IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 3, 2022, accepted March 22, 2022, date of publication March 30, 2022, date of current version April 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3163292

A Comprehensive Review of Arabic Text Summarization

ASMAA ELSAID[1,2], AMMAR MOHAMMED[1], LAMIAA FATTOUH IBRAHIM[1,3], AND MOHAMMED M. SAKRE[2]

[1]Department of Computer Science, Faculty of Graduate Studies of Statistical Researches, Cairo University, Giza 12613, Egypt
[2]Higher Institute of Computer Science and Information Technology, El Shorouk 11566, Egypt
[3]Department of Computer Science, Faculty of Information Systems and Computer Science, October 6 University, Cairo 12511, Egypt

Corresponding author: Asmaa Elsaid (12422019534235@pg.cu.edu.eg; asmaa.alsaeed@sha.edu.eg)

ABSTRACT The explosion of online and offline data has changed how we gather, evaluate, and understand data. It is frequently difficult and time-consuming to comprehend large text documents and extract crucial information from them. Text summarization techniques address the mentioned problems by compressing long texts while retaining their essential contents. These techniques rely on the fast delivery of filtered, high-quality content to their users. Due to the massive amounts of data generated by technology and various sources, automated text summarization of large-scale data is challenging. There are three types of automatic text summarization techniques: extractive, abstractive, and hybrid. Regardless of these previous techniques, the generated summaries are a long way from the summarization produced by human experts. Although Arabic is a widely spoken language that is frequently used for content sharing on the web, Arabic text summarization of Arabic content is limited and still immature because of several problems, including the Arabic language's morphological structure, the variety of dialects, and the lack of adequate data sources. This paper reviews text summarization approaches and recent deep learning models for this approach. Additionally, it focuses on existing datasets for these approaches, which are also reviewed, along with their characteristics and limitations. The most often used metrics for summarization quality evaluation are ROUGE1, ROUGE2, ROUGE L, and Bleu. The challenges that are encountered during Arabic text summarizing methods and approaches and the solutions proposed in each approach are analyzed. Many Arabic text summarization methods have problems, such as the lack of golden tokens during testing, being out of vocabulary (OOV) words, repeating summary sentences, lack of standard systematic methodologies and architectures, and the complexity of the Arabic language. Finally, providing the required corpora, improving evaluation using semantic representations, the lack of using rouge metrics in abstractive text summarization, and using recent deep learning models to adopt them in Arabic summarization studies is an essential demand.

INDEX TERMS Text summarization, arabic natural language processing, machine learning, extractive text summarization, abstractive text summarization, and deep learning models.

I. INTRODUCTION

Automatic text summarization has recently gotten great press. Because the internet generates vast volumes of text every day in various forms, existing text data is accessible electronically through the internet or on corporate or personal computers. Text summaries were created to solve the problem of having to read long texts on the same topic in order to grasp the key concept, saving time by creating a shorter text version of the text that contains the same ideas [1]. It also saves money when compared to a skilled human summary.

Applications of natural language processing include information retrieval, machine translation, questions and answers, and text summarization. Compared to Arabic, there is a lot more NLP research on Latin languages, notably summarization [2]. [3] highlighted a lack of Arabic language studies in NLP, particularly in summarization. Using automated summarization instead of human expertise saves money.

Most automatic text summarizing approaches are extractive, abstractive, or hybrid [4]. Some assessment techniques require extracting the text's most essential bits

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

(usually sentences). The length of the resultant summary is usually specified, either explicitly or implicitly. So, with a text of 40 phrases, an extractive algorithm can choose 10 out of 25 key sentences. However, abstractive summarization works as a human might. The algorithm examines the text, deciphers its content, and then utilizes its word combinations to characterize the document. This strategy might theoretically provide better and more compact memory. In practice, this is difficult because it requires understanding the material at the level of a well-educated human reader as well as the ability to apply it correctly. For these reasons, extractive summarization still benefits. Arabic is one of the world's oldest languages, with a wealth of information that archaeologists are still trying to uncover. Arabic is now an official language in 26 nations, with 280 million speakers worldwide. It is very descriptive language. A word in Arabic may mean various things. This is due to the complexities of the Arabic language's syntactic and morphological structures, as well as the compression ratio observed when summarizing multiple texts rather than a single document. There are no standard summaries for Arabic. There are no Arabic benchmark corpora, lexicons, or machine-readable dictionaries [3]. It is forbidden for non-expert Arabic scientists to research in Arabic since it is complex and difficult for them. ATS is a difficult issue. ATS causes several issues for the scientific community. Identifying the most informative text segments in the product, a summary of several documents; evaluation of the computer-generated summary without comparing it to a human-made summary; generation of an abstractive summary comparable to a human-made summary Researchers are still looking for an ATS system that can accurately summarize the main themes of a text. It is devoid of redundant or repetitive material and is easily understood by users. Ever since ATS research started in the late 1950s, many ATS surveys and methodologies have recently been published. Most research focuses on extractive summarization methods. Because abstractive summarization necessitates a high level of NLP, various methodologies and systems related to the automatic summarization of Arabic text were used in this research. paper [5]–[7].This paper describes various challenges that may be experienced while performing text summarization techniques and methodologies. Reference [8] discusses the ATS system and extensively mentions that of the Arabic language summarization. This proposed method presents a corpus study of the Arabic text summarization model, which will enable the researcher to specify a set of relations among the rhetorical features throughout the following empirical observations in the rhetorical frames. It proposes a method for automatically summarizing Arabic text and describes various challenges encountered while implementing text summarization techniques and methodologies.

The Arabic NLP (ANLP) continues to expand. The difficulty of the Arabic language constitutes the main obstacle to developing new techniques for automatic summarization. The NLP task was to carry out orthographic transliteration, orthographic normalization, and automatic diacritics [9] Other challenges with ANLP for Arabic summarization include the fact that the majority of ANLP tools were developed in the West for security reasons. The need for ANLP tools that can scan different Arabic documents to find names, places, dates, spelling corrections, and other things has become clear. Machine learning (ML) gave good results for non-Arabic speakers. However, due to the presence of sparse entities and structures in Arabic, the ML ingredient did not have enough data to make the proper generalization. Additionally, the Arab-developed ANLP had different aims and commonly used rule-based and machine-learning techniques [10]. The challenges for ANLP are, therefore, the normalization of the scripts, diglossia, agglutination, ambiguity, non-concatenative Arabic morphology, lack of capitalization, optional short vowels, and the syntactic structure of Arabic [11]. In addition to the above linguistic issues, also The Arabic summary is difficult to evaluate. Summarization evaluation may be done manually or automatically. However, Arabic has no gold-standard summary. Automatic evaluation is difficult because there aren't enough Arabic benchmark corpora, lexicons, and machine-readable dictionaries to use. The key challenge is establishing a gold standard against which the system's output may be measured. It's also difficult to define a good summary since it's subjective. In addition, other challenges such as being out of vocabulary (OOV) words, repeating summary sentences, lack of standard systematic methodologies and architectures, the complexity of the Arabic language, the lack of golden tokens during testing, The lack of using rouge metrics in abstractive text summarization, the lack of a gold standard corpus, a high reduction rate, input document length, and the summarization process's stop criteria will be discussed in section IV

The contribution of this paper is multifold. The first provides an overview of the three ATS approaches extractive, abstractive, and hybrid. The second represents each approach's overall architecture, benefits, and drawbacks. The third is the challenge of Arabic text summarization for extractive and abstractive text summarization and how it can be solved. The fourth provides an overview of automated text evaluation metrics and the datasets for the Arabic language.Finally, it provides an overview of the role of deep learning in Arabic text summarization with recent techniques.

The structure of this paper is organized as follows. SectionII depicts the classifications of ATS systems. SectionIII An illustration of how ATS systems approach. SectionIV provides an overview of Arabic summarization and challenges associated with it. Section V provides an overview of the standard Arabic datasets, as well as manual and automatic evaluation criteria and tools for computer-generated summaries, The Role of Deep Learning in Text Summarization, SectionVI.Finally, section VII concludes the paper.

II. AUTOMATIC TEXT SUMMARIZATION SYSTEMS CLASSIFICATION

This section gives an outline of the distinctive types used to design and implement ATS systems. For summarized texts, there is no unit categorization, and summaries can be classified according to a variety of criteria. Some of these characteristics were examined in detail in many studies. in Figure 1 illustrates the steps in extractive and abstract text summarization using DL models.

A. PREPROCESSING PROCEDURES

This process involves the identification of the input text specifications, which are genre, language, multilingual, subject specificity, size of summary, and type [12]. This process is required because, usually, texts in their raw state are not well structured. First, rows with NULL values in either new content or summary have been removed, and duplicate rows have also been removed. Then The most common techniques employed are segmentation, tokenization, stemming, stop word removal, removing any unwanted characters, normalization, lemmatizing data, and normalization data.

B. REPRESENTING DATA

In order to overcome the limitations of deep learning and neural networks in that they only take numbers as input, and since the text is a string (not a number), word embedding is employed to tackle this issue. The word embeddings were built by concatenating each new piece of content with its summary from a dataset of unique Arabic news. The dictionary now includes special tokens such as UNK, PAD, EOS, and SOS, which are used to substitute less frequent or unknown words, pad short sentences, start sentences, and end sentences. As a result, the model may be trained faster by analyzing the length of texts and summaries. This saves time and effort. If there were more than one UNK in the news content or any UNKs in the summary, some news would not have been included. This is designed to ensure the model is developed using relevant data. Finally, each phrase in news articles and summaries is an integer set.

C. SPLITTING DATA

In this phase, the dataset has been divided into three sets: training, validating, and testing. The training set was used to train our model, the validation set for validation, and the testing set (unseen set) for testing and evaluation.

D. BUILDING AND TRAINING MODEL

In this phase, they build deep learning models by setting up the model by defining the architecture, which is Seq2Seq. Bi-LSTM, RNN, BERT, and so on. Define the learning parameters such as metric of accuracy, loss function, optimizer, and the number of layers. These layers allow us to specify the sequence of transformations we want to perform on our input. Training the Model: Now that we have constructed the model architecture, we need to train the model. Training involves making a prediction based on the current state of the model, calculating how incorrect the prediction is, and updating the weights or parameters of the network to minimize this error and make the model predict better. We repeat this process until our model has converged and can no longer learn by using training hyper-parameters such as learning rate, epochs, batch size, and early stopping. Model validation is often referred to as the process where a trained model is evaluated with a validation data set. The validation data set is a separate portion of the same data set from which the training set is derived. The main purpose of using the validation data set is to test the generalization ability of a trained model. Model validation is carried out after model training. Model training, model validation aims to find an optimal model with the best performance.

E. INFERENCE MODEL

Model testing is often referred to as the process where the performance of a fully trained model is evaluated on a testing set. The testing set, consisting of a set of testing samples, should be separated from both the training and validation sets, but it should follow the same probability distribution as the training set. For testing and evaluating all variations of the testing set, the testing set is fed into the inference model to predict the summary.

F. EVALUATION

Three standard metrics are used to evaluate the quality of all variations of the DL model. The F-measure, ROUGE, and BLUE, which are nondifferentiable metrics qualified for comparing the generated summary to the reference summary and discussed in this sectionV.

Text summarization comes in three types. Generally, different document techniques (or multi-documents) involve information aggregation. Thus, most summary generation methods are abstractive, extractive, or hybrid [13], as illustrated in Figure 2. Automatic text summarization may take many forms. The main distinction is probably between extracts, abstracts, and hybrid summaries [2].

