IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 May 2025, accepted 15 June 2025, date of publication 1 July 2025, date of current version 11 July 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3584855

RESEARCH ARTICLE

A Review of Arabic Text Summarization: Methods, Datasets, and Evaluation Metrics, With a Proposed Solution

ZEYAD EZZAT[1,2], GHADA KHORIBA[1,3], (Member, IEEE), AND AYMAN KHALAFALLAH[2,4]

[1]Center for Informatics Science, Information Technology and Computer Science School, Nile University, Giza 12677, Egypt
[2]Applied Innovation Centre, Alexandria 21500, Egypt
[3]Faculty of Computers and Artificial Intelligence, Helwan University, Cairo 11795, Egypt
[4]Computer and Systems Department, Faculty of Engineering, Alexandria University, Alexandria 21544, Egypt

Corresponding author: Zeyad Ezzat (zezzat@nu.edu.eg)

ABSTRACT This survey comprehensively reviews Arabic text summarization, examining state-of-the-art methodologies, commonly used datasets, and evaluation practices. Despite notable progress, the field faces challenges such as fragmented benchmarking, inconsistent metric use, and lacking resources for long-document summarization. We categorize existing summarization methods into traditional, Transformer-based, and hybrid approaches, highlighting their strengths and limitations. We introduce Mukhtasar, a novel dataset supporting short and long summaries across diverse genres to address significant gaps. Additionally, we propose six standardized evaluation splits tailored to distinct summarization goals, promoting reproducibility and fair comparison. To address inconsistencies, we also recommend a consistent reporting protocol using ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-S. While lexical overlap metrics dominate evaluation practices, we identify the absence of neural-based metrics for Arabic as a significant limitation and call for future development in this area. Our contributions aim to unify evaluation protocols, enrich available resources, and guide the community toward more interpretable and scalable Arabic summarization research.

INDEX TERMS Text summarization, Arabic NLP, transformers, deep learning, summarization datasets.

I. INTRODUCTION

There are several challenges in research related to Natural Language Processing (NLP) tasks in the Arabic language, some of which include: Lack of resources: Compared to other languages, Arabic has limited high-quality resources such as annotated corpora, lexicons, and tools as highlighted in previous research [1], [2], which makes it challenging to develop and evaluate NLP models.

Morphological complexity: Arabic has a complex morphology with a rich system of inflection and derivation, which makes it challenging to process Arabic text, especially in tasks such as part-of-speech tagging, named entity recognition, and machine translation [1].

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

Dialectal variation: Arabic is spoken in different dialects across the Middle East and North Africa, each with variations in pronunciation, grammar, and vocabulary, which poses a significant challenge for developing NLP models that handle multiple dialects [3].

Summarization poses a significant challenge in text processing. It involves condensing an article's content into a concise version. While it may be tempting to select only a sentence or two, such an approach often fails to capture the essence and depth of the original article. Summarization can be divided into two distinct categories: extractive and abstractive.

In extractive summarization, the task involves selecting the most important sentences from the original article for the summary. On the other hand, abstractive summarization involves generating entirely new text that

comprehensively captures the article's essence after a thorough understanding [4].

Another crucial aspect to consider is the article's length, which needs to be summarized. This dimension introduces variations such as short article summaries and lengthy article summaries; the latter, also referred to as document summaries, involves significantly lengthier articles. Dealing with longer texts adds a new layer of complexity that traditional methods cannot address [5].

This research investigates existing methods and datasets, offering a fresh perspective on their utilization while exploring the most effective approaches to building well-trained models. The contributions of our paper can be summarized as follows:

- We conducted a comprehensive survey of methods used in Arabic summarization literature.
- We compiled all available Arabic summarization datasets and carried out an extensive intrinsic evaluation of their quality.
- We introduced Mukhtasar, a novel dataset tailored for summarization tasks emphasizing summary length.
- We fine-tuned a multilingual pre-trained model using our newly developed dataset and evaluated its performance across all available Arabic summarization datasets.

The structure of this paper is organized as follows: The Literature Search and Selection Process outlines the methodology used to identify relevant studies. The Summarization Methods section presents a comprehensive review of existing approaches to Arabic text summarization. This is followed by an in-depth discussion of evaluation metrics, addressing the assessment of summarization quality and the intrinsic characteristics of the datasets used in evaluation.

Next, we present three dedicated sections on datasets: (1) a review of all available Arabic summarization datasets, (2) an in-depth analysis of these datasets, and (3) the introduction of a new dataset, referred to as Mukhtasar. We encourage the Arabic research community to leverage the Mukhtasar dataset in future research endeavors.

In the Experiments section, we compare the performance of two models trained on Mukhtasar with scores reported in the literature. Finally, the paper concludes with a discussion and summary of our findings.

II. LITERATURE SEARCH AND SELECTION PROCESS

To ensure the rigor and reproducibility of our systematic review, we adopted a structured methodology for identifying, selecting, and analyzing relevant literature on Arabic text summarization. This section outlines the search strategy, data sources, inclusion/exclusion criteria, and the research questions guiding our analysis.

A. SEARCH STRATEGY

We constructed a set of search strings to capture a broad range of work related to Arabic text summarization. The primary keywords used in various combinations included the following queries.

First query: (''Arabic text summarization'' OR ''Arabic summarization models'' OR ''Arabic NLP summarization'' OR ''Arabic summarization datasets'' OR ''Arabic summarization evaluation'' OR ''Arabic abstractive summarization'' OR ''Arabic extractive summarization'')

To narrow and specify further (e.g., within NLP context or publication type), we extend the query to be:

(''Arabic text summarization'' OR ''Arabic summarization models'' OR ''Arabic NLP summarization'' OR ''Arabic summarization datasets'' OR ''Arabic summarization evaluation'' OR ''Arabic abstractive summarization'' OR ''Arabic extractive summarization'') AND (''natural language processing'' OR NLP).

Boolean operators (`AND`, `OR`) were used to refine results and capture different summarization types and evaluation perspectives.

B. DATABASES CONSULTED

We conducted our search across multiple reputable academic databases and digital libraries to ensure comprehensive coverage: IEEE Xplore, Google Scholar, ACL Anthology, and Scopus/Web of Science. The search was conducted between **[2010]** and **[2024]**. We have done our best to cover the literature review of the summarization evolution in Arabic literature.

C. INCLUSION AND EXCLUSION CRITERIA

The following criteria were applied to filter the retrieved papers:

Inclusion Criteria:

- Studies focused on automatic Arabic text summarization.
- Papers that proposed new models, datasets, or evaluation strategies.
- Peer-reviewed journal and conference papers.
- Articles available in English or Arabic.

Exclusion Criteria:

- Papers on general NLP or non-Arabic summarization without a specific Arabic component.
- Non-peer-reviewed sources (e.g., blog posts, opinion pieces).
- Duplicate studies or extended versions of earlier work without significant new contributions.

D. RESEARCH QUESTIONS

The following key research questions guided our review:

RQ1: What are the main approaches (traditional, neural, hybrid) used in Arabic text summarization?

RQ2: What are the characteristics and limitations of existing Arabic summarization datasets?

RQ3: What are the major gaps in the field, and how can new resources (such as our proposed *Mukhtasar* dataset) address them?

III. ARABIC TEXT SUMMARIZATION METHODS

Summarizing Arabic text is difficult due to its intrinsic complexity and the scarcity of extensive datasets. In this section, we will provide a historical overview of the methods and the evolution of the summarization task in the Arabic language. Arabic summarization methods can be categorized into two distinct eras, similar to many other tasks in Natural Language Processing (NLP): the world before and after the transformer architecture. The introduction of the transformer architecture has truly revolutionized the entire field.

Before describing the methods, we would like to clarify that the right-to-left (RTL) Arabic script has minimal impact on the summarization models. This is because the models operate at the token level, not based on the visual layout of the text.

