IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 16, 2020, accepted December 19, 2020, date of publication December 22, 2020,
date of current version December 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3046494

Extractive Multi-Document Arabic Text Summarization Using Evolutionary Multi-Objective Optimization With K-Medoid Clustering

RANA ALQAISI, WASEL GHANEM, (Member, IEEE), AND AZIZ QAROUSH
Department of Electrical and Computer Engineering, Birzeit University, Birzeit 71939, Palestine
Corresponding author: Wasel Ghanem (ghanem@birzeit.edu)

ABSTRACT The increasing usage of the Internet and social networks has produced a significant amount of online textual data. These online textual data led to information overload and redundancy. It is important to eliminate the information redundancy and preserve the time required for reading these online textual data. Thus, there is a persistent need for an automatic text summarization system, which extract the relevant and salient information from a collection of documents, that sharing the same or related topics. Then, presenting this extracted information in a condensed form to preserve the main topics. This paper proposes an automatic, generic, and extractive Arabic multi-document summarization system. The proposed system employs the clustering-based and evolutionary multi-objective optimization methods. The clustering-based method discovers the main topics in the text, while the evolutionary multi-objective optimization method optimizes three objectives based on coverage, diversity/redundancy, and relevancy. The performance of the proposed system is evaluated using TAC 2011 and DUC 2002 datasets. The experimental results are compared using ROUGE evaluation measure. The obtained results showed the effectiveness of the proposed system compared to other peer systems. The proposed system outperformed other peer systems for all ROUGE metrics using TAC 2011. We achieved an F-measure of 38.9%, 17.7%, 35.4%, and 15.8% for Rouge-1, Rouge-2, Rouge-L, and Rouge-SU4, respectively. In addition, the proposed system with DUC 2002 dataset achieved an F-measure of 47.1%, 23.7%, 47.1%, 20.4% for Rouge-1, Rouge-2, Rouge-L, and Rouge-SU4, respectively.

INDEX TERMS Natural language processing, extractive text summarization, multi-objective optimization, maximum coverage and relevancy, less redundancy.

I. INTRODUCTION

The significant amount of the information on the Internet, such as the news articles posted on the websites, has increased the complexity of extracting useful information. In addition, online forums and social networks have become the most popular platform for users to share their experiences. Nowadays, People find it distributive to read many articles with redundant information. Thus, it is important to have an automated summarization system, that can help in identifying the most important and salient information quickly. Automatic summarization systems have been applied for different domains including search engines, web pages, news, and all forms of online reviews. For example, Qumsiyeh and Ng [1] proposed a query-based summarizer to enhance the web search engine results, Modaresi *et al.* [2] presented a study that shows the effect of using query-based extractive summarization approach for media monitoring and media response analysis.

Text Summarization is one of the most important applications of Natural Language Processing (NLP). It aims to create a shorter version from one or more related text documents while preserving the content and overall meanings. Summarization methods can be classified based on the input, approach, language, generality, and output as shown in Figure 1 [3], [4]. The summarization systems are

The associate editor coordinating the review of this manuscript and approving it for publication was Kathiravan Srinivasan.

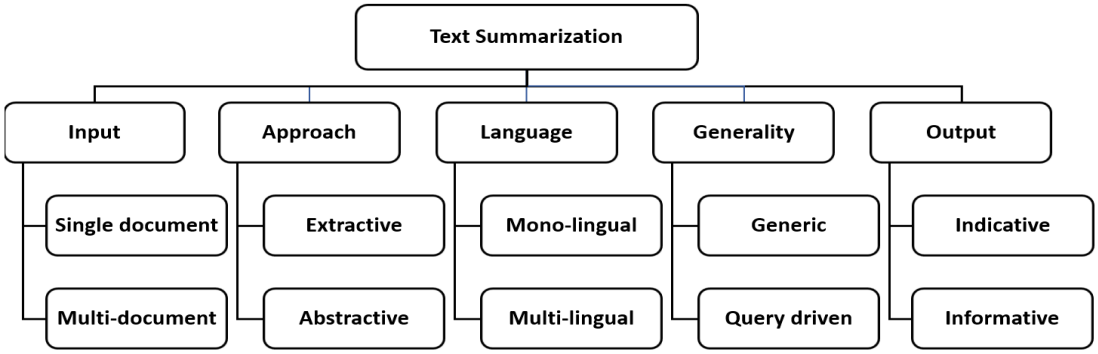


FIGURE 1. Categories of text summarization.

classified based on the input into a single document or multi-document summarization. Single-document summarization tries to summarize a single document, while a set of related documents from different sources is processed in multi-document summarization. Thus, a single document does not exhibit inconsistency problems, because it has only one author or group of authors, who wrote it according to a common consensus [3], [4]. However, a set of problems in multi-document summarization is raised such as inconsistency, redundancy, and conflicting ideas by the different authors. As a result, dealing with multi-document summarization is harder than single-document [3]–[5]. According to the language, summarization systems are classified as monolingual or multilingual. In monolingual summarization systems, all documents have the same language, while in multilingual different languages can be seen in the input documents and the output summaries [3], [4]. Also, the summary may be generic which addresses the whole community of readers, or query-driven which focuses on the important topics related to a user's query. The output is an important parameter in classifying the summarization system into informative or indicative. Informative summaries cover the content of all topics appeared in the input documents, while indicative summaries present the general idea of the source text to the user, and highlight the topics addressed in the text [3], [4].

Finally, the generated summary may be extractive or abstractive. In extractive summarization, the summary is formed by selecting the important sentences based on statistical and linguistic features, and presenting them in the form of summary to the user. In contrast, the abstractive summary depends on understanding the text using the NLP techniques to generate novel sentences that hold the main ideas appeared in the source text [3], [4]. Despite that abstractive summaries are more readable and similar to human summaries, it needs deep knowledge of the text and lexical resources such as parsers and language generators. In traditional text summarization approaches, researches focus mainly on extractive text summarization [6], [7]. On the other hand, neural-based techniques such as transfer learning were employed in abstractive text summarization and produces fairly good results. Extractive summarization selects the most important sentences based on a predefined set of features, then those sentences are combined to from the summary. With multi-document summarization, the redundancy issue is raised since sentences are extracted from different documents. Thus, a technique is needed to handle the redundancy issue. Moreover, with limited summary length, and many important sentences, a strategy is needed to select the best summary rather than the best sentences. Selecting the best summary which contains the most important sentences with maximum coverage and minimum redundancy is considered a global optimization problem [8].

Arabic NLP is considered much more complex than English language and other European languages. The main reason for this complexity is the nature of Arabic language which is highly derivational and has rich morphology. Thus, Arabic NLP has many challenges that prevent the advance of research compared to other languages, which include the following [9], [10]: i) Arabic language is highly derivational and inflectional, this highly affects NLP task such as stemming and lemmatization, ii) the absence of diacritics in written documents, where diacritics play an important rule in determining the word meaning and ease the task of tokenization and parsing the text, iii) no capitalization in Arabic language which hardens the identification of proper nouns, titles, and abbreviations. This also affects the task of named-entity recognition, iv) and the lack of resources such as lexicons and NLP tools.

Most of the available summarization approaches have targeted the English language and other European languages, while little works have been introduced in Arabic language. In addition, most of the previous related approaches deal with redundancy and coverage as a single objective, which represented as a weighted sum of these two objectives making the solution not pure multi-objectives optimization. At the same time, the sentence relevancy or score objective which include important features such as sentence location and sentence length is ignored in such optimization systems. Moreover, in clustering-based approach, which is widely used with multi-document summarization to eliminate the redundancy, most of these methods failed to consider the number of clusters which highly affects the coverage of the generated summary. In this paper, we propose an extractive Arabic multi-document summarization approach that employs

a clustering-based and an evolutionary multi-objective optimization methods. The proposed approach goes through a sequence of stages to select the sentences that form the summary. First, we applied a set of pre-processing operations including, tokenization, normalization, stop words removal and stemming to the set of related documents to transform the original text into a unified form. Followed by a set of informative features with novel representation were extracted from each sentence as a representation of sentence relevancy or score function. The next stage of the proposed approach uses the k-medoid clustering algorithm with a Silhouette method to identify the main topics appearing in the original set of documents. In the last stage of the proposed approach, the NSGA-II algorithm was adopted as a multi-objectives optimization process to simultaneously maximizes three stand alone objectives namely, coverage, relevancy, and diversity. We evaluated the proposed system on the DUC2002 and TAC 2011 data sets, and the results showed that our system outperforms other peer systems based on the ROUGE metrics. Hence, the main contributions of this paper include:

i) Studying the effect of using different tokenization and stemming methods in Arabic multi-document text summarization.
ii) Handling the Arabic multi-document summarization as a real multi-objective optimization problem that try to simultaneously optimize three separated objectives.
iii) Introducing sentence relevancy with novel features representation as a third objective to be maximized, which is to the best of our knowledge this work is the first one that try to simultaneously maximize diversity, coverage, and relevancy.
iv) The evaluation results showed that our proposed approach outperforms other peer systems in terms of precision, recall, and F-measure.

The rest of the paper is organized as follows: section 2 presents the related work around the multi-document summarization. Section 3 presents the proposed methodology. In section 4, experiments and results are illustrated. Finally, section 5 concludes the work and presents the future work.

II. RELATED WORK

In the literature, text summarization can be classified into two main approaches: traditional/classical and deep learning approaches.

A. CLASSICAL APPROACHES

In Extractive text summarization, classical approaches are further classified into two approaches to select the most relevant sentences the greedy approach which selects one sentences at a time and the global approach which searches for the best summary instead of the best sentences. The optimization process is considered an NP-hard problem, and it is necessary to approximate the solution using meta-heuristics techniques such as Genetic Algorithms (GA) and population based methods [11], [12]. Several techniques are proposed in the literature for both greedy and global text summarizing approaches.

1) GREEDY-BASED TEXT SUMMARIZATION

In this approach, only one sentence at a time is chosen based on a predefined set of features to be included in the output. This approach is considered fast and simple, but it barely produces the best summary where the generated summary may suffer from data redundancy. Many techniques are proposed in this approach such as statistical-based, and machine learning-based approach.

a: STATISTICAL-BASED APPROACH

Statistical methods are widely used in text summarization which are based on the concept of relevance score and Bayesian classifier [13]. In this approach, a set of features like Term Frequency (TF), keyphrases, sentence length, and position are used to reflect the importance of each sentence in the original text [14]–[19]. Statistical methods are used for both single and multi-document summarization. Also, it can be used to enhance the selection of important sentences or the elimination of redundant sentences. However, it fails to understand the text, since it only depends on statistical measures [14].

b: MACHINE LEARNING BASED APPROACH

In this approach, text summarization is considered as a binary classification problem, where a set of documents and their extractive summaries are used as a training set, and each sentence is classified as a summary sentence or non-summary based on statistical, semantic features or a combination of them [20]–[23]. According to Nenkova, A. *et al.* [23], machine learning approaches are well suited for single document more than multi-document summarization. Moreover, studies have shown the effectiveness of this approach [24]. However, this approach needs labeled data (training dataset), and the creation of such dataset is time-consuming task. Also, the generated summary may suffer from redundancy.

