Informatics Bulletin, Faculty of Computers and Artificial Intelligence, Helwan University
Published Online *Vol 6 Issue 2, July 2024*
(https://fcihib.journals.ekb.eg)

Exploring Advances in Arabic Long-Text Summarization Strategies: A Survey

Mostafa Magdy[1], Salwa O. Slim[1], Ensaf Hussein Mohamed[1,2]
[1]Computer science department, Faculty of Computers and Artificial Intelligence, Helwan university, Cairo, Egypt
[2]School of Information Technology and Computer Science (ITCS), Nile University, Giza, Egypt
mostafa_magdy@fci.helwan.edu.eg, salwaosama@fci.helwan.edu.eg ,enmohamed@nu.edu.eg

Abstract— **The number of documents, textbooks, and articles is growing exponentially. Thus, the text summarization process aids in recalling the preceding part of a novel before reading the subsequent section. It also facilitates time-saving by allowing readers to peruse summarized versions of lengthy articles or books. This survey aims to present recently published studies on Arabic long-text summarization. Text summarization poses a significant challenge within the domain of Natural Language Processing (NLP). Constructing an effective summary requires accurate text analysis, encompassing complex tasks such as semantic and lexical analysis. Moreover, a quality summary should encapsulate vital details while maintaining conciseness, and it must also consider factors like non-redundancy, relevance, coverage, coherence, and readability. In academic research, various approaches to text summarization are employed, including extractive summarization, abstractive summarization, and hybrid methods. Extractive summarization has reached a level of maturity, leading to a shift in research emphasis towards abstractive summarization and the development of real-time summarization techniques. According to this survey, we found that the abstractive approach is recently used but has many limitations, such as summarizing long text and allowing the user to determine the compression ratio for summarizing the original text. Therefore, the hybrid approach is recommended.**

Index Terms— **NLP, Long-text, text summarization, Abstractive Summary, Extractive Summary, hybrid summarization**

I. INTRODUCTION

THE number of documents and books on the internet is growing exponentially every day. Because of the large volume of documents, it is difficult to get the required information needed from them. Summarizing the text is, therefore, necessary to easily extract the most important information from a large text [1]. A manual text summarization process is an effective way to get a summary with preserving meaning; however, this is a time-consuming activity.

Automatic text summarization (ATS) is one of the most important applications in the field of Natural Language Processing (NLP). Its primary objective is to distill the most vital meanings from a document within a limited space [2]. It aims to derive the most important meanings in a limited size [3].

ATS can be classified based on two viewpoints, as illustrated in Figure 1. The first perspective revolves around the approach/output types, which encompass extractive, abstractive, and hybrid summarization. In the extractive approach, the summary comprises the most crucial sentences extracted from the documents. These significant sentences are then combined to form the summary, with each sentence originating from the original document. On the other hand, in abstractive summarization, the summary generates new sentences and phrases that diverge from the document's original text.

The abstractive approach requires a semantic analysis of the text. Therefore, it is more complicated than extractive summarization [4]. In the hybrid approach, both extractive and abstractive methods are combined to create a hybrid text summarizer [5].

The second perspective is based on the length of the text or type of documents, such as single-document, multi-document, and long-document summarization. In single document summarization, researchers focus on one short document which considers a short text [6]. Multi-document summarization, on the other hand, entails the summarization of multiple documents as input to create a cohesive summary that encompasses all the input documents [2]. Long-document summarization deals with the division of lengthy texts into smaller, more manageable segments.

In our paper long text could be based on factors such as the number of chapters, the total word count across all chapters, or the overall reading time required to cover the entire book.

Due to the limited availability of research papers focused on Arabic long text summarization, our approach has involved exploring related areas. Specifically, we have extensively studied papers on English long text summarization. Initially, this might seem off-track, but it's crucial to realize that English summarization techniques can be modified for Arabic with the right adjustments.

Furthermore, recognizing that long texts may ultimately need to be distilled into shorter summaries for various purposes, such as enhancing readability or accommodating length constraints.

We have also dedicated a significant portion of our efforts to Arabic short text summarization. This strategic choice allows us to not only excel in summarizing standalone short texts but also provides us with valuable insights and methodologies that can be easily integrated into the Arabic long text summarization process. In essence, Arabic short text summarization techniques serve as building blocks to enhance our proficiency in handling lengthy Arabic documents effectively.

In our study, we are particularly concerned with long-document summarization, where we consider book summarization as a prime example. The model takes the entire book as input and attempts to summarize each chapter individually, then combines the summaries of the chapters to create a comprehensive summary of the book [7]. Long document summarization holds a significant position within the field of Arabic language processing. Arabic is known for being a bit complex in its language, with lots of words, tricky grammar, and different ways of putting sentences together. As a result, longer Arabic texts may be challenging to understand, even for skilled readers.

Summarization becomes a valuable tool, simplifying complex information into a more understandable format, and making it more accessible and comprehensible for a wider audience. Summarization benefits the media sector by aiding journalists in simplifying complex news stories into concise, informative briefs. Additionally, legal experts use summarization to expedite the analysis of extensive legal documents, enhancing both efficiency and compliance [39].

The primary objective of this study is to analyze, compare, and evaluate different techniques employed in Arabic long document summarization. Additionally, our study aims to assess various datasets utilized for model training, examine the metrics used to measure performance, and evaluate the achieved performance levels.

The structure of the paper is outlined as follows: Section 2 focuses on the specifics and challenges of the Arabic language. Section 3 presents applications and tools of Automatic Text Summarization (ATS). Section 4 outlines ATS approaches and related works. The discussion is elaborated upon in Section 5, while Section 6 concludes the paper.

II. ARABIC LANGUAGE SPECIFICS AND CHALLENGES

The unique characteristics of the Arabic language that impact summarization encompass its intricate morphology, flexible syntax, and layered semantics. These aspects introduce challenges and considerations when generating effective summaries [13]:

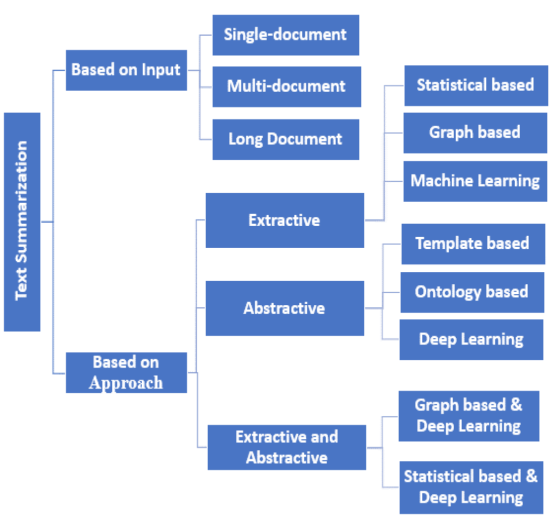

Figure 1. Categorization of Document Summarization.

Morphology stands as a defining feature of Arabic. It is characterized by a rich system of affixes. These affixes convey intricate grammatical and semantic differences. The transformation of words through various inflections can lead to the loss of essential details. For example, consider the Arabic root "كتب", which means "to write." Through morphology, this root can transform into words such as "كتاب" meaning "book," "كاتب" meaning "writer," or "كُتِب" meaning "was written." Each form carries distinct semantic connotations .This happens when trying to summarize the long text. Therefore, careful consideration is required during the summarization process [40].

Syntax: Arabic's sentence structure allows for flexibility due to case endings and conjugations indicating grammatical relations. This syntactic variability poses a challenge in identifying and summarizing the main ideas from convoluted sentence constructions [41]. For example, the sentence " ذهب الولدُ إلى المدرسة" and "الولدُ ذهب إلى المدرسة" both mean "The boy went to school," but with different syntax. Ensuring that the summarized content retains logical flow and core meaning necessitates careful dissection of intricate syntax.

Semantic challenges in text summarization for Arabic involve navigating polysemy and addressing ambiguity stemming from complex sentence structures [41]. For example, the Arabic word "قطع" can mean "to cut," "to interrupt," or "to end," depending on the context. Similarly. These intricacies demand the application of advanced natural language processing techniques. The goal is to ensure that summaries proficiently capture the essential meaning.

Lexical Richness the Arabic language is known for having lots of words that mean similar things, as well as words that sound the same but mean different things. This makes it tricky to summarize text because some words can have more than one meaning. For example, the word "سَكين" can mean both "knife" and "tranquility," depending on how it's used. So, when summarizing, it's important to pick the right words carefully.

III. APPLICATIONS AND TOOLS FOR AUTOMATIC TEXT SUMMARIZATION (ATS)

Recently, ATS has found applications in various domains, including question-answering, information extraction, and information retrieval.