In traditional approaches, words are used as tokens, while modern transformer-based models often tokenize text into subwords—sometimes into characters or whole words—depending on the tokenizer used. In most cases, subword tokenization is employed, which helps handle Arabic's rich morphology. Consequently, the RTL nature of Arabic does not pose a technical challenge for these models.

A. TRADITIONAL APPROACHES

Before the introduction of the transformer, the available methods for Arabic summarization were considerably limited. They predominantly relied on extractive techniques. Most of these methods aimed to extract and rank sentences using various algorithms.

1) TOPIC MODELING AND REDUNDANCY REDUCTION METHODS

Researchers utilized vector space representations for sentences to minimize redundancy through methods such as Latent Dirichlet Allocation (LDA) or Minimum Redundancy Maximum Relevance, as detailed in [6], [7], and [8]. This procedure entails first statistically representing sentences and then extracting essential sentences or words using the earlier methods for sentence representation.

2) OPTIMIZATION METHODS

Several researchers have approached sentence ranking as an optimization problem, employing various optimization algorithms such as genetic algorithms, particle swarm optimization (PSO), and firefly algorithm as applied in [9], [10], and [11], respectively. These approaches have diverged in their representations of sentences, employing distinct statistical features for sentence characterization.

3) GRAPH BASED METHODS

Drawing inspiration from Google's page ranking algorithm [12], researchers predominantly employed statistical features to represent sentences. They transformed these representations into a graph structure where sentences functioned as nodes, with edges denoting similarity between sentences. Connection creation between sentences relied on weights surpassing predefined thresholds. Subsequently, sentence ranking was achieved using the TextRank algorithm [13].

Noteworthy studies showcasing these methodologies include [14], [15]. Additionally, some researchers explored supplementary graph algorithms, such as A* introduced by Bahloul et al. [16], and extended these techniques to multi-document summarization [17], proposed by Abdulateef et al. Moreover, researchers leveraged Arabic WordNet to generate refined summaries, as evident in studies like [18], [19]

Researchers such as Elayeb et al. [20] explored a novel approach by employing analogical proportions in the summarization process. Conversely, Qaroush et al. [21] expanded sentence representation by incorporating semantic features, a departure from the previous reliance solely on statistical features. These innovative perspectives provide a valid and distinct view of the summarization problem.

B. TRANSFORMERS VS TRADITIONAL METHODS

1) HYBRID METHODS

Preceding the dominance of end-to-end learning approaches, there were endeavors to bridge deep and non-deep learning methods. Some researchers explored hybrid techniques by employing variational autoencoders to represent sentences in latent spaces, followed by a re-representation process.

This approach is exemplified in studies such as those conducted by Alami et al. [22], [23]. Additionally, other researchers adopted a more traditional approach by generating candidate summaries, later refining them using Long Short-Term Memory (LSTM) [24] networks, as proposed by Fadel et al. [25]. These hybrid methodologies aimed to combine the strengths of different techniques to enhance the summarization process.

The advent of the deep learning era has significantly transformed the landscape, markedly improving the existing pipelines. This period can be distinctly divided into two sub-eras within the research field. In the initial phase, Recurrent Neural Networks(RNN) [26], [27] and Long Short-Term Memory(LSTM) [24] models were the conventional choices in Natural Language Processing. In the current phase, the Transformer architecture has become the standard choice, marking a significant shift in the field.

2) END-TO-END

The emergence of end-to-end deep learning models shifted the focus towards leveraging datasets without the need for intricate feature engineering. This era predominantly employed LSTM [24] and RNN [26], [27] architectures, with variations in the number of layers and transitions from LSTM [24] to Bidirectional LSTM models [84]. Studies by Helmy et al. [28], Al-Maleh et al. [29], and

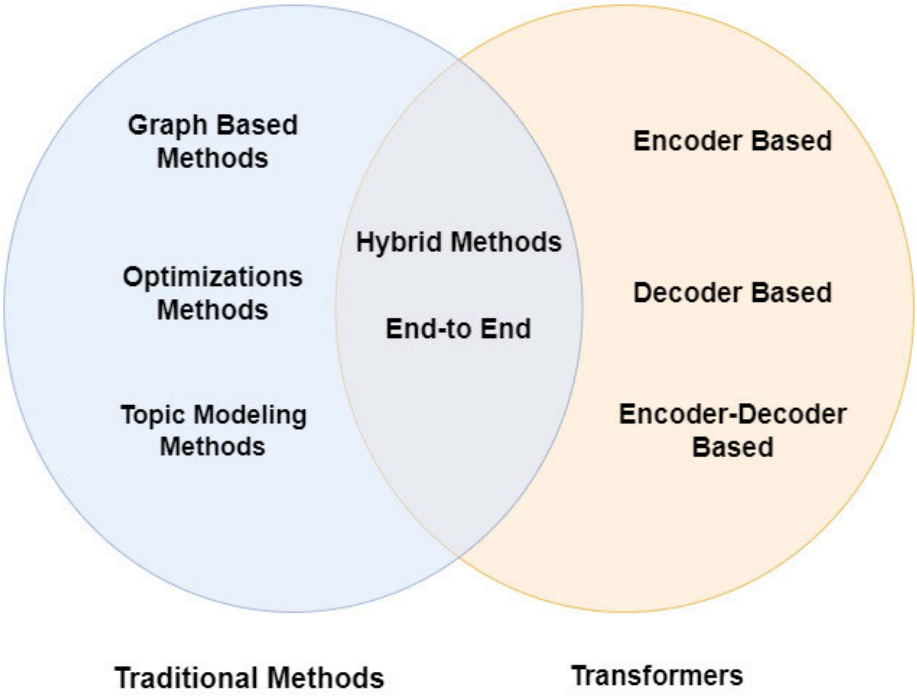

FIGURE 1. Taxonomy of the summarization methods.

Suleiman and Awajan [30] exemplify this trend. Additionally, Wazery et al. [31] explored the point-generator architecture proposed by See et al. [4], enabling a combination of extractive and abstractive summarization techniques. This phase underscored the significance of quality datasets in representing the summarization task.

C. TRANSFORMER-BASED APPROACHES

The rise of Transformers has led to categorizing tasks based on their community architecture usage. Traditionally, Transformers' encoders are primarily employed to understand tasks, making them well-suited for classification. Consequently, they can be used as a sentence classification methodology in extractive summarization.

Conversely, encoder-decoder architectures are employed for understanding and generation tasks, making them suitable for tasks requiring both functions. Surprisingly, even the decoder-only architecture, believed to be appropriate only for generation tasks, has found application in summarization tasks, challenging the traditional belief about its limitations [32].

1) ENCODER-BASED APPROACHES

Within the realm of encoder-based approaches, it is evident that models have been employed for sentence classification. This classification mandates an extractive summary according to the method's definition. Alternatively, some strategies involve leveraging decoder models to enhance the generated summary further, enabling a more abstractive form. This trend is observable in the works referenced next [33], [34], [35].

2) ENCODER-DECODER BASED

Encoder-decoder-based methods serve as the standard in sequence-to-sequence approaches, attracting the primary focus of the literature. This domain predominantly concentrates on two categories: (1)fine-tuning without pertaining and (2) pre-training and fine-tuning.

- **Fine-tuning**: Fine-tuning is the process of training a model on a smaller, task-specific dataset, as compared to the larger dataset used in pretraining (which will be discussed in the upcoming paragraphs). During fine-tuning, the goal is to adapt the model to perform a specific task, often referred to as a downstream task, such as summarization and translation.
  This Process is essential in sequence-to-sequence tasks and is crucial for achieving a well-performing model tailored to the specific task. The success of fine-tuning significantly relies on the dataset utilized, which is typically smaller than the dataset used during pre-training. Notably, methods such as [36], [37], and [38] have entirely developed new datasets tailored to their respective tasks, which were also utilized by upcoming models in this context.
  Recently, with the growing trend of augmenting parameter counts, even fine-tuning processes have become considerably costly. As a result, efforts have arisen to discover more cost-effective fine-tuning methods that

ensure a high-quality final model. In this context, Qin et al. [39] have extensively experimented with various parameter-efficient fine-tuning approaches available.