Extractive summarization Techniques of extractive summarization attempt to evaluate the importance of the identified sentences or words within the original text. It takes a fixed or perceived number of top-scoring sentences which is determined by the length of the original text. Finally, they rearrange the selected sentences to keep the original text's order no additional sentences are generated.

Abstractive summarization attempt to function in a manner similar to that of a human. They analyze the text and attempt to comprehend its meaning in order to extract the most important information and topics. Contextual knowledge is frequently required either through a knowledge base or, more recently, through the use of a machine learning model (most often various types of neural networks). Usually, they need an extensive data set for training the model to understand the concepts behind the words and sentences. Then, abstractive summarizers attempt to paraphrase the original text, which means they attempt to generate a grammatically correct and

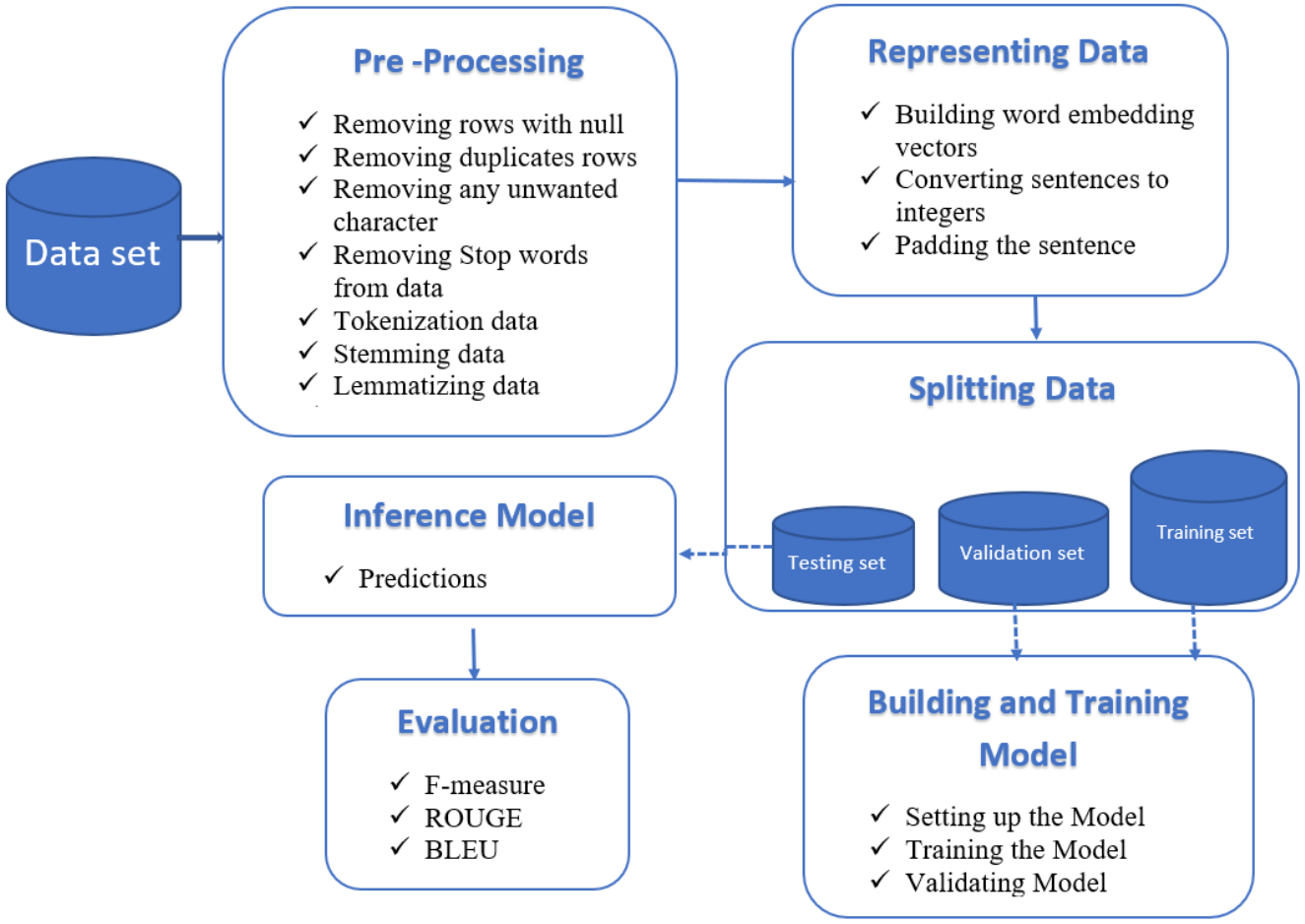

FIGURE 1. The architecture of Arabic text summarization systems.

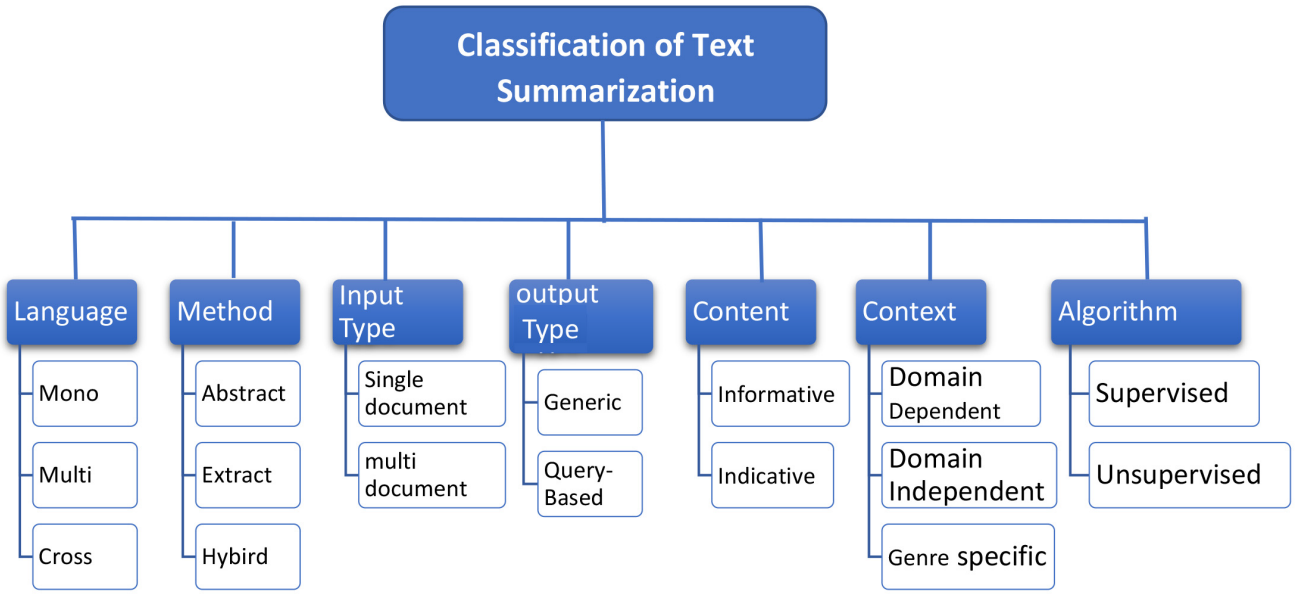

FIGURE 2. Classification of the ATS systems.

meaningful summary that conveys all of the most important information. In abstractive summarization, new sentences are generated as output. The sentences generated through abstractive summarization might differ from those present in the original text, as one might guess [14].

Hybrid summarization strategy combines the abstractive and extractive approaches typical of the traditional design of a hybrid text summarizer [15]. The generation of natural language text is still quite challenging, which is probably the main reason why abstractive summarizers usually do not perform well in general. They are, however, starting to achieve perfect results on some more restricted summarization tasks, like, for example, short article headline generation. There are many methods of summarizing based on the length of the final summary.

The *headline summarization* technique [16], [17] generates a one-sentence summary. Often only the first few lines are required for a system to perform well in this area.

Summarization of keywords [16], [17]. It generates summaries comprised of sentences and a list of the most significant words or brief phrases in the text. Abstractive summarizers often excel at this job since they are not required to provide grammatically correct and relevant material. The summary may be suggestive, informative, or critical, depending on the goal, type of details, and output style [5], [8]. The document's main concept is presented.

Inductive summaries offer a sense of the text's subject matter without imparting precise material, while *Informative summaries* provide a condensed version of the text's substance. In the case of a scientific publication, for example, it might offer a viewpoint. The most feasible type to automate is the indicative summary, and the least one is the critical summary [17], which can express an opinion of the document, scientific paper, and others. Another distinction is between general and query-based summaries [16], [17], which is based on the value of the content in the original text. The most common type is the *generic summarization* that imposes no limitations on the length of the original text or the summary. It makes no assumptions about the genre, purpose, reader, or task information [16], [17].On the other hand, in text-based summarizing, the user must first establish the topic of the original text as the basis of a query before beginning the summarizing process. In these cases, the user has a general understanding of the text and is looking for particular information, which is frequently a response [8], [12]. As a result, when a user submits a query, the system merely pulls certain pieces of information from the text and shows them as a summary. Summarization can also be divided based on the length and type of the input document. There are two commonly used terms: *single document summarization*, which focuses on input that consists of only one document, which is usually not very long [18] and *multi-document summarization*, which takes an input collection of documents and attempts to construct a summary from the set as a whole rather than summarizing each input document individually, which may result in a lot of redundancy [19]. If the reader is interested in a specific topic, some systems may accept a search query and specifically produce a summary focused on the parts of the original text that are most relevant to the search query. This type of summarization is called *query-focused* or *topic-focused* summarization. This may be particularly valuable in the case of search engines, which can show short, relevant snippets from the pages they return as results of the search query [13], [20]. Summaries can also be divided based on the type or genre of the input document. There are news summaries, literary summaries (where the input is a narrative document), specialized summaries, comprehensive summaries, specialized summaries (for specific domains, such as sports), and social media summaries (Twitter, Facebook, blogs). Additionally, summarizers are classified according to their input and output languages. The most often used summarizer is *monolingual summarizer*, which accepts input in a single language (English, Arabic, and others) and provides output in that language. Moreover, *multilingual summarizers* provide output in the same language as the input but can accept many languages, or even any language. Furthermore, there are so-called *cross-lingual summarizers* that can analyze inputs in many languages and generate a summary in another. This last kind is not supported by conventional extractive summarizers [8] and [5]. Another classification based on the context of an input text is t summaries, which are divided into three types: domain-dependent, genre-specific, and self-contained.

Genre-specific systems only accept a specific type of text as input, and the text template is restricted. Newspaper articles, scientific papers, tales, instructions, and other types of templates are available.

Domain-independent systems, on the other hand, have no predetermined constraints and accept a variety of texts [8], [12].

Furthermore, some systems only summarize texts so that their subject may be defined in the domain of the system; these systems are *domain-dependent*. These systems impose certain restrictions on the topic matter of papers. Such systems are well-versed in a certain area and take advantage of it.

This proposed research paper by [21] presents a survey based on the automatic text summarization system and is heavily dependent on Arabic language summarization. This proposed method presents a corpus study of the Arabic text summarization model, which will enable the researcher to specify a set of relations among the rhetorical features throughout the following empirical observations in the rhetorical frames. After that, the paper proposes a method that will automatically summarize the Arabic text. This paper [22] describes various challenges that may be encountered while performing text summarization techniques and methodologies.

III. AUTOMATIC TEXT SUMMARIZATION APPROACHES

This section will discuss many ways of extracting text summaries, which may be classified into the following

categories: statistical, graph, machine-learning, fuzzy-logic, and latent semantics approaches; and additionally, discourse approaches, which come from or are based on one or more of the previous approaches. Based on the type of learning, these approaches can be classified as supervised, semi-supervised, and unsupervised.

A. EXTRACTIVE AUTOMATIC TEXT SUMMARIZATION TECHNIQUES

Early research in extractive summarization focused on

1) STATISTICAL-BASED METHODS

These algorithms select the most important sentences and words from the original text based on a statistical analysis of certain features, methodology for calculating word frequency. It is the most commonly used method for scoring sentences [23], [24] [25]: The sentence's score is determined by the number of frequencies and avoiding all stop words. The proposed game plan to produce news titles by joining word-frequency, sentence position, and similarity gauges is drawing near. In information retrieval, TFIDF, short for term frequency-inverse report frequency, may be a numerical measurement planned to highlight a term's importance in a collection or corpus report. Summarization is generated based on features including similarity to a centroid sentence, in which the centroid sentence is captured based on TF-IDF. Then, each sentence is calculated to have a similarity value with the centroid based on cosine similarity, and then the feature values for each sentence are added together to get the sentence scores. A detailed review of techniques based on statistical approaches is discussed in [26] and [27].

2) GRAPH-BASED METHODS

These approaches use sentence-based graphs to describe a text as a cluster. Clustering is a method for identifying the most salient phrases from a text and eliminating duplication to create an effective summary. To reduce redundancy and avoid selecting sentences from the same cluster at the same time, cluster similar sentences in one group to eliminate them from the selection process. After that, you select sentences with a high score from each cluster to reduce redundancy and avoid selecting sentences from the same cluster at the same time. To eliminate repetition and boost relevance, they extract and give scores to the most important and distinctive phrases in the document. The method is based on a Page-Ranking algorithm [28], in which text as words or sentences are represented as nodes in a weighted graph with weighted edges determined by similarities between nodes. Both Text Rank and Lex Rank are graph-based approaches. In-Text Rank [29], the importance scores of nodes are determined using voting-based weighting, while in Lex Rank [29], it is a cosine-transform-based weighting algorithm. Text Rank and Lex Rank are fully unsupervised algorithms as they do not rely on the training set. However, instead, they depend on the entire text.

3) SEMANTIC-BASED METHODS

Latent Semantic Analysis (LSA) is a completely unsupervised approach for learning and capturing the contextual use and meaning of words using statistical calculations. By utilizing the semantic content of words, it avoids the issue of synonymy [30]. LSA is composed of three main steps: the generation of an input matrix, the use of singular value decomposition (SVD), and sentence selection. The input document is represented as a matrix in which the columns correspond to sentences, the rows correspond to words, and the cells correspond to the significance of words in sentences. A weight function is a function that calculates cell values. It may be normal, or DF-IDF, IDF, or entropy weight function [24], [31]. In singular value decomposition, the input matrix is decomposed into three other matrices to model the relationship between words and sentences (the first and third matrices represent the vector of extracted values for the original rows and original columns, respectively; the second matrix represents scaling values and the third matrix represents original columns as the vector of extracted values). In sentence selection, important sentences are selected from SVD results. Many other semantic-based techniques are used, as in [32]. Various techniques based on

Machine Learning Approaches are proposed, which can be classified into supervised, semi-supervised, or unsupervised approaches. To learn and detect essential aspects of sentences using supervised techniques, training data sets (labeled data) must be represented in a collection of texts with their human summaries. Regression, multi-layer neural networks, decision trees, support vector machines, genetic algorithms, and the Naive Bayesian Classifier are all examples of supervised learning approaches. Semi-supervised approaches depend on labeled and unlabeled data to produce a convenient classifier; for instance, Support Vector Machine (SVM) and Naive Bayes Classier are used as semi-supervised learning techniques [33]. On the other hand, unsupervised approaches generate summaries without needing training data. Unsupervised learning techniques (RBM, Autoencoder, Seq2seq, RNN, Transformer, Bert) are instances of unsupervised learning techniques. The model employs the numeric approach but for the Giga word. TF-IDF BERT, fine-tuning [34]Analogical Proportions [35] Word2Vec and Clustering [28]Natural Language Processing [18] F-RBM [18] Clustering algorithm are based on the Ara BERT model [36].To reduce redundancy, multi-document Arabic text summarization based on clustering and Word2Vec is used. Automatic Summarization of Arabic Documents use Unsupervised Deep Learning New approaches to the age of automated headline features in Arabic documents [29].Many new algorithms are also there to solve the problem of text summarization, like RNN [37], LSTM [38], Encode-Decode [39], [40], Attention [39], Transformer, Bert [22], [41]. Moreover, at the start of the model in the year 2021, the researchers published a new language model named Pegasus, and they evaluated this model in this field of text summarization. As of late, Google

AI Language specialists distributed another pre-prepared language portrayal technique. Transformers' bidirectional encoder representations are also known as Transformers' bidirectional encoder representations (BERT). It built a language comprehension model based on a massive content corpus [42]. Such models have demonstrated exceptionally effective language understanding by demonstrating that persuasion brings about most NLP undertakings [43].This approach depends on the interaction of computer programs with a dynamically active environment like a multiplayer game [44].

4) FUZZY-LOGIC APPROACHES

can model common sense reasoning in addition to dealing with uncertainty in an unsupervised manner. On the other hand, the classification solution is another task that appears using fuzzy logic to summarize the text. For example, in [45], the fuzzy-rough set aided in the extraction of critical sentences, in which the sentences are ranked according to their relevance using fuzzy relevance clustering. The relevance of each vector of these features maintains the following sentences: sentence position, length, TF-ISF, and semantic pattern. After that, these vectors are clustered by a fuzzy c-mean algorithm (FCM), and the relevance of the score is calculated for each sentence. Finally, choose candidate sentences with a relevance score higher than 0.5 and then the highest-scoring sentence from each cluster to create the final summary. By relying on senses rather than raw words, this strategy addresses the issue of sentences with the same semantic meaning but expressed in synonyms that are interpreted differently. In [46], a single document summarization approach is discussed based on nine features, including sentence centrality, position, length, the number of proper nouns, and others, using the combination of fuzzy rules and sets to pick up sentences based on their features. On the other hand, some researchers suppose that integrating fuzzy logic with other approaches will give better results, such as the previously mentioned approach, which integrates fuzzy sets with rough sets [47]. Another integration approach was proposed in [48], which incorporated fuzzy logic with swarm intelligence where feature weights are obtained from the swarm algorithm to adjust feature scores and use them as inputs for the fuzzy inference system to gather the final scores.

B. ABSTRACTIVE AUTOMATIC TEXT SUMMARIZATION TECHNIQUE

This section describes three types of abstractive automatic text summarization methods. The first is based on structure, such as graphs, trees, rules, and anthologies; the second is based on semantics, such as semantic text representation and common dialect period frameworks. The third kind of strategy is one that is based on deep learning (e.g., based on information items, predicate arguments, and semantic charts). [49]classifies abstractive techniques as neural-based or classical, which refers to any method that is not neural-based.

1) METHODS BASED ON GRAPHS

In [49], they suggest an abstractive summarizer called ''Opinosis,'' which makes use of a chart display. Each node acts as a word, and nodes are linked by positional information. The structure of sentences is represented by coordinated edges. The graph-based method's preparation processes include graph creation, constructing a textual graph to represent the original text, and summary creation. It may be used in any domain and does not need the involvement of human expertise [49]. [50]. By connecting all words on a word graph route, a new phrase is created [51]. The disadvantage of this strategy is that word charts do not represent the meaning of the words. Due to the way nodes are stored, sentences composed of nodes cannot be combined [51].

2) METHODS BASED ON TREES

These algorithms find similar comparison statements and combine them to generate the abstractive summary [52]. Similar sentences are represented by a tree. Dependency trees are the most frequently used tree-form representations for text. Trees are managed through pruning, linearization (converting trees to strings), and other methods [52]. The method's advantages include enhanced quality generated summaries since language generators provide fewer redundant and fluent summaries [52]It is not feasible to discern relationships between sentences without first locating common terms. Because it ignores the context, it misses a variety of important phrases within the material. The effectiveness of this technique is limited by the available parsers. It is more concerned with syntax than semantics. [52].

3) METHODS BASED ON RULES

These methods need to establish the rules and categories in order to determine the most essential ideas in the input text. This method's stages are as follows: To construct an abstractive summary, one must first categorize the input text based on words and ideas, then formulate the questions based on the input text's domain, and at that point react to the questions by searching the text for terms and ideas. The generated summaries are high in information [52]. The ability to handle additional data is by increasing abstraction scheme complexity and variety.

4) METHODS BASED ON ONTOLOGIES

Each domain has its own set of articles, each with its own information structure. Data arrangement an ontology such as [53] may be expressed. The basic idea is to utilize an ontology to extract relevant information from a text and construct an abstract summary. It is based on publications from a certain domain. It can deal with text uncertainty [52] and provide logical summaries [54]. To do so, it needs a domain-specific ontology, which takes time to construct [52]. So, more time is needed.

5) METHODS BASED ON SEMANTICS

Traditional dialect period frameworks are utilized as a verb and noun phrases to generate the final abstractive summary [52].In [55], they propose a multi-document abstractive summarizer that 1) utilizes SRL to talk to input documents 2) uses SRL to communicate with output documents 3) cluster semantically identical predicate-argument structures throughout the content 4) order the predicate-argument structures based on the semantic proximity metric. The SRL may be used to bind words together [50]. The quality of the output summary is determined by the input text's semantic representation.

6) DEEP-LEARNING-BASED APPROACHES

sequence-to-sequence learning (seq2seq) [56] has made abstract summarization possible. Seq2seq has been used successfully in NLP applications such as machine translation, speech recognition, and conversation systems. Deep learning still has concerns with 1) creating repeated words or phrases and 2) not dealing with terms that are not in the vocabulary. RNN models use attention encoder-decoder to summarize material effectively. However, deep learning approaches still suffer from challenges such as 1) creating repeated words or phrases and 2) the inability to cope with words out of vocabulary (OOV) (i.e., unusual and limited-occurrence words) [56]. The summarizing mechanism of [56] goes like this: 1) Separate the actual items (for example, news reports) and their summaries. 2) After preparing the data with a sub-word display, do word segmentation. 3) using a pre-trained Genism toolkit to initialize the word vectors, with one layer of Bilstm for the encoder and a unidirectional LSTM layer for the decoder. The cost function was optimized using the Adam optimizer (loss). Deep learning models are typically employed for brief text summaries [51]. Combining multiple approaches and tactics are advised to develop improved abstractive summaries. It is very promising to combine results from numerous ATS techniques to provide considerably better summaries than those created by individual algorithms [57]. These techniques are typically used in hybrid summaries, while semantic or deep learning-based methods are used in abstractive summaries [52]. These methods might be employed in preprocessing to extract key terms from the input text and then utilized to generate the abstractive summary [52]. Reference [51] that creates an abstractive ATS system using semantic data transformations and encoder-decoder deep learning models. Seq2seq excels at short text summarization. RNN-based Seq2Seq models require extensive training and are incapable of detecting removed dependencies in long sequences [58]. It creates repetitive compounds and incorrect information when applied to noisy social media [58].

C. HYBRID AUTOMATIC TEXT SUMMARIZATION TECHNIQUE

They begin with the extracted phrases that are then submitted to one of the abstractive text-summarizing algorithms. Reference [53] presents the "EA-LTS" hybrid approach for summarizing long texts. The system has two stages: extraction (using a graph model) and abstraction (using an RNN-based encoder-decoder, a printer, and attention approaches). The hybrid summarization approach may be investigated. Reference [53] researchers create hybrid ATS systems that combine extractive and abstractive approaches. Extractive and abstractive procedures are employed to improve the summary's quality.

IV. ARABIC TEXT SUMMARIZATION AND ASSOCIATED CHALLENGES

This section outlines a few of the basic challenges of Arabic text summarization for both extractive and abstractive methods. Despite the early work on text summarization in English that started as early as 1958 [59], the attempts to make automatic Arabic summarization started very late. Arabic has a complicated morphology based on "root-and-pattern". A root is a group of consonants. A word's meaning is defined by its root and embedded in a pattern. If this adds to the interpretation process, NLP stemming may help in several languages (for example, to reduce the dimensionality of vector-space models). However, determining a word's root, or stem, in Arabic is difficult to automate. In addition, many words' roots have an abstract meaning that is not suitable for NLP. Furthermore, Arabic words may be "borrowed" from different contexts, creating ambiguity and confusing mechanical interpretation. Reference [60] Even though research on such topics has been scanty in comparison with the English language, for example, the Arabic particulars (e.g., morphological richness and orthographic ambiguity because of the optional diacritics) may lead to a more significant number of homographs and, therefore, more ambiguity than in English [9]. These include the following:

- Arabic has twenty-eight letters. The language has eight diacritics that produce different phonetics of alphabetic letters. It requires a sophisticated analysis to determine the correct diacritic, which helps to gain the appropriate meaning of the word and sentence. This yields a high degree of ambiguity, morphologically and syntactically [61].
- Arabic's lack of capitalization makes it difficult to distinguish proper names, titles, acronyms, and abbreviations.
- Arabic includes 28 letters each of which has a different form depending on its placement.
- Other challenges are in Arabic semantics, which is the science of the meanings of a text. The incredible complexity of the Arabic language is an obstacle to NLP.
- Arabic has several morphosyntactic distinctions from other languages. Broken plurals are also an issue. English split plurals differ from singular forms.
- The Arabic language has been missing from several operating systems. The main challenges for the Arabic language were partially solved through the encoding

process (or encoding design). However, the encodings had several flaws that made natural language processing (NLP) difficult.

A. CHALLENGES AND SOLUTIONS

Arabic text summarization approaches have faced various challenges, and although some have been solved, others still need to be addressed. These challenges include: The complexity of the Arabic language: Comprehensive morphological analysis with reasonable accuracy is essential. A preprocessing stage should be created to remove the morphological complexity from the data before it is used. An accurate preprocessor needs to know how to use stop words, modern Arabic rules, and different dialects of Arabic to be accurate.

lack of gold standard corpus: The fundamental problem with the text-summarizing dataset is the quality of the reference summary (Golden summary). Consequently, obtaining an excellent dataset takes considerable time and effort. A multi-sentence dataset for abstractive summarization is also not accessible for several languages, including Arabic. Single sentences are available in Arabic for abstractive text summarization.

The lack of using rouge metrics in abstractive text summarization : A common evaluation metric (the ROUGE score) cannot be used to evaluate abstract summaries since it assesses n-gram matching. Moreover, the abstract summaries may include terms not found in the original texts. therefore, a new evaluation measure must be proposed to consider the context of the words.

Out-of-Vocabulary (OOV) words One of the problems that may happen during testing is that the main words in the test document may not be used or seen during training. These words are called OOV words.

Summary Sentence Repetition and Inaccurate Information Summary. It's also difficult to define a good summary since it's subjective. Another difficulty is the subjective nature of the summary; its quality varies from person to person. The quality of a summary varies depending on the reader's interest in the text's body.

the lack of golden tokens during testing When training, previous tokens in the headline may be input into the decoder using ''golden tokens. During testing, the golden tokens are not available, so the next step in the decoder is restricted to input from the previously generated output word. To resolve this problem, which becomes more difficult when dealing with small datasets, we need to use a different approach as long as at least the training step receives the same input as the testing step, there are a lot of different ways to solve this problem. In all cases, the decoder's first input is the ''EOS'' token, and the same calculation is used to determine the loss. In addition, the mass convolution of the QRNN is used in [62] because it is hard to predict how words will be linked in the future.

Another issue is the high reduction rate: single document summarizer extracts aim to be 5 to 30 shorter than the original text. However, multi-document summaries for handheld devices have substantially lower compression rates. This is a difficult task since such a high decrease rate requires specialist expertise.

Another difficulty is with Input Document Length: The majority of ATS systems are designed to work with very short text documents. For instance, a news article is shorter in length than a novel chapter. While ATS approaches perform well when summarizing short texts, they perform poorly when summarizing large texts.

Another difficulty is with the summarization process's stop criteria: humans use an iterative method to summarize documents. After generating the first summary, the author (or the system) must select whether to stop or continue the summarizing process. The most common method is to use a retention rate as the basis for decision-making. The retention rate is not similar across all texts. It should vary depending on the content and style of the text. It is critical to propose a more effective method for halting the summary.

Another difficulty is with text summarization using deep learning: Large-scale structured training data is required for deep learning models, such as the seq2-seq, AraBert, and RNN, as described in the section VI on the summary generation step. In real-world NLP applications, the necessary training data isn't always readily accessible. The use of classic NLP approaches such as syntactic, grammatical, and semantic analysis to develop an ATS system with little training data is an interesting research issue.

Another difficulty is with text summarization approaches. Abstractive and hybrid summarization systems need more research, not just the extraction method [63].

V. TEXT SUMMARIZATION EVALUATION METRICS AND DATA SETS

This section includes automated text evaluation metrics and the data-sets for the Arabic language. All the standard methods use some variation of comparing the automatically created summary with a set of human-created, or so-called golden summaries. While several standard testing datasets and their golden summary sets for English exist, it is challenging to find something similar to the Arabic language.

A. STANDARD DATASETS FOR ARABIC TEXT SUMMARIZATION

The accuracy of automatic text summarization relies on data collection, text size, and sentence count. However, unfortunately, the Arabic language does not contain benchmarks like other languages, and most of the available data sets for Arabic contain text with short sentences, not enough to generate an accurate summary. The data set size used in Arabic text needs a more extensive vocabulary to give better results, like in other languages. According to the majority of polls [22], [74],this will provide an overview of the summarizing corpora. English, Chinese, and other ATS systems' original standard benchmark data sets are presented here. Table 1 shows

TABLE 1. Standard datasets for Arabic text summarization.

Dataset Name	Document Description	Domains	Single document	Multi-document	Abstractive	Extractive	Notes
Duc2004 [64]	100*10	News	√	×	√	×	the summary in DUC2004 is written in English instead of Arabic, while the text is written in Arabic.
EASC [60]	150 document *765 human-generated summaries	News	√	×	×	√	
KALI MAT [65]	contains 20,291 for a single document and 2,057 for multi-document	News	√	√	×	√	
Giga word 5 [66]	2716995 Documents	News	×	√	√	×	Not free
OSAC [67]	22.428 document	News	√	×	√	×	
SANAD [68]	190000 document	News	√	×	×	√	
RTA news [69]	23,837 document	News	√	×	×	√	
NADA [70]	13,066 document	News	√	×	√	×	
Multi-document Summaries Corpora [19]	30 document	News	×	√	×	√	
XL_Sum [71]	46897 document	News	√	x	√	x	
WikiHow [72]	29,229 document	News	√	x	√	x	
AHS [38]	300k document	News	√	x	√	x	
AMN [73]	265k document	News	√	x	√	x	

the most often used benchmark data sets for Arabic automatic text summarizing systems evaluation including

Document Understanding Conference Datasets (DUC2004) [64] is a dataset for abstractive single-document summarization. It has 100 Arabic news articles with four human-written summaries apiece. It consists of 50 TREC document clusters. Each cluster has 10 documents on average. However, the summary is written in English, whereas the text is written in Arabic. These datasets don't have enough data to train neural network models. They are usually used to evaluate the ATS systems, but not enough to train the models with.

Essex Arabic Summaries Corpus (EASC) [60] It is a popular dataset for single-document extractive summarization research. It contains 150 articles and 765 human-generated summaries of those articles [34], [39] [28], [35] [40], [43] [75].

The KALIMAT Dataset [65] can extract summary summaries from single or multiple documents. It contains 20,291 single documents for texts with summaries and another 2,057 for multi-documents. The dataset was created from the Omani newspaper Alwatan. The data is news documents [34].

Gigaword 5 Dataset [66] An additional collection of news articles used for summarizing is the Gigaword 5 Dataset. There are no summaries associated with the source articles in the dataset. However, some previous work used a small part of this dataset and made pairs of summaries by taking the first line of an article and its headline. This makes the dataset good for short text summarization tasks, but it is not free.

The OSAC Dataset [67]is a popular dataset for single-document abstractive summarization. It contains 22,429 news documents.

The SANAD Dataset [68] is a large collection of Arabic news articles and can be used for single-document extractive summarization. The articles were collected from three popular news websites: Al-Khaleej, Al-Arabiya, and Akhbarona. SANAD contains a total number of 190,000 articles [76].

The RTAnews Dataset [69] is a collection of multi-label Arabic texts, collected from Russia Today in the Arabic news for single-document extractive summarization. RTAnews It contains a total of 23,837 articles, spread over 40 categories

New Arabic Dataset for Text Classification (NADA) [70] contains a set of newswire texts which are taken from two existing corpora, OSAC, and DAA, for single-document abstractive summarization. The data set contains 13,066 articles. It is used in text-based parts of NLP, like text classification, text summarization, and so on.

Multi-Document Summaries Corpora [19] It can be used for generic extractive summarization of both Arabic and English multi-documents. They translated the English gold-standard summary into Arabic by using Google translate. The data set contains 30 articles.

The Xl_Sum Dataset [71] can be used for a single document for abstractive summarization. It contains 1 million texts with short summaries. This dataset was created from the BBC website's news using a set of well-designed algorithms. The dataset contains 44 languages, ranging in resource availability from low to high, several of which lack a publicly accessible dataset. As shown by human and intrinsic evaluation. It contains 46897 Arabic articles and the human-generated abstractive summaries of these articles.

The WikiHow Dataset [72] contains 770,000 articles and summary pairings from WikiHow in 18 languages. Images used to show how-to steps in an article were matched so we could find gold standard alignments across languages. It includes 29,229 Arabic newswire texts and a summary of a single abstractive document.

The Arabic Headline Summary (AHS) Dataset [38] can be used for a single document for abstractive summarization. It contains 300k texts and their titles without any summary. Consider the title as a summary of it. This dataset was created from the Mawdoo3 website's news [77].

The Arabic Mogalad Ndeef (AMN) Dataset [73] can be used for a single document for abstractive summarization. It contains 265 k news texts and their summaries.

Table 1 defines the following attributes for each dataset: The dataset name, the number of documents, the data domain (e.g., news or blogs), whether single-document or multi-document summarising is supported, and whether the summary is extractive or abstractive, The summarization dataset has "100 × 10" documents, meaning it has 10 clusters of documents, each with about 100 documents. As demonstrated in Table 1, most existing datasets focus on the news domain, so additional datasets that support the Arabic language and cover other data domains are required. If the researchers test their proposed ATS systems on a lot of different datasets, they will spend a lot of time. They usually use only one or a few corpora in their research in the field of ATS.

B. SUMMARY EVALUATION

Evaluation of automatically generated summaries is a rather complex problem of its own. There are two basic approaches: either fully manual or semi-automatic. Table 2 shows a summary of the evaluation measures and methods in the surveyed literature used for Arabic automatic text summarizing systems evaluation.

1) MANUAL EVALUATION

Manual evaluation needs human judges to read the original text, the summary, and then subjectively rate the summary's quality. The National Institute of Standards and Technology (NI ST) specifies various criteria for judges to evaluate the summary in terms of linguistic non-redundancy, referential clarity, focus, structure, and coherence. It also defines a 1 (worst) to 5 (best) point qualitative scale. However, there is no perfect summation, and all evaluations are subjective [22].

2) AUTOMATIC EVALUATION

An excellent automatic metric for summarization needs to rank the quality of the selected information content and potentially the fluency of the output summary. Evaluating abstractive text creates additional challenges since the output summary may contain words that are not part of the input article. Despite these obstacles, various automated measures allow easy comparison of different summary algorithms and give some insight into the generated summaries' quality.

ROUGE is a set of measures and a software framework for evaluating automated summarization [78]. It is perhaps the most well-known and widely used software for this task, and it comes with standard text datasets and golden summaries. Each text needs a set of human-produced (golden) summaries. The most popular varieties are ROUGE-N, ROUGE-L (Longest Common Subsequence), ROUGE-W (Weighted Longest Common Subsequence), ROUGE-S (Skip-Bigram Co-Occurrence Statistics), and ROUGE-SU (Extension of ROUGE-S). This project does not cover the functionality of each measure in depth. Also,

TABLE 2. Evaluation measures used in the surveyed literature.

Reference	Year	Model	Automatic Evaluation Metrics
[77]	2022	Seq2Seq	Rouge1= 0.5149 , Rouge2= 0.12, RougeL= 0.343 and BLEU =0.41
[76]	2022	Seq2Seq	Rouge1= 0.384, ROUGE1NOORDER=0.462, ROUGE1-STEM=0.526 and ROUGE1-CONTEXT=0.581
[24]	2021	Supervised Machine learning and score-based algorithms	Rouge1= 0.643, Rouge2= 0.617
[34]	2020	BERT Model	Rouge1= 0.42, Rouge2= 0.2459 and RougeL= 0.422
[28]	2020	Word2Vec,Clustering and W- PC	Rouge1= 0.664, Rouge2= 0.559
[36]	2020	Ara BERT Model and Clustering algorithm	Rouge1= 0.54 and Rouge2= 0.51
[41]	2020	Arabic Pretrained , and LSTM	Rouge1= 0.38 and ROUGE1NOORDER=0.46
[35]	2020	Bert for Multi Lingual	ROUGE-1 = 0.75 and BLEU-1 = 0.47 for the First algorithm ROUGE-1 = 0.74 and BLEU-1 = 0.49 for the Second one
[38]	2020	Seq2Seq	ROUGE-1 = 0.443
[43]	2019	Restricted Boltzmann Machine (RBM)	ROUGE-1 = 0.44 and Rouge2= 0.34
[40]	2019	Linear Discriminant Analysis (LDA)	ROUGE-1 = 0.450 and Rouge2= 0.307
[40]	2018	Autoencoder	ROUGE-1 = 0.450 and Rouge2= 0.307

although certain ROUGE measures outperform human evaluation, this may not be true for other human evaluation methodologies or corpora, such as Arabic. The underlying principle is the comparison of word (or n-gram) distribution with golden summaries. It is defined according to formula1

$$\text{ROUGE-N} = \frac{\sum_{S\in\ Refsum} \sum_{gramn\in s} countMatch(gramn)}{\sum_{S\in\{Refsum\}} \sum_{gramn\in s} count(gramn)} \tag{1}$$

where n refer to the length of the n-gram, gram n, and Count match (gram n) is the greatest number of n-grams co-occurring in a generated summary and ground truth. Comprehensive summarization approach. It is suitable for small datasets where human effort is involved in summarizing the original text and comparing it to the system summary. Additional necessary measures for measuring the quality of document summarization are

Precision, Recall, and F-Measure [78]. Precision, the proportion of relevant documents retrieved, is defined according to formula 2. Similarly, the reference summaries evaluate the accuracy of the correctness. The recall evaluates how much information in the reference summaries is covered by the automated summary, is defined according to the formula in3. Note the trade-off precision and recall (the increase in one tends to decrease the other). Finally, the F-measure the combined precision and recall is defined according to the formula in 4.

$$Precision = Correct/(Correct + Wrong) \tag{2}$$

$$Recall = Correct/(Correct + Missed) \tag{3}$$

where "correct" is the number of sentences that are the same in both the human and system-generated summaries, and "wrong" is the number of sentences that are provided in the system-generated summary but not in the human-generated summary. The number of sentences that do not exist in a system-generated summary but do appear in a human-produced summary is called "missed." The F-measure is then calculated as follows:

$$F = (2 * (Recall * precision))/((Recall * precision)) \tag{4}$$

The fact that these metrics are easy to compute is their main benefit. However, using these measurements has several drawbacks. Initially, they just compare the summary to the reference summaries. Lacking a proper reference summary, this judgment may be skewed. These measures also penalize summaries that employ sentences that the reference summaries do not use, even if they are comparable. For example, in multi-document summaries, reference summaries choose a single line from a group of similar sentences in the documents. Bleu's precision measurements are also important for document summary quality evaluation.

BLEU (The Bilingual Evaluation Understudy) Score evaluates a generated sentence against a reference sentence. The machine-generated sentences (and n-grams) appeared in the human reference [79]. The fact that these indicators are simple to calculate is their main advantage. Adopting these metrics has significant disadvantages, including a lack

of meaning. Do not explicitly consider sentence structure, and handle morphologically rich languages effectively. The results from human references must have a high Bleu if discovered with many terms from the system, and a high Rouge if found with many terms from the system. BLEU incorporates a shortness penalty term and computes the n-gram match for a range of n-gram sizes (unlike ROUGE-n, where there is only one chosen n-gram size). The Bleu is then calculated as follows56: First, compute a brevity penalty which looks for the reference with the most similar length by

$$BP = \begin{cases} 1 & c > r \\ e^{1-r} & c \leq r \end{cases} \tag{5}$$

where c and r are a candidate summary and a reference summary. Finally, the BLEU score is computed by

$$BLEU = BP * \exp(\sum_{n=1}^{N} w_n \log P_n) \tag{6}$$

wherep_n is the n-gram precisions score and w_n is positive weights Finally, we might utilize the F1 metric to link the measures together, as in formula 7.

$$F1 = 2 * (Bleu * Rouge)/(Bleu + Rouge) \tag{7}$$

VI. DEEP LEARNING-BASED ATS

Deep learning is a representation learning approach that employs a cascade of several nonlinear processing units to execute transformations and feature extractions, with the output of one layer being used as an input to the next [39], [80]. Through multiple levels known as ''feature layers,'' deep learning algorithms may learn from inputs in a supervised or unsupervised manner. Humans do not specify or create the feature layers; instead, deep learning algorithms are automatically learned through a generalized learning process. Deep learning is a set of algorithms that focus on learning abstract representations of data at several levels using a variety of nonlinear transformations. Following the development of neural networks, it has become one of the most prominent fields in recent years. Traditionally, text summarizing methods involve explicitly extracting terms from the textual material. Among other things, stop words are removed, noun clusters are identified, and lemmatization is performed.

The most significant drawback of traditional approaches is that the resulting summary may contain unnecessary words. Words may repeat themselves in the summary, as they do in the main text because there is no record of the words that have previously been chosen. In addition, the link between the produced summary and the document is shallow in traditional techniques [81]. As a result, consumers may find it challenging to understand the document from the summarized content. To solve these drawbacks, deep learning approaches for text summarization are used.An RNN model can work with inputs of any length. An RNN model is designed to recall each piece of information throughout time, which is very useful in any time series prediction. Even if the input size is enormous, the model size remains constant, and the weights may be shared between time steps. The disadvantages to problems that can arise in training RNN are ''exploding gradients'' and ''vanishing gradients.'' To solve this problem, we use LSTM. LSTM is a great tool for anything with a sequence. Because the meaning of a word is determined by the words that come before it, the path is paved for NLP and text analysis to use Neural Networks. The cons of LSTMs are that LSTMs take longer to train, need more memory, and are easily overfit. Dropout is more difficult in LSTM because of its inconsistent weight initializations. Seq2seq performs sequence-related tasks like time series. The problem of Seq2seq It is challenging to implement it in long sentences. Bi-LSTM It solves the fixed sequence to sequence prediction issue. The input and output of a vanilla RNN have the same size, and If input and output text have different widths, or text summarization has a different length, then machine translation is a problem. The cons of Bi-LSTM are Since Bi-LSTM has double LSTM cells, it is costly and not a good fit for speech recognition. Attention tasks increase accuracy and can work effectively for long sentences. The cons of attention add extra weight factors to the model, increasing training time, particularly if the input data is lengthy sequences. Attention handles fixed-length text strings. The text is separated into pieces or chunks before being delivered into the system. Transformer Unlike transformer models, LSTM or RNN models are sequential and must be processed in sequence. Due to their parallelization capabilities, transformer models can handle substantially more data in the same amount of time. The cons of the transformer attention can only deal with fixed-length ''text strings.'' The text is separated into parts or chunks before being sent into the system as input. This text chunking produces context fragmentation. Bert, this is the best summary yet. No additional training is necessary because BERT models are pre-trained on massive datasets. It employs a flat design with inter-sentence transforming layers to provide the best summary results. The cons of Bert It is compute-intensive at inference time, so using it at scale may be expensive. One limitation is the availability of big Arabic summarization corpora. Text summarization uses unsupervised deep learning to construct a compressed version of the original document. Unsupervised learning uses unlabeled data. It is a machine learning algorithm that draws from datasets consisting of unlabeled input data. Deep learning may be used for unsupervised or supervised learning. Arabic is solved using deep learning algorithms [82].When working with natural language, the textual data must be expressed in a way that a machine can understand. Word embeddings are a common method of representing text. Embedding is a mathematical method of mapping objects from one domain to objects from another. It is used as a tool in NLP activities. Embeddings can be created in either a supervised or unsupervised manner. Unsupervised embeddings are more general and can be used in a wider range of situations. An object can be described by other objects. Word embeddings have been in use since the 1960s. A popular type of embedding is word

embeddings such as word2vec, Glove, Centroid Net, Arabic, Fast Text, and others. In Word2vec, researchers were able to train word embeddings in a short amount of time using large datasets [41] and [38]. Word embeddings work by combining words to encode them. Word2vec uses two approaches to do this: CBOW [10], [10] and skipping grammes [70].The CBOW method predicts the current word from nearby words. The skip-gram approach predicts neighboring words based on the present word. Fast text provided sub-word embeddings, which allowed it to create an embedding even if the term had never been seen before [83]. Global Vectors for Word Representation uses combinations of word vectors that describe the probability of these words' co-occurrence in the text [33] for the English Language. For the Arabic NLP research community, AraVec is a free-to-use distributed word representation (word embedding) open-source tool [84]. Concept Net: The number group can be seen as a substitution for other precomputed embeddings, such as word2vec and glove, that do not incorporate the graph-style information in Concept Net. The number group beats these datasets on benchmarks of word similarity [85]. And large vocabulary. However, even with a large vocabulary, there is a possibility that you will come across certain terms that are not in your vocabulary. Using byte pair encoding instead of words may help prevent OOV [86]. (BPE). Internally, this approach generates embeddings for these sub-word units. Recurrent neural network (RNN): In an RNN, the new yield is subordinate to the past yield. Because of this RNN feature [50], we try to summarize our content as human-like as possible. Long short-term memory (LSTM) is a form of RNN that can learn long-term dependencies, solving the problem of a simple RNN's short-term dependencies. An RNN cannot grasp the context behind the input when trying to do this using an LSTM [38], [41]. The two basic components of a sequence2sequence RNN architecture are an encoder and a decoder [39]. When the inputs and outputs have varying lengths, these are used. Different LSTMs are configured successively to encode and then decode the input in this approach [36], [39].It converts the entire input into a format that can be processed. This component converts the processed input into a stable and required output. Bidirectional LSTM, or bi LSTM, is a sequence model that consists of two LSTMs: the first passes the input in a forward direction and the second in a backward direction. In the input sequence, Bidirectional LSTM trains two LSTMs instead of one. The first is in the input sequence, while the second is inverted [38]. Bahdanau and Luong, two scientists, suggested ''attention'' as a solution to the Seq2Seq challenge. As the model progresses, it might concentrate on various sections of the input sequence [87]. Human visual attention processes are roughly modeled in neural networks [39]. To tackle sequence-to-sequence issues, NLP's transformer design handles long-range relationships easily. Transformers use several attention techniques to compute values [34], [84]. BERT might be used as a transformer to overcome RNN and other neural system limitations. It is a bidirectional pre-trained model [34]. BERT word embedding makes use of a transformer bidirectional encoder multilayer [84], [88]. A transformer-based based on par-BERT combines the representations of words and sentences to generate a single-layer transformer. Additionally, a significant amount of unsupervised objective text was utilized to train BERT. BERT is a feature-based method that may be fine-tuned to meet the objectives of certain functions. Furthermore, learning the significance of word pairs in self-attention improves the transformer [84]. Transformers are used to learn contextual representations of language from large datasets. BERT is one of the new language representations that extend word embedding models. Two tokens are placed in the text in BERT. The initial token (CLS) is used to aggregate the information about the whole text sequence. The second token is (SEP); this token is used to express it after each sentence. The resulting text is composed of tokens, each of which is given one of three kinds of embeddings: token, segmentation, or location. Token embedding is used to provide information about the meaning of a token. Segmentation embedding is used to identify the sentences, whereas position embedding is used to determine the token's location. The bidirectional transformer is fed the sum of the three embeddings as a single vector. Pretrained vectors of word embeddings are more accurate and include a huge number of semantic features. BERT has the benefit of being fine-tuned (following the aims of certain tasks) and using feature-based methods. In addition, transformers use self-attention to figure out how the input and output presentation should look, which allows people to learn about the ''word-pair'' significance [89].

Reference [39] They suggested a novel methodology for Arabic text summarizing, demonstrating better extraction capability and higher summary quality. They adopted the offered strategy. Two ways of summarizing are presented: a graph-based approach and a query-based approach. A query-based strategy: They discovered that the models had improved. Both summarizing approaches are used. The primary downsides of this technique are that training huge datasets takes a very long time and that determining the optimal parameters for the network is a difficult challenge.

Reference [34] They proposed a new method for both abstractive and extractive summarization using pre-trained BERT and encoder BERTSUM. The result illustrates how multilingual BERT may be used to summarize Arabic material in low-resource scenarios.

Reference [28] They used unsupervised techniques, focusing on text summarization. Problems such as noisy information, redundancy elimination, and sentence order the K-means clustering technique was used to pick the important phrase. The method has achieved an F-score of 0.644.

Reference [36] They proposed a new method for extractive text summarization by combining NLU (Ara BERT) and clustering algorithms. The experiments show a rough F-measure score of 0.51 and, by the expertise, a measure score of 0.52. The suggested system's drawbacks include a drop in accuracy when the text is too long and the extracted sentences contain linguistic terms that cause the summary to be misunderstood.

Reference [24] They have used extractive summarization methods to generate the summary, such as score-based and supervised machine learning techniques.

Reference [41] They used the abstractive summarization method. They have used AraVec pre-trained for word embedding. After that, they used the encoder-decoder sequence mechanism to produce the summary. It is an LSTM RNN architecture. As a result, this model is more effective than some of the other generalized models for the text summarization of the Arabic language.

Reference [35] The algorithms that they used are called BERT for multilingualism. They have used it for extractive and well-abstractive purposes. A word2vec model is being used for the comparison of the systems that have been created. This marks the quality of the model and compares it to the quality of the text summarization technique. The quality of word embedding does highly affect the quality of the generalized summary. Experimentation has shown that a ROUGE-1 = 0.75 and BLEU-1 = 0.47 for the First algorithm and ROUGE-1 = 0.74 and BLEU-1 = 0.49 for the Second one.

Reference [77] They proposed an abstractive summarization system using a seq2seq model using GRU, LSTM, and BILSTM, with global attention to developing the encoder and decoder. The AraBERT preprocessing stage has been used to enhance the model's understanding of Arabic words and get the best possible results. The skip-gram and a continuous bag of words (CBOW) word2vec word embedding models were also compared. The experimental results evaluated by ROUGE-1 have an F-score of 44.28, ROUGE-2 has an F-score of 18.35, ROUGE-L has an F-score of 32.46, and BLEU has an F-score of 0.41, showing that the BILSTM achieves the best performance and using the skip-gram word2Vec model outperforms models that use the CBOW word2Vec model.

Reference [43] Extractive text summarization utilizes a clustering technique with Latent Semantic Analysis (LSA) and Deep Restricted Boltzmann Machine (DRBM). After doing the manual evaluation, they found that DRBM performs better than all other algorithms. Finally, a word2vec model is being used for the comparison of the systems that have been created. After the comparison has been made, it has been found that the summarization of the technique employed for the system built using the clustering method based on latent semantic analysis outperforms the system created using the deep learning-based restricted Boltzmann machine (RBM).

Reference [40] They propose a multilingual text summarization approach that is based on LDA, linear discriminant analysis, and modified PageRank. They used the k-means clustering technique to choose the essential sentences based on similarity metrics from a document set written in seven languages: English, Arabic, Greek, French, Hindi, and Hebrew. A separate subject has been produced and prepared for each of these languages, for each of these documents. The approach in this suggested technique for MDS is based on LDA and modified page rank. The improved performance of this system is based on the concept of removing unneeded sentences and disregarding sentences that are not important, resulting in shorter sentences. As a result, this system may be considered an effective technique in the classification phase of the Arabic text summarization model.

Reference [38] They propose an abstractive text summarization approach that is based on the Seq2Seq model with an attention mechanism. They have used two different systems, The first system that has been created is the Arabic query-based text summarization system, which will use the standard retrieval method for mapping a query that has been mapped against that of a document collection and has been used for creating a summary of the text in the document. This proposed method shows that it is better than the other related works in the task summarization of the Arabic language. This automatic text summarization method is beneficial and produces an effective result.

Reference [76] They proposed an abstractive summarization system using a seq2seq model using multi and single encoder layers, LSTM, with global attention to developing the encoder and decoder. The resulting summary's quality is evaluated and qualitatively evaluated. In addition to ROUGE1, three additional evaluation metrics for the quality of the produced summary are proposed: ROUGE1-NO ORDER, ROUGE1STEM, and ROUGE1-CONTEXT. Experimentation has shown that a multi-layer encoder model provides the best results, with the suggested model having a ROUGE1 of 38.4, aROUGE1NOORDER of 46.2, a ROUGE1-STEM of 52.6, and a ROUGE1-CONTEXT of 58.1.

Reference [75]They have used extractive summarization methods to generate the summary by applying the modified Page Rank algorithm to enhance the performance and quality of summaries for single documents from the EASC corpus by using the Al-Khalil morphological analyzer. This is used to overcome the problems of Arabic structural complexity. The method has achieved values of 72.94, 68.75, and 67.99 for recall, precision, and measurement prospectively.

The sentences are evaluated using ''improved futures'' as a criterion and formalized The summary that was written The new method has an accuracy of 0.7 and a recall of 0.63, both of which are greater than the previous method. They compare and contrast the human-and system-generated summaries. ROUGE-1 is being used since it has a high recall significance test. Their technology has an accuracy of 85, according to the F-measure. They proposed a new solution for a single document in [53], as well as the previous method, which merely uses RBM. Both approaches' produced summaries are compared. The resulting summaries are evaluated using rough evaluation. Precision, recall, and the F measure are the performance evaluation metrics. To increase the accuracy of the summary, RBM is employed as an unsupervised learning method combined with fuzzy logic. It has been noted that the suggested method produces brief and precise summaries with no unnecessary content. On average, the experimental

TABLE 3. Comparison among different techniques for Arabic summarization.

Reference	Model	Document Type	Methodology	Evaluation Metrics	Dataset
[39]	Autoencoder	Single document	Using a graph-based approach and a A query-based approach.	ROUGE metric.	EASC
[34]	BERT Model	Single document with multilingual	Using pre-trained BERT and encoder BERTSUM.	ROUGE metric.	EASC and a KALIMAT
[28]	Word2Vec, Clustering and W- PC	Single document	They used unsupervised techniques and K-means clustering technique	Rouge -1 Rouge -2	EASC
[36]	Ara BERT Model and Clustering algorithm	Single document	Combining NLU (Ara BERT) and clustering algorithms.	ROUGE Manually by expertise	They have prepared the data set (multiple articles in different domains and it's summary)
[24]	Supervised machine learning and score-based algorithms	Single document	Using score-based and supervised machine learning techniques.	Rouge f-measure	EASC
[41]	Arabic Pretrained , and LSTM	Single document	AraVec is pre-trained for word embedding and encoder-decoder sequences.	Rouge 1	They have collected the dataset from various sources like Reuters, Aljazeera.
[35]	Bert for Multi Lingual	Single document	BERT for multilingual and the word2vec model.	ROUGE metric.	EASC
[77]	Seq2Seq model	Single document	using a seq2seq model, GRU, LSTM, and BILSTM, with global attention.	ROUGE-1, ROUGE-2, ROUGE-L, and BLEU	AHS and AMN
[43]	Restricted Boltzmann Machine (RBM)	Single document with multilingual	Latent Semantic Analysis (LSA) and Deep Restricted Boltzmann Machine (DRBM).	ROUGE metric.	They have manually prepared the data through scrapping and the EASC dataset.
[40]	Linear discriminant analysis (LDA)	Single document	Using linear discriminant analysis (LDA), modified PageRank, and the k-means clustering technique.	Rouge -1 Rouge -2	EASC
[38]	Seq 2Seq	Single document	Using a Seq2Seq model with an attention mechanism.	Rouge and f-measure	They have scrapped the dataset named it the Arabic headline summary dataset called (AHS).
[76]	Seq2Seq	Single document	Using a seq2seq model using multi and single encoder layers, LSTM, with global attention	Rouge and Suggested new method	They have scrapped the dataset from Aljazeera, containing 10932 news articles and SANAD_SUBSETdatase.
[75]	Modified, PageRank Algorithm	Single document	Modified Page Rank algorithm	Precision Recall F-measure	.EASC

outcome was 88 accuracies, 80 recall, and 84 F measures. This was obtained with the data set for a news article from Kaggle. They presented the Summ Coder in [54], a unique approach for extracting text from single documents using the Autoencoder architecture. The improved technique uses rouge metrics. In [15], for a single document. Using the word frequency feature, the AE tries to detect and learn the features and then ranks sentences using a cosine measure with subjects or critical phrases. Unlike other deep learning techniques, which may suffer from sparse input representation, this technique proposes solutions to reduce this problem via two techniques. First, developing local word representations (a bag-of-words (BOW) representation) consisting of input representations of each sentence in the document, and second, adding random noise to word representation weight An auto uses an ensemble method called "Ensemble Noisy." In an encoder (ENAE), the model is run numerous times on the same input, each time with a different quantity of random noise added. This led to other extractive summaries and then aggregated the rankings of these different experiments. After that, sentences that occur most frequently are obtained to form the final summary. They are using ROUGE for evaluation. For multi-document summarization, they suggested employing CNN and spreading phrases into dispersed representations [55] then using cosine similarity measurement to describe and model sentence redundancy. Then, using the diverse selection as an optimization problem, choose high-quality phrases by reducing the prestige and variety costs. For single document extractive summarization, they suggested using RNN based on gated recurrent unit

neural networks (GRU) in [50]. Each sentence is given a binary choice (based on the preceding decision) to determine whether it should be picked or not. Table 3 compares different Arabic summarization techniques, taking into consideration models and methodology.

VII. CONCLUSION

Currently, there are a growing number of Arabic documents available online. Arabic Natural Language Processing (ANLP) is the technique that is mainly used to perform the task. However, several challenges face the full achievement of this task, one of which is that research in this area has been scanty compared to the English language. The Arabic particulars (e.g., morphological richness and orthographic ambiguity because of the optional discretization) may lead to more homographs and, therefore, more ambiguity than English. Since there is still a huge quality gap between automatic and human-written summaries, we need good summarizers that consider all the semantically important information described in the source documents. This emphasizes the necessity of proposing abstractive-level summarization approaches across all types of text and domains. We also need effective evaluation metrics to assess the newly generated summaries. As we move toward abstract text summarization, we hope to make this task more adaptable to a wide range of users. various text sources and text types and improved evaluation using semantic representations. Recent techniques that apply deep learning for abstractive and extractive text summarization, datasets, and evaluation metrics for these approaches were reviewed in this paper. Moreover, the challenges encountered when employing various approaches and their solutions were discussed and analyzed. As a result, more work, experimenting, and research are required. A common problem during the process of summarizing was not having a ''golden standard'' to compare it to. In addition, the lack of standard systematic methodologies and architectures They were working on an automated Arabic text summarization model using deep learning to overcome the challenges of Arabic text summarization. the most common challenges faced during the summarization process were In addition, other challenges such as being out of vocabulary (OOV) words, repeating summary sentences, lack of standard systematic methodologies and architectures, the complexity of the Arabic language, the lack of golden tokens during testing, The lack of using rouge metrics in abstractive text summarization, the lack of a gold standard corpus, a high reduction rate, input document length, and the summarization process's stop criteria. In addition, there are several challenges to be taken into consideration while abstracting and extracting Arabic text summarization. containing the data set, evaluation metrics, and the generated summary's quality.

REFERENCES

[1] H. N. Fejer and N. Omar, ''Automatic Arabic text summarization using clustering and keyphrase extraction,'' in *Proc. 6th Int. Conf. Inf. Technol. Multimedia*, Nov. 2014, pp. 293–298.
[2] M. El-Haj, U. Kruschwitz, and C. Fox, ''Exploring clustering for multi-document Arabic summarisation,'' in *Proc. Asia Inf. Retr. Symp.* Springer, 2011, pp. 550–561.
[3] D. Miller, ''Leveraging BERT for extractive text summarization on lectures,'' 2019, *arXiv:1906.04165*.
[4] R. Witte, R. Krestel, and S. Bergler, ''Generating update summaries for DUC 2007,'' in *Proc. Document Understand. Conf.*, 2007, pp. 1–5.
[5] J. Pennington, R. Socher, and C. Manning, ''Glove: Global vectors for word representation,'' in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[6] M. Afsharizadeh, H. Ebrahimpour-Komleh, and A. Bagheri, ''Query-oriented text summarization using sentence extraction technique,'' in *Proc. 4th Int. Conf. Web Res. (ICWR)*, Apr. 2018, pp. 128–132.
[7] M. Mohamed and M. Oussalah, ''SRL-ESA-TextSum: A text summarization approach based on semantic role labeling and explicit semantic analysis,'' *Inf. Process. Manage.*, vol. 56, no. 4, pp. 1356–1372, Jul. 2019.
[8] A. B. Soliman, K. Eissa, and S. R. El-Beltagy, ''AraVec: A set of Arabic word embedding models for use in Arabic NLP,'' *Proc. Comput. Sci.*, vol. 117, pp. 256–265, Jan. 2017.
[9] N. Y. Habash, ''Introduction to Arabic natural language processing,'' *Synth. Lectures Hum. Lang. Technol.*, vol. 3, no. 1, pp. 1–187, Jan. 2010.
[10] R. Sennrich, B. Haddow, and A. Birch, ''Neural machine translation of rare words with subword units,'' 2015, *arXiv:1508.07909*.
[11] K. Shaalan, ''A survey of Arabic named entity recognition and classification,'' *Comput. Linguistics*, vol. 40, no. 2, pp. 469–510, Jun. 2014.
[12] J. Brownlee, ''A gentle introduction to text summarization,'' in *Machine Learning Mastery*, vol. 29. 2017.
[13] K. F. Ardhi, ''Sentiment analysis of smartphone accounting application users,'' *J. Appl. Accounting Taxation*, vol. 6, no. 2, pp. 161–174, Oct. 2021.
[14] R. Z. Al-Abdallah and A. T. Al-Taani, ''Arabic single-document text summarization using particle swarm optimization algorithm,'' *Proc. Comput. Sci.*, vol. 117, pp. 30–37, Jan. 2017.
[15] S. Wang, X. Zhao, B. Li, B. Ge, and D. Tang, ''Integrating extractive and abstractive models for long text summarization,'' in *Proc. IEEE Int. Congr. Big Data (BigData Congr.)*, Jun. 2017, pp. 305–312.
[16] F. Dernoncourt, M. Ghassemi, and W. Chang, ''A repository of corpora for summarization,'' in *Proc. 11th Int. Conf. Lang. Resour. Eval. (LREC)*, 2018, pp. 1–7.
[17] S. Banerjee and A. Lavie, ''METEOR: An automatic metric for MT evaluation with improved correlation with human judgments,'' in *Proc. ACL Workshop Intrinsic Extrinsic Eval. Measures Mach. Transl. Summarization*, 2005, pp. 65–72.
[18] A. A. Bialy, M. A. Gaheen, R. ElEraky, A. ElGamal, and A. A. Ewees, ''Single Arabic document summarization using natural language processing technique,'' in *Recent Advances in NLP: The Case of Arabic Language*. Springer, 2020, pp. 17–37.
[19] L. Li, C. Forăscu, M. El-Haj, and G. Giannakopoulos, ''Multi-document multilingual summarization corpus preparation, Part 1: Arabic, English, Greek, Chinese, Romanian,'' in *Proc. Multiling Workshop Multilingual Multi-Document Summarization*, 2013, pp. 1–12.
[20] K. Al-Sabahi, Z. Zhang, J. Long, and K. Alwesabi, ''An enhanced latent semantic analysis approach for Arabic document summarization,'' 2018, *arXiv:1807.11618*.
[21] L. M. Al Qassem, D. Wang, Z. Al Mahmoud, H. Barada, A. Al-Rubaie, and N. I. Almoosa, ''Automatic Arabic summarization: A survey of methodologies and systems,'' *Proc. Comput. Sci.*, vol. 117, pp. 10–18, Jan. 2017.
[22] 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.
[23] R. M. Duwairi, R. Marji, N. Sha'ban, and S. Rushaidat, ''Sentiment analysis in Arabic tweets,'' in *Proc. 5th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2014, pp. 1–6.
[24] A. Qaroush, I. Abu 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.*, vol. 33, no. 6, pp. 677–692, Jul. 2021.
[25] A. Khan, N. Salim, and H. Farman, ''Clustered genetic semantic graph approach for multi-document abstractive summarization,'' in *Proc. Int. Conf. Intell. Syst. Eng. (ICISE)*, Jan. 2016, pp. 63–70.
[26] F. Alotaiby, ''New approaches to automatic headline generation for Arabic documents,'' *J. Eng. Comput. Innov.*, vol. 3, no. 1, pp. 11–25, Feb. 2012.
[27] P. Verma and A. Verma, ''A review on text summarization techniques,'' *J. Sci. Res.*, vol. 64, no. 1, pp. 251–257, 2020.

[28] S. Abdulateef, N. A. Khan, B. Chen, and X. Shang, "Multidocument Arabic text summarization based on clustering and Word2 Vec to reduce redundancy," *Information*, vol. 11, no. 2, p. 59, Jan. 2020.

[29] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.

[30] M. M. Boudabous, M. H. Maaloul, and L. H. Belguith, "Digital learning for summarizing Arabic documents," in *Proc. Int. Conf. Natural Lang. Process.* Springer, 2010, pp. 79–84.

[31] A. Elnagar, R. Al-Debsi, and O. Einea, "Arabic text classification using deep learning models," *Inf. Process. Manage.*, vol. 57, no. 1, Jan. 2020, Art. no. 102121.

[32] N. S. Ranjitha and J. S. Kallimani, "Abstractive multi-document summarization," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 1690–1694.

[33] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[34] K. N. Elmadani, M. Elgezouli, and A. Showk, "BERT fine-tuning for Arabic text summarization," 2020, *arXiv:2004.14135*.

[35] B. Elayeb, A. Chouigui, M. Bounhas, and O. B. Khiroun, "Automatic Arabic text summarization using analogical proportions," *Cognit. Comput.*, vol. 12, no. 5, pp. 1043–1069, Sep. 2020.

[36] A. M. A. Nada, E. Alajrami, A. A. Al-Saqqa, and S. S. Abu-Naser, "Arabic text summarization using arabert model using extractive text summarization approach," *Int. J. Academic Inf. Syst. Res.*, vol. 4, no. 8, pp. 6–9, Aug. 2020.

[37] W. Yin and Y. Pei, "Optimizing sentence modeling and selection for document summarization," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015.

[38] M. Al-Maleh and S. Desouki, "Arabic text summarization using deep learning approach," *J. Big Data*, vol. 7, no. 1, Dec. 2020.

[39] N. Alami, M. Meknassi, N. En-nahnahi, Y. El Adlouni, and O. Ammor, "Unsupervised neural networks for automatic Arabic text summarization using document clustering and topic modeling," *Expert Syst. Appl.*, vol. 172, Jun. 2021, Art. no. 114652.

[40] Z. H. Ali and A. P. D. S. Malallah, "Multilingual text summarization based on LDA and modified pagerank," *J. Inf. Technol. V*, vol. 9, no. 3, p. 2018, 2019.

[41] S. Encoder, "Deep learning based abstractive Arabic text summarization using two layers encoder and one layer decoder," *J. Theor. Appl. Inf. Technol.*, vol. 98, no. 16, pp. 3233–3244, 2020.

[42] D. R. Radev, E. Hovy, and K. McKeown, "Introduction to the special issue on summarization," *Comput. Linguistics*, vol. 28, no. 4, pp. 399–408, 2002.

[43] L. Al Qassem, D. Wang, H. Barada, A. Al-Rubaie, and N. Almoosa, "Automatic Arabic text summarization based on fuzzy logic," in *Proc. 3rd Int. Conf. Natural Lang. Speech Process.*, 2019, pp. 42–48.

[44] C. Sun, L. Lv, G. Tian, Q. Wang, X. Zhang, and L. Guo, "Leverage label and word embedding for semantic sparse web service discovery," *Math. Problems Eng.*, vol. 2020, Mar. 2020, Art. no. 5670215.

[45] M. Yousefi-Azar and L. Hamey, "Text summarization using unsupervised deep learning," *Expert Syst. Appl.*, vol. 68, pp. 93–105, Feb. 2017.

[46] S. Gupta and S. Gupta, "Abstractive summarization: An overview of the state of the art," *Expert Syst. Appl.*, vol. 121, pp. 49–65, 2019.

[47] H. T. Le and T. M. Le, "An approach to abstractive text summarization," in *Proc. Int. Conf. Soft Comput. Pattern Recognit. (SoCPaR)*, Dec. 2013, pp. 371–376.

[48] P.-E. Genest and G. Lapalme, "Fully abstractive approach to guided summarization," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2012, pp. 354–358.

[49] N. Moratanch and S. Chitrakala, "A survey on abstractive text summarization," in *Proc. Int. Conf. Circuit, Power Comput. Technol. (ICCPCT)*, Mar. 2016, pp. 1–7.

[50] T. Vodolazova and E. Lloret, "The impact of rule-based text generation on the quality of abstractive summaries," in *Proc. Int. Conf. Recent Adv. Natural Lang. Process. (RANLP)*, 2019, pp. 1275–1284.

[51] A. Khan, N. Salim, and Y. Jaya Kumar, "A framework for multi-document abstractive summarization based on semantic role labelling," *Appl. Soft Comput.*, vol. 30, pp. 737–747, May 2015.

[52] L. Hou, P. Hu, and C. Bei, "Abstractive document summarization via neural model with joint attention," in *Proc. Nat. CCF Conf. Natural Lang. Process. Chin. Comput.* Springer, 2017, pp. 329–338.

[53] T. Cai, M. Shen, H. Peng, L. Jiang, and Q. Dai, "Improving transformer with sequential context representations for abstractive text summarization," in *Proc. CCF Int. Conf. Natural Lang. Process. Chin. Comput.* Springer, 2019, pp. 512–524.

[54] A. Mahajani, V. Pandya, I. Maria, and D. Sharma, "A comprehensive survey on extractive and abstractive techniques for text summarization," in *Ambient Communications and Computer Systems*. 2019, pp. 339–351.

[55] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017.

[56] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," 2016, *arXiv:1603.07252*.

[57] P. Kouris, G. Alexandridis, and A. Stafylopatis, "Abstractive text summarization based on deep learning and semantic content generalization," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5082–5092.

[58] S. Verma and V. Nidhi, "Extractive summarization using deep learning," 2017, *arXiv:1708.04439*.

[59] H. P. Luhn, "The automatic creation of literature abstracts," *IBM J. Res. Develop.*, vol. 2, no. 2, pp. 159–165, Apr. 1958.

[60] M. El-Haj, U. Kruschwitz, and C. Fox, "Creating language resources for under-resourced languages: Methodologies, and experiments with Arabic," *Lang. Resour. Eval.*, vol. 49, no. 3, pp. 549–580, Sep. 2015.

[61] A. M. Azmi and R. S. Almajed, "A survey of automatic Arabic diacritization techniques," *Natural Lang. Eng.*, vol. 21, no. 3, pp. 477–495, May 2015.

[62] J. Bradbury, S. Merity, C. Xiong, and R. Socher, "Quasi-recurrent neural networks," 2016, *arXiv:1611.01576*.

[63] A. Bawakid, "Automatic documents summarization using ontology based methodologies," Ph.D. dissertation, Univ. Birmingham, Birmingham, U.K., 2011.

[64] D. Harman and P. Over, "The effects of human variation in DUC summarization evaluation," in *Text Summarization Branches Out*. 2004, pp. 10–17.

[65] M. El-Haj and R. Koulali, "KALIMAT a multipurpose Arabic corpus," in *Proc. 2nd Workshop Arabic Corpus Linguistics (WACL)*, 2013, pp. 22–25.

[66] C. Napoles, M. R. Gormley, and B. Van Durme, "Annotated gigaword," in *Proc. Joint Workshop Autom. Knowl. Base Construct. Web-Scale Knowl. Extraction (AKBC-WEKEX)*, 2012, pp. 95–100.

[67] M. K. Saad and W. M. Ashour, "OSAC: Open source Arabic corpora," in *Proc. 6th ArchEng Int. Symp. (EEECS)*, vol. 10, 2010.

[68] O. Einea, A. Elnagar, and R. Al Debsi, "SANAD: Single-label Arabic news articles dataset for automatic text categorization," *Data Brief*, vol. 25, Aug. 2019, Art. no. 104076.

[69] B. Al-Salemi, M. Ayob, G. Kendall, and S. A. M. Noah, "Multi-label Arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms," *Inf. Process. Manage.*, vol. 56, no. 1, pp. 212–227, Jan. 2019.

[70] N. Alalyani and S. Larabi, "NADA: New Arabic dataset for text classification," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 9, pp. 1–7, 2018.

[71] T. Hasan, A. Bhattacharjee, M. S. Islam, K. Mubasshir, Y.-F. Li, Y.-B. Kang, M. S. Rahman, and R. Shahriyar, "XL-sum: Large-scale multilingual abstractive summarization for 44 languages," in *Proc. Findings Assoc. Comput. Linguistics (ACL-IJCNLP)*, 2021, pp. 4693–4703.

[72] C. C. Faisal Ladhak, E. Durmus, and K. McKeown, "Wikilingua: A new benchmark dataset for multilingual abstractive summarization," in *Proc. Findings EMNLP*, 2020.

[73] A. M. Zaki, M. I. Khalil, and H. M. Abbas, "Deep architectures for abstractive text summarization in multiple languages," in *Proc. 14th Int. Conf. Comput. Eng. Syst. (ICCES)*, Dec. 2019, pp. 22–27.

[74] H.-H. Huang, Y.-H. Kuo, and H.-C. Yang, "Fuzzy-rough set aided sentence extraction summarization," in *Proc. 1st Int. Conf. Innov. Comput., Inf. Control (ICICIC)*, vol. 1, Aug./Sep. 2006, pp. 450–453.

[75] 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.

[76] D. Suleiman and A. Awajan, "Multilayer encoder and single-layer decoder for abstractive Arabic text summarization," *Knowl.-Based Syst.*, vol. 237, Feb. 2022, Art. no. 107791.

[77] Y. M. Wazery, M. E. Saleh, A. Alharbi, and A. A. Ali, "Abstractive Arabic text summarization based on deep learning," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–14, Jan. 2022.

[78] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. 2004, pp. 74–81.

[79] C. Callison-Burch, M. Osborne, and P. Koehn, "Re-evaluating the role of BLEU in machine translation research," in *Proc. 11th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2006, pp. 1–8.
[80] R. Wason, "Deep learning: Evolution and expansion," *Cognit. Syst. Res.*, vol. 52, pp. 701–708, Dec. 2018.
[81] S. N. Turky, A. S. A. Al-Jumaili, and R. K. Hasoun, "Deep learning based on different methods for text summary: A survey," *J. Al-Qadisiyah Comput. Sci. Math.*, vol. 13, no. 1, p. 26, 2021.
[82] S. Rauniyar, "A survey on deep learning based various methods analysis of text summarization," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, Feb. 2020, pp. 113–116.
[83] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017.
[84] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[85] C. Kumar, P. Pingali, and V. Varma, "Generating personalized summaries using publicly available web documents," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, Dec. 2008, pp. 103–106.
[86] W. Xu, C. Napoles, E. Pavlick, Q. Chen, and C. Callison-Burch, "Optimizing statistical machine translation for text simplification," *Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 401–415, Dec. 2016.
[87] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.
[88] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, 2019, vol. 2, no. 4, p. 5.
[89] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2016, pp. 93–98.
[90] Y. HaCohen-Kerner, D. Miller, and Y. Yigal, "The influence of preprocessing on text classification using a bag-of-words representation," *PLoS ONE*, vol. 15, no. 5, May 2020, Art. no. e0232525.
[91] M. S. Binwahlan, N. Salim, and L. Suanmali, "Swarm based text summarization," in *Proc. Int. Assoc. Comput. Sci. Inf. Technol. (Spring Conf.)*, Apr. 2009, pp. 145–150.

ASMAA ELSAID received the M.Sc. degree in computer science from the Arab Academy for Science, Technology, and Maritime Transport, Cairo, Egypt, in 2012. She is currently pursuing the Ph.D. degree in computer science with the Faculty of Graduate Studies for Statistical Research, Cairo University. She is currently a Teaching Assistant with the Higher Institute of Computer Science and Information Technology, Elshrouk Academy, Egypt. Her research interests include deep learning, NLp, advanced database management, knowledge-based systems, big data, and data science.

AMMAR MOHAMMED received the bachelor's and master's degrees in computer science from Cairo University, Egypt, and the Ph.D. degree in computer science from the University of Koblenz-Landau, Germany, in 2010. He worked as a Researcher and a Research Fellow with the Artificial Intelligence (AI) Research Group, University of Koblenz-Landau. He is currently an Associate Professor of computer science with Cairo University and Misr International University, Egypt. He supervised a group of Ph.D. and master's students and established the Machine/Deep Learning Research Group, Department of Computer Science, Faculty of Graduate Studies for Statistical Research, Cairo University. His research interests include machine and deep learning techniques, methods, algorithms, and their applications in several domains.

LAMIAA FATTOUH IBRAHIM received the B.Sc. degree from the Computer and Automatic Control Department, Faculty of Engineering, Ain Shams University, in 1984, the master's degree from the Ecole National Superieur de Telecommunication, ENST Paris, in 1987, the master's degree from the Computer and Systems Engineering Department, Faculty of Engineering, Ain Shams University, in 1993, and the Ph.D. degree from the Faculty of Engineering, Cairo University, in 1999. Previously, she was with the Information Technology Department, Faculty of Computing and Information Technology, King Abdulaziz University. She is currently a Professor of artificial intelligence and the Vice Dean for education and student affairs with the Faculty of Information Systems and Computer Science, October 6 University. Previously, she was the Head of the Department of Computer Science, Faculty of Graduate Studies for Statistical Research, Cairo University. She has over 33 years of experience in the fields of network design engineering and artificial intelligence, focusing on applying knowledge base and data mining techniques to wired and wireless network planning. She has published papers in many international journals and international conferences in the areas of networks, data mining, and wired and mobile network planning.

MOHAMMED M. SAKRE received the bachelor's and master's degrees in computer engineering from the Military Technical College, Egypt, in 1979 and 1987, respectively, and the Ph.D. degree in computer science from the University of Cranfield, U.K., in 1992. He worked as a Researcher and a Lecturer in the fields of computer science and information technology at the Egyptian Army, from 1992 to 2003. He is currently a Professor of computer science with Al-Shorouk Academy, Cairo, Egypt. His research interests include natural language processing of Arabic language in different applications, including machine translation, text summarization, intelligent search engine, question answering systems, and many other areas.