A. We highlight some of the applications that utilize ATS, such as summarizing books, tweets, news articles, emails, and financial reports:

1) Books Summarization

ATS primarily focuses on summarizing lengthy texts, particularly books. Traditional methods often struggle to provide book summarization [7]. Summarizing books can help readers quickly grasp the content, aiding them in making informed decisions before purchasing.

2) Tweet Summarization

In the present day, millions of tweets and posts are shared across platforms like Twitter and other social networks. ATS has become a valuable tool for extracting useful information from this abundance [8].

3) News Summarization

People tend to read multiple news articles on a specific topic because a single news article may not contain all important information. Therefore, ATS helps to get a summary of all articles related to specific news which will save time and energy to get information [9][10].

4) *Email Summarization*

ATS assists in automatically summarizing email threads by presenting a paragraph comprising the most crucial sentences. This aids the user in grasping the main point of the email [11].

5) *Research Paper Exploration*

Arabic researchers and academics can benefit from summarization by quickly assessing the main points and contributions of lengthy research papers. This accelerates the process of identifying relevant studies for their research [56].

6) *Financial Reports Summarization*

Many companies produce a variety of reports containing numerical information during the financial year. The readers of financial reports use those reports and their information for making decisions regarding the allocation of resources. Therefore, there is a need for ATS to reduce the time and efforts of investors in decision making [12].

B. In addition, we'll spotlight some APIs that make text summarization easier:

1) *TextRazor API*

TextRazor provides an Arabic summarization API that developers and businesses can integrate into their applications [52]. It leverages machine learning to generate summaries while considering semantic context and entity relationships within the text.

2) *Lakhasly API*

Lakhasly is a machine-based Summarizer, which can summarize long well-structured Arabic or English documents [53]. Lakhasly uses different algorithms to summarize text based on the input language.

3) *MeaningCloud API*

MeaningCloud offers a comprehensive text analysis API that includes summarization capabilities for Arabic and other languages [54]. It provides both extractive and abstractive summarization methods.

IV. RELATED WORK AND APPROACHES TO AUTOMATIC TEXT SUMMARIZATION

Text summarization approaches can be categorized based on their output, namely extractive, abstractive, and hybrid summarization. In the following sections, we will discuss each approach and explore previous studies within each category.

A. Extractive summarization

1) Extractive summarization Approaches

This technique attempts to assess and assign scores to the importance of sentences and words in the original text. It selects several top-scoring sentences, determined by the length of the original text while maintaining the sequence of selected sentences from the original text. All summary sentences are retained in the original text without any structural modifications [14]. Some extractive summarization techniques are:

a) Statistical Based

It is interesting to examine statistical features of sentences that aid in extracting important sentences from documents. These features include the position and similarity of a sentence with other sentences [38].

This approach offers the advantage of rapidity in generating summaries, rendering it suitable for real-time applications like news aggregation and chatbots. However, there are also disadvantages, particularly in the realm of abstraction. Statistical models, primarily reliant on statistical patterns, encounter challenges when tasked with generating abstract or conceptual summaries [51].

b) Graph-Based

Represent the document as a connected graph. Each sentence is represented as a vertex, and the similarity between them is depicted as edges. The edge between two vertices signifies the similarity between them [7].

This approach offers the advantage of enhanced abstraction, enabling graph-based summarization to effectively handle abstract concepts and pivotal themes. Moreover, it allows the extension of graph-based techniques to summarize multiple documents by creating a unified graph that integrates data from various sources. However, there are also disadvantages, particularly in its scalability, as constructing and processing intricate graphs can become unmanageable when dealing with exceedingly large documents or datasets [43].

Figure 2. The summarization system that relies on a graph follows a clear structure with five key steps, as outlined below [47]. First, there's the Data Preprocessing stage, where the data is prepared for analysis. Next is the Text Graph-based Representation, which creates a visual representation of relationships between different elements, forming smaller graphs to represent specific aspects of the text. Then comes Sentence ranking, where the importance of each sentence is determined. Finally, in the Summary generation stage, the system produces a concise summary based on the processed information.

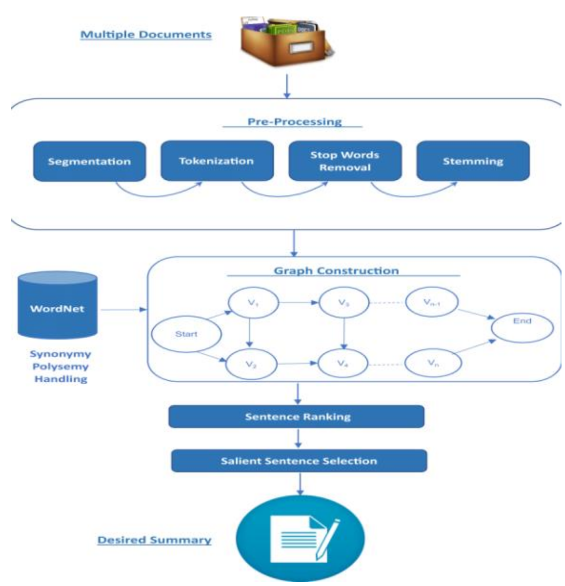

Figure 2. Overview of Triangle-Graph-Based Summarization [55]

A triangle graph is a specific type of graph where connections between nodes (sentences) are considered in the context of triangular structures (sets of three nodes).

This helps to capture the interrelationships more robustly than simple pairwise connections.

c) Machine Learning

This categorization can be divided into two main groups: supervised and unsupervised extractive techniques. In supervised methods, such as SummaRunner as outlined in [49], the training process requires a dataset containing texts alongside human-generated summaries. In contrast, unsupervised methods, like LexRank introduced in [50] and the Clustering technique introduced in [2], generate summaries without relying on any training data.

This approach offers advantages in customization. It allows for fine-tuning and tailoring to specific domains or user preferences, providing substantial flexibility in the summarization process. However, there are also disadvantages, particularly in its capacity to handle abstraction. Some machine learning models may encounter challenges when tasked with generating abstract or conceptual summaries [32].

Figure 3. Illustrates the principal eight steps involved in the unsupervised K-means clustering technique [2]:

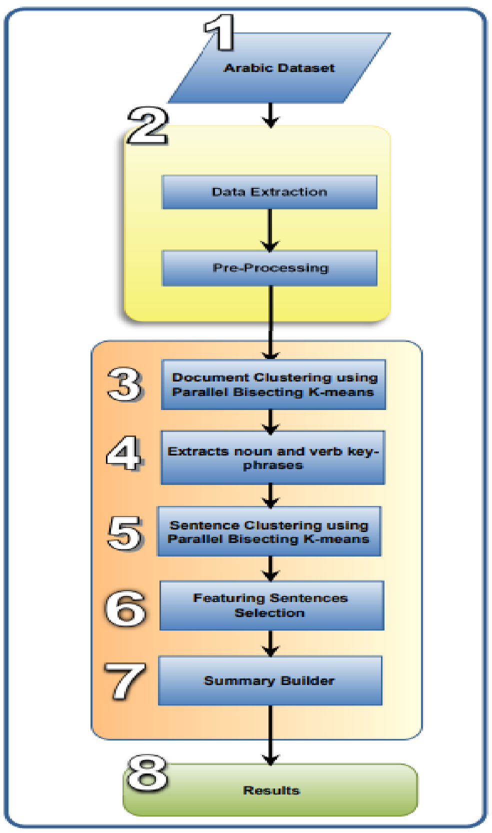

Figure 3. The flowchart of K-means Clustering Technique [2]

2) Extractive Summarization Related Work:

Khulood et al. [2] proposed a novel model for Arabic multi-document text summarization based on K-clustering. The model comprises two clustering stages. In the first stage, document clustering is employed to group each topic, followed by ranking each document based on its size and the scores of the encompassing documents. The second stage involves creating a list of extracted sentences and using a build checker to re-rank them by testing the content of the sentences.

Rada et al. [7] introduced a novel model for textbook summarization. They employed a graph-based segmentation algorithm using normalized cuts to partition the book into segments. Each segment encompasses a group of sentences. Subsequently, they generated an individual summary for each segment. They incorporated a segment ranking approach that assigns scores to reflect the significance of each segment.

Elbarougy et al. [14] introduced a novel model employing a modified PageRank algorithm. The model encompasses three key stages. Initially, text is extracted from a document, followed by preprocessing tasks such as tokenization, normalization, and the removal of stop words. In the second stage, desired features are extracted from sentences, and the document is modeled as a graph. The third stage involves utilizing the modified PageRank algorithm to rank each sentence and generate a summary based on their ranks. When applying the PageRank algorithm, each sentence is assigned an initial value equal to its noun count. These initial values contribute to achieving optimal performance.

Alqaisi et al. [15] presented a multi-document text summarization model based on clustering. The model aims to extract the most significant sentences encompassing the primary topic of the original text. Each sentence is standardized and then expressed through a bag-of-words (BOW) representation. A collection of features is extracted from each sentence, and subsequently clustering is employed to identify topics present in the original text.

Bowen et al. [16] introduced two novel models for extractive and abstractive text summarization. In their extractive summarization approach, they employed k-nearest neighbor to extract sentences that capture comprehensive semantic meanings from abstracts. For abstractive summarization, their model utilized a pre-trained distilled version of GPT-2, which was built with 12 attention heads and 6 transformer decoder layers.

Ladhak et al. [17] proposed a new approach for alignment chapter sentences. They employed a stable matching algorithm to select the most optimal alignments. Each chapter is summarized independently and assessed separately, without merging individual summaries at the conclusion. Chapters containing over 700 sentences were ignored. The experiments demonstrated that sentence-level stable-matched alignment outperformed the summary-level alignment utilized in the Mihalcea study [7]. The model was trained using three extractive systems: CNN-LSTM extractor (CB), seq2seq with attention (K), and RNN (N).

Samer et al. [18] introduced an innovative model for Arabic multi-document text summarization that combines clustering and Word2Vec. This unsupervised, score-based method consists of six key stages: data collection, text preprocessing, discriminative document selection, sentence tokenization, sentence weight mapping, and the selection of important sentences for summary based on optimal weight. The clustering technique is divided into two sections. The first involves classifying documents into distinctive and non-distinctive sets.

In the second part, document sentences are tokenized and clustered within distinctive and non-distinctive sets based on cosine similarity.

Zuhair et al. [19] introduced a fresh model for Arabic text summarization utilizing Linear Discriminant Analysis (LDA) and a modified PageRank approach. They employed the LDA classifier to categorize document sentences into important and non-important ones. The selected important sentences are then processed by constructing a graph and employing a modified PageRank algorithm to assign weights to each sentence. Ultimately, a summary is generated based on the assigned sentence weights.

Alaidine et al. [20] introduced an innovative model for single-document summarization utilizing the knapsack balancing algorithm. The initial stage involves dividing the document into segmentations, where each segment is associated with significant topics and an effective retention score. The subsequent step employs the knapsack algorithm to optimize segmentations. Ultimately, the summary is generated by maximizing effective retention.

Raed et al. [21] introduced a novel model based on the Firefly algorithm. The model comprises several steps: preprocessing, computation of sentence similarity scores, construction of a graph for candidate solutions, and utilization of the Firefly algorithm to select significant sentences for the summary. The proposed Firefly algorithm plays a crucial role in extracting the optimal path from candidate paths in the DAG graph, with each path representing a summary.

Merniz et al. [22] introduced a fresh model for Arabic multi-document text summarization, which operates across three distinct phases. The initial phase involves obtaining thematic annotations for the documents. Documents are segmented, with each segment corresponding to significant topics. The second phase focuses on graph representation, aiming to avoid redundancy while encompassing all segments associated with significant topics. The final step entails graph reduction, where a modified PageRank algorithm is employed on the constructed graph to select only crucial sentences for the ultimate summary.

B. *Abstractive Summarization*

1) *Abstractive Summarization Approaches*

This technique endeavors to simulate human behavior. It analyzes the text, aiming to grasp its meaning in a shorter form. Its goal is to produce a grammatically accurate and meaningful summary by rephrasing the original text [23]. Unlike extractive summarization, the generated summary sentences may differ from those found in the original text. Some abstractive summarization techniques are:

a) *Template Based*

This approach enables the end-user to formulate a template outlining what should be included in the summary. The end-user can supply multiple templates as required for generating summaries [28].

This approach offers the advantage of providing precise control over the summary format and ensuring essential information is consistently included. However, there are also disadvantages, particularly in its dependency on predefined knowledge, which may not always align seamlessly with the dynamic nature of textual data, potentially limiting its adaptability and responsiveness to different contexts.

b) *Ontology-Based*

This approach is employed to capture the semantic connections among texts and extract crucial sentences. It includes representing sentences as vertices and representing the semantic relationships between them as edges [28].

This approach offers the advantage of leading to better content extraction by utilizing ontological relationships between concepts, resulting in more precise and pertinent summaries. However, there are also disadvantages, particularly in that creating and maintaining ontologies can be demanding and adapting them to diverse topics can be challenging. Additionally, there may be gaps between ontological representations and natural language nuances, which can limit the system's understanding.

c) *Deep Learning / Encoder-Decoder Network*

This approach involves both an Encoder and Decoder. The encoder's role is to process the input and produce a final state vector, while the decoder is tasked with generating the final summary based on the final state vector [4].

This approach offers the advantage of understanding the context and how words relate to each other. This ensures that the summaries it creates make more sense and fit well with the context [25]. However, there are also disadvantages, particularly in that it requires a significant amount of data and computing power to perform well, especially when employing large structures like Transformers. This can be a problem when you don't have a lot of data or a powerful computer, and it might not do as well in those situations [42].

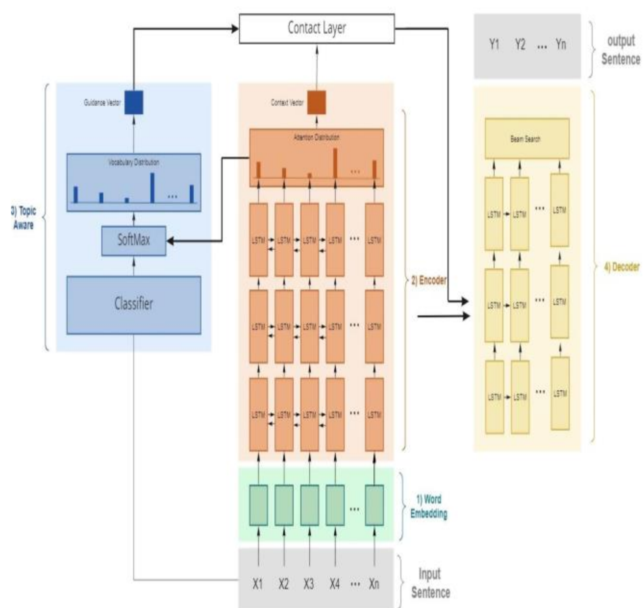

Figure 4. The Architecture of Encoder-Decoder Summarizer [48]

Figure 4. Illustrates the encoder (left side) processes input data and extracts meaningful representations, while the decoder (right side) generates output sequences based on these encoded representations.

2) Abstractive Summarization: Related Work

Azmi et al. [3] proposed a user-controlled summarizing ratio. The model started first by dividing the content of the document into multiple paragraphs called segments where each segment contains a bulk of sentences. Then, it generates a title for each segment. Finally, generates an abstractive summary using the sentence reduction technique. The model allows users to choose a compression ratio for a summary from the original text.

Wazery et al. [4] introduced a model based on a sequence-to-sequence architecture. The model was composed of an encoder and decoder, constructed using different layers of Gated Recurrent Units, Long Short-Term Memory, and Bidirectional Long Short-Term Memory. The implementation was carried out using the Keras library and executed on Google Colab. The experimental results indicated that the best performance was achieved with three layers of Bidirectional Long Short-Term Memory. Additionally, the skip-gram Word2Vec model outperformed the CBOW Word2Vec model.

Dima et al. [23] introduced a model based on a sequence-to-sequence recurrent neural network (RNN) architecture. The model was built with a multi-layer encoder and a single-layer decoder. The encoder layer employed LSTM, while the decoder layer utilized Bi-LSTM.

Moussa et al. [24] introduced a novel model called AraBert. This innovative approach involves end-to-end pre-training of both the encoder and decoder components, inspired by the Bart model and utilizing BERTScore. While maintaining the structure of the Bart model, they added an extra layer on top of both the encoder and decoder. The model was fine-tuned for 3 epochs using the Adam optimizer. Following common practices in Arabic, they normalized the output summary by removing discretization, normalizing Alef/Yaa, and segregating punctuation for evaluation.

Ashwathy et al. [31] introduced a novel dataset for interactive digital narrative text (IDN). This dataset was generated from transcripts of two narrative games, namely 'Before the Storm' and 'The Wolf Among Us.' The authors employed three baseline models: SummaRuNNer, BertSum, and Longformer. Among these models, SummaRuNNer demonstrated superior performance compared to the others.

Chatterjee et al. [32] introduced an architectural approach for a multi-layer long text summarizer (MLLTS). Their methodology begins by segmenting the lengthy document into multiple parts, denoted as P. These parts are then distributed across various layers within the multi-layer architecture. Each part is subjected to summarization using three distinct techniques: TextRank, LexRank, and Distil-BART. This results in Px3 summarization outputs from the different summarizers. Subsequently, they employ the VoteSumm technique to optimize the generation of the final summary.

Kashyap et al. [33] introduce a recursive approach to summarization involving the finetuning of a pre-trained model known as LongFormerEncoderDecoder (LED). Their process commences by segmenting the lengthy text document. Subsequently, they employ the LED model to create summaries for each of these segmentations. These partial summaries are then combined into an intermediate summary. This intermediate summary is subsequently treated as input to the LED model to generate a summary for it. This recursive process continues until a specific summary length is achieved or there is no change observed between the generated summary and the intermediate summary.

Kumar et al. [34] performed fine-tuning on a pre-trained Bart model using the SamSum dialogue summarization dataset. Their approach begins with segmenting dialogues into chunks, considering two factors. The first factor is based on SCENE_CHANGE, and the second factor involves segmenting based on the maximum tokens permissible in the pre-trained model, which, in their specific case, amounted to 1024 tokens.

Upadhyay et al. [35] introduce a fine-tuned Bart model applied to a substantial abstractive screenplay summarization dataset. Their summarization approach consists of three phases. Initially, they preprocess the text to eliminate redundancies and segment it into chunks, ensuring each chunk adheres to a token-length limit of {512,1024} tokens. Secondly, they leverage the Bart model, trained using the SummScreen dataset. Finally, the summary lines are ranked using TextRank, and the bottom 15% of lines are excluded to derive the final gold summary.

Etaiwi et al. [44] present a semantic graph (SemG-TS) model based on abstractive summarization techniques for the Arabic language. SemG-Ts has four phases. First, it starts by representing the original text as a semantic graph. Second, the graph embedding technique is applied to extract structural details from the semantic graph. Third, suitable Arabic language vectors are created using the semantic features of the Arabic text stored in the semantic graph. Fourth, the resulting vectors are sent to a deep neural network (NN) to generate the final text summary. The dataset's articles were collected from AlJazeera.net.

El Moatez et al. [45] present an innovative benchmark known as ARGEN for generating summaries in the Arabic language. They employ the standard T5-base and T5-small architectures, each featuring 12 layers in both the encoder and decoder, with 12 attention heads and 768 hidden units. The researchers pre-train three robust variations of the text-to-text transformer (T5) model specifically designed for Modern Standard Arabic (MSA) and various Arabic dialects.

Kahla et al. [46] present a fine-tuned multilingual BERT model for abstractive Arabic text summarization using a newly collected dataset. An important feature of BERT it has a multilingual model available. They followed the same approach using English training.

Alahmadi et al. [48] present a topic-aware abstractive summarization model (TAAM), based on a deep recurrent neural network (RNN) for the Arabic text language. The

TAAM consists of four modules: 1) word embedding model utilized to transform the input text into vectors of varying dimensions. 2) an encoder model that includes layers of RNN with LSTM and an attention mechanism to highlight essential words in the input text. 3) a topic-aware module comprising an RNN-based topic classifier that establishes data features with enhanced informativeness. 4) a decoding module composed of several RNN layers with LSTM gates to compute the probability of each word.

C. *Hybrid Summarization*

Wang et al. [25] introduced a hybrid text summarization model that leverages the strengths of both extractive and abstractive methods. The model operates in two stages: In the initial stage, they partition a document into sentences and transform them into a topological structure utilizing a complete graph. Each graph edge is assigned a weight based on the similarity between two sentences, which are then ranked using PageRank. The extraction process selects the highest-ranked sentences. In the subsequent stage, the sentences extracted in the first stage are inputted into an RNN model based on an encoder-decoder architecture to generate the final summary.

Fadel et al. [26] introduced a novel hybrid model for long text, amalgamating both extractive and abstractive summarization techniques. The extractive model comprises three phases. The initial phase involves sentence processing, followed by feature extraction through a set of formulations in the second phase. The third phase entails sentence selection to identify pivotal sentences. The abstractive model is constructed upon an encoder-decoder LSTM framework. Within the encoder, they employed multiple layers of bidirectional LSTM, while the decoder utilized a recurrent neural network with an attentional model. The encoder takes the output of the extractive model as its input, converting words into distributed representation vectors. Subsequently, the decoder employs the encoder's output and generates the final hidden state produced by the encoder.

Ji Pei et al. [27] introduced a hybrid text summarization model, crafted through the fusion of two recurrent neural network (RNN) layers – one for an extractive model and another for a sequence-to-sequence attentional abstractive model. This model unfolds in two stages: In the initial stage, the document is segmented into sentences and subjected to preprocessing. In the second stage, the sentences are inputted into the Extractive model, which assigns a score to each sentence and selects the top-ranked sentence to generate a summary through a 2-RNN layer process. The third stage involves taking the top-ranked sentences generated by the extractive model and using them as input for the abstractive model.

Kim et al. [36] introduce a hybrid summarization model that incorporates both abstractive and extractive summarization techniques. The primary objective is to summarize movie scripts. For the abstractive model, they fine-tuned the DialogLM model using the FD-Dataset. Additionally, they employed a pre-trained BertSum model for the extractive summarization model. Their hybrid model encompasses four distinct stages: scene segmentation, abstractive summarization, important scene selection, and extractive summarization.

Dongqi et al. [37] present a novel model designed for summarizing movie scripts, employing a two-stage hierarchical architecture. In the initial stage, extractive summarization is employed through the heuristic extraction method. This approach contributes to reducing the average input length of movie scripts by 66%, trimming it down from 24k tokens to 8k tokens. Subsequently, in the second stage, they incorporated a pre-trained model, LongerFormerEncoderDecoder (LED), to generate the final summary. The LED model was augmented through two fine-tuning methods, BitFit and NoisyTune. Additionally, they imposed constraints on the encoding input length (8k tokens) and decoding length (1024 tokens).

Elsaid et al. [47] present a hybrid model consisting of a Modified Sequence-To-Sequence (MSTS) model and a transformer-based model. MSTS was modified by adding multi-layer encoders and a one-layer decoder to its structure. MSTS produces an extractive summarization using bi-directional LSTM. The extractive summarization is manipulated by a transformer-based model to generate an abstractive summary using MT5 transformers.

V. DISCUSSION

This paper presented a comprehensive survey of Arabic text summarization research conducted over the past five years. Our survey encompassed studies targeting single, multi, and long-document summarization, with a particular focus on long-document summarization due to its relative scarcity in the current research landscape.

Two key trends emerged from our analysis. Firstly, research has steadily shifted from single-document summarization to encompass both single and multi-document scenarios. This shift acknowledges the increased complexity and importance of multi-document summarization, where information redundancy across multiple documents requires careful consideration to avoid repetition and generate concise summaries. Secondly, while extractive approaches remain dominant, a growing trend towards exploring abstractive summarization methodologies is evident. Despite the increased complexity, abstractive approaches demonstrate the potential to generate higher-quality summaries and introduce novel phrases, as opposed to simply extracting existing sentences [24]. Notably, most studies addressing long-document summarization favor abstractive or hybrid approaches [16, 14].

Table 2 provides a comparative overview of the included studies, highlighting their utilized datasets, performance metrics, and achieved performance levels.

In conclusion, a thorough understanding of Arabic text summarization necessitates a closer examination of its key components. These include summarization techniques, the influence of source inputs, training and evaluation datasets, employed evaluation methodologies, preprocessing strategies, document segmentation, and the identification of research gaps and future directions. Each of these aspects significantly impacts the effectiveness and efficiency of text summarization. By delving deeper into these areas, we can gain valuable insights into the current state of the field and pave the way for future advancements in Arabic text summarization.

A. *Summarization Techniques.*

This section delves into three primary summarization approaches: extractive, abstractive, and hybrid. We will analyze the theoretical underpinnings of each technique, identifying their specific strengths and weaknesses to provide a nuanced understanding of their suitability for different summarization tasks. Furthermore, we will explore the latest advancements within each category, highlighting their potential to enhance summarization performance and expand the scope of applicable contexts.

1) *Extractive Summarization:*

- Extractive summarization aims to select and extract the most important sentences or phrases from the source text to compose a summary.
- This technique is particularly well-suited for summarizing short texts, such as articles and news [9].
- Based on our study, graph-based techniques have emerged as the most used methods for extractive summarization in recent years [43].

2) *Abstractive Summarization:*

- Abstractive summarization aims to generate summaries by rewriting and rephrasing the content from the source text [4].
- This technique is suitable for summarizing both short and long texts. Abstractive methods offer increased flexibility in producing concise summaries and have the potential to create more human-like summaries [25].
- Based on our study, encoder-decoder techniques have become the most widely used method for abstractive summarization in recent years.

3) *Hybrid Summarization:*

- Hybrid summarization combines both extractive and abstractive techniques to harness the strengths of both approaches.
- This technique is especially well-suited for summarizing lengthy texts because it seeks to find a balance between the precision of extractive summarization and the adaptability of abstractive summarization [26].

Figure 5 shows the distribution of summarization techniques used in our survey.

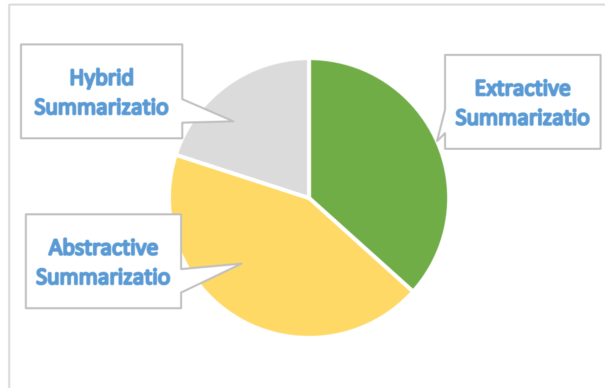

Figure 5. Distribution of Summarization Techniques

Figure 6 shows that, in recent years, researchers have headed towards utilizing transformer techniques over other methods.

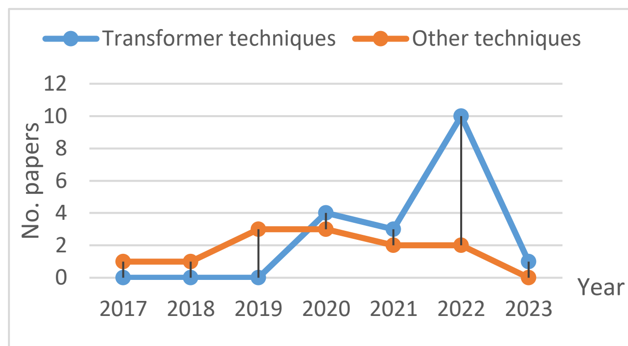

Figure 6. Transformers and Other Techniques Used Per Year

B. *Source Inputs to Summarization*

The choice of summarization approach and methods may vary depending on the type of source input and the specific summarization goals. Based on our study of previous studies mentioned in Table 2, some important source inputs commonly used for summarization include:

1) *Short-Single Document Summarization:*

- This type of summarization involves condensing a single source document into a brief and more concise version.
- It is commonly applied to summarizing news articles, research papers, blog posts, and individual documents [6].
- One of its primary challenges lies in selecting which sentences or phrases to include in the summary, ensuring that the most important information is retained.

2) *Multi-Document Summarization:*

- This type of summarization involves summarizing multiple source documents (e.g., articles, and reports) that are related to a common topic [15].
- It finds frequent use in aggregating news coverage, summarizing research on a specific subject, or crafting literature reviews.
- It faces challenges such as avoiding redundancy when summarizing similar documents and ensuring that the summary effectively captures diverse perspectives and viewpoints from multiple sources [22].

3) *Long-Document Summarization:*

- This type of summarization focuses on summarizing extensive documents, including books, theses, and extended reports. It is also employed for lengthy legal documents, providing concise summaries for lawyers or judges [39].
- It combines both extractive and abstractive techniques to effectively condense the content.
- One of the key challenges in long-document summarization is ensuring that the summary maintains coherence and comprehensively covers the essential points of the lengthy source document.

Figure 7 shows the distribution of input source types for papers used in our survey.

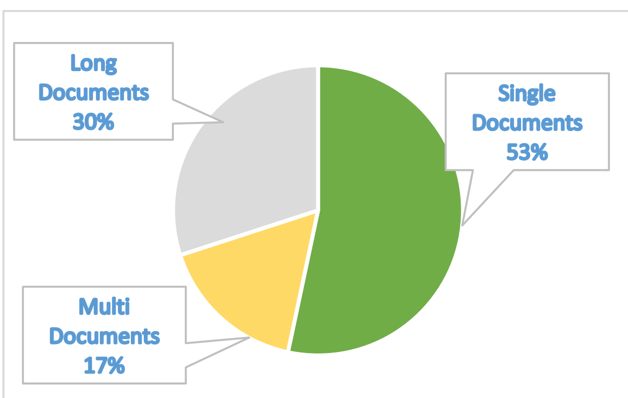

Figure 7. Distribution of Input Source for Summarization

C. Dataset

In the field of research, the benefit of a dataset becomes essential for evaluating the effectiveness of a proposed approach. In the context of text summarization research, a diverse range of datasets has been employed to assess performance and outcomes.

Figure 8 shows the distribution of Datasets for papers used in our survey.

Among the 30 selected studies on text summarization research, 12 focused on long English text, while 18 centered on short Arabic text. According to our study, the most favored dataset for Arabic text summarization is the Essex Arabic Summaries Corpus, with six studies. Meanwhile, for English long-text summarization, the preferred dataset is the SummScreen dataset with four studies, and for book-text summarization, the prominent dataset is the BookSum dataset with two studies.

1) *The* Essex Arabic Summaries Corpus (EASC) is a collection of Arabic documents and their corresponding human-generated summaries [14]. The EASC Dataset is likely to include a variety of text genres. It may consist of news articles, reports, and other types of documents, along with summaries that capture the main points and key information from the original texts. Researchers and developers often utilize such corpora for tasks like text summarization, machine translation, information retrieval, and sentiment analysis, among others [20].

It is suitable for short-text summarization as it contains articles with their corresponding summaries.

2) *The* SummScreen Dataset is an abstractive summarization *dataset* combining TV series transcripts and episode recaps. It is constructed from fan-contributed websites. it combines long source inputs, large numbers of speakers, and a moderate number of instances [34].

3) *The* BookSum Dataset contains a series of datasets designed for the summarization of extensive narratives [33]. Encompassing literary works like novels, plays, and stories, this dataset features accurately crafted abstractive summaries presented across three escalating levels of intricacy: paragraph, chapter, and book.

It has organized data that presents special challenges for *summarization* systems. These challenges include dealing with really long documents, complicated cause-and-effect relationships, time-based connections, and rich discourse structures [33].

It is suitable for long-text summarization as it contains books with their chapters and paragraphs, all labeled with corresponding summaries.

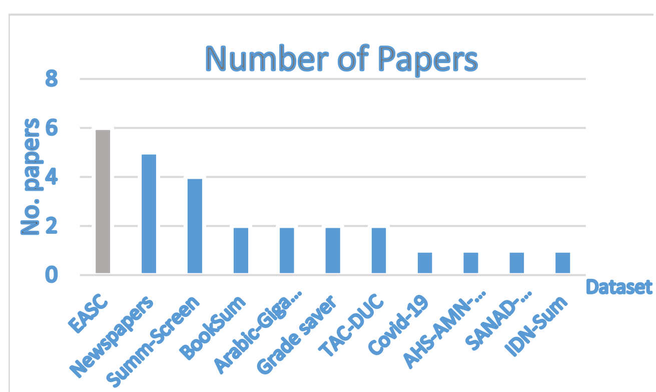

Figure 8. Distribution of Datasets for Summarization

Table 1. shows a comparative table between the three datasets:

Feature	The Essex Arabic Summaries Corpus (EASC)	The SummScreen Dataset	The BookSum Dataset
Language	Arabic	English	English
Type of Content	News articles and their summaries	TV show transcripts and summaries	Books and their summaries
Purpose	Evaluating Arabic summarization systems	Evaluating summarization models for TV show scripts	Evaluating book summarization models
Size of Dataset	4,000+ summaries	Over 20,000-episode summaries	12,000+ book summaries
Domain	News	Entertainment (TV shows)	Literature
Source of Summaries	Human-generated summaries	Human-generated summaries	Human-generated summaries
Additional Features	Focus on Arabic NLP research	Useful for dialogue-based summarization research	Includes long-form text summarization challenges

Table 1. Comparative table of the Three Datasets

D. *Evaluations in text summarization*

In the context of Arabic text summarization,it refers to the process of assessing or evaluating the quality of generated summaries. Evaluating the effectiveness of summarization models and techniques is essential to ensure that the generated summaries are coherent, informative, and faithful to the original text. Based on our study, a range of methods have been employed to assess the outcomes of machine-generated summaries. These approaches encompass evaluations based on key content extraction, sentence extraction, content relevance, and alignment with specific tasks:

1) ***ROUGE*** (Recall-Oriented Understudy for Gisting Evaluation): provide a quantitative measure of how well the generated summary captures the main content of the reference summary, it primarily relies on text overlapping. This means that summaries achieving high ROUGE scores may not necessarily reflect semantic coherence or understanding [16]. Therefore, while ROUGE is valuable for comparing summarization models and assessing improvements, it's important to supplement its findings with metrics that consider semantic relevance. They are widely used in research and development to compare different summarization models, assess improvements, and guide the fine-tuning of summarization systems, including those focused on Arabic text summarization.

2) ***BLEU*** (Bilingual Evaluation Understudy): Originally developed for machine translation, BLEU measures the precision of n-grams in the generated summary compared to reference summaries [47]. Like ROUGE, it primarily focuses on text overlapping, which may not fully capture semantic nuances. As such, while BLEU can indicate the quality and adequacy of the generated summary, it's essential to complement its findings with metrics that delve into semantic coherence and understanding.

3) ***Human Evaluation***: Human judges assess the quality of generated summaries based on various criteria such as coherence, relevance, fluency, and informativeness [31]. Human evaluation provides insights into the overall quality from a human perspective.

E. *Preprocessing*

Preprocessing is a crucial step in Arabic text summarization to prepare the input data for effective summarization [18]. Due to the complexities of the Arabic language, preprocessing plays a significant role in enhancing the quality of summaries. Based on our study of previous works mentioned in Table 2, some important preprocessing steps for Arabic text summarization include [21]:

1) *Text Tokenization:* Break down the input Arabic text into individual words or tokens. It is crucial because it enables the text to be processed at a granular level, facilitating analysis, and understanding by algorithms and systems.

2) *Stopword Removal:* Remove common stopwords (high-frequency function words) in Arabic to reduce noise and focus on content-bearing words during summarization, a critical preprocessing step that enhances the quality and relevance of the generated summaries.

3) *Stemming and Lemmatization:* Apply stemming or lemmatization to reduce words to their root forms, addressing the rich morphological variations in Arabic words, which is essential for improving the consistency and accuracy of the summarization process.

4) *Sentence Segmentation:* Divide the text into sentences, considering the various sentence-ending punctuation marks and potential challenges in detecting sentence boundaries, a crucial step that enables the summarization algorithm to process the text at the sentence level, facilitating coherent and informative summarization.

5) *Removing Noisy Text:* Eliminate noisy elements like HTML tags, special characters, and extraneous symbols that might hinder summarization, ensuring that the summarization algorithm focuses solely on the meaningful content of the text, thus improving the quality and clarity of the generated summaries.

F. *Segmentation in text summarization*

Segmenting books or documents is a crucial step in the summarization process. It involves breaking down the text into smaller, manageable chunks to generate meaningful summaries [17]. The choice of segmentation method depends on the nature of the document and the specific requirements of the summarization task. In many cases, a combination of these segmentation techniques may be used to produce a well-rounded summary that captures the key information and insights from the text.

Based on our study of previous works mentioned in Table 2, here are several ways to segment books or documents for summarization:

1) Chapter-*Level Segmentation:*
Divide the text into chapters or sections [17]. Each chapter can be summarized individually, providing a structured summary of the entire book.

2) Paragraph-Level Segmentation*:*
Break the document into paragraphs. Summarize each paragraph individually to capture the main points and ideas [22].

3) Sentence-Level Segmentation*:*
Segment the text into individual sentences. Summarize each sentence or group of sentences to create concise summaries.

4) Topic-Based Segmentation*:*
Identify key topics or themes in the document. Segment the text based on these topics and generate summaries for each topic separately.

G. Gaps and implications for research

Our study presented some important gaps that needed to be filled, especially in the Arabic domain. Several problems have evolved into significant challenges that researchers have worked to tackle and overcome.

There is a lack of text summarization methods specifically designed for books or novels, and there is a limited availability of datasets suitable for the Arabic domain in this context. The lack of datasets specifically designed for summarizing lengthy Arabic texts represents a significant challenge within the field of natural language processing [41]. This shortage of suitable datasets obstructs the development and evaluation of effective summarization models designed to handle the unique linguistic and contextual characteristics of longer Arabic documents.

Another problem is extraction and segmentation because it is a remarkably complex challenge [49]. Within the context of in-text summarization, extraction refers to the process of retrieving information from a data source, which could be either structured or unstructured data, for subsequent processing to generate a concise summary.

Another challenge that remains a limitation in prior research regards semantics. The objective of automated text summarization is to generate a summary containing high-quality essential content [44]. This objective is intricately tied to the inherent meaning encapsulated within the summarized sentences. Particularly in lengthy documents, and notably in scenarios involving multiple documents, the presence of ambiguous sentences, polysemous words (with multiple meanings), or synonyms can arise.

Another issue is that there aren't many studies or systems that let users pick how much they want to shorten a summary from the original text. Right now, a lot of summarization tools use fixed rules to make summaries, so users don't have much say in how they turn out. This is a big deal because it means these systems aren't very user-friendly. People might want different amounts of shortening based on why they're making the summary and what the content is like. So, not being able to pick the compression level can be a problem for different users and different situations.

Addressing these gaps holds far-reaching implications, benefiting applications like news analysis, legal document review, and academic research, thereby supporting the pivotal role of accurate summarization in enhancing information extraction, comprehension, and decision-making. Moreover, we need more research to gather large sets of data for summarization and create models that work well with long texts.

VI. CONCLUSION AND FUTURE WORK

Text summarization plays a crucial role in assisting readers in capturing the key essence of lengthy texts, thereby enhancing comprehension and optimizing time efficiency.

This research project provides an extensive overview of the growing field of automatic Arabic text summarization. Examining existing research in this area is crucial due to the rapid advancements and maturation of extractive techniques. Notably, there is a noticeable shift towards abstractive summarization methods and the exploration of real-time summarization capabilities. This transition is driven by the inherent complexity of abstractive summaries, which require greater computational resources compared to extractive methods. However, the demand for extractive summaries persists due to their predictability and proven effectiveness, as evidenced by sustained research efforts in this domain.

Our survey meticulously examines various methods employed for both summary generation and evaluation, with a focus on studies published between 2018 and 2023. Specifically, we delve into the intricacies of Arabic single, multi-document, and long-document summarization. Recognizing the lack of research targeting Arabic long-text and textbook summarization, we strategically incorporate relevant studies from the English domain for comparative analysis. Interestingly, our findings reveal a prevalent preference for abstractive summarization over extractive approaches. Furthermore, PageRank remains the dominant algorithm for sentence ranking, empowering users to tailor the level of summarization compression. Acknowledging the inherent complexities of automatic text summarization and the potential imperfections in generated summaries, our research underscores a critical challenge within Arabic text summarization: the scarcity of gold-standard reference summaries, particularly for book-length texts. The domain of Arabic long-text summarization presents an exciting and dynamic research frontier.

One promising avenue for future exploration involves delving into the development of hybrid models that synergistically leverage the strengths of both extractive and abstractive approaches.

TABLE 2
A SUMMARY OF STUDY METHODS AND THEIR EVALUATION

(Author, year) [Reference]	**The used approach**	**Language**	**Input Text**	**Dataset**	**Performance**
(Khulood et al., 2018) [2]	Extractive	Arabic	Multi doc	Arabic Giga-word	The performance is tested using R, P and F-measure
(Rada et al., 2007) [7]	Extractive	English	Long doc / Book	Grade Saver - Cliff's Notes	R1: [P:0.472, R:0.366 F:0.412] R2: [P:0.069, R:0.054 F:0.061] R-SU4: [P:0.148, R:0.115 F:0.129]
(Elbarougy et al., 2020) [14]	Extractive	Arabic	Single doc	EASC	P: 68.75, R: 72.94, F-measure: 67.99
(Alqaisi et al., 2020) [15]	Extractive	Arabic	Multi doc	TAC 2011 - DUC 2002	TAC: [R1:38.9, R2:17.7, RL:35.4, R-SU4:15.8] DUC: [R1:47.1, R2:23.7, RL:47.1, R-SU4:20.4]
(Bowen et al., 2020) [16]	Extractive	English	Multi doc	COVID-19 dataset	Experiment show that 40% abstraction tend has ROUGE scores lower than the 60% group Experiment show that using only verbs as keywords has very low ROUGE scores
(Ladhak et al., 2020) [17]	Extractive	English	Long doc / Book	(Grade Saver) BB-BW-CN-GS-NG	R1: 35.9, R2: 7.0, RL: 35.2
(Samer et al., 2020) [18]	Extractive	Arabic	Multi doc	EASC	F1-Score: 0.644
(Zuhair et al., 2019) [19]	Extractive	Arabic	Single doc	TAC-2011	P: 0.25, R: 0.37, F-Score: 0.30
(Alaidine et al.,2021)[20]	Extractive	Arabic	Single doc	EASC	R1-F1: 0.56, R2-F1: 0.47
(Raed et al., 2019) [21]	Extractive	Arabic	Single doc	EASC	P: 0.57, R: 0.60, F-measure: 0.57
(Merniz et al.,2021)[22]	Extractive	Arabic	Multi doc	Al sulaiti corpus [29]	R1: 0.59, R2:0.36, R-SU4: 0.41
(Azmi et al., 2019) [3]	Abstractive	Arabic	Single doc	newspapers (Ar-Riyadh - Al-Jazirah)	For a ratio of 31% P: 0.69, R: 0.68, F-measure: 0.69 For a ratio of 25% P: 0.72, R: 0.56, F-measure: 0.62
(Wazery et al., 2022) [4]	Abstractive	Arabic	Single doc	AHS - AMN	For AHS: R1: [P:54.95, R:50.48, F1:51.49] R2: [P:13.1, R:13.1, F1:12.01] RL: [P:37.48, R:35.19, F1:34.37] BELU:0.41
(Dima et al., 2021) [23]	Abstractive	Arabic	Single doc	SANAD_SUBSET	R1: 38.4, R1-N: 46.2, R1-S: 52.6, R1-C: 58.1
(Moussa et al., 2022) [24]	Abstractive	Arabic	Single doc	Arabic Giga-word - XL-sum.	R1:42.4, R2:28.8, R3:4.3, BS:69.8
(Wang et al., 2017) [25]	Hybrid	English	Single doc	Chinese websites	R1: 36.6, R2: 25.5, RL: 35.1
(Fadel et al.,2020)[26]	Hybrid	Arabic	Single doc	EASC - Abu El-khair [30]	R1-F1:0.54, R1-F1:0.53
(Ji Pei et al., 2020) [27]	Hybrid	English	Single doc	newspapers (CNN – DailyMail)	R1: 43.92, R2: 20.70, RL: 40.67
(Ashwathy et al,2022)[31]	Abstractive	English	Long doc	IDN-Sum	R1F1: BS:0.16 , LF:0.31, SRL:0.42
(Chatterjee et al. 2022)[32]	Abstractive	English	Long doc / Book	BookSum	For validation set: R1: 0.159, R2: 0.014 For test set: R1: 0.2643, R2: 0.0471, RL:0.2436

TABLE 2
A SUMMARY OF STUDY METHODS AND THEIR EVALUATION

(Author, year) [Reference]	Model output	Language	Input Text	Dataset	Performance achieved
(Kashyap et al. 2022)[33]	Abstractive	English	Long doc / Book	BookSum	R1-F1 score of 29.75. R2-F1 score of 7.89. BERT F-1 score of 54.10.
(Kumar et al, 2022)[34]	Abstractive	English	Long doc	SummScreen-FD	R1: 0.2469, R2: 0.0408, RL: 0.2300
(Upadhyay et al, 2022) [35]	Abstractive	English	Long doc	XSum - SAMSum - SummScreen-FD	R1: 0.3921, R2: 0.0909, RL: 0.3794
(Kim et al, 2022) [36]	Hybrid	English	Long doc	SummScreen-FD	R1: 0.4144, R2: 0.0823, R3: 0.3963
(Dongqi et al,2022) [37]	Hybrid	English	Long doc	SummScreen-FD	R-1 F1: 46.34, , R-2 F1: 11.58
(Etaiwi et al,2022) [44]	Abstractive	Arabic	Single doc	News-AlJazeera.net	P: 0.45, R: 0.50, F-measure: 0.47
(El Moatez et al,2022) [45]	Abstractive	Arabic	Single doc	News- Twitter Dataset - EASC	R-1: 54.61, R-2: 43.58, R-3: 54.55
(Kahla et al,2021)[46]	Abstractive	Arabic	Single doc	News- CNN, BBC…	R-1: 16.363, R-2: 2.524, R-3: 16.363
(Elsaid et al,2023) [47]	Hybrid	Arabic	Single doc	HASD (News)	R-1: 0.637, R-2: 0.490, R-L: 0.6047 Bleu: 0.44
(Alahmadi et al,2022)[48]	Abstractive	Arabic	Single doc	News-MSA	R-1: 71.6, R-2: 58.6, R-L: 70.1

Terms: Precision (P), Recall (C), ROUGE-1 (R1), ROUGE-2 (R2), ROUGE-L (RL), BERTScore (BS), Rouge-SU4 (R-SU4), ROUGE1-NOORDER (R1-N), ROUGE1-STEM (R1-S), ROUGE1-CONTEXT (R1-C), Essex Arabic Summaries Corpus (EASC), Arabic Headline Summary (AHS), Arabic Mogalad_Ndeef (AMN), arronsBookNotes (BB), BookWolf, (BW), CliffsNotes (CN), GradeSaver (GS), NovelGuide (NG), Large-Scale Multilingual Abstractive Summarization (XL-Sum),BerSum(BS), Longformer (LF), SummaRuNNer(SRL), SummScreen forever dream (SummScreen-FD), hybrid Arabic text summarization dataset (HASD), Modern Standard Arabic (MSA).

VII. REFERENCES

[1] Saziyabegum, Saiyed, and Priti S. Sajja. "Literature review on extractive text summarization approaches." International Journal of Computer Applications 156, no. 12 (2016).

[2] Maria, Khulood Abu, Khalid Mohammad Jaber, and Mossab Nabil Ibrahim. "A new model for Arabic multi-document text summarization." International Journal of Innovative Computing, Information and Control 14, no. 4 (2018): 1443-1452.

[3] Azmi, Aqil M., and Nouf I. Altmami. "An abstractive Arabic text summarizer with user controlled granularity." Information Processing & Management 54, no. 6 (2018): 903-921.

[4] Wazery, Yaser Maher, Marwa E. Saleh, Abdullah Alharbi, and Abdelmgeid A. Ali. "Abstractive Arabic text summarization based on deep learning." Computational Intelligence and Neuroscience 2022 (2022).

[5] Wang, Shuai, Xiang Zhao, Bo Li, Bin Ge, and Daquan Tang. "Integrating extractive and abstractive models for long text summarization." In 2017 IEEE international congress on big data (BigData congress), pp. 305-312. IEEE, 2017.

[6] Bialy, Asmaa A., Marwa A. Gaheen, R. M. ElEraky, A. F. ElGamal, and Ahmed A. Ewees. "Single Arabic document summarization using natural language processing technique." Recent Advances in NLP: The Case of Arabic Language (2020): 17-37.

[7] Mihalcea, Rada, and Hakan Ceylan. "Explorations in automatic book summarization." In Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), pp. 380-389. 2007.

[8] Vijay Kumar, N., and M. Janga Reddy. "Factual instance tweet summarization and opinion analysis of sport competition." In Soft Computing and Signal Processing: Proceedings of ICSCSP 2018, Volume 2, pp. 153-162. Springer Singapore, 2019.

[9] Gupta, Vishal, and Gurpreet Singh Lehal. "A survey of text summarization extractive techniques." Journal of emerging technologies in web intelligence 2, no. 3 (2010): 258-268.

[10] Alwis, Varuni. "Intelligent E-news summarization." In 2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer), pp. 189-195. IEEE, 2018.

[11] Zhang, Shiyue, Asli Celikyilmaz, Jianfeng Gao, and Mohit Bansal. "EmailSum: Abstractive email thread summarization." arXiv preprint arXiv:2107.14691 (2021).

[12] Vanetik, Natalia, Marina Litvak, and Sophie Krimberg. "Summarization of financial reports with tiber." Machine Learning with Applications 9 (2022): 100324.

[13] Al Qassem, Lamees Mahmoud, Di Wang, Zaid Al Mahmoud, Hassan Barada, Ahmad Al-Rubaie, and Nawaf I. Almoosa. "Automatic Arabic summarization: a survey of methodologies and systems." Procedia Computer Science 117 (2017): 10-18.

[14] Elbarougy, Reda, Gamal Behery, and Akram El Khatib. "Extractive Arabic text summarization using modified PageRank algorithm." Egyptian informatics journal 21, no. 2 (2020): 73-81.

[15] Alqaisi, Rana, Wasel Ghanem, and Aziz Qaroush. "Extractive multi-document Arabic text summarization using evolutionary multi-objective optimization with K-medoid clustering." IEEE Access 8 (2020): 228206-228224.

[16] Kieuvongngam, Virapat, Bowen Tan, and Yiming Niu. "Automatic text summarization of covid-19 medical research articles using bert and gpt-2." arXiv preprint arXiv:2006.01997 (2020).

[17] Ladhak, Faisal, Bryan Li, Yaser Al-Onaizan, and Kathleen McKeown. "Exploring content selection in summarization of novel chapters." arXiv preprint arXiv:2005.01840 (2020).

[18] Abdulateef, Samer, Naseer Ahmed Khan, Bolin Chen, and Xuequn Shang. "Multidocument Arabic text summarization based on clustering and Word2Vec to reduce redundancy." Information 11, no. 2 (2020): 59.

[19] Malallah, Suhad, and Zuhair Hussein Ali. "Multilingual Text Summarization based on LDA and Modified PageRank." Iraqi Journal of Information Technology 9, no. (2019).

[20] Ayed, Alaidine Ben, Ismaïl Biskri, and Jean-Guy Meunier. "Arabic text summarization via Knapsack balancing of effective retention." Procedia Computer Science 189 (2021): 312-319.

[21] Al-Abdallah, Raed Z., and Ahmad T. Al-Taani. "Arabic text summarization using firefly algorithm." In 2019 amity international conference on artificial intelligence (AICAI), pp. 61-65. IEEE, 2019.

[22] Merniz, Amina, Anja Habacha Chaibi, and Henda Hajjami Ben Ghézala. "Multi-document Arabic Text Summarization based on Thematic." (2021).

[23] Suleiman, Dima, and Arafat Awajan. "Multilayer encoder and single-layer decoder for abstractive Arabic text summarization." Knowledge-Based Systems 237 (2022): 107791.

[24] Eddine, Moussa Kamal, Nadi Tomeh, Nizar Habash, Joseph Le Roux, and Michalis Vazirgiannis. "Arabart: a pretrained arabic sequence-to-sequence model for abstractive summarization." arXiv preprint arXiv:2203.10945 (2022).

[25] Wang, Shuai, Xiang Zhao, Bo Li, Bin Ge, and Daquan Tang. "Integrating extractive and abstractive models for long text summarization." In 2017 IEEE international congress on big data (BigData congress), pp. 305-312. IEEE, 2017.

[26] Fadel, Abdullah, and G. Bora Esmer. "A hybrid long Arabic text summarization system based on integrated approach between abstractive and extractive." In Proceedings of the 2020 6th International Conference on Computer and Technology Applications, pp. 109-114. 2020.

[27] Pei, Ji, Rim Hantach, Sarra Ben Abbès, and Philippe Calvez. "Towards hybrid model for automatic text summarization." In 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 987-993. IEEE, 2020.

[28] Elsaid, Asmaa, Ammar Mohammed, Lamiaa Fattouh Ibrahim, and Mohammed M. Sakre. "A comprehensive review of arabic text summarization." IEEE Access 10 (2022): 38012-38030.

[29] Al-Sulaiti, Latifa, and Eric Steven Atwell. "The design of a corpus of contemporary Arabic." International journal of corpus linguistics 11, no. 2 (2006): 135-171.

[30] El-Khair, Ibrahim Abu. "1.5 billion words arabic corpus." arXiv preprint arXiv:1611.04033 (2016).

[31] T Revi, Ashwathy, Stuart Middleton, and David Millard. "Idn-sum: A new dataset for interactive digital narrative extractive text summarisation." (2022).

[32] Chatterjee, Niladri, Aadyant Khatri, and Raksha Agarwal. "Summarization of Long Input Texts Using Multi-Layer Neural Network." In Proceedings of The Workshop on Automatic Summarization for Creative Writing, pp. 13-18. 2022.

[33] Kashyap, Prerna. "COLING 2022 Shared Task: LED Finteuning and Recursive Summary Generation for Automatic Summarization of Chapters from Novels." In Proceedings of The Workshop on Automatic Summarization for Creative Writing, pp. 19-23. 2022.

[34] Kumar, Rishu, and Rudolf Rosa. "TEAM UFAL@ CreativeSumm 2022: BART and SamSum based few-shot approach for creative Summarization." In Proceedings of The Workshop on Automatic Summarization for Creative Writing, pp. 24-28. 2022.

[35] Upadhyay, Aditya, Nidhir Bhavsar, Aakash Bhatnagar, Muskaan Singh, and Petr Motlicek. "Automatic summarization for creative writing: BART based pipeline method for generating summary of movie scripts." In Proceedings of The Workshop on Automatic Summarization for Creative Writing, pp. 44-50. 2022.

[36] Kim, EunChong, TaeWoo Yoo, Gunhee Cho, Suyoung Bae, and Yun Gyung Cheong. "The CreativeSumm 2022 shared task: a two-stage summarization model using scene attributes." In Proceedings of The Workshop on Automatic Summarization for Creative Writing, pp. 51-56. 2022.

[37] Pu, Dongqi, Xudong Hong, Pin-Jie Lin, Ernie Chang, and Vera Demberg. "Two-stage movie script summarization: An efficient method for low-resource long document summarization." In Proceedings of The Workshop on Automatic Summarization for Creative Writing, pp. 57-66. 2022.

[38] Dedhia, Parth Rajesh, Hardik Pradeep Pachgade, Aditya Pradip Malani, Nataasha Raul, and Meghana Naik. "Study on abstractive text summarization techniques." In 2020 international conference on emerging trends in information technology and engineering (ic-ETITE), pp. 1-8. IEEE, 2020.

[39] Alwis, Varuni. "Intelligent E-news summarization." In 2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer), pp. 189-195. IEEE, 2018.

[40] Al-Sughaiyer, Imad A., and Ibrahim A. Al-Kharashi. "Arabic morphological analysis techniques: A comprehensive survey." Journal of the American society for information science and technology 55, no. 3 (2004): 189-213.

[41] Shaalan, Khaled, Sanjeera Siddiqui, Manar Alkhatib, and Azza Abdel Monem. "Challenges in Arabic natural language processing." In Computational linguistics, speech and image processing for arabic language, pp. 59-83. 2019.

[42] Rane, Neha, and Sharvari Govilkar. "Recent trends in deep learning based abstractive text summarization." Int. J. Recent Technol. Eng 8, no. 3 (2019): 3108-3115.

[43] AL-Khassawneh, Yazan Alaya, and Essam Said Hanandeh. "Extractive Arabic text summarization-graph-based approach." Electronics 12, no. 2 (2023): 437.

[44] Etaiwi, Wael, and Arafat Awajan. "SemG-TS: Abstractive arabic text summarization using semantic graph embedding." Mathematics 10, no. 18 (2022): 3225.

[45] Nagoudi, El Moatez Billah, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. "AraT5: Text-to-text transformers

for Arabic language generation." arXiv preprint arXiv:2109.12068 (2021).

[46] Kahla, Mram, Zijian Győző Yang, and Attila Novák. "Cross-lingual fine-tuning for abstractive Arabic text summarization." In Proceedings of the international conference on recent advances in natural language processing (ranlp 2021), pp. 655-663. 2021.

[47] ELSAID, Asmaa, AMMAR MOHAMMED, Lamiaa Fattouh, and MOHAMMED SAKRE. "Hybrid Arabic text summarization Approach based on Seq-to-seq and Transformer." (2023).

[48] Alahmadi, Dimah, Arwa Wali, and Sarah Alzahrani. "TAAM: Topic-aware abstractive arabic text summarisation using deep recurrent neural networks." Journal of King Saud University-Computer and Information Sciences 34, no. 6 (2022): 2651-2665.

[49] Nallapati, Ramesh, Feifei Zhai, and Bowen Zhou. "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents." In Proceedings of the AAAI conference on artificial intelligence, vol. 31, no. 1. 2017.

[50] Erkan, Günes, and Dragomir R. Radev. "Lexrank: Graph-based lexical centrality as salience in text summarization." Journal of artificial intelligence research 22 (2004): 457-479.

[51] Chandra, Munehs, Vikrant Gupta, and Santosh Kr Paul. "A statistical approach for automatic text summarization by extraction." In 2011 International Conference on Communication Systems and Network Technologies, pp. 268-271. IEEE, 2011.

[52] “TextRazor - Automatically Extract Meaning from All Your Text," TextRazor, accessed December 12, 2024. Available: https://www.textrazor.com/docs/rest.

[53] Douzidia, Fouad Soufiane, and Guy Lapalme. "Lakhas, an Arabic summarization system." In Proceedings of DUC, vol. 4, pp. 128-135. 2004.

[54] "Summarization - MeaningCloud," MeaningCloud, accessed December 12, 2023. Available: https://www.meaningcloud.com/developer/documentation.

[55] Jalil, Zakia, Jamal Abdul Nasir, and Muhammad Nasir. "Extractive multi-document summarization: a review of progress in the last decade." IEEE Access 9 (2021): 130928-130946.

[56] Yasunaga, Michihiro, Jungo Kasai, Rui Zhang, Alexander R. Fabbri, Irene Li, Dan Friedman, and Dragomir R. Radev. "Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks." In Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 01, pp. 7386-7393. 20