- **Pre-training and Fine-tunning:** Pre-training has become an essential methodology for achieving high-quality models in transformers. It involves training models on language modeling tasks to grasp the essence of language, a process that demands substantial computational resources, time, and large datasets. On the contrary, fine-tuning adapts pre-trained models to specific tasks like summarization, which is significantly less computationally demanding than pre-training but remains a crucial step in the process.
  Subsequently, numerous models have adopted the pre-training above and a fine-tuning pipeline. Each model employed distinct datasets for pre-training, as seen in studies such as [40], [41], [42], and [43]. After pre-training, these models utilized fine-tuning datasets provided by the methods in the previous part to evaluate their performance.

We didn't specifically highlight the utilized models in the previously discussed methods, primarily due to the lack of substantial changes. Most employed models, such as MBART or Mt5, are both variants of transformers with minor modifications. However, some researchers ventured into altering the architecture of the base model. Reference [44] attempted to modularize the T5 model, incorporating language-specific components. Moreover, efforts were made by [45] and [46] to extend the capabilities of T5 to handle longer sequences, which the base model didn't originally support.

3) DECODER-BASED APPROACHES

Recent advancements in generative Large Language Models (LLMs) have garnered significant attention, particularly due to their massive parameter counts and multilingual capabilities. For instance, Meta introduced three versions of the LLaMA model series [47], [48], [49]. The first version, LLaMA-1, supported only the English language. Subsequent iterations, including LLaMA-2 and LLaMA-3, extended support to additional languages, including Arabic. LLaMA-3 demonstrated substantial improvements in Arabic language tasks, as evaluated by Khondaker et al. [50]. Another significant contribution is BLOOM, introduced by Scao et al. [51], which includes a one-shot fine-tuning method for abstractive cross-lingual summarization. Although Arabic was included in their experiments, the authors did not report specific results for the language. Google has also made strides in this area with their PaLM model [52], evaluated on summarization tasks in a one-shot setting using the XLSum dataset. More recently, the Aya model, developed by Üstün et al. [53], focused on enhancing multi-linguality with an emphasis on LLM-based approaches.

Efforts to enhance the multilingual capabilities of LLMs have seen contributions from several researchers. Ji et al. [54] and Bhattacharjee et al. [55] proposed two distinct approaches to extend the LLaMA-2 model [48], with both reporting results on the XLSum dataset. Whitehouse et al. [56] conducted further experiments to improve multilingual summarization capabilities.

As mentioned earlier, the field of summarization literature can be categorized into two primary eras, with one further divided into distinct categories based on the methodologies employed.

The first era is characterized as the initial phase of summarization, marked by a scarcity of available data. During this period, researchers dedicated their efforts to exploring novel techniques, primarily concentrating on extractive summarization due to the limitations of the available methods. Given the task's nature, the challenge revolved around extracting essential sentences, which was essentially necessary sentence extraction. Consequently, much of the focus was on enhancing sentence feature representation and developing algorithms for ranking these sentences.

With the rise of neural networks and the introduction of recurrent neural networks presented by [26] and [27] and LSTM presented by [24], the research landscape found itself between evolving methods and expanding datasets. While datasets saw a significant increase, the evolution of methods in this period was relatively less dynamic compared to the previous era. Researchers often oscillated between LSTM [24] and RNN [26], [27], occasionally combining statistical sentence features with word embeddings.

In the recent era, the primary emphasis has been on datasets rather than methodological changes. Methods employed during this period have predominantly revolved around the Transformer architecture, with minor adjustments. Contributions in this phase have chiefly concentrated on two aspects: either initiating a model's training entirely anew on larger datasets of superior quality or fine-tuning a pre-existing model on similarly enhanced datasets.

As mentioned earlier, specific techniques used in the field are categorized as sentence classification methods, while others are seen as text generation methods. This duality has led to inconsistencies in the datasets employed and the evaluation metrics applied.

Considering the limitations inherent in transformer models, which heavily hinge on their training data, and the disparities observed in the evaluation metrics, we will introduce the evaluation criteria employed and further explore the available datasets in the following section III. These discrepancies are evident in Table 1.

Finally, to highlight the disparity, we selected one traditional method and one transformer-based method to illustrate the difference between them. Details of both methods can be found in the appendices.

IV. EVALUATION METRICS

In this section, we will explore two evaluation metrics, which will be used to evaluate the summary itself, like ROUGE and BertScore, and on the other hand, we will explore the intrinsic

TABLE 1. Text summarization techniques evaluated on arabic language datasets.

Category	Sub-Category	Method Used	Dataset Used	Evaluation Metric used	Year
Tradional Methods	Topic Modeling	El-Haj et al. [6]	DUC–2002	Rouge 1 F,P,R	2011
		Oufaida et al. [7]	EASC , TAC 2011	Rouge 1,2 F,P,R	2014
		Al-Sabahi et al. [8]	EASC , DUC–2002	Rouge 1,2 F	2018
	Optimization Methods	Al-Abdallah et al. [9]	EASC	Rouge 1,2 F,P,R	2017
		Al-Radaideh et al. [10]	EASC , KALIMAT	Rouge 1,2 F,P,R	2018
		Al-Abdallah et al. [11]	EASC	Rouge 1 F,P,R	2019
	Graph Methods	Elbarougy et al. [14]	EASC	F,P,R	2020
		AL-Khassawneh et al. [15]	EASC	Rouge 2 , F,P,R	2023
		Bahloul et al. [16]	EASC	Rouge 1,2,s,L F,P,R	2020
		Abdulateef et al. [17]	EASC	Rouge 1,2 F,P,R	2020
		Alami et al. [18]	EASC	F,P,R	2018
		Alami et al. [19]	EASC	Rouge 1 F,R,P	2021
	Misc	Elayeb et al. [20]	Arabic News Texts (ANT), EASC	Rouge 1, Bleu 1	2020
		Qaroush et al. [21]	EASC	Rouge 1,2 F,P,R	2021
Between Traditional Methods and Transformers	Hybrid	Alami et al. [22]	EASC and internal DS	ROUGE 1 R	2018
		Alami et al. [23]	EASC	Rouge F,R	2021
		Fadel et al. [25]	EASC and Al-khair news	Rouge 2 F,P,R	2020
	End-to-End	Helmy et al. [28]	AKEC	P,R,F	2018
		Al-Maleh et al. [29]	AHS	Rouge 1 P,R,F	2020
		Suleiman et al. [30]	newly generated dataset	Rouge1 , ROUGE1-NOORDER	2020
		Wazery et al. [31]	AHS , AMN	Rouge 1,2,L BLEU	2022
Transformers	Encoder	Reda et al. [33]	EASC	rouge 1, F,R,P	2022
		Elmadani et al. [34]	EASC, KALIMAT	Rouge 1,2,L F	2020
		Abu Nada et al. [35]	Internal Dataset	Rouge 1,2, R P F	2020
	Encoder-Decoder(fine-tunning)	Ladhak et al. [36]	Wikilingua	Rouge 1,2,L F	2020
		Hasan et al. [37]	XL-sum	Rouge 1,2,L F	2021
		Qin et al. [39]	XL-SUM	Rouge 2 F	2022
		Kahla et al. [38]	AraSum	Rouge 1,2,L F	2023
	Encoder-Decoder(Pre-training)	Ma et al. [40]	Wikilingua	Rouge 1,2,L F	2021
		Nagoudi et al. [41]	EASC , Wikilingua	Rouge 1,2,L F	2021
		Ghaddar et al. [42]	Wikilingua , EASC	Rouge 1,2,L F	2022
		Alghamdi et al. [43]	Wikilingua , EASC	Rouge 1,2,L F	2023
	Decoder	Scao et al. [51]	WikiLingua	Rouge 2 F	2022
		Nagoudi et al. [57]	Wikilingua	perplixity	2022
		Anil et al. [52]	WikiLingua , XL-SUM	SentencePiece Rouge 2 F	2023
		Gemini Team [58]	WikiLingua	bleurt score F	2023
		Gemini Team [58]	WikiLingua	bleurt score F	2023
		Khondaker et al. [50]	XL-SUM	Rouge-L	2024
		Üstün et al. [53]	XL-SUM	Rouge L Sum	2024
		Ji et al. [54]	XL-SUM	Rouge L	2024
		Lai et al. [59]	XL-SUM	Rouge L	2024
		Whitehouse et al. [56]	XL-SUM	Rouge L	2024

evaluation of the summarization dataset, like compression ratio and abstractivity.

A. SUMMARIZATION EVALUATION METRICS

This subsection will explore the evaluation metrics utilized in summarization research, such as ROUGE, BERTScore, METEOR, and BLEU.

Lin [60] introduced the Recall-Oriented Understudy for Gisting Evaluation (ROUGE), a set of metrics designed to assess the quality of a summary by comparing it to reference summaries crafted by humans. ROUGE includes several variants:

- **ROUGE-N:** Evaluates the overlap of n-grams between the generated summary and the reference summaries.
- **ROUGE-L:** Focuses on the longest common subsequence, capturing sentence-level structure and order.
- **ROUGE-W:** A weighted variant of ROUGE-L that emphasizes consecutive matches.
- **ROUGE-S:** Measures overlap of skip-bigrams, allowing gaps between words.

For instance, ROUGE-N is formally expressed in equation 1:

$$\text{ROUGE-N} = \frac{\sum_{\text{reference summaries}} \sum_{\text{n-grams}} \text{Count}_{\text{match}}(n\text{-gram})}{\sum_{\text{reference summaries}} \sum_{\text{n-grams}} \text{Count}(n\text{-gram})} \quad (1)$$

where $\text{Count}_{\text{match}}$ refers to the number of n-grams that overlap between the system and reference summaries, and Count denotes the total number of n-grams in the reference summaries.

ROUGE metrics have become widely adopted in summarization tasks due to their effectiveness and simplicity in quantifying content similarity and capturing key aspects of summary quality.

While ROUGE is not necessarily the most advanced or comprehensive evaluation metric, we report it primarily because of its widespread use in the literature, which allows for consistent comparison with existing Arabic summarization methods.

Additionally, unlike in English, no well-established neural evaluation models are available for the Arabic language, and existing multilingual models have not been validated for Arabic. This makes ROUGE the most practical and comparable choice for our setting.

Recently, following the emergence of the transformer model, a new metric called BERTScore has been introduced by Zhang et al. [61] as an alternative to the traditional ROUGE metric. BERT score involves obtaining contextual embeddings for both the generated and reference summaries and comparing their similarity. The previously mentioned approach has been found to better align with human judgment.

In the same way as BERTScore, Sellam et al. [62] introduced a new metric called BLEURTScore as an alternative

to the traditional ROUGE metric. This metric takes a pair of sentences as input—a reference and a candidate—and returns a score indicating the extent to which the candidate is fluent and conveys the meaning of the reference.

Although BLEURTScore and BERTScore provide nuanced evaluations and leverage deep learning models, they remain relatively new compared to ROUGE. As a result, they are less commonly used in practice. Furthermore, neural metrics like the BLEURTScore can be computationally expensive compared to traditional metrics, which may limit their widespread adoption.

Although not as frequently used as the metrics mentioned earlier, machine translation evaluation methods such as BLEU [63] and METEOR [64] have also been used to evaluate summarization.

B. INTRINSIC EVALUATION OF SUMMARIZATION DATASET

In this part, we will explore the intrinsic evaluation of summarization datasets, which mainly have been adopted from Bommasani and Cardie [65]; we will go one by one in the following.

1) COMPRESSION

Compression is measured at both the word (w) and sentence (s) levels as follows in equations 2 3:

$$\mathrm{CMP}_w(D_i, S_i) = 1 - \frac{|S_i|}{|D_i|} \tag{2}$$

$$\mathrm{CMP}_s(D_i, S_i) = 1 - \frac{\|S_i\|}{\|D_i\|} \tag{3}$$

Here, $|S_i|$ represents the number of words in the summary, and $|D_i|$ represents the number of words in the document. Similarly, $\|S_i\|$ and $\|D_i\|$ denote the number of sentences in the summary and document.

For sentence-level compression, an additional parameter specifies how sentences are segmented. In our case study, articles were segmented based on punctuation marks.

2) TOPIC SIMILARITY

A topic model M is trained on the training corpus T with k topics using LDA [66]. Topic similarity is quantified by comparing the inferred topic distributions $\theta_{D_i|M}$ and $\theta_{D_i|M}$ for a given document and summary, which can be viewed in equation 4:

$$\mathrm{TS}(D_i, S_i) = 1 - \mathrm{JS}(\theta_{D_i|M}, \theta_{S_i|M}) \tag{4}$$

where JS is the Jensen-Shannon distance. We set $k = 20$ and $T = D$.

3) ABSTRACTIVITY

Bommasani and Cardie [65] introduced the concept of fragments $F(D_i, S_i)$, which are greedily matched spans shared between D_i(document) and S_i (summary) which can be viewed in equation 5. Abstractivity is quantified as a normalized measure of the aggregate fragment length. This definition extends the earlier concept introduced by Grusky et al. [67].

$$\mathrm{ABS}_p(D_i, S_i) = 1 - \frac{\sum_{f \in F(D_i,S_i)} |f|^p}{|S_i|^p} \tag{5}$$

4) REDUNDANCY

Bommasani and Cardie [65] demonstrated that ROUGE [60] is effective for identifying redundancy due to its emphasis on overlapping spans. Redundancy is quantified as the average ROUGE-L F-score computed across all pairs of distinct sentences within the summary, which can be seen in equation 5.

$$\mathrm{RED}(S_i) = \mathrm{mean}_{(x,y)\in S_i \times S_i, x \neq y} \mathrm{ROUGE}(x, y) \tag{6}$$

5) SEMANTIC COHERENCE

The semantic coherence of multi-sentence summaries is evaluated by predicting the probability of each successive sentence conditioned on the previous one. This is achieved using BERT [68], a powerful language model pre-trained specifically for this objective, as seen in equation 7.

$$\mathrm{SC}(S_i) = \frac{1}{\|S_i\| - 1} \sum_{j=2}^{\|S_i\|} \mathrm{BERT}(S_i^j | S_i^{j-1}) \tag{7}$$

V. ARABIC DATASETS FOR TEXT SUMMARIZATION TASK

In this section, we have gathered comprehensive information on the datasets utilized in the existing literature[1,2]

A. ARABIC HEADLINES GENERATION

This subsection discusses datasets designed explicitly for Arabic headline generation, a task closely related to title generation. These datasets focus on producing concise and informative headlines that accurately reflect the core content of an article.

Arabic Headlines Summary(AHS) [29]: It is employed for the abstractive summarization of an individual document. The dataset was sourced from the Mawdoo3 website [69]. It has 300,000 texts. The first sentences of the introduction paragraph are considered the original material, but their titles function as the summary.

Arabic Mogalad_Ndeef (AMN) dataset [70] have merged between the following:

- Arabic News Dataset: (236,000 news articles) A mix of many Arabic datasets from diverse news items constituted this extensive Arabic News dataset, as referenced in [70].
- Saudi newspapers [70]: this dataset contains 31,030 Arabic newspaper articles, headlines, and extra metadata extracted from various online Saudi newspapers.

[1] You can access all of the datasets through this link.
[2] The repository is currently private until the publication is accepted.

B. TEXT SUMMARIZATION

This subsection presents key datasets available for Arabic text summarization, an essential area in Arabic NLP aimed at condensing longer texts while preserving their main ideas.

1) ESSEX ARABIC SUMMARIES CORPUS (EASC) [71]

The EASC is a compilation of Arabic language resources consisting of 153 articles in Arabic and 765 extracted summaries of those articles written by humans via Mechanical Turk [72].

2) KALIMAT [73]

The Multipurpose Arabic Corpus Dataset comprises 20,291 articles in the Arabic language, sourced from the Omani newspaper Alwatan. The dataset includes extractive summaries generated by single-document and multi-document systems, as well as articles that have been annotated with named entities. The articles are classified into six categories: culture, economy, local news, international news, religion, and sports.

The authors of **WikiLingua** [36] presented a novel multilingual dataset for abstractive summarization. The dataset was created by automatically collecting procedural steps from the WikiHow website and automatically transforming them into abstractive summaries. On average, the dataset includes 42,783 articles and their summaries in 18 languages. Specifically, it contains 29,229 article-summary pairs in Arabic. Many recent models have used this dataset to evaluate their performance in abstractive summarization.

XL-SUM [37] is a multilingual abstractive summarization dataset that covers 44 languages. The authors collected the data from various BBC news websites and their parallel translations. The dataset contains 40,327 articles in the Arabic language for abstractive summarization.

Bhattacharjee et al. [55] recently created the **CrossSum** dataset. This dataset is dedicated to cross-lingual summarization, comprising approximately 76,348 Arabic instances and corresponding summaries in various languages. The authors regard it as a significant cross-lingual summarization dataset due to its extensive coverage and diverse linguistic representations.

MassiveSumm [74] is a news summarization dataset created by scraping 370 news sources across 92 languages. It contains 31 million article-summary pairs in total, including 432,384 pairs in the Arabic language.

Alhamadani et al. [75] introduced a large-scale news summarization dataset called **LANS**. The collection comprises 8,443,484 articles and associated summaries from 22 publications across 19 Arab nations, covering 1999 to 2019. LANS provides a diverse range of Modern Standard Arabic (MSA) from the 19 Arab countries included in the dataset.

Tikhonov et al. [76] presented a **WikiMulti** dataset that was collected based on Wikipedia's "good article" concept. Good articles are articles that have been approved by the community and are considered to be of high quality. Each of these articles includes a summary in the first paragraph. The dataset provides article-summary pairs in 15 languages, including 7,476 Arabic articles. The authors noted that they excluded Egyptian Arabic articles from the dataset because they typically contain two or fewer paragraphs, which is unsuitable for the summarization task.

Indeed, there is a plethora of datasets available for the summarization task. Additionally, you may notice the inclusion of datasets for headline generation and key phrase generation in the list. This addition stems from prior studies treating these datasets as relevant to the summarization task. The datasets are primarily sourced from the news domain, which serves as the primary reservoir for summarization datasets.

However, with the advent of deep learning and the prominence of Transformer models, the approach to data collection has shifted. Transformer models, being data-intensive, prompted communities to amass extensive datasets rapidly, often without stringent quality checks. Consequently, this rush led to the creation of numerous models, some of which had poor quality.

While Transformer models possess remarkable capabilities, their effectiveness is limited by the quality and integrity of the data. There is no magic; a poorly curated or noisy dataset results in an inferior model. Because of the definition above, we will assess whether these datasets genuinely represent the summarization task. Towards the end, we will introduce a new version of these datasets that we believe accurately represents the requirements of a summarization task.

VI. DATASETS COMPARATIVE ANALYSIS

We have used the same methods explored in the intrinsic evaluation of the summarization dataset subsection. These methods align well with our goals. We plan to utilize all metrics outlined in this subsection except semantic coherence. These metrics will be applied following the parameters outlined in the section.

The mathematical notation utilized in our study is as follows: $|D_i|$,$|S_i|$ represent the average number of words in documents and sentences, respectively. Similarly, at the sentence level, we have$\|D_i\|$ and $\|S_i\|$ indicating compression rates for sentences and words. We denote sentence-level and word-level compression rates as CMP_s and CMP_w, respectively. Topic Similarity, Abstractivity, and Redundancy are denoted by TS, ABS_1, and RED.

We performed a comprehensive analysis, considering datasets such as EASC, Kalimat, WikiLingua, XL-Sum, and CrossSum, as well as the recently introduced WikiMulti. Notably, WikiMulti is distinctive for its focus on long text summarization. The outcomes of this analysis are detailed in Table 2.

The analysis used the training sets from Wikilingua, CrossSum, and XL-SUM. Notably, these datasets offer comprehensive training, validation, and test subsets. In contrast,

TABLE 2. Comparison of datasets based on intrinsic evaluation metrics: sentence/word compression ratio, topic similarity, abstractivity, and redundancy. The comparison includes EASC, Kalimat, WikiLingua, XLSum, CrossSum, and WikiMulti—all in Arabic. The table also reports the average number of words and sentences in articles and summaries.

	EASC	Kalimat	WikiLingua	XL-Sum	CrossSum	WikiMulti
# ex	1535	18256	20441	37519	37371	7476
avg $\|D_i\|$	381	515	343	428	422	824
avg $\|S_i\|$	122	178	29	25	25	139
avg $\|\|D_i\|\|$	29	15	33	39	38	79
avg $\|\|S_i\|\|$	10	3	6	3	3	14
CMP_w	0.66	0.43	0.86	0.9	0.9	-1.5813
CMP_S	0.61	0.54	0.72	0.86	0.86	-2.4778
TS	0.70	0.80	0.35	0.36	0.35	0.42
ABS_1	0.13	0.018	0.54	0.54	0.54	0.70
RED	0.23	0.16	0.28	0.03	0.03	0.28

EASC, Kalimat, and WikiMulti lack separate validation and test sets, or we could not locate them. Hence, our analysis was performed on the available data from all these sources.

EASC and Kalimat exhibit a favorable word compression ratio, CMP_w (0.43 and 0.66, respectively), making them promising choices for summarization datasets. In contrast, WikiLingua, XL-Sum, and Cross-Sum possess exceptionally high compression ratios (0.90, 0.90, and 0.86, respectively), as shown in Table 2.

These higher ratios align more closely with headline or small-summary generation tasks, which inherently omit significant details from the original article rather than traditional text summarization. WikiLingua and XL-Sum remain widely utilized within the summarization research community despite their high compression ratios.

It's worth noting that this observation correlates with our previous assertion that recent summarization datasets lack quality assurance processes. Unlike earlier datasets like EASC, which were meticulously curated manually, the newer datasets might not adhere to the conventional summarization standards.

Another noteworthy aspect of the compression ratio is its inherent connection to the sentence compression ratio, a generally expected correlation.

Unexpected findings arose in WikiMulti, available in table 2, where all compression ratios(word and sentence) were negative (-1.5813, -2.4778), indicating that the summaries were longer than the original articles. This outcome directly contradicts the fundamental definition of the summarization task and initially led us to question the reliability of our evaluation method. To address this, we conducted a comprehensive manual review.

Upon further analysis, we uncovered significant errors in the article-summary pairs within the dataset. Many entries lacked proper articles, with headlines mistakenly categorized as articles, while the corresponding summaries were disproportionately lengthy. This issue was observed in 2,806 out of 7,476 instances, accounting for approximately 37% of the dataset.

Additionally, it's worth noting that the sentence compression ratio was 1.5 times the word compression ratio. This discrepancy occurred because we occasionally considered a word to be a sentence, which was incorrect. This misclassification arose due to excessive punctuation, particularly commas and semicolons, which we used as separators during sentence segmentation.

Another crucial dimension that warrants exploration is the abstraction of each dataset. EASC and Kalimat, for instance, emerge as the least abstractive datasets, being entirely extractive. This aligns with the research methods employed during the initial era of Arabic summarization methods.

Conversely, at first glance, XL-SUM, CrossSum, and Wikilingua might appear more suited for abstractive text summarization. However, it's essential to note that the term abstractive can be deceptive. For example, if we have a 50% abstraction percentage within a summary of only 20 words, the overall abstraction might be disproportionately high. Hence, it's crucial not to be misled by numerical values alone.

While it might seem that Wikimulti is the most abstract summary dataset, this is not entirely accurate. In reality, Wikimulti consists of numerous short texts in the article section and exceptionally long sequences in the summary section. Consequently, its abstractiveness score appears high. To put it in perspective, imagine the situation in XL-SUM, but in a reversed manner.

Examining the redundancy value is crucial, as high redundancy contradicts the essence of summarization, which aims for concise, non-repetitive representations of more extended text sequences. You can find a correlation between the length of the summary and the redundancy because whenever you have a small summary, it's so hard to have repetitive words, unlike the longer one, when you have the opportunity to repeat some of them.

Topic similarity between the original article and its summary is a critical indicator of content fidelity. To quantitatively assess this, we employed the Jensen-Shannon distance, a widely used measure for comparing probability distributions. By extracting topic distributions from both the source articles and their generated summaries—using a topic modeling technique—we could evaluate how closely the summaries reflected the thematic content of the original texts. This comparison served as a proxy for testing whether the contextual meaning was preserved.

Our analysis revealed that as the Jensen-Shannon distance increased, indicating a greater divergence in topic similarity, the quality and coherence of the summaries tended to decrease. In other words, summaries that deviated significantly from the topic distribution of their corresponding articles were more likely to omit or misrepresent key information.

Thus, this metric functions as a numerical indicator of semantic alignment and a practical validation tool to ensure that the summarization process does not dilute or distort the original meaning. This reinforces the broader principle that high-quality summarization should maintain linguistic clarity and contextual integrity.

In conclusion, the datasets present various challenges, and none can be deemed the standard choice for summarization. Considering this, the idea of merging these datasets arises. If one dataset alone cannot suffice, the possibility of integrating them all to extract the best features becomes intriguing. This merging approach will be further discussed in the upcoming section.

VII. PROPOSED ARABIC TEXT SUMMARIZATION DATASET MUKHTASAR

Several crucial aspects need to be considered to enhance the summarization dataset. Firstly, the length of the articles being summarized is of paramount importance. We intend to encompass three distinct tasks: one for short articles, another for medium-length articles, and a third for long articles. These task divisions have been derived from the existing literature on transformer models.

For instance, text-to-text models are commonly designed for context lengths of 512, 1024, or even larger, although literature resources exceeding the 1024-word limit are scarce.

As per our definition, small articles consist of up to 330 words, medium-length articles range between 331 and 680 words, and large texts comprise more than 680 words. These specific word counts have been chosen, assuming the model's tokenizer will generate contexts matching the defined lengths.

Secondly, let's consider the compression rate, a factor often misunderstood. There's a common misconception that increasing the compression rate always results in better summarization. However, this notion is not entirely accurate. After conducting a detailed analysis, we established specific criteria for three distinct tasks. For long summarization, the compression ratio should ideally fall between 20% and 35% at most.

When generating short summaries, the compression ratio is recommended to be between 5% and 20%. Also, these ratios have been chosen empirically after many trials; we have found that less than 6% wouldn't be sufficient and more than 35% would be too much. This process has been visualized in the figure 2.

Additionally, we will filter out any samples in all datasets containing more than 5% English characters, as there is a significant presence of English samples, and to get a better quality, we made sure that the divergence (topic similarity)between the document and its summary can't be more than 0.6 which guarantees a better quality without sacrificing all of the data.

Therefore, we have developed a dataset on creating short summaries for short, medium, and long articles, and comprehensive summaries for short, medium, and long texts can be viewed in figure 2; we didn't generate any dataset for headline generations.

In our Pipeline, we didn't use any of the datasets mentioned except the latest largest ones, LANS and MassiveSum.

The newly generated Mukhtasar dataset statistics can be found in Table 3. To complete our dataset-related work, we have also generated the train-valid-test split so whoever is going to use should have a standard and to open the Arabic literature for a better comparison between methods, the data have been split into 80%,4%, and 16% train, valid and test respectively with the target not to have a large valid dataset so It could be easily used during training.

TABLE 3. Comparison of Mukhtasar dataset parts based on intrinsic evaluation metrics: sentence/word compression ratio, topic similarity, abstractivity, and redundancy. The average number of words and sentences in the articles and summaries is also reported.

Article length	Short	Medium	Long	Short	Medium	Long
Summary length	Short			Long		
# ex	1049100	645205	136918	1034195	71209	4796
avg $\|D_i\|$	200	458	909	193	420	853
avg $\|S_i\|$	27	45	74	63	105	211
avg $\|\|D_i\|\|$	17	39	79	12	35	69
avg $\|\|S_i\|\|$	3	5	7	4	9	17
CMP_w	0.86	0.89	0.91	0.73	0.74	0.75
CMP_S	0.76	0.86	0.89	0.61	0.71	0.73
TS	0.65	0.64	0.63	0.60	0.65	0.67
ABS_1	0.55	0.50	0.49	0.63	0.60	0.60
RED	0.05	0.118	0.19	0.086	0.24	0.31

We want to highlight that a part of this dataset has been used in previous research [77], and we are leveraging our research to publish all the details of this dataset.

VIII. MUKHTASAR DATASET GENERALIZATION EXPERIMENTS

We have a goal to replace all of the available datasets, so to replace all of them, you need to satisfy that the dataset can replace them, so that's the target of this subsection. In this part, we have restricted the dataset and the model to 1024 tokens only for some reasons

- All of the Available datasets don't contain any samples with tokens of more than 1024
- Most of the Encoder-Decoder Models that have been used in Arabic don't exceed 1024 tokens

We trained two distinct models, each designed for a specific summarization goal: one for generating short summaries and the other for long summaries. Accordingly, data selection was tailored to these objectives. The model aimed at producing short summaries is called ours-short, while the one targeting long summaries is ours-long.

A. TRAINING SETTINGS

1) TRAINING INFRASTRUCTURE

We consistently utilized a single machine node equipped with four V100 GPUs throughout our experiments. Employing a distributed data-parallel technique during training, we allocated a batch size of one per GPU. Our training procedure was executed using the Hugging Face Accelerate framework. We have used Adam with a learning rate equal to 2e-5 with a linear learning rate scheduler without warm-up steps. These parameters have been added to the table 4.

2) MODEL USED

We used a model provided by Meta, originally designed for translation but capable of supporting multiple languages,

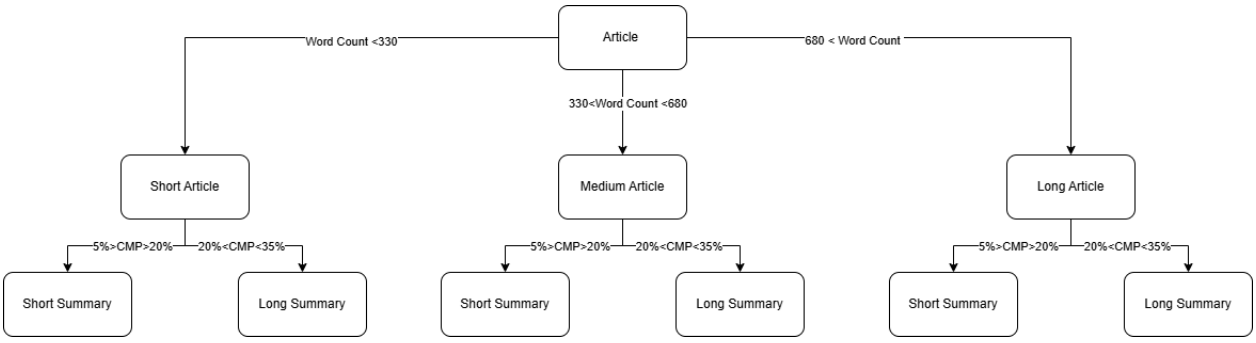

FIGURE 2. The data pipeline begins by creating the dataset based on the word count of the articles. Each article-summary pair is categorized into one of three groups: short, medium, or long articles. Subsequently, each group is further classified based on the compression ratio: 5% to 20% for short summaries and 20% to 35% for long summaries.

TABLE 4. Training parameters for mBART training. The same settings were used to train the model with short and long summarization targets.

Parameter	Value
Optimizer	Adam
Epochs	1
Num Machines	1
batch size per gpu	1
total batch size	1
num gpus	4
Warm-up	0
lr	2e-5

including Arabic. This multilingual model allowed us to show that while specialized Arabic models like AraBART might enhance performance, competitive results can still be achieved without using models specifically trained for Arabic.

3) DATASET USED

Since we have two distinct models, each trained on different subsets of the data, the model referred to as ours-short was trained on the medium article-short summary and short article-short summary portions of the Mukhtasar dataset; in contrast, the model called ours-long was trained on the medium article-long summary and short article-long summary portions.

All details about the dataset itself are discussed in the Mukhtasar Dataset Creation section.

This choice was made for two main reasons. First, to ensure comparability with current benchmarks, which do not include lengthy articles. Second, the model we selected has a 1024-token limit, requiring truncation for longer articles. In many summarization tasks, key summary content often relies on the concluding sections of the article, and truncation could omit this information, complicating practical training.

B. EVALUATION PROCESS

Our Evaluation Process has been split into two parts, one where we use the validation and test set of Mukhtasar Dataset, and the other to compare our model with the rest of the models, which can also help as a survey of the methods.

1) MUKHTASAR DATASET EVALUATION

In this part, we applied the same method using the validation and testing sets of the Mukhtasar Dataset (see 5).

Our analysis highlights a frequently overlooked issue: in some cases, the ''summary'' is so brief that it resembles a title rather than a meaningful summary. This phenomenon is observed when evaluating our models on short summary-short article and long summary-short article tasks, where both ours-short and ours-long produce very similar results.

However, as expected, our short score was higher than our long score for a medium article's short summary. Similarly, in the case of a medium article-long summary, ours-long outperforms ours-short, aligning with its intended objective.

This occurs because the original article in the short article parts of the datasets may be very short and not require summarization. Consequently, parts of the original short article appear in both the short and long summaries, which reduces the impact of summary length on ROUGE scores, leading to very similar results available in the table 5.

This outcome aligns with expectations, confirming that our dataset is meaningfully segmented, with each part serving a distinct purpose. It is also evident that medium-length articles, longer than short articles, yield higher scores when the model is trained to generate more extended summaries, especially in cases where the target is an extended summary. Conversely, for datasets with a short summary as the target, models trained with short summaries perform better.

2) COMPARISON WITH THE REST OF METHODS

This section reports our results on four of the most widely used datasets in Arabic summarization literature: EASC, Kalimat, WikiLingua, and XLSum. Additionally, we provide all the reported results on these datasets, offering a comprehensive reference for future researchers working on summarization.

We have also separated the LLM results on the available datasets to align with the latest LLM research trend.

Evaluation Settings: In this part, we did not train any extra steps. We just evaluated the validation and the test set (check

TABLE 5. Results of our two models on the Mukhtasar dataset, reported in terms of ROUGE-1, ROUGE-2, and ROUGE-L scores. We define three summarization tasks based on article length: short (< 331) words, medium (331 – –680), and long (> 680), aligning with common transformer context sizes. Compression ratios were also considered: 5%–20% for short and 20%–35% for long summaries, based on empirical evaluation.

Model / Summary Type Dataset	Short			Long		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
Short Summary -Short Article	39.0	22.0	31.0	39.8	22.6	32.0
Short Summary -Medium Article	39.0	22.0	31.0	32.0	13.0	21.0
Short Summary -Long Article	-	-	-	-	-	-
Long Summary -Short Article	41.0	25.0	35.0	42.0	24.9	35.0
Long Summary -Medium Article	28.0	12.0	20.0	31.0	11.0	18.0
Long Summary -Long Article	-	-	-	-	-	-

TABLE 6. Evaluation results of our models on the WikiLingua and XL-Sum datasets, reported regarding ROUGE-1, ROUGE-2, and ROUGE-L scores. The models were evaluated in a zero-shot setting, with no training samples from these datasets used during training. The evaluation includes both validation and test splits, with the scores shown here corresponding to the test split. For Kalimat and EASC, the label *(full)* indicates that the entire dataset was used for testing.

Model / Summary Type Dataset	Short			Long		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
Kalimat (full)	31.0	22.0	27.0	47.0	37.0	41.0
EASC (full)	58.0	46.0	55.0	50.0	40.0	47.0
XL-Sum (val)	34.9	18.0	29.0	35.0	18.5	29.0
XL-Sum (test)	35.0	18.0	29.0	35.0	18.0	29.0
WikiLingua (val)	30.0	18.0	29.0	29.0	18.0	28.0
WikiLingua (test)	30.0	18.0	29.0	29.0	18.0	28.0

table 6). In the case of EASC and Kalimat, we couldn't find a valid test set them, so we have used the training set for evaluation (check table 6), As previously highlighted in the introduction of this subsection, it is clear that the problem we are facing does not occur in these datasets. This is because these datasets primarily consist of short article sequences rather than long ones. As a result, some of their summaries are simply headlines, not complete summaries. Nevertheless, we used these datasets to ensure consistency with the existing literature. In the upcoming part, we will have a small section for each dataset and the reported results of the literature.

EASC is the most commonly used dataset for summarization, having served as the standard for many classical methods. It is also noticeably more straightforward than the other datasets we used, as reflected by the consistently high scores of various methods. Initially, we suspected an issue with our evaluation process due to the high scores, but this trend was confirmed across other methods.

This may be because this dataset has been generated to serve the traditional methods, which mostly involve extractive summarization, so there are a lot of common tokens between the source and target.

All methods reported in the literature are listed in Table 7. Notably, our two models achieve performance within the same range as the other methods despite not using this dataset during training. This underscores the generalization capability of our models and suggests robust performance across datasets.

Table 7 has been organized into six columns. The first column lists the model names (if available), followed by the second column, which includes the authors' names. The remaining columns display the ROUGE metrics, with the final column indicating the year of evaluation.

In Table 7, we present a summary of reported results across various methods. However, several issues warrant further discussion. Firstly, the evaluation setups differ significantly between studies. For instance, each research group uses distinct training and test splits, with some training traditional models applicable only to this dataset. Additionally, parameter configurations vary widely, with most methods tuned specifically to demonstrate the efficacy of their approach.

Although we followed similar practices to validate our results, it is clear that inconsistent experimental setups hinder reliable comparisons across methods. Therefore, we encourage future research to adopt our dataset as a benchmark, ensuring more standardized comparisons across studies. This consistency will help avoid inflated or arbitrary performance differences and provide a more precise measure of model effectiveness.

Kalimat: Kalimat has not been used a lot in comparison with EASC; we have found that it has been used only once in Al-Radaideh et al. [10], they have used an extractive summarization approach, and they have used a mix between experts' knowledge and ranking of sentences to use it, which's why they have fairly higher scores than us. Also, if you have checked the compression ratio mentioned in the dataset analysis, the summaries are considered long, which is why long models have better scores, as shown in Table 8

WikiLingua: This dataset has been widely utilized in the era of neural methods. Notably, two neural models have reported results on it. The first is AraT5-Big, with results from Ghaddar et al. [42]. The second is AraT5, evaluated twice: initially by its original authors [80] and subsequently by Ghaddar et al. [42]. The first study reported an exceptionally high ROUGE-1 score of 74, which is unusually high for this dataset.

TABLE 7. Reported results of methods that have used the EASC dataset over the years. For each method, the table includes the model name (if unique), author, ROUGE-1, ROUGE-2, ROUGE-L scores, and the publication year.

Name	Author	ROUGE-1	ROUGE-2	ROUGE-L	Year
–	El-Haj et al. [6]	80.0	74.0	_	2014
–	Oufaidaet et al [7]	_	_	_	2014
–	Al-Abdallah et al [9]	55.0	45.0	_	2017
–	Alami et al [18]	58.2	_	_	2018
–	Al-Sabahi et al. [8]				2018
–	Qasem et al. [10]	54.0	42.0	_	2018
–	Al Qassem et al. [78]	44.0	34.0	_	2019
–	Al-Abdallah et al. [11]	57.0	_	_	2019
–	Fadel et al. [25]	54.0	53.0	_	2020
ArA*	Bahloul et al. [16]	56.0	48.0	51.0	2020
–	Abdulateef et al. [17]	64.0	_	_	2020
–	Elayeb et al. [20]	74.0	_	_	2020
–	Elbarougy et al. [14]	34.0	_	_	2020
Distilbert	Alshanqiti et al. [79]	49.0	44.0	52.0	2021
–	Qaroush et al. [21]	64.0	61.0	_	2021
–	Alami et al. [19]	59.0	_	_	2021
–	Alami et al. [23]	_	41.0	_	2021
AratT5	Elmadany et al. [80]	60.0	48.0	60.0	2021
–	Tanfouri et al. [81]	45.5	_	_	2022
–	Reda et al. [33]	49.1	_	_	2022
SemGTs	Etaiwi et al. [82]	4.7	_	_	2022
At5	Ghaddar et al. [42]	12.6	3.5	11.3	2022
Arat5 (reevaluated)	Ghaddar et al. [42]	10.7	2.7	9.3	2022
–	AL-Khassawneh et al. [15]	61.7	_	_	2023
Aramus	Alghamdi et al. [43]	16.1	6.7	13.3	2023
ours-SHORT		58.0	46.0	55.0	2024
ours-LONG		50.0	40.0	47.0	2024

TABLE 8. Reported results of methods that have used the Kalimat dataset over the years in summarization. For each method, the table includes the model name (if unique), author, ROUGE-1, ROUGE-2, ROUGE-L scores, and the publication year.

Name	Author	ROUGE-1	ROUGE-2	ROUGE-L	Year
–	Al-Radaideh et al. [10]	52.0	40.0	_	2018
ours-SHORT		31.0	22.0	27.0	2024
ours-LONG		47.0	37.0	41.0	2024

TABLE 9. Reported results of methods that have used the WikiLingua dataset over the years, excluding the Large Language Models papers. For each method, the table includes the model name (if unique), author, ROUGE-1, ROUGE-2, ROUGE-L scores, and the publication year.

Name	Author	ROUGE-1	ROUGE-2	ROUGE-L	Year
At5B	Ghaddar et al. [42]	26.1	10.5	23.2	2014
Ara-t5	Elmadany et al. [80]	74.0	67.0	74.0	2021
Ara-t5 (reevaluated)	Ghaddar et al. [42]	25.1	10.5	22.5	2022
ours-SHORT		30.0	18.0	29.0	2024
ours-LONG		29.0	18.0	28.0	2024

This raised concerns, prompting Ghaddar et al. [42] to reevaluate the model on other datasets, such as EASC and Kalimat. We suspect that some evaluation settings might have been omitted or misrepresented in AraT5's summarization evaluation, as we are aware of how effective these models typically are in the Arabic language, check table 9

It is also worth noting that there are large language models (LLMs) that have reported their results on the same dataset (see Table 10). Wikilingua is considered the first multilingual summarization dataset that includes Arabic, making it a valuable resource for research. Additionally, the metrics reported for these LLMs represent an average across all languages. Furthermore, these models use BLEURTScore rather than ROUGE, which means their results are not directly comparable to ours.

TABLE 10. Reported results of methods that have used the WikiLingua dataset and Large Language Models. For each method, the table includes the model name (if unique), author, BLEURT Score, and the publication year.

Name	Author	BLEURT score	Year
PALM-2-L(3 shots)	Gemini Team [58]	50.4	2023
Gemini-pro(5 shots)	Gemini Team [58]	47.8	2023
Gemini-Ultra(5 shots)	Gemini Team [58]	48.9	2023

XLSum: This dataset is the most widely used benchmark for Arabic neural summarization models. It is also the least explored with traditional methods, highlighting a shift towards modern approaches. The trend of utilizing newly available large language models is evident here, with models like PaLM [52] and modular architectures such as [44],

representing a relatively new direction even within neural methods.

Our method achieved scores very similar to those reported by other methods, indicating that our dataset contains information comparable to that of XLSum, which can be viewed in table 11.

You can also examine the LLM results in Table 12. Part of these scores are averaged across the multilingual dataset XL-SUM, the most widely used in recent studies. It is important to note that some models utilize few-shot methods to achieve better scores.

On average, however, the scores of LLMs are typically lower than those of smaller, specialized models. This is because LLMs are general-purpose models designed for various tasks and not exclusively for summarization. Nonetheless, their performance could improve significantly if explicitly fine-tuned for the summarization task.

IX. CONCLUSION AND OPEN ISSUE

Arabic summarization research has made considerable strides in recent years, especially with the rise of neural approaches and multilingual datasets. However, several persistent issues continue to hinder progress. In this work, we identify core challenges facing the field and propose targeted contributions to address each, aiming to provide a roadmap for more consistent, rigorous, and impactful future research.

A. FINDINGS AND OUR CONTRIBUTIONS

1) **Lack of Standardized Evaluation Protocols:**
   Evaluations in Arabic summarization are often performed using ad hoc splits and settings, making results across studies difficult to compare.
   Our Contribution: We introduce six carefully designed evaluation splits, each aligned with a specific summarization objective. These splits offer a structured framework that encourages consistency and comparability across research efforts.
2) **Inconsistencies in Metric Usage:**
   The widespread but inconsistent use of ROUGE metrics (e.g., variations in reporting F-score, precision, or recall) undermines result reproducibility.
   Our Contribution: We advocate for the systematic use of ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-S, with standardized configurations, to enhance consistency and interpretability in performance reporting.
3) **Scarcity of Long-Context Summarization Data:**
   Arabic summarization has predominantly focused on short and medium-length texts, with limited resources for long-document summarization.
   Our Contribution: We present **Mukhtasar**, a new dataset explicitly designed to support short and long summaries. We demonstrate Mukhtasar's versatility and generalizability across summarization tasks by training and evaluating models for different summary lengths.
4) **Fragmentation in Dataset Adoption:**
   The community currently relies on various datasets (e.g., EASC, Kalimat, XLSum, WikiLingua), with no unified benchmarking tradition, causing fragmented progress.
   Our Contribution: Through comprehensive cross-evaluation, we establish Mukhtasar as a viable benchmark and highlight the trends in dataset usage, particularly the community's shift from traditional datasets to neural-friendly ones like XLSum and WikiLingua.
5) **Lack of Neural-Based Evaluation Metrics:**
   Despite the dominance of neural summarization methods, evaluation relies heavily on lexical overlap metrics. No neural-based or semantic evaluation metrics (e.g., BERTScore, BLEURT) have been developed or adapted specifically for Arabic summarization.
   Our Contribution: We highlight this gap and call for future research into developing neural-based evaluation metrics tailored for Arabic, which could align more closely with human judