scalability of the proposed methods. Secondly, the assessment was based only on the ROUGE-1 F-score that mostly represents unigram overlap and might not be a complete measure of the quality or coherence of the summaries. It would be better to include more evaluation metrics, including ROUGE-L or human evaluation, to have a more comprehensive view of the system's performance. Irrespective of these limitations, this study provides valuable data on the best preprocessing, word embedding methods, ranking algorithms, and compression ratios to determine the optimal method of generating Arabic text summaries. There are various areas of significance that the research will focus on in the future in order to improve the proposed framework. The validation on larger and more heterogeneous corpora will be performed first in order to evaluate the scalability and generalizability of the model to other areas and Arabic dialects. Moreover, the human assessment will be introduced to supplement the automated measures to give a more detailed analysis of summary coherence, readability, and quality in general. Moreover, the framework will be presented in cases of multilingual summarization where the application of the methodology will be applied to other languages, and, thereby, the framework will have a greater potential impact and applicability in cross-linguistic text summarization. Such developments will also help in increasing the strength, flexibility, and human-oriented summarization systems.

REFERENCES

[1] El-Kassas WS, Salama CR, Rafea AA, Mohamed HK. Automatic text summarization: A comprehensive survey. Expert systems with applications. 2021 Mar 1; 165:113679. https://doi.org/10.1016/j.eswa.2020.113679

[2] Sharma KP, Yajid MS, Gowrishankar J, Mahajan R, Alsoud AR, Jadhav A, Singh D. A systematic review on text summarization: techniques, challenges, opportunities. Expert Systems. 2025 Apr;42(4): e13833. https://doi.org/10.1111/exsy.13833

[3] Watanangura P, Vanichrudee S, Minteer O, Sringamdee T, Thanngam N, Siriborvornratanakul T. A comparative survey of text summarization techniques. SN Computer Science. 2023 Dec 2;5(1):47. https://doi.org/10.1007/s42979-023-02343-6

[4] Abdelqader KJ, Mohamed A, Shaalan K. Systematic review of automatic Arabic text summarization techniques. InInternational conference on Variability of the Sun and sun-like stars: from asteroseismology to space weather 2023 (pp. 783-796). Springer, Singapore. https://doi.org/10.1007/978-981-99-3416-4_63

[5] Alami N, Meknassi M, En-nahnahi N, El Adlouni Y, Ammor O. Unsupervised neural networks for automatic Arabic text summarization using document clustering and topic modeling. Expert Systems with Applications. 2021 Jun 15; 172:114652. https://doi.org/10.1016/j.eswa.2021.114652

[6] Elsaid A, Mohammed A, Ibrahim LF, Sakre MM. A comprehensive review of arabic text summarization. IEEE Access. 2022 Mar 30; 10:38012-30. https://doi.org/10.1109/ACCESS.2022.3163292

[7] Abdulateef S, Khan NA, Chen B, Shang X. Multidocument Arabic text summarization based on clustering and Word2Vec to reduce redundancy. Information. 2020 Jan 23;11(2):59. https://doi.org/10.3390/info11020059

[8] Elbarougy R, Behery G, KHATIB AE. Graph-Based Extractive Arabic Text Summarization Using Multiple Morphological Analyzers. Journal of Information Science & Engineering. 2020 Mar 1;36(2). https://doi.org/10.6688/JISE.202003_36(2).0013

[9] Elsaid A, Mohammed A, Ibrahim LF, Sakre MM. A comprehensive review of arabic text summarization. IEEE Access. 2022 Mar 30; 10:38012-30. https://doi.org/10.1109/ACCESS.2022.3163292

[10] Elbarougy R, Behery G, El Khatib A. Extractive Arabic text summarization using modified PageRank algorithm. Egyptian informatics journal. 2020 Jul 1;21(2):73-81. https://doi.org/10.1016/j.eij.2019.11.001

[11] Alselwi G, Taşcı T. Extractive arabic text summarization using pagerank and word embedding. Arabian Journal for Science and Engineering. 2024 Sep;49(9):13115-30.

https://doi.org/10.1007/s13369-024-08890-1

[12] Qaroush A, Farha IA, Ghanem W, Washaha M, Maali E. An efficient single document Arabic text summarization using a combination of statistical and semantic features. Journal of King Saud University-Computer and Information Sciences. 2021 Jul 1;33(6):677-92. https://doi.org/10.1016/j.jksuci.2019.03.010

[13] Elbarougy R, Behery G, KHATIB AE. Graph-Based Extractive Arabic Text Summarization Using Multiple Morphological Analyzers. Journal of Information Science & Engineering. 2020 Mar 1;36(2). https://doi.org/10.6688/JISE.202003_36(2).0013

[14] Chouikhi H, Alsuhaibani M. Deep transformer language models for Arabic text summarization: A comparison study. Applied Sciences. 2022 Nov 23;12(23):11944. https://doi.org/10.3390/app122311944

[15] Burmani N, Alami H, Lafkiar S, Zouitni M, Taleb M, Nahnahi NE. Graph based method for Arabic text summarization. In2022 International Conference on Intelligent Systems and Computer Vision (ISCV) 2022 May 18 (pp. 1-8). IEEE. https://doi.org/10.1109/ISCV54655.2022.9806127

[16] Aarab A, Oussous A, Saddoune M. A review on recent arabic information retrieval techniques. EAI Endorsed Transactions on Internet of Things. 2022 Oct 1;8(3): e5. https://doi.org/10.4108/eetiot. v8i3.2276

[17] Al-Shammari E, Lin J. A novel Arabic lemmatization algorithm. InProceedings of the second workshop on Analytics for noisy unstructured text data 2008 Jul 24 (pp. 113-118). https://doi.org/10.1145/1390749.1390767

[18] Matrane Y, Benabbou F, Ellaky Z, Zaoui C. An Automatic Stop Words Removal in Maghrebi Arabic Dialect Text Classification Using Part of Speech Tagging. InThe Proceedings of the International Conference on Smart City Applications 2025 Oct 1 (pp. 187-196). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-88653-9_19

[19] AL-Khassawneh YA, Hanandeh ES. Extractive Arabic text summarization-graph-based approach. Electronics. 2023 Jan 14;12(2):437. https://doi.org/10.3390/electronics12020437

[20] Al-Numai A, Azmi A. Lemma-rouge: An evaluation metric for arabic abstractive text summarization. The Indonesian Journal of Computer Science. 2023 Apr 30;12(2):470-81. https://doi.org/10.33022/ijcs.v12i2.3190

[21] Wazery YM, Saleh ME, Alharbi A, Ali AA. Abstractive Arabic text summarization based on deep learning. Computational Intelligence and Neuroscience. 2022;2022(1):1566890. https://doi.org/10.1155/2022/1566890

[22] Alayba AM. Arabic Natural Language Processing (NLP): A Comprehensive Review of Challenges, Techniques, and Emerging Trends. Computers. 2025 Nov 15;14(11):497. https://doi.org/10.3390/computers14110497

[23] Sousa N, Oliveira N, Praça I. Machine reading at scale: A search engine for scientific and academic research. Systems. 2022 Apr;10(2):43. https://doi.org/10.3390/systems10020043

[24] Gammarano ID, Arruda Filho EJ. Human vs virtual influencers: what elements influence the followers in the hyperconnected environment. Qualitative Market Research: An International Journal. 2025 Nov 18;28(5):871-913. https://doi.org/10.1108/QMR-04-2024-0062

[25] Qaroush AM, Naser L, Mali M, Naji A. Semantic-aware hybrid graph-based extractive summarization for arabic texts. Journal of King Saud University Computer and Information Sciences. 2025 Dec;37(10):349. https://doi.org/10.1007/s44443-025-00359-x

[26] Hatch BT, Richardson SD. Semitic Root Encoding: Tokenization Based on the Templatic Morphology of Semitic Languages in NMT. InProceedings of The Third Arabic Natural Language Processing Conference 2025 Nov (pp. 26-41). https://doi.org/10.18653/v1/2025.arabicnlp-main.3

[27] Mohammed ZK, Abdullah NA. Survey for Arabic part of speech tagging based on machine learning. Iraqi Journal of Science. 2022 Jun 30:2676-85. https://doi.org/10.24996/ijs.2022.63.6.33

[28] Bourahouat G, Abourezq M, Daoudi N. Word embedding as a semantic feature extraction technique in arabic natural language processing: an overview. The International Arab Journal of Information Technology. 2024 Mar 1;21(2):313-25.

[29] Bilal ZS, Gargouri A, Mahmood HF, Mnif H. Comparison of Collective Diverse Arabic Sign Language Dataset. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 2024;15(4):133-50. https://doi.org/10.58346/JOWUA.2024.I4.009

[30] Alrashidi B, Jamal A, Alkhathlan A. Abusive content detection in Arabic tweets using multi-task learning and transformer-based models. Applied Sciences. 2023 May 9;13(10): 5825. https://doi.org/10.3390/app13105825

[31] Bordbar S, Shirazi AH. Satire Terminology in Persian and Arabic literature. International Academic Journal of Innovative Research. 2019;6(1):66-76. https://doi.org/10.9756/IAJIR/V6I1/1910006

[32] Krishnaveni P, Balasundaram SR. Automatic Text Summarization by Providing Coverage, Non-Redundancy, and Novelty Using Sentence Graph. Journal of Information Technology Research (JITR). 2022 Jan 1;15(1):1-8. https://doi.org/10.4018/JITR.2022010108

[33] Egger R. Text representations and word embeddings: Vectorizing textual data. InApplied data science in tourism: Interdisciplinary approaches, methodologies, and applications 2022 Jan 31 (pp. 335-361). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-88389-8_16

[34] Sabbeh SF, Fasihuddin HA. A comparative analysis of word embedding and deep learning for Arabic sentiment classification. Electronics. 2023 Mar 16;12(6):1425. https://doi.org/10.3390/electronics12061425