2) GLOBAL-BASED TEXT SUMMARIZATION

On the other hand, this approach searches for the best summary rather than the bests sentences. This approach produces better summary than greedy approach, but it is more complicated and time consuming. Many techniques are proposed in this approach including graph-based, cluster-based, lexical and semantic-based, discourse theory, and an optimization-based approach.

a: GRAPH BASED APPROACH

In this approach, each document is represented as directed graph G=(V, E), where V represents the set of vertices, and E is the edge between two vertices. Each sentence of the document is a node (vertex) in the graph, and an edge connects two sentences if there is a relation between them. The weight of the edge corresponds to the similarity between these two sentences. The cosine similarity is

widely used to measure the relation between two sentences, and an edge exists between two nodes if their similarity is greater than a predefined threshold [25]–[28], [30]. The document's sub-graphs represent the different topics covered in the document, so this approach works fine for both query-based and generic-based summaries. For query-based summaries, sentences are only connected from pertinent sub-graph, while for generic summaries sentences are selected from each sub-graph for best coverage [29]. However, the graph-based approach fails to understand the text since it depends only on statistical measures.

b: CLUSTER-BASED APPROACH

This approach is used to group similar objects in one cluster, while dissimilar ones into different clusters. Each object represents a sentence, and the cluster is a set of related sentences. The cosine similarity is widely used to measure the similarity between two sentences, where each sentence is represented generally using Term Frequency-Inverse Document Frequency (TF-IDF) vector [25], [26]. Clustering approaches can be classified as agglomerative, and partitional based on the initial state. Agglomerative clustering is a bottom-up approach and it represents each sentence as a cluster then tries to merge similar clusters until stopping criteria. On the other hand, partitional clustering starts with one cluster that contains all sentences, then tries to divide it into different clusters. The k-means is considered the common partitional clustering algorithm [26], [31]–[36]. This approach is widely used in multi-document text summarization since similar sentences from different documents are grouped into the same cluster. Thus, the selection will be one sentence from many similar ones, as a result, this will reduce the redundancy. However, it generates an un coherent summary, since it is based on statistical measures and cannot capture contextual information [26].

c: LEXICAL AND SEMANTIC BASED APPROACHES

The aim of these approaches is to find relations between sentences. Many techniques exist in the state of art, including textual entailment, semantic clustering, co-reference, and lexical chains and semantic [37]–[41]. Text entailment has used to determine if a sentence can infer the meaning of another one. Only sentences that are not inferred by any other sentences are included in the summary. Also, lexical cohesion is used to determine the important sentences and how it contributes to the summary with cosine similarity to reduce the redundancy. Also, the root and semantic relations between senses of words are used in to extract the common words [38]. Also, ontologies are used to capture the semantic information of a specific domain e.g. Arabic WordNet (AWN) is a form of ontologies, that groups synonym words into sets, and records the different semantic relations into these sets. Moreover, Imam *et al.* [39] used the AWN to expand the user's query and adding the knowledge base of a specific domain, then the decision tree algorithm is used to generate the summary. This approach can produce a coherent, non-redundant, and informative summaries. However, ontologies and NLP resources are not available for all domains which are used to capture the semantic and lexicon relations. Moreover, constructing these resources manually is a time-consuming task.

d: DISCOURSE THEORY

Discourse theory is represented by a set of approaches to produce more informative and representative summaries by describing the relations between text units. These approaches include Rhetorical Structure Theory (RST) [42]–[45], Cross-document Structure Theory (CST) [46], and Segmented Discourse Representation Theory (SDRT) [47]. RST describes the main aspects of the text and the relations between sentences. It represents the coherent text as a tree of a nuclear node which represents an important proposition, and satellite which is considered as additional information. On the other hand, CST describes the semantic connection among units of related texts. It is widely used in multi-document summarization, and it represents the coherent text as a graph. Also, SDRT allows attachment between non-adjacent discourse units and for multiple attachments to a given discourse unit, and it represents the discourse structures as an acyclic graph. This approach produces more informative and coherent summaries, since it is based on analyzing the relations between text units. However, it fails to deal with multi-document issues such as redundancy elimination.

e: OPTIMIZATION-BASED APPROACH

Multi-document summarization is considered by many researchers as an optimization problem, where a set of objectives are considered to produce a good summary, including maximum coverage, minimum redundancy (maximum diversity), coherence, and balance. Coverage means that a summary should contain all important aspects that appear in the documents, while diversity aims to reduce the similar sentences in the output summary. On the other hand, coherence aims to generate a coherent text flow. Moreover, balance means that a summary should have the same relative importance of different aspects in the original documents [48]–[52].

Optimization algorithms divided into single-objective and multi-objectives optimization. Single-objective optimization aims to find the best solution that minimizes or maximizes a single objective which accumulates all objective functions into one. Many algorithms are used to solve the single-objective optimization problems such as Particle Swarm Optimization (PSO) [53], [54], binary differential evolution algorithm [55], and Cuckoo search approach [57] which is used to generate a summary that maximizes coverage, cohesion, and readability together [57]. On the other hand, in Multi-objective optimization more than one objective function are optimized simultaneously. Recently, multi-objective evolutionary algorithms have attracted a lot of researches by their ability to approximate a set of Pareto solutions (non-dominated solutions) [58] such as Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [8], [56], Multi-Objective Artificial Bee Colony [59], and Ant

Colony optimization [11]. The results of this approach are very promising. Moreover, there are little researches conducted on the Arabic language. However, it needs a adequate and accurate formulation of the objective functions.

B. DEEP LEARNING APPROACHES

Due to the evolve of deep learning techniques, neural-based text summarization has attracted considerable attention. Compared to classical method of text summarization, deep learning methods achieved better results with less human intervention [60]. However, deep learning text summarization requires a large-scale structured training data set. Generally, most of the deep learning text summarization (extractive or abstractive) follow similar pipeline of (i) representing words as continuous vector using word embeddings such as Word2vec and GloVe, (ii) encoding sentence or document using word embeddings which can be used as encoders for extracting sentence features, and (iii) the sentence or document representations are then fed to a regressors model for ranking or selection (extractive summarization) or decoder model for generation (abstractive summarization) [60].

Neural-based text summarization models as well as deep learning techniques were employed recently in both extractive and abstractive text summarization [61], [62]. Extractive text summarization is a selection-based method which require handling two main challenges, sentence representation and sentence ranking and selection considering maximizing coverage and diversity. Different neural-based extractive text summarization models are presented recently in literature. They are spanning a large range of approaches [63] such as encoder-decoder framework using Recurrent Neural Network (RNN) [64], Transformers [65], or Gated Recurrent Unit (GRU) networks [66] as encoders, or non-auto regressive [67] or auto regressive as decoders [68].

On the other hand, abstractive summarization focuses on capturing the salient features of the text or the meaning of the text and then generate an abstractive summary like human-generated summaries based on this representation. Different deep learning models were used for abstractive text summarization where sequence-to-sequence using encoder-decoder architectures based on RNNs has become the dominant framework [69]–[72]. In this framework, the encoder is responsible for representing token in the input source, while the decoder is responsible for generating words that form the summary and this is dependent on the vector representation returned by the encoder. In order to find the best sequence of the words that form the summary, a beam search algorithm is commonly used. The RNNs of the encoder and decoder can be implemented with bidirectional RNN, attention mechanisms, Elman RNN, Long-Short Term Memory (LSTM), GRU networks, or using Transformers [60]–[62], [69], [70].

The main challenges of abstractive text summarization based on deep learning, in general, is the lack of the quality of the reference summary (Golden summary) as well as the quality of datasets [70], [71]. For example in the Arabic language, there is no multi-sentence dataset for abstractive text summarization. Another challenge is the use of ROUGE in evaluation is not enough, especially when measuring relevance, and readability, as ROUGE depends on exact matching between words, while abstractive summarization may rephrase the original words and use different words with the same meaning. Further, abstractive summarization may generate also fake facts, as 30% of summaries generated using this technique undergoes from this problem. Other challenges like summary sentence repetition, sentence inaccuracy are also reported [61], [62], [69], [70].

In summary, several summarization approaches are proposed in the literature and each one has it owns limitations. For example, statistical and graph-based approaches depend on statistical measures, so that it fails to understand the meaning behind the text. In contrast, lexical and semantic approaches can handle linguistic features. However, these approaches highly depend on the ontologies and NLP resources, which are not available for all domains, and constructing them manually is a time consuming task. For clustering-based approach, it is widely used with multi-document summarization to eliminate the redundancy. However, clustering techniques have many issues that affect the quality of the generated summary including the number of clusters, how to order them, how to select sentences, and finally how to merge the selected sentences to form the summary. These parameters are rarely considered together by researchers. Regarding multi-objective optimization approach used for Arabic multi-document summarization, all systems deal with the contradictory objectives using the weighted sum approach. Also, the sentence's score is ignored in such systems, while it plays an important role to spur sentences that are important and not similar to other sentences to appear in the output summary. Finally, although recent neural-based summarization achieved better results with less human interaction compared to traditional methods. However, these methods requires a large-scale structured data. In addition, these techniques have several challenges related to the generated summary such as the stopping criterion of the summarization process, quality of the generated summary, and the evaluation of generated summary [3], [70].

III. PROBLEM DEFINITION AND FORMULATIONS

A. PROBLEM DEFINITION AND MATHEMATICAL REPRESENTATION

As an input, we have a set of topic-related documents collection $D = d_1, d_2, \ldots, d_{\mathrm{m}}$, where m represents the number of documents. Each document has a set of sentences $S_{di} = s_1, s_2, \ldots, s_n$, where n represents the number of sentences in document d_i. The goal is to generate a summary $\overline{D} \subset D$ (e.g. $\overline{D}$ represents a set of selected sentences from collection D) taking into account the following four text summarization objectives:

- **Relevance:** selecting the most relevant, important, or informative sentences (e.g. sentences with high score) from a set of topic-related documents collection.

- **Coverage:** the selected sentences should cover all important aspects (e.g. sub-topics) from topic-related documents collection as much as possible. In other words, the generated summary must include the information provided in the original documents set.
- **Redundancy:** the selected sentences shouldn't contain redundant information.
- **Length:** the generated summary should have abounded length (e.g summary ratio), which must be specified in advance to maximize coverage and minimize redundancy.

Extractive text summarization can be formalized as a global optimization problem, where the main goal is to find the subset of relevant sentences that cover the main sub-topics or contents with minimum information redundancy. Optimizing these objectives jointly is a challenging task. Thus, a multi-objective optimization method seems to be the natural way to handle this type of optimization problems.

B. SENTENCE REPRESENTATION

Sentence representation is one of the main tasks of natural language processing. It aims to encode sentence information into a real-valued representation vector. Several methods have been outlined in the literature(add the section) for sentence representation, such as TF-IDF and word embedding [73]. In a document or topic-related documents collection, each sentence is represented by a vector that is defined as a bag-of-words. Let $T = t_1, t_2, \ldots, t_{\mathrm{m}}$ represents all unique terms (e.g. words) occurred in a document D, where m is the number of unique terms. Using the vector space model representation, each sentence is represented by the weights of the terms that it contains, ignoring the order of the words and any punctuation. Each sentence $S_j = w_{\mathrm{j1}}, w_{\mathrm{j2}}, \ldots, w_{\mathrm{jk}}$ is represented as a vector in m dimensional space, where w_{jk} is the weight of term t_k in sentence S_j. Different weighting schemes are available in the literature. Here, the term weight is calculated using *Term Frequency - Inverse Sentence Frequency* scheme (TF-ISF) [74]. TF-ISF is a special version of TF-IDF. TF-IDF is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. On the other hand, TF-ISF is a numerical statistic that is intended to reflect how important a word is to a sentence in a collection or corpus. TF-IDF normalized by dividing on the total number of documents containing term k, while TF-ISF normalized by dividing on the total number of sentences containing term k, which is what we need as away of measuring sentence relevancy. TF-ISF scheme combines term frequency along with inverse sentence frequency, to produce a composite weight for each term in each sentence. Indeed, the *TF* is used to measures the local importance of the term in a given sentence (how many times a term appears in a sentence), while the *ISF* is used to measure the global importance among all sentences in the document (how many sentences of the document contain the term). The *TF-ISF* weight of term t_k in sentence S_j is calculated as follows:

$$W_{kj} = TF_{kj} \times log_2 \frac{N}{n_k} \tag{1}$$

where TF_{kj} is the number of occurrences of term t_k in sentence S_j, N is the total number of sentences, and n_k is the number of sentences containing term t_k. The weight will be higher when term t_k occurs many times within a small number of sentences n_k, lower when the term t_k occurs fewer times in a sentence S_j, or occurs in many sentences n_k, and lowest when the term occurs in all sentences ($n_k = N$). It is worth mentioning that, in general TF-ISF is outperformed by word embedding representation. However, word embeding need large structured dat and generally used with abstractive deep learning techniques. Besides, our approach is an extractive method which didn't need to understand the semantic information of the sentence. Moreover, we used Arabic WordNet along with TF-ISF for better sentence representation.

C. SIMILARITY MEASURE

There are several measurement are available to calculate the similarity between textual units (e.g. sentences) such as euclidean distance, cosine similarity, and Jaccard correlation [75]. However, cosine similarity is the most widely used [8], [53], [55], [57], [59], [76], [77]. The cosine similarity is used to measure the similarity between sentences by performing the inner product between their vectors, then the product normalized by the length of their vectors. Given two sentences S_i and S_j, where each sentence is represented by the vector space model and the TF-ISF weighting method, the cosine similarity is calculated using the following equation:

$$similarity(\vec{s_{\mathrm{i}}}, \vec{s_{\mathrm{j}}}) = \frac{\vec{s_{\mathrm{i}}} \cdot \vec{s_{\mathrm{j}}}}{|\vec{s_i}| \times |\vec{s_{\mathrm{j}}}|} = \frac{\sum_{k=1}^{m} w_{ik} w_{\mathrm{jk}}}{\sqrt{\sum_{k=1}^{m} w_{ik}{}^2 \cdot \sum_{k=1}^{m} w_{jk}{}^2}} \tag{2}$$

D. SENTENCE CLUSTERING

The general purpose of clustering is to group similar items (e.g. sentences) into one cluster, while dissimilar ones into different clusters. In text summarization, the aim of clustering is to find the main topics and sub-topics in the document or documents collection. Thus, each item is represented by a sentence, and the output cluster contains a set of related sentences, which represent a topic or sub-topic. The clustering algorithms partition the input data (e.g. sentences) into k clusters based on a similarity or dissimilarity measure. Given a set of sentences $S = s_1, s_2, \ldots, s_n$ (related to document or documents collection) represented as vectors. The goal is to partition these sentences into k clusters $(C_1, C_2, \ldots, C_{\mathrm{k}})$ considering the following five objectives [78], [79]:

1) Each cluster should have at least one sentence, $C_{\mathrm{p}} \neq \phi, \forall p \in (1, 2, \ldots, k)$.
2) Each sentence should definitely assigned to a cluster, $\bigcup_{p=1}^{k} C_{\mathrm{p}} = S$.

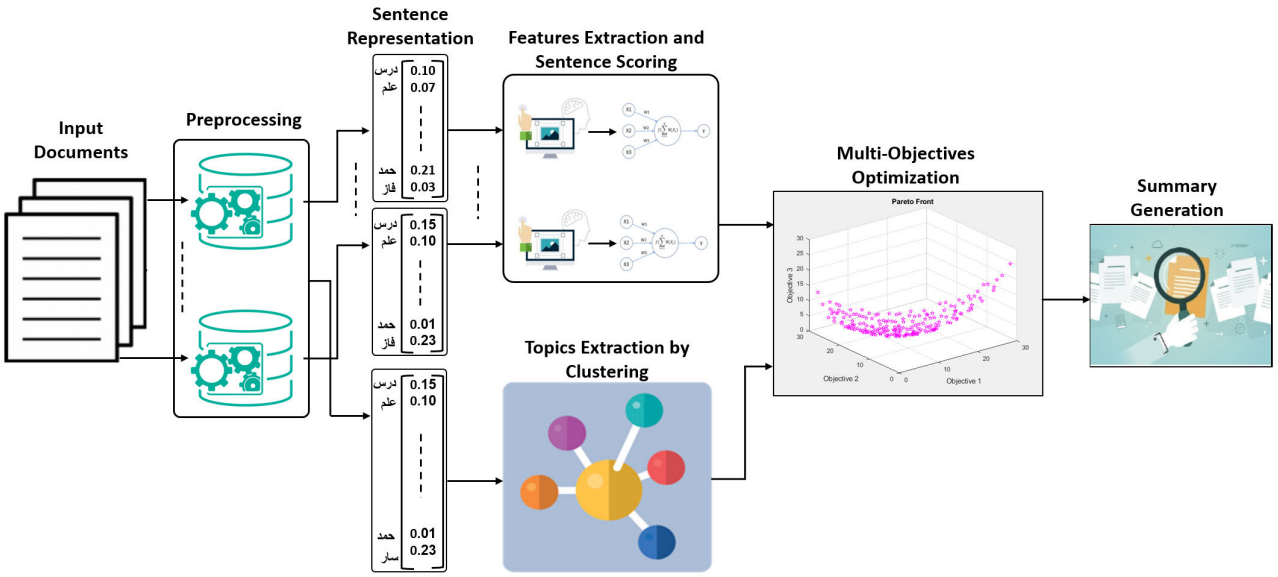


FIGURE 2. Flow of the main framework stages.

3) Different clusters should not have sentences in common, $C_p \cap C_q = \phi, \forall p, q \in (1, 2, \ldots, k)$.
4) Maximize similarity between sentences in the same cluster, $MAX(sim(s_i, centroid\ of\ C_p)), \forall s_i \in C_p$.
5) Minimize similarity between clusters, $MIN(sim(centroid\ of\ C_p, centroid\ of\ C_q)), \forall p, q \in (1, 2, \ldots, k)$.

Here, we represent each sentence by the vector space model and the TF-ISF weighting method. In addition, we use the cosine similarity measure to compute the similarity between sentences.

IV. PROPOSED TEXT SUMMARIZATION APPROACH

We present a generic, extractive, multi-document text summarization approach that employs an evolutionary multi-objective optimization and clustering-based approach. The system tries to extract the most important sentences that cover the main topics of the original source text while eliminating the redundant information from the generated summary. Figure 2 shows the stages of the proposed system. The input text documents transformed into a unified form by applying a set of text prepossessing tasks. In the next stage, each sentence is represented using a bag-of-words with the TF-ISF weighting method. Then, a set of informative features were extracted to express the importance of each sentence, followed by using clustering to identify the topics that appeared in the original text. In the next stage, the extractive summarization process is handled as a multi-objective optimization approach to simultaneously maximize coverage, diversity, and relevancy. In last, a set of sentences are selected to generate the summary. The next sections will describe the work-flow of the proposed approach in more detail.

A. TEXT PREPROCESSING

Preprocessing aims to handle some of the Arabic NLP challenges by transforming the original text into a unified form that facilitates working with the next stages such as computing sentence similarity and sentence score. Preprocessing aims to reduce the ambiguity of words and reduce inconsistency for a better word and sentence representation. This stage includes four sequenced methods, tokenization, normalization, stop word removal, and stemming [80], [81]. We relied on published recent studies to choose the best preprocessing techniques. Besides, in order to speed up the development process, we relied on well-known Arabic NLP tools which implements up to date preprocessing algorithms to handle these preprocessing methods. Moreover, we studied experimentally the effect of different tokenization and stemming techniques on text summarization. It is worth mentioning that, in this stage we did not handle issues like typos and mistakes.

1) TOKENIZATION

Tokenization aims to split the document into small units such as paragraphs, sentences, and words [82]. This task is highly related to the morphological analysis. Thus, it is a non-trivial task. Besides, things got worst when dealing with languages that have rich and complex morphology such as Arabic. Here, text tokenization is performed at two levels; sentence level to compute sentence score, which based on the punctuation marks ".,!,;, and ?" as a sentence delimiter, and at the word level to represent sentences as bag-of-words, using the white space as delimiter. In addition, we studied the effect of using semantic tokenization using the Stanford CoreNLP tool instead of relying on punctuation marks tokenization. This tokenization approach is usfull when the existence of

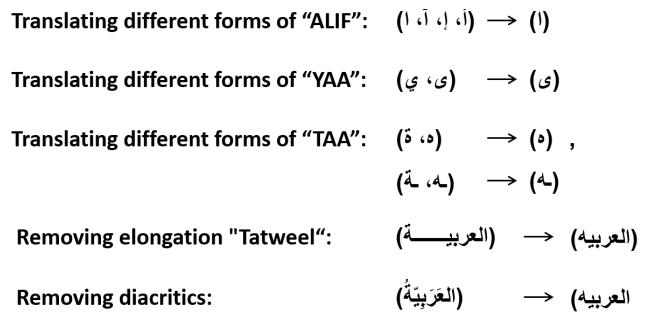


FIGURE 3. Examples on text normalization operations.

punctuation errors. Figure 4 shows an example of an input text and its tokenized version (sentence level) using punctuation marks and semantic tokenization. Note that, the punctuation marks tokenization returns two sentences while the semantic tokenizer produces one sentence.

2) NORMALIZATION

In the Arabic language, the same character or term can have a set of variations because of using Arabic dots and diacritics. Hence, characters may appear in different forms and can be used instead of other characters because they have similar shapes. This will affect sentence representation, computation of some important features such as term frequency, and the computation of text similarity. Thus, normalization is the process of making the text more consistent either by replacing (e.g. unifying the different forms of the same character to avoid variations) or removing (e.g. removing punctuation) [83]. Here, the normalization is done as follows: (i) removing punctuation marks, (ii) removing non Arabic words and non-Arabic letters such as special symbols, (iii) removing diacritics, (iv) removing elongation ''Tatweel'', and (v) translating different forms of ''ALIF'', ''TAA'', and ''YAA''. Figure 3 shows examples of theses normalization operations and Figure 4 shows the output produced by the normalization step.

3) STOP-WORDS REMOVAL

Stop-words (e.g. pronouns, prepositions, conjunctions, etc.) are frequently occurring words in natural languages. They are used to complete forming sentences by connecting their different parts together. Stop-words are not informative and don't help in identifying documents topics. Thus, they are considered as unimportant in some NLP applications like classification, clustering, summarization, etc [14]. Removing stop-words shortens the length of the document and can increase the performance since some of the measures are based on the words' frequencies in the sentence/document. Features will be simplified and become more relevant and accurate by removing stop words. In general, there is no uniform or general list of stop-words incorporated by all Arabic NLP tasks. Besides, some NLP task has its own domain of interest and thus it has its own preferred list of stop-words. Here, the general stop-words list [84], and the Khoja's stop-words list [85] are combined and used. Figure 4 shows the output after removing stop-words.

4) STEMMING

Arabic is a highly inflectional and derivational language characterized by a complex set of morphological features and grammatical rules. This means that Arabic words can have many different forms but share the same abstract meaning. This will eventually affect sentence representation (e.g. building bag-of-word model) and thus affect computing sentence similarity [86]. Stemming is the process of transforming (e.g. removing affixes) all the inflected forms of a word into unified and canonical form (e.g. stem) [86]. In Arabic, there are two major approaches for stemming; light stemming which known as affixes removal stemming, and morphological analysis stemming which further classified as root-based stemming and lemma-based stemming [86]. The work presented in [86], [87] compares these two approaches regarding text summarization. Their experiments showed that, in Arabic text summarization, morphological analysis stemming performed better than light stemming. Based on those finding, we experimented with Khoja stemmer [85] as a root-based stemming and MADAMIRA stemmer as a lemma-based stemming. Figure 4 shows the output after applying root-based stemming and lemma-based stemming.

B. FEATURE EXTRACTION AND SENTENCE SCORING

After the preprocessing stage, a set of features were extracted for each sentence to compute the sentence score. In fact, sentence score plays an important role to express the importance or relevancy of each sentence and will be added later as a third objective to be maximized. This is because coverage doesn't capture important features such as similarity with title, location, and length. Selecting and designing these features will greatly affect the quality of the produced summaries. Here, we select and redesign four informative features namely similarity with title, key phrases, sentence location, and sentence length. The selected features include word level and sentence level features. In addition, it includes statistical and semantic features. It is worth mentioning that the selection is based on our observations as well as the studies, experiments, and results made in other related works [3], [4], [24], [88]. The features used in the proposed approach are explained below.

1) SIMILARITY WITH TITLES

This feature measures the similarity or the overlapping between document titles and each sentence. The importance of this feature comes from the fact that if a sentence contains words appearing in the title, then it might be an important sentence since it indicate the subject of the document. In other words, sentences containing words or terms that appear in the document title indicate the theme of the document. This is based on the hypothesis that an author chooses the title to reflect the subject matter of the document. Besides, if a sentence share key-phrases with the title, this will significantly increase its importance. Based on these observations,

FIGURE 4. Output of text preprocessing method.

we defined the similarity with title feature as:

$$title\ similarity(s_i, t) = sim\ (\vec{s_i}, \vec{t}) + KPt\ \cap\ KPs_i \qquad (3)$$

where $\vec{s_i}$ is the TF-ISF representation of sentence s_i, $\vec{t}$ is the TF-ISF representation of all titles (titles of documents related to the same topic), KPt is the list of Key-phrases that appear in the documents titles, KPs_i is the list of Key-phrases extracted from sentence s_i, $KPt \cap KPs_i$ is the intersection value that will be normalized by dividing on the maximum intersection value, and $sim(\vec{s_i}, \vec{t})$ is the degree of similarity between $\vec{s_i}$ and the documents titles $\vec{t}$ computed by cosine similarity measure.

2) KEY-PHRASES

Key-phrases are a list of important and topical keywords that provide a condensed summary of the main topic in the related documents. Key-phrases such as proper nouns might be a single word or consist of multiple words. The existence of key-phrases in a sentence increases its importance w.r.t. to other sentences as it contains valuable information [89], [90]. This feature is calculated by counting the number of key-phrases that appear in a sentence and then normalized by the total number of key-phrases extracted from all related documents, which mathematically defined as:

$$keyphrases(s_i) = \frac{No.\ of\ keyphrases\ in\ s_i}{Total\ number\ of\ keyphrases} \qquad (4)$$

3) SENTENCE LOCATION

The location of the sentence always shows the importance of sentences regardless of the document topic. Leading sentences of documents especially the first sentence are always important and should be included in the summary. This is based on the hypothesis that says the most important sentences of the document occur very early [88]. For example, the first sentence in a document is the most important sentence [91]. We model the sentence location score based on [24] as:

$$location(s_i) = \begin{cases} 3, & \text{first sentence in first paragraph} \\ 2, & \text{first sentence in last paragraph} \\ 1, & \text{first sentence in any paragraph} \\ \frac{1}{\sqrt{i}}, & \text{Other sentences in first or last paragraph} \\ \frac{1}{\sqrt{i+j^2}}, & \text{other sentences in the document} \end{cases} \qquad (5)$$

where i represent sentence index and j is the paragraph index. The location score will be normalized by dividing on 3. It is worth mentioning that, in this formulation, the first sentence in the first paragraph gets the higher score.

4) SENTENCE LENGTH

Sentence length can be used to measure the information contents in a sentence. Long sentences will increase its information content, while short sentences tend to include less crucial information compared to other sentences and thus they are less important [3]. We are not considering short sentences as an important one. This feature counts the number of terms appear in a sentence normalized by the length of the longest sentence, which defined mathematically as:

$$length(s_i) = \frac{No.\ of\ terms\ in\ s_i}{|No.\ terms\ in\ the\ longest\ sentence|} \qquad (6)$$

Several methods presented in the literature to compute the sentence score. According to [7], which compares the performance of different sentence-based voting methods such as BordaFuse, CombMNZ, expCombANZ, etc., we adopt a weighted linear sum of normalized features scores to evaluate each sentence in the document defined as:

$$Score(s_i) = w_1 \cdot title\ similarity(s_i, t) + w_2 \cdot key\ phrases(s_i) + w_3 \cdot location(s_i) + w_4 \cdot length(s_i) \quad (7)$$

where $\sum w_i = 1$. The weight of each feature reflects its importance and thus affect the computation of the total score. Based on conducted experiments and statistical analysis (like the mean and standard deviation), we set weights to be 1, 3, 1, $\frac{1}{2.5}$ for w_1, w_1, w_3, and w_4, respectively.

C. TOPICS IDENTIFICATION BY CLUSTERING

Each set of input related documents has a set of topics. To identify these topics, we employ a clustering-based method. Several clustering methods have been presented in the literature for text summarization [26], [31]–[36]. We chose k-medoid (also called as Partitioning Around Medoid) [92] clustering algorithm. This algorithm is widely used to overcome the weaknesses of the k-means clustering method. K-medoids algorithm uses the medoid as a representation for each cluster. A medoid can be defined as the point in the cluster, whose dissimilarities with all the other members in the cluster are minimum. Medoids are similar in concept to means or centroids, but medoids are always members of the data set. The dissimilarity of the medoid C_i and object P_i is calculated by using any distance measure like Manhattan distance $|C_i - P_i|$. K-medoid chooses the medoid for each cluster that minimizes the summation of distances from it to all the other data points. Formally, let k represent the number of clusters, S represent the set of sentences of all related documents where each sentence S_i is represented using bag-of-words weighted by TF-ISF method, and C_j represent the medoid of cluster k_j where $j \in k$, the k-medoids algorithm for sentence clustering can be summarized by the following steps [92]:

1) Choose number of clusters K
2) Randomly k-sentences are chosen from S to be the initial clusters medoids.
3) Assign each sentence $S_i \in S$ to the cluster K_j with the closest medoid using Manhattan distance measure, $s_i \in K_j$ where $|C_j - s_i|$ is minimum $\forall K_j \in (1, 2, \ldots, k)$.
4) Recalculate the medoid C_j for each cluster K_j by choosing the sentence that minimizes the summation of distances from it to all the other sentences, choose C_j where sum of differences $= \sum_{C_j} \sum_{s_i \in C_j} |C_i - s_i|$ is minimum
5) Repeat steps 2 and 3 until the medoids become unchanged.

The number of clusters k indicates the number of different topics that exist in the original set of related documents. Since k is user defined, it is hard and time consuming to obtain the optimal number of clusters manually. We employed the Silhouette as an automatic method of determining the optimal number of clusters. Silhouette measures the quality of a clustering in terms of cohesion, which measures how closely related are objects in a cluster, and separation, which measures how distinct or well-separated a cluster is from other clusters. For each sentence s_i, let $a(i)$ be the average distance from sentence s_i to all other sentences in K_j cluster. For every $C \neq C_i$, let $d(s_i, C)$ be the average distance from sentence s_i to all other objects in that cluster. After computing this value for all $C \neq C_i$, let $b(i)$ represents the minimum distance, $b(i) = min_c d(s_i, C)$. Finally the silhouette coefficient of sentence s_i is defined by:

$$s(i) = \frac{b(i) - a(i)}{max(b(i), a(i))} \quad (8)$$

This value measures how the sentence s_i fits K_j cluster or its neighbor cluster. A negative value means that the sentence is miss-classified and a value equals to zero indicates that a neighbor cluster is more suitable for sentence s_i. If it is close to one, it means that sentence s_i fits well in its cluster. The average value of $s(i)$ for all sentences in a cluster K_j is called the average Silhouette width of that cluster. Moreover, the mean of $s(i)$ for all sentences is called the average Silhouette width for the entire data set and is denoted by $\bar{s}\ (k)$, where k represents the number of clusters. Choosing k which maximizes $\bar{s}\ (k)$ represents the optimal number of clusters [92]. The output of this stage is a set of clusters each one expresses a topic, and each cluster is represented by its medoid.

D. MULTI-OBJECTIVE OPTIMIZATION

We formalize the multi-document extractive summarization as a global optimization problem of maximizing a set of objective functions that assets summary quality. Our objectives include coverage, diversity, and sentence relevance. However, due to the limitation in the summary length, we want to maximize the coverage and relevancy while minimizing redundancy. Improving coverage and relevancy objectives may lead to the deterioration of diversity. Thus, a single solution, which can optimize all these objectives simultaneously, does not exist. The proposed approach involves the simultaneous optimization of these contradictions objectives. The Multi-objective Optimization (MOO) approach seems to be the natural way to handle this type of problems. MOO optimization handles simultaneously more than one objective function to solve a particular problem. It provides a set of non-dominating solutions as opposed to the single objective optimization approach. Mathematically, MOO can be formulated as:

$$\text{maximize/minimize } F(x) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_m(\vec{x})] \quad \text{s.t. } \vec{x} \in X \quad (9)$$

where X is a set of decision vectors $\vec{x}, \vec{x}, \ldots, \vec{x}$ and m is the the number of objective functions to be maximized/minimized. Several MOO optimization techniques have been presented in the literature to solve real-world

problems [93]. Here, the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) is used to optimize the objective functions [94]. NSGA-II is one of the most popular multi-objective optimization approaches. It is an extension and improvement on the earlier multi-objective evolutionary algorithm NSGA. NSGA-II has three special characteristics, fast non-dominated sorting method, fast crowded distance estimation method, and simple crowded comparison operator. In NSGA-II the population is sorted and partitioned into fronts ($F1, F2, etc.$), where each front contains a set of solutions with the same fitness value. The solutions with the highest fitness values will be in the better fronts. The crowding distance metric which measures the distance of two neighboring solutions is used to distinguish solutions on the same front. Solutions with different non-domination levels and better fitness values will be taken. Otherwise, the one with a higher crowded distance will be chosen to form the optimal Pareto-front. The formulation, parameters, and the main steps of the NSGA-II algorithm are described below:

1) ENCODING OF THE INDIVIDUALS

In text summarization, individuals represent the candidate set of sentences to form the summary. our extractive summarization approach is based on binary optimization, where each solution or chromosome is represented as a binary coded vector $\vec{x}^i = [\vec{x}^{i,1}, \vec{x}^{i,2}, \ldots, \vec{x}^{i,N}]$, where N is the size of the vector, which represent the number of sentences in the related set of documents. For example, in a document of 7 sentences, $\vec{x}^p = [0, 1, 1, 0, 0, 0, 0]$ means that in solution $\vec{x}^p$, sentence 1, 4, 5, 6, and 7 are not included in the summary, while sentence 2, and 3 are included.

2) POPULATION INITIALIZATION

Any evolutionary algorithm, starts with a set of initial solutions $P = [\vec{x}^1, \vec{x}^2, \vec{x}^3, \ldots, \vec{x}^p]$, so-called population, where, p is the number of solutions or population size. The initial population of individuals is generated randomly and uniformly between predefined search ranges, n^{th} component of the p^{th} population member is a uniform random number between 0 and 1 and is instantiated independently. In this case, a discretization is needed to transform the generated real-coded random number into binary-coded [55]. The transformation is based on the following rule:

$$\vec{x}^{p,n} = \begin{cases} 1, & if\ rand_{\mathrm{p,n}} \leq sigm(\vec{x}^{p,n}) \\ 0, & otherwise \end{cases}$$

where $sigm(x)$ represents the sigmoid function.

3) OBJECTIVE FUNCTIONS

To measure the quality of each solution and also to rank them, three objective functions were computed including coverage, diversity, and relevancy. The description and formulation of each objective are given below.

- **Coverage.** Coverage means that the summary should contain all important contents or topics that appear in the related set of documents. We formulated the content coverage as the similarity between sentence s_i where $s_i \in Summary$ and the extracted topics, which represented by the medoids C of the clusters K generated by the k-medoids algorithm with the silhouette method. Thus, the following function should be maximized:

$$f_{\mathrm{coverage}}(X) = \sum_{s_i \in Summary} \sum_{c_j \in C} sim(s_i, c_j) \tag{10}$$

  where, sim represents the cosine similarity measure, s_{i} represent the i^{th} sentence, c_{j} represent the medoid of the j^{th} cluster K_j.
- **Diversity.** Since the generated summary constraint in length, it should not contain multiple sentences having the same information. Diversity aims to reduce information redundancy in the output summary. We calculated the redundancy as the similarity between sentences in the output summary:

$$f_{\mathrm{redundancy}}(X) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} sim(s_i, s_j) \tag{11}$$

  where, sim represents the cosine similarity measure and N is the number of sentences in the output summary. The system tries to minimize this objective to generate a good summary with less redundancy.
- **Relevancy.** Relevancy which is measured by sentence score indicates the importance of each sentence in the output summary. We added the relevancy as a third objective to be maximized since content coverage didn't cover important features such as sentence location. This objective will promote the sentences with a high score to be included in the summary. We calculated the score objective function as:

$$f_{\mathrm{Score}}(X) = \sum_{s_{\mathrm{i}} \in Summary} Score(s_{\mathrm{i}}) \tag{12}$$

In summary, the proposed system can be considered as a maximization problem described as following:

$$f(X) = maximize\ (f_{\mathrm{coverage}}, f_{\mathrm{Score}}, \frac{1}{f_{\mathrm{redundancy}}}) \tag{13}$$

s.t.:

$$lenght(S) \leq l \tag{14}$$

where S is the output summary and l represents the required summary length. It is worth mentioning that, the length of the summary is a constraint, and the evolutionary MOO algorithms treat it as an objective. The length is considered as the fourth objective to be maximized.

a: CROSSOVER

Crossover is used to increase diversity in the population [55], [95]. Several crossover techniques have been presented in the literature. Among them, the Simulated Binary Crossover (SBX) is a competitive one [96]. The SBX uses a probability density function to simulates the single-point

crossover operator of binary-coded representation. The probability distribution is controlled by the distribution index η_c where large η_c indicates a higher probability to create near parent solutions, and a small η_c allows children solutions distant from their parents. The SBX operator biases solutions, which are near each parent more favorably than the solutions, which are away from the parents. Here, we set η_c to be 20.

b: MUTATION

A mutation operator is used as a mechanism for maintaining diversity in the population. Here, the mutation method used follows the classical method of mutation, where one variable at a time with a pre-defined mutation probability p_m is mutated. We set p_m equal $1/N$ so that on average, one variable $\vec{x}^{i,n}$ will be mutated per individual, where N is the number of variables. To perform the mutation, a random number $u \in [0, 1]$ is generated for every variable $\vec{x}^{i,n}$ for an individual $\vec{x}^i$ and if $u \leqslant p_m$ the variable is mutated using the Polynomial Mutation (PM) operator [97], which is a well-known operator for evolutionary MOO algorithms. Similar to the SBX operator, the PM operator biases offspring near each parent more favorably. We set the distribution index η_m, which controls the shape of the offspring distribution equal to 150. It is worth mentioning that while generating the initial solutions and applying crossover and mutation operators, the constraint on the summary length is taken into consideration by adding or removing sentences from the summary.

c: RANKING AND SELECTION

After producing the offspring using crossover and mutation operators, a new population space is formed, which combines offspring with the old population P with a size equal $2P$, where P is the size of the old population. To keep the population space constant over subsequent generations, a selection of P solutions is needed to be used in the next generation. The selection process determines which one of the offspring and the parents will survive for the next generation. To perform this operation, non-dominated sorting (NDS) and crowding distance operator (CDO) are utilized. The NDS assigns a ranking based on the objective functions computed and puts them on different fronts. The CDO measures how close an individual with respect to its neighbors. Finally, parents are selected using binary tournament selection based on the rank and the crowding distance. The individual is selected if the rank is lesser than the other or if the crowding distance is greater than the other when the rank for both individuals is the same.

d: STOPPING CRITERIA

The process of producing offspring and then the selection of parents for the next generation will continue until a stopping criterion is met, such as the maximum number of iterations, CPU time limits, the best objective functions are not changed and achieving a predefined objective function. We adopt the maximum number of iterations as a stopping criterion, in which the algorithm terminates when the maximum number of generations g_{max} is reached.

Algorithm 1 Multi-objective Optimization Algorithm

Input: Population Size P**, Generations** g_{max}
Initialize population P
Compute Objectives Values of the Individuals
Assign Rank Based on NDS Pareto Sort
Generate Offspring Population
  Binary Tournament Selection
  Crossover and Mutation
Compute Objectives Values of the Offspring
Merge the Populations of Parents and Offspring
for $i = 1$ *to* g_{max} **do**
  for each parent and child in the population **do**
    Assign Rank Based on NDS Pareto Sort
    Generate sets of Non-dominated fronts
    Compute Crowding distance Between Members of each Front
  end for
  Select points (elitist) on the lower fronts with high crowding distance
  Generate Next Solution
    Binary Tournament Selection
    Crossover and Mutation
  Compute Objectives Values of the Offspring
  Merge the Populations of Parents and Offspring
end for
Output: Pareto Optimal Solutions

FIGURE 5. Multi-objective's optimization algorithm.

Based on the formalization above, the pseudo-code of the multi-objective optimization stage can be summarized as given in Algorithm 1. The algorithm shows briefly the flow of the genetic operators and the flow of the optimization process. The algorithm takes the population size P and the maximum generation g_{max} as an input and returns the solutions having the highest rank and high crowding distance from the final set of Pareto optimal solutions.

E. SUMMARY GENERATION

The output of the optimization process is a set of optimal Pareto solutions that are non-dominated by others in terms of the objective functions. Thus, there is a need to select the best solution based on user-defined requirements. We adopted a majority voting approach to combine the solutions, which is a simple method and performs very well with real problems [98]. To generate the final solution, we performed the majority voting approach over all the non-dominated solutions, where the summary is formed by choosing the set of sentences that appeared in most of the solutions as shown in Figure 6. When the majority voting output is longer than the desired summary length, sentences with the lowest score are deleted .

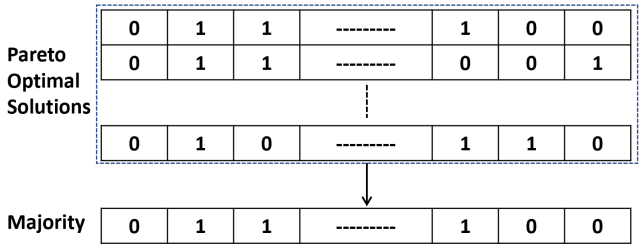


FIGURE 6. Majority voting approach.

TABLE 1. Description of the data sets [14].

	DUC–2002 (Arabic)	TAC-2011
Number of Documents	567	100
Number of Sentences	17,340	1,573
Number of Words	199,423	30,908
Number of Distinct Words	19,307	9,632
Number of Reference Sets	59	10
Documents per Reference Set	10 on average	10
Number of Gold-standard summaries	2 for each reference set	3 for each reference set

V. EXPERIMENT AND RESULTS

In this section, the effectiveness of the proposed summarization approach is evaluated using a set of conducted experiments. Data sets are describe in section A. Evaluation measures are describe in section B. Experiments setup described in section C. The experiments, results, and discussion are reported in section D. Finally, comparing with other related systems is presented in section E.

A. DATA SETS

To evaluate our approach, we used two publicity available data sets; TAC-2011 Multi-Ling and DUC–2002. We used TAC-2011 Multi-Ling as the main dataset for our experiments, since the Arabic version is ready and available for research. On the other hand, the Arabic version of DUC–2002 is not available and thus we have obtained the raw data set and translated it using Google translator. We manually checked and validated the translation. Table 1 provides statistics about the two data sets.

B. EVALUATION MEASURES

We used the the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [99] metric to evaluate the performance of the proposed approach. ROUGE is a widely accepted metric and considered the official evaluation metric for automatic evaluation of text summarization by DUC and TAC. ROUGE is an automatic method that measure the quality of the generated summary by computing the similarity between the generated summaries and the ground truth human generated summaries. Computing similarity could be by counting overlapping terms such as the N-gram (e.g. ROUGE-N, N = 1 − 4), word sequences (e.g. ROUGE-L and ROUGE-W) and word pairs (e.g. ROUGE-S and ROUGE-SU). In our evaluation, we used four metrics of ROUGE: ROUGE-N (ROUGE-1 and ROUGE-2), ROUGE-L, and ROUGE-SU. The ROUGE-N measure compares N-grams of two summaries (generated and reference) and then counts the number of matches. It is calculated as:

$$ROUGE-N = \frac{\sum_{S \in Sum_{ref}} \sum_{\text{N-gram} \in S} Count_{\text{match}}(\text{N-gram})}{\sum_{S \in Sum_{ref}} \sum_{\text{N-gram} \in S} Count(\text{N-gram})} \quad (15)$$

where N is the length of the N-gram (e.g. N equals 1 for ROUGE-1), $Count_{match}$(N-gram) is the maximum number of N-grams that are co-occurring in a candidate summary and a set of reference summaries, and Count(N-gram) is the number of N-grams in the set of reference summaries. The ROUGE-L calculates the ratio between summaries' longest common sub-sequence (LCS) and the length of the ground truth reference summary which is defined as:

$$F_{LCS}(R, S) = \frac{(1+\beta^2) P_{LCS}(R, S) R_{LCS}(R, S)}{\beta^2 P_{LCS}(R, S) + R_{LCS}(R, S)} \quad (16)$$

where, $LCS(R, S)$ is the length of a LCS of R and S, $P_{LCS}(R, S)$ is the precision of $LCS(R, S)$ which equal $LCS(R, S)/|S|$, $R_{LCS}(R, S)$ is the recall of $LCS(R, S)$ which equal $LCS(R, S)/|R|$, $|S|$ is the length of the candidate S sentence summary, $|R|$ is the length of the reference R, and β is the relative importance of $P_{LCS}(R, S)$ and $R_{LCS}(R, S)$ which equal $P_{LCS}(R, S)/R_{LCS}(R, S)$. Finally, ROUGE-SU is the extended version of ROUGE-S and defined as a weighted average between ROUGE-S and ROUGE-1. ROUGE-S measures the interfere ratio of skip-bigrams between a candidate summary and a set of reference summaries, where skip-bigram is any pair of words or terms in their sentence that allowing for any arbitrary gaps. ROUGE-S defined as:

$$F_{LCS}(R, S) = \frac{(1+\beta^2) P_{SKIP2}(R, S) R_{SKIP2}(R, S)}{\beta^2 P_{SKIP2}(R, S) + R_{SKIP2}(R, S)} \quad (17)$$

where, $SKIP2(R, S)$ is the number of the matches between R and S, $P_{SKIP2}(R, S)$ and $R_{SKIP2}(R, S)$ is the precision an recall of $SKIP2(R, S)$, and β is the relative importance of $P_{SKIP2}(R, S)$ and $R_{SKIP2}(R, S)$.

C. EXPERIMENTS SETUP AND TOOLS

We used Java as the main language for implementing several tasks such as sentence representation and features extraction. Also, we used Java to integrate the tools used since the selected tools are Java based (published as JAR files). Table 2 provides a summary of the methods, parameters, and tools used in the proposed approach. It is worth mentioning that the selection tools and their settings are based on the studies made on [11], [80], [86], [100], [101].

D. RESULTS

The first set of experiments study the effect of pre-processing techniques in text summarization. The TAC 2011 MultiLing and DUC datasets are tokenized using two types of tokenization; semantic using Stanford CoreNLP tool and punctuation marks using AraNLP tool. Besides, the datasets

TABLE 2. The variable parameters in our system.

Tasks/Tools	Our Usage
Preprocessing tools	Normalization: AraNLP. Tokenization: Stanford semantic tokenizer, AraNLP punctuation marks tokenizer. Stemming: Madamera Lemma stemmer, Khoja root stemmer. Stop word list: General stop word list
Key-Phrase extraction	KP-Miner
Sentence representation	Bag-of-words with TF-ISF
Clustering method	K-medoids
Clusters validation	Silhouette measure
MOO algorithm	NSGA-II
Objectives	Coverage, Diversity, and Relevance.
NSGA-II parameters	Encoding: Binary. Population size 100. Generation of initial populations: Random (Uniform). Crossover: BLX operator. Crossover Probability: 0.95. BLX distribution index: 20. Mutation: Polynomial operator. Polynomial distribution index: 150. Mutation probability: 0.05. Ranking and selection: NDS and CDO. Max iteration:25,000. Summary generation: Majority voting.
ROUGE tool	ROUGE-1.5.5
Optimization tool	jMetal

are stemmed using two stemmers: MADAMIRA lemma and Khoja root stemmers. Table 3 and Table 4 show the effect of pre-processing techniques on the performance of text summarization. They show the results of four different combinations of pre-processing techniques (semantic tokenization + lemma stemming, semantic tokenization + root stemming, punctuation marks tokenization + lemma stemming, and punctuation marks tokenization + root stemming). It is clear from Table 3 and Table 4 that the best results achieved when the punctuation marks tokenizer and Khoja root stemmer are used. They acheived an average F-measure of 0.471, 0.237, 0.471, and 0.204 for R-1, R-2, R-L, and R-SU, respectively. In fact, punctuation marks tokenizer is more stable when the data is written with the correct usage of punctuation marks, while semantic tokenizer can be more effective if the data written as long line without punctuation marks. In our case, the datasets are written with full usage of punctuation marks, so this explains why punctuation marks tokenizer outperforms semantic tokenizer. On the other hand, Khoja root stemmer beats lemma stemmer. This can be explained as follows: Khoja stemmer is a root-based stemmer which retrieves the root of a word and this increases the semantic similarity between sentences. Based on these results, all subsequent experiments are conducted with punctuation marks tokenizer and Khoja root stemmer. It is worth mentioning that since the optimization process uses random variables to control the different operators such as crossover, and mutation. Thus, we can not rely on the results of a single run, so that we use the average F-measure of 10-Independent runs to obtain the results of our approach and also compare with related systems participating with these datasets.

The second set of experiments study the effect of adding relevancy (sentence score) as a third objective to be maximized. To show the effectiveness of adding relevancy objective, an experiment with only coverage and diversity objectives is conducted. Table 5 shows the results of optimizing only coverage and diversity objectives along with the results of optimizing coverage, diversity, and relevancy on both datasets. The results shows that adding relevancy as a third objective improve results significantly for both

TABLE 3. The average F-measure values of ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-L (R-L), and ROUGE-SU (R-SU) of semantic tokenizer with lemma and root stemmers.

	Semantic + Lemma				Semantic + Root			
Data set	R-1	R-2	R-L	R-SU4	R-1	R-2	R-L	R-SU4
TAC	0.317	0.101	0.297	0.104	**0.355**	**0.117**	**0.317**	**0.128**
DUC	0.323	0.125	0.317	0.118	**0.381**	**0.135**	**0.365**	**0.133**

TABLE 4. The average F-measure values of ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-L (R-L), and ROUGE-SU (R-SU) of punctuation marks tokenizer with lemma and root stemmers.

	Punctuation marks + Lemma				Punctuation marks + Root			
Data set	R-1	R-2	R-L	R-SU4	R-1	R-2	R-L	R-SU4
TAC	0.346	0.129	0.311	0.125	**0.389**	**0.177**	**0.354**	**0.158**
DUC	0.372	0.162	0.372	0.144	**0.471**	**0.237**	**0.471**	**0.204**

TABLE 5. The average F-measure values of ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-L (R-L), and ROUGE-SU (R-SU) with and without relevancy objective.

	Coverage and Diversity				Coverage, Diversity, and Relevance			
Data set	R-1	R-2	R-L	R-SU4	R-1	R-2	R-L	R-SU4
TAC	0.278	0.078	0.263	0.086	**0.389**	**0.177**	**0.354**	**0.158**
DUC	0.360	0.139	0.350	0.124	**0.471**	**0.237**	**0.471**	**0.204**

TABLE 6. Relative improvement of adding relevancy objective on the F-measure values of ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-L (R-L), and ROUGE-SU (R-SU).

	TAC Dataset				DUC Dataset			
Objectives	R-1	R-2	R-L	R-SU4	R-1	R-2	R-L	R-SU4
Covearge and Diversity	0.278	0.078	0.263	0.086	0.360	0.139	0.350	0.124
Coverage, Diversity, and Relevence	0.389	0.171	0.354	0.158	0.471	0.237	0.471	0.204
Relative improvments	**+39.5%**	**+119%**	**+34.8%**	**+83.4%**	**+30.8%**	**+70.0%**	**+30.9%**	**+64.3%**

TABLE 7. Systems participated with DUC-2002 and TAC 2011 datasets.

ID	System	Approach	Features	Dataset
ID1 [11]	Baseline	Clustering-based approach	-	TAC 2011
ID2 [102]	Global and Local Models for Multi-Document Summarization	Unsupervised models for latent structure discovery	TF-ISF weighting	TAC 2011
ID3 [103]	The CIST Summarization System at TAC 2011	hierarchical Latent Dirichlet Allocation (hLDA)	Title similarity, keywords, name entity, sentence coverage, and word abstractive level	TAC 2011
ID4 [104]	LIF at TAC Multiling: Towards a Truly Language Independent Summarizer Arabic/English multi-document summarization with CLASSYthe past and the future	Maximal Marginal Relevance (MMR)	TF-ISF weighting	TAC 2011
ID5 [105]	University of Essex at the TAC 2011 Multilingual Summarisation Pilot	Clustering-based approach	TF-ISF weighting	TAC 2011
ID6 [106]	Arabic/English multi-document summarization with CLASSYthe past and the future	Clustering, Linguistics, and Statistics for Summarization Yield (CLASSY)	The signature terms, and the probability of a term occurs in the sentences	TAC 2011
ID7 [107]	Guided and Multilingual Summarization Tasks	LSA-based summarizer	TF weighting	TAC 2011
ID8 [11]	Ant Colony System for Multi-Document Summarization	Ant Colony optimization algorithm to maximize the summary coverage	TF-ISF weighting with PageRank and HITS ranking	TAC 2011
ID9 [11]	Topline	Genetic algorithm	-	TAC 2011
ID10 [108]	Extractive Automatic Text Summarization Based on Lexical-Semantic Keywords	Lexical-Semantic Keywords	TF-IDF, OHE, and DOC2VEC	TAC 2011
ID1 [14]	Multi-document Arabic Text Summarisation	Cluster-based summrization	Similarity using three models: VSM, LSA, and Dice	DUC 2002
ID2 [31]	Automatic Multi-Document Arabic Text Summarization Using Clustering and Keyphrase Extraction	Combined clustering method to group the documents into clusters	Sentence count, TF, First/last occurrence in text, and C-value	DUC 2002
Our approach	Multi-Document Text Summarization using Evolutionary Multi-Objective Optimization with K-mediod Clustering	Evolutionary Multi-Objective Optimization with Clustering	Three objectives: coverage, diversity, and relevancy. Features: similarity with title, key-phrases, sentence location, and sentence length.	TAC 2011, DUC 2002

TABLE 8. The F-measure values ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-L (R-L), and ROUGE-SU4 (R-SU4) for the participating systems and the proposed approach for the Arabic version of the 2011 MultiLing dataset. The highest values among those of them are written in bold.

	Results				Relative Improvement			
Systems	**R-1**	**R-2**	**R-L**	**R-SU4**	**R-1**	**R-2**	**R-L**	**R-SU4**
ID1 [11]	0.231	0.095	0.212	0.097	+68.4%	+80%	+66.98%	+62.89%
ID2 [102]	0.224	0.086	0.214	0.099	+73.66%	+98.84%	+65.42%	+59.6%
ID3 [103]	0.232	0.089	0.220	0.099	+67.67%	+92.13%	+60.91%	+59.6%
ID4 [104]	0.263	0.086	0.239	0.107	+47.91%	+98.84%	+48.12%	+47.66%
ID5 [105]	0.268	0.097	0.248	0.115	+45.15%	+76.29%	+42.74%	+37.39%
ID6 [106]	0.292	0.103	0.273	0.133	+33.22%	+66.02%	+29.67%	+18.8%
ID7 [107]	0.300	0.128	0.272	0.151	+29.67%	+33.59%	+30.15%	+4.64%
ID8 [11]	0.308	0.149	0.269	0.155	+26.3%	+14.77%	+31.6%	+1.94%
ID9 [11]	0.312	0.120	0.284	0.130	+24.68%	+42.5%	+24.65%	+21.54%
ID10 [108]	0.339	–	–	–	+14.70%	–	–	–
Semantic tokenization + Lemma stemming	0.317	0.101	0.297	0.104	+22.71%	+69.31%	+19.19%	+51.92%
Punctuation mark tokenization + Lemma stemming	0.346	0.129	0.311	0.125	+12.43%	+32.56%	+13.83%	+26.4%
Semantic tokenization + Root stemming	0.355	0.117	0.317	0.128	+9.58%	+46.15%	+11.67%	+23.44%
Punctuation mark tokenization + Root stemming	**0.389**	**0.171**	**0.354**	**0.158**	–	–	–	–

datasets with an average relative improvement of 39.5%, 119%, 34.8%, and 83.4% for R-1, R-2, R-L, and R-SU, respectively for TAC dataset, and 30.8%, 70.0%, 30.9%, and 64.3% for R-1, R-2, R-L, and R-SU, respectively, for DUC dataset as shown in Table . This improvement comes from the important features covered by the relevancy objective (e.g. sentence position) which are not covered by the coverage and the diversity objectives.

E. COMPARING WITH OTHER RELATED WORKS

Table 7 provides a subjective comparison between the proposed system against other Arabic extractive text summarization methods presented in the literature in terms of summarization approach, features, and datasets. Table 8 and Table 9 provide an objective comparisons with relative improvements between our proposed approach and other related approaches on both datasets using the ROUGE measures. The results show that our approach outperform all systems participating with TAC 2011 and DUC-2002 datasets. Our system showed a relative improvements of +24.68%, +42.5%, +24.65%, and +21.54% over the top-ranked system participated in TAC in terms of Rouge-1, Rouge-2, Rouge-L, and Rouge-SU4 respectively. In addition, with DUC 2002 dataset, our system beats all systems participating

TABLE 9. The F-measure values of ROUGE-1 results for the participating systems and the proposed approach for the Arabic version of DUC 2002 dataset. The highest values among those of them are written in bold.

	Results			Relative Improvement		
Systems	**Recall**	**Precision**	**F-measure**	**Recall**	**Precision**	**F-measure**
ID1 [14]	0.395	0.384	0.390	+15.75%	+26.02%	+20.64%
ID2 [31]	0.452	0.418	0.434	+1.15%	+15.77%	+8.41%
Punctuation mark tokenization + Lemma stemming	0.352	0.391	0.372	+29.89%	+23.76%	+26.48%
Punctuation mark tokenization + Root stemming	**0.4572**	**0.4839**	**0.4705**	-	-	-

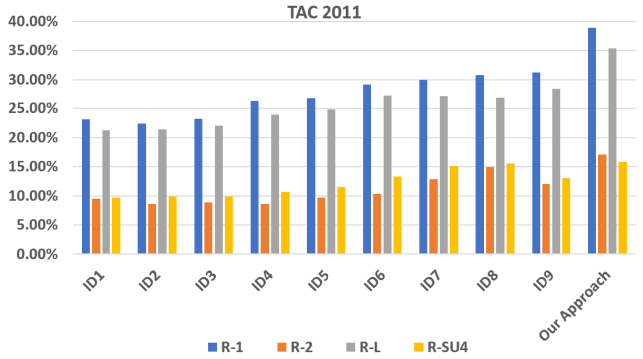


FIGURE 7. Comparison of TAC 2011 results.

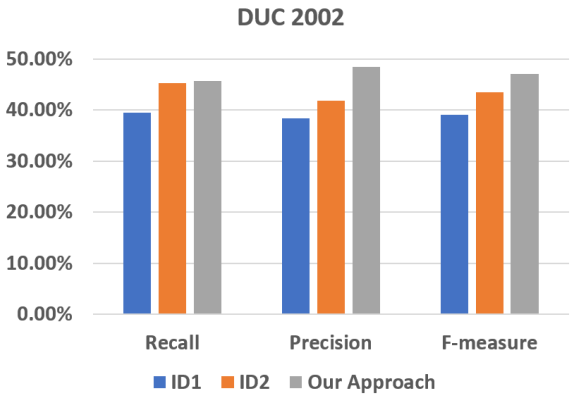


FIGURE 8. Comparison of DUC 2002 results.

with this dataset in terms of all ROUGE metrics. Our system achieved a relative improvements of +1.15%, +15.77%, +8.41% over the top-ranked system in terms of recall, precision, and F-measure of Rouge-1 respectively. It is worth mentioning that our approach performance is better than other peer systems, which it is clear from ROUGE-2 results which is bi-gram matching. Figure 7 and 8 show a comparison of our system to the most recent related work with both datasets.

VI. CONCLUSION

To conclude, in this paper we proposed the multi-document text summarization as a multi-objective's optimization approach. The presented approach utilizes four stages of preprocessing, feature extraction, clustering, and multi-objectives optimization. First, to represent sentences in a unified form, four preprocessing methods were applied namely tokenization, normalization, stop word removal, and stemming. In the second stage, a set of statistical and semantic features were extracted to be employed for scoring each sentence as a measure of sentence relevancy. Next, topics of the related set of documents were extracted using k-medoid clustering method with Silhouette measure. Finally, to create an optimal document summary, an evolutionary multi-objectives optimization method was employed to simultaneously optimize three objectives. The optimization process tries to maximize coverage and sentence relevancy while eliminating information redundancy. Results on standard datasets including TAC 2011 and DUC 2002 proved clearly the efficacy of our proposed techniques compared to the state-of-art in terms of ROUGE measures.

Since all research studies have some limitations, the limitations for the presented approach can be summarized in (i) In computing sentence score, we determined the weight of the features experimentally, it could be determined using Genetic Algorithm to find the optimal weights, and (ii) we did not encounter coherency or readability of the generated summary, it could be added as a fourth objective to be maximized.

As a future work, authors will work on (i) testing with other languages (ii) studying the effect of using other sentence representation and sentence similarity measures, (iii) using Genetic Algorithm to find the optimal weights of the features in the score equation, () using differential evolution MOO approach, (iv) developing other modification of MOO algorithm in order to find the best summary more effectively, and (vi) enhancing readability or cohesion of the generated summary as a post processing step.

REFERENCES

[1] R. Qumsiyeh and Y.-K. Ng, "Searching Web documents using a summarization approach," *Int. J. Web Inf. Syst.*, vol. 12, no. 1, pp. 83–101, Apr. 2016.

[2] P. Modaresi, P. Gross, S. Sefidrodi, M. Eckhof, and S. Conrad, "On (commercial) benefits of automatic text summarization systems in the news domain: A case of media monitoring and media response analysis," 2017, *arXiv:1701.00728*. [Online]. Available: http://arxiv.org/abs/1701.00728

[3] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113679, doi: 10.1016/j.eswa.2020.113679.

[4] A. P. Widyassari, S. Rustad, G. F. Shidik, E. Noersasongko, A. Syukur, A. Affandy, and D. R. I. M. Setiadi, "Review of automatic text summarization techniques & methods," *J. King Saud Univ.-Comput. Inf. Sci.*, 2020.

[5] Y.-H. Hu, Y.-L. Chen, and H.-L. Chou, "Opinion mining from online hotel reviews—A text summarization approach," *Inf. Process. Manage.*, vol. 53, no. 2, pp. 436–449, Mar. 2017.
[6] R. Ferreira, L. de Souza Cabral, R. D. Lins, G. P. E. Silva, F. Freitas, G. D. C. Cavalcanti, R. Lima, S. J. Simske, and L. Favaro, "Assessing sentence scoring techniques for extractive text summarization," *Expert Syst. Appl.*, vol. 40, no. 14, pp. 5755–5764, Oct. 2013.
[7] Y. K. Meena and D. Gopalani, "Efficient voting-based extractive automatic text summarization using prominent feature set," *IETE J. Res.*, vol. 62, no. 5, pp. 581–590, Sep. 2016.
[8] C. Jung, R. Datta, and A. Segev, "Multi-document summarization using evolutionary multi-objective optimization," in *Proc. Genetic Evol. Comput. Conf. Companion*, Jul. 2017, pp. 31–32.
[9] A. B. Al-Saleh and M. E. B. Menai, "Automatic arabic text summarization: A survey," *Artif. Intell. Rev.*, vol. 45, no. 2, pp. 203–234, Feb. 2016.
[10] K. S. Al Harazin, "Multi-document arabic text summarization," *Multi-Document Arabic Text Summarization*, 2015.
[11] A. Al-Saleh and M. E. B. Menai, "Ant colony system for multi-document summarization," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 734–744.
[12] A. Al-Saleh and M. Menai, "Solving multi-document summarization as an orienteering problem," *Algorithms*, vol. 11, no. 7, p. 96, 2018.
[13] V. Patil, M. Krishnamoorthy, P. Oke, and M. Kiruthika, "A statistical approach for document summarization," Dept. Comput. Eng., Fr. C. Rodrigues Inst. Technol., Navi Mumbai, India, Tech. Rep., 2004.
[14] M. El-Haj, U. Kruschwitz, and C. Fox, "Multi-document arabic text summarisation," in *Proc. 3rd Comput. Sci. Electron. Eng. Conf. (CEEC)*, Jul. 2011, pp. 40–44.
[15] R. M. Alguliev and R. M. Aliguliyev, "Effective summarization method of text documents," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Sep. 2005, pp. 264–271.
[16] J. M. Conroy and J. D. Schlesinger, "CLASSY and TAC 2008 metrics," in *Proc. TAC*, 2008.
[17] H. Morita, T. Sakai, and M. Okumura, "Query snowball: A co-occurrence-based approach to multi-document summarization for question answering," *Inf. Media Technol.*, vol. 7, no. 3, pp. 1124–1129, 2012.
[18] D. Patel, S. Shah, and H. Chhinkaniwala, "Fuzzy logic based multi document summarization with improved sentence scoring and redundancy removal technique," *Expert Syst. Appl.*, vol. 134, pp. 167–177, Nov. 2019, doi: 10.1016/j.eswa.2019.05.045.
[19] R. Elbarougy, G. Behery, and A. El Khatib, "Extractive arabic text summarization using modified PageRank algorithm," *Egyptian Informat. J.*, vol. 21, no. 2, pp. 73–81, Jul. 2020, doi: 10.1016/j.eij.2019.11.001.
[20] M. A. Fattah and F. Ren, "GA, MR, FFNN, PNN and GMM based models for automatic text summarization," *Comput. Speech Lang.*, vol. 23, no. 1, pp. 126–144, Jan. 2009.
[21] R. Belkebir and A. Guessoum, "A supervised approach to arabic text summarization using Adaboost," in *New Contributions in Information Systems and Technologies*. Cham, Switzerland: Springer, 2015, pp. 227–236.
[22] Q. A. Al-Radaideh and D. Q. Bataineh, "A hybrid approach for arabic text summarization using domain knowledge and genetic algorithms," *Cognit. Comput.*, vol. 10, no. 4, pp. 651–669, Aug. 2018.
[23] A. Nenkova and K. McKeown, "A survey of text summarization techniques," in *Mining Text Data*. Boston, MA, USA: Springer, 2012, pp. 43–76.
[24] A. Qaroush, I. A. Farha, W. Ghanem, M. Washaha, and E. Maali, "An efficient single document Arabic text summarization using a combination of statistical and semantic features," *J. King Saud Univ.-Comput. Inf. Sci.*, 2019.
[25] V. Gupta and G. S. Lehal, "A survey of text summarization extractive techniques," *J. Emerg. Technol. Web Intell.*, vol. 2, no. 3, pp. 258–268, Aug. 2010.
[26] Y. J. Kumar and N. S. Salim, "Automatic multi document summarization approaches," *J. Comput. Sci.*, vol. 8, no. 1, pp. 133–140, Jan. 2012.
[27] D. R. Radev, E. Hovy, and K. McKeown, "Introduction to the special issue on summarization," *Comput. Linguistics*, vol. 28, no. 4, pp. 399–408, Dec. 2002.
[28] A. T. Al-Taani and M. M. Al-Omour, "An extractive graph-based arabic text summarization approach," in *Proc. Int. Arab Conf. Inf. Technol.*, Jordan, 2014.
[29] S. Lagrini, M. Redjimi, and N. Azizi, "Automatic arabic text summarization approaches," *Int. J. Comput. Appl.*, vol. 164, no. 5, pp. 31–37, Apr. 2017.
[30] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "EdgeSumm: Graph-based framework for automatic text summarization," *Inf. Process. Manage.*, vol. 57, no. 6, Nov. 2020, Art. no. 102264, doi: 10.1016/j.ipm.2020.102264.
[31] H. N. Fejer and N. Omar, "Automatic multi-document arabic text summarization using clustering and keyphrase extraction," *J. Artif. Intell.*, vol. 8, no. 1, pp. 1–9, Dec. 2014.
[32] A. Haboush, M. Al-Zoubi, A. Momani, and M. Tarazi, "Arabic text summarization model using clustering techniques," *World Comput. Sci. Inf. Technol. J.*, 2012.
[33] K. Sarkar, "Sentence clustering-based summarization of multiple text documents," *Int. J. Comput. Sci. Commun. Technol.*, vol. 2, no. 1, pp. 325–335, 2009.
[34] H. J. Jain, M. S. Bewoor, and S. H. Patil, "Context sensitive text summarization using K means clustering algorithm," *Int. J. Soft Comput. Eng.*, vol. 2, no. 2, pp. 301–304, 2012.
[35] S. A. Waheeb and H. Husni, "Multi-document arabic summarization using text clustering to reduce redundancy," *Int. J. Adv. Sci. Technol.*, vol. 2, no. 1, pp. 194–199, 2014.
[36] S. Abdulateef, N. A. Khan, B. Chen, and X. Shang, "Multidocument arabic text summarization based on clustering and Word2Vec to reduce redundancy," *Information*, vol. 11, no. 2, p. 59, Jan. 2020.
[37] F. Al-Khawaldeh and V. Samawi, "Lexical cohesion and entailment based segmentation for arabic text summarization (LCEAS)," *World Comput. Sci. Inf. Technol. J.*, vol. 5, no. 3, pp. 51–60, 2015.
[38] D. Tatar, A. Mihis, D. Lupsa, and E. Tamaianu-Morita, "Entailment-based linear segmentation in summarization," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 19, no. 8, pp. 1023–1038, Dec. 2009.
[39] I. Imam, N. Nounou, A. Hamouda, and H. A. A. Khalek, "An ontology-based summarization system for arabic documents (OSSAD)," *Int. J. Comput. Appl.*, vol. 74, no. 17, pp. 38–43, Jul. 2013.
[40] K. Al-Sabahi, Z. Zhang, J. Long, and K. Alwesabi, "An enhanced latent semantic analysis approach for arabic document summarization," *Arabian J. Sci. Eng.*, vol. 43, no. 12, pp. 8079–8094, Dec. 2018.
[41] L. Cagliero, P. Garza, and E. Baralis, "ELSA: A multilingual document summarization algorithm based on frequent itemsets and latent semantic analysis," *ACM Trans. Inf. Syst.*, vol. 37, no. 2, pp. 1–33, Mar. 2019, doi: 10.1145/3298987.
[42] W. AlSanie, A. Touir, and H. Mathkour, "Towards an infrastructure for Arabic text summarization using rhetorical structure theory," M.S. thesis, King Saud Univ., Riyadh, Saudi Arabia, 2005.
[43] A. M. Azmi and S. Al-Thanyyan, "A text summarizer for arabic," *Comput. Speech Lang.*, vol. 26, no. 4, pp. 260–273, Aug. 2012.
[44] H. I. Mathkour, "A novel rhetorical structure approach for classifying arabic security documents," *Int. J. Comput. Theory Eng.*, vol. 1, no. 3, pp. 195–200, 2009.
[45] A. Ibrahim, T. Elghazaly, and M. Gheith, "A novel Arabic text summarization model based on rhetorical structure theory and vector space model," *Int. J. Comput. Linguistics Natural Lang. Process.*, vol. 2, no. 8, pp. 480–485, 2013.
[46] P. C. Cardoso and T. A. Pardo, "Multi-document summarization using semantic discourse models," *Procesamiento del Lenguaje Natural*, vol. 56, pp. 57–64, 2016.
[47] I. Keskes, "Discourse analysis of arabic documents and application to automatic summarization," Ph.D. dissertation, Univ. de Toulouse, Univ. Toulouse III-Paul Sabatier, Toulouse, France, 2015.
[48] K. Umam, F. W. Putro, G. Q. O. Pratamasunu, A. Z. Arifin, and D. Purwitasari, "Coverage, diversity, and coherence optimization for multi-document summarization," *Jurnal Ilmu Komputer dan Informasi*, vol. 8, no. 1, pp. 1–10, 2015.
[49] K.-Y. Chen, S.-H. Liu, B. Chen, and H.-M. Wang, "Improved spoken document summarization with coverage modeling techniques," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 6010–6014.
[50] L. Li, K. Zhou, G.-R. Xue, H. Zha, and Y. Yu, "Enhancing diversity, coverage and balance for summarization through structure learning," in *Proc. 18th Int. Conf. World Wide Web (WWW)*, 2009, pp. 71–80.
[51] X. Cai, W. Li, and R. Zhang, "Enhancing diversity and coverage of document summaries through subspace clustering and clustering-based optimization," *Inf. Sci.*, vol. 279, pp. 764–775, Sep. 2014.
[52] W. Luo, F. Zhuang, Q. He, and Z. Shi, "Exploiting relevance, coverage, and novelty for query-focused multi-document summarization," *Knowl.-Based Syst.*, vol. 46, pp. 33–42, Jul. 2013.

[53] R. M. Alguliev, R. M. Aliguliyev, M. S. Hajirahimova, and C. A. Mehdiyev, "MCMR: Maximum coverage and minimum redundant text summarization model," *Expert Syst. Appl.*, vol. 38, no. 12, pp. 14514–14522, Nov. 2011.

[54] R. Z. Al-Abdallah and A. T. Al-Taani, "Arabic single-document text summarization using particle swarm optimization algorithm," *Procedia Comput. Sci.*, vol. 117, pp. 30–37, Jan. 2017.

[55] R. M. Alguliev, R. M. Aliguliyev, and M. S. Hajirahimova, "GenDocSum+MCLR: Generic document summarization based on maximum coverage and less redundancy," *Expert Syst. Appl.*, vol. 39, no. 16, pp. 12460–12473, Nov. 2012.

[56] J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "Comparison of automatic methods for reducing the Pareto front to a single solution applied to multi-document text summarization," *Knowl.-Based Syst.*, vol. 174, pp. 123–136, Jun. 2019, doi: 10.1016/j.knosys.2019.03.002.

[57] R. Rautray and R. C. Balabantaray, "An evolutionary framework for multi document summarization using cuckoo search approach: MDSCSA," *Appl. Comput. Informat.*, vol. 14, no. 2, pp. 134–144, Jul. 2018.

[58] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, Mar. 2011.

[59] J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach," *Knowl.-Based Syst.*, vol. 159, pp. 1–8, Nov. 2018.

[60] Y. Dong, "A survey on neural network-based summarization methods," 2018, *arXiv:1804.04589*. [Online]. Available: http://arxiv.org/abs/1804.04589

[61] N. Raphal, H. Duwarah, and P. Daniel, "Survey on abstractive text summarization," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2018, pp. 513–517.

[62] Y. Dong, "A survey on neural network-based summarization methods," 2018, *arXiv:1804.04589*. [Online]. Available: http://arxiv.org/abs/1804.04589

[63] M. Zhong, P. Liu, Y. Chen, D. Wang, X. Qiu, and X. Huang, "Extractive summarization as text matching," 2020, *arXiv:2004.08795*. [Online]. Available: http://arxiv.org/abs/2004.08795

[64] Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, and T. Zhao, "Neural document summarization by jointly learning to score and select sentences," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 654–663.

[65] D. Wang, P. Liu, M. Zhong, J. Fu, X. Qiu, and X. Huang, "Exploring domain shift in extractive text summarization," 2019, *arXiv:1908.11664*. [Online]. Available: http://arxiv.org/abs/1908.11664

[66] D. Wang, P. Liu, Y. Zheng, X. Qiu, and X. Huang, "Heterogeneous graph neural networks for extractive document summarization," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020.

[67] K. Arumae and F. Liu, "Reinforced extractive summarization with question-focused rewards," in *Proc. ACL Student Res. Workshop*, 2018, pp. 105–111.

[68] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 3721–3731.

[69] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization," 2019, *arXiv:1912.08777*. [Online]. Available: http://arxiv.org/abs/1912.08777

[70] D. Suleiman and A. Awajan, "Deep learning based abstractive text summarization: Approaches, datasets, evaluation measures, and challenges," *Math. Problems Eng.*, vol. 2020, pp. 1–29, Aug. 2020, doi: 10.1155/2020/9365340.

[71] W. Qi, Y. Yan, Y. Gong, D. Liu, N. Duan, J. Chen, R. Zhang, and M. Zhou, "ProphetNet: Predicting future N-gram for sequence-to-sequence pre-training," 2020, *arXiv:2001.04063*. [Online]. Available: http://arxiv.org/abs/2001.04063

[72] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7871–7880, doi: 10.18653/v1/2020.acl-main.703.

[73] Z. Liu, Y. Lin, and M. Sun, "Sentence representation," *Represent. Learn. Natural Lang. Process.*, pp. 59–89, 2020.

[74] A. Ayedh, G. Tan, K. Alwesabi, and H. Rajeh, "The effect of preprocessing on arabic document categorization," *Algorithms*, vol. 9, no. 2, p. 27, 2016.

[75] M. K. Vijaymeena and K. K. Kavitha, "A survey on similarity measures in text mining," *Mach. Learn. Appl., Int. J.*, vol. 3, no. 1, pp. 19–28, Mar. 2016, doi: 10.5121/mlaij.2016.3103.

[76] R. M. Alguliev, R. M. Aliguliyev, and C. A. Mehdiyev, "Sentence selection for generic document summarization using an adaptive differential evolution algorithm," *Swarm Evol. Comput.*, vol. 1, no. 4, pp. 213–222, Dec. 2011.

[77] M. Peyrard and J. Eckle-Kohler, "A general optimization framework for multi-document summarization using genetic algorithms and swarm intelligence," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers (COLING)*, 2016, pp. 247–257.

[78] R. M. Alguliyev, R. M. Aliguliyev, N. R. Isazade, A. Abdi, and N. Idris, "COSUM: Text summarization based on clustering and optimization," *Expert Syst.*, vol. 36, no. 1, p. e12340, 2019.

[79] J. Soler, F. Tencé, L. Gaubert, and C. Buche, "Data clustering and similarity," in *Proc. 26th Int. FLAIRS Conf.*, May 2013.

[80] A. Ayedh, G. Tan, K. Alwesabi, and H. Rajeh, "The effect of preprocessing on arabic document categorization," *Algorithms*, vol. 9, no. 2, p. 27, Apr. 2016, doi: 10.3390/a9020027.

[81] R. Elbarougy, G. M. Behery, and A. El Khatib, "A proposed natural language processing preprocessing procedures for enhancing arabic text summarization," in *Recent Advances in NLP: The Case of Arabic Language*. 2020, doi: 10.1007/978-3-030-34614-0_3.

[82] M. A. Attia, "Arabic tokenization system," in *Proc. Workshop Comput. Approaches Semitic Lang., Common Issues Resour.* Stroudsburg, PA, USA: Association for Computational Linguistics, Jun. 2007, pp. 65–72.

[83] M. Rouhia, H. Mousa, and M. Hussein, "Improving arabic text categorization using normalization and stemming techniques," *Int. J. Comput. Appl.*, vol. 135, no. 2, pp. 38–43, Feb. 2016, doi: 10.5120/ijca2016908328.

[84] Ranks.nl. (2018). *Arabic*. Accessed: Jan. 6, 2018. [Online]. Available: https://www.ranks.nl/stopwords/arabic

[85] S. Khoja and R. Garside, "Stemming arabic text," Dept. Comput., Lancaster Univ., Lancaster, U.K., Tech. Rep., 1999.

[86] M. Mustafa, A. S. Eldeen, S. Bani-Ahmad, and A. O. Elfaki, "A comparative survey on arabic stemming: Approaches and challenges," *Intell. Inf. Manage.*, vol. 9, no. 2, pp. 39–67, 2017.

[87] N. Alami, M. Meknassi, S. A. Ouatik, and N. Ennahnahi, "Impact of stemming on arabic text summarization," in *Proc. 4th IEEE Int. Colloq. Inf. Sci. Technol. (CiSt)*, Oct. 2016, pp. 338–343, doi: 10.1109/CIST.2016.7805067.

[88] Y. K. Meena, P. Dewaliya, and D. Gopalani, "Optimal features set for extractive automatic text summarization," in *Proc. 5th Int. Conf. Adv. Comput. Commun. Technol.*, Feb. 2015, pp. 35–40.

[89] T. El-Shishtawy and F. El-Ghannam, "Keyphrase based arabic summarizer (KPAS)," in *Proc. 8th Int. Conf. Inform. Syst. (INFOS)*, 2012, p. 7.

[90] N. H. Hassan, I. Al-Kabi, M. Mahmoud, and M. B. Issa, "Automatic keyphrase extractor from arabic documents," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 2, pp. 192–199, 2016, doi: 10.14569/IJACSA.2016.070226.

[91] Y. Ouyang, W. Li, Q. Lu, and R. Zhang, "A study on position information in document summarization," in *Proc. Coling*, 2010, pp. 919–927.

[92] A. P. Reynolds, G. Richards, and V. J. Rayward-Smith, "The application of k-medoids and pam to the clustering of rules," in *Proc. Int. Conf. Intell. Data Eng. Automated Learn.* Berlin, Germany: Springer, Aug. 2004, pp. 173–178.

[93] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural Multidisciplinary Optim.*, vol. 26, no. 6, pp. 369–395, Apr. 2004, doi: 10.1007/s00158-003-0368-6.

[94] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[95] C. Lin, A. Qing, and Q. Feng, "A comparative study of crossover in differential evolution," *J. Heuristics*, vol. 17, no. 6, pp. 675–703, Dec. 2011.

[96] K. Deb, K. Sindhya, and T. Okabe, "Self-adaptive simulated binary crossover for real-parameter optimization," in *Proc. 9th Annu. Conf. Genet. Evol. Comput.*, 2007, pp. 1187–1194, doi: 10.1145/1276958.1277190.

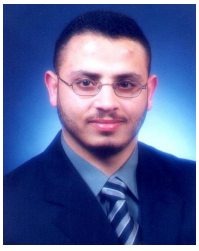
[97] K. Deb and D. Deb, "Analysing mutation schemes for real-parameter genetic algorithms," *Int. J. Artif. Intell. Soft Comput.*, vol. 4, no. 1, pp. 1–28, 2014, doi: 10.1504/IJAISC.2014.059280.

[98] A. M. Narasimhamurthy, "A framework for the analysis of majority voting," in *Proc. Scand. Conf. Image Anal.* Berlin, Germany: Springer, Jun. 2003, pp. 268–274.

[99] C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using N-gram co-occurrence statistics," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Hum. Lang. Technol. (NAACL)*, 2003, pp. 71–78.
[100] A. J. Nebro and J. J. Durillo, "jMetal: A java framework for multi-objective optimization," *Adv. Eng. Softw.*, vol. 42, no. 10, pp. 760–771, 2011.
[101] A. Nebro, M. López-Ibáñez, C. Barba-González, and J. García-Nieto, "Automatic configuration of NSGA-II with jMetal and irace," in *Proc. Genet. Evol. Comput. Conf. Companion*, 2019, pp. 1374–1381, doi: 10.1145/3319619.3326832.
[102] P. Das and R. K. Srihari, "Global and local models for multi-document summarization," in *Proc. TAC*, 2011.
[103] H. Liu, W. H. P. Liu, W. Heng, and L. Li, "The CIST summarization system at TAC 2011," in *Proc. TAC*, 2011.
[104] F. Hmida and B. Favre, "LIF at TAC multiling: Towards a truly language independent summarizer," in *Proc. TAC*, 2011.
[105] M. El-Haj, U. Kruschwitz, and C. Fox, "University of essex at the TAC 2011 multilingual summarisation pilot," Tech. Rep., 2011,
[106] J. D. Schlesinger, D. P. O'leary, and J. M. Conroy, "Arabic/English multi-document summarization with CLASSY—The past and the future," in *Proc. Int. Conf. Intell. Text Process. Comput. Linguistics*. Berlin, Germany: Springer, Feb. 2008, pp. 568–581,
[107] J. Steinberger, M. A. Kabadjov, R. Steinberger, H. Tanev, M. Turchi, and V. Zavarella, "JRC's participation at TAC 2011: Guided and multilingual summarization tasks," in *Proc. TAC*, vol. 11, 2011, pp. 1–9.
[108] A. Hernandez-Castaneda, R. A. Garcia-Hernandez, Y. Ledeneva, and C. E. Millan-Hernandez, "Extractive automatic text summarization based on lexical-semantic keywords," *IEEE Access*, vol. 8, pp. 49896–49907, 2020, doi: 10.1109/ACCESS.2020.2980226.

RANA ALQAISI received the bachelor's degree in computer systems engineering and the master's degree in computing from Birzeit University, Birzeit, Palestine, in 2015 and 2018, respectively. She has served as a Teaching Assistant for the Department of Electrical and Computer Engineering, Birzeit University, from 2015 to 2018. She is currently a Software Engineer with Asal Technologies. Her research interests include natural language processing, information retrieval, and machine learning.

WASEL GHANEM (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the Jordan University of Science and Technology, in 1990 and 1993, respectively, and the Ph.D. degree in electrical and computer engineering-electron devices & sensors from the University Elangen-Nuremburg and Fraunhofer Institute of Applied Research, Germany, in 1999. He is currently a Senior Faculty Member with the Department of Electrical and Computer Engineering, since 1999 through this period, he has been serving as the Dean of engineering and technology and the Head for the Department of Electrical and Computer Engineering, Birzeit University. He has also served as the Director for Palestine Education Initiative (PEI) to promote ICT in education within the schools systems by the Ministry of Education and Higher Education in Palestine. His current research interests include machine learning, natural language processing, smart grids, smart systems, and ICT in education.



AZIZ QAROUSH received the bachelor's and master's degrees in computer engineering from the Jordan University of Science and Technology, Irbid, Jordan, in 2003 and 2006, respectively. He has been an Assistant Professor with the Department of Electrical and Computer Engineering, Birzeit University, Birzeit, Palestine, since 2006. His research interests include machine learning, information retrieval, and natural language processing. He has published many articles in these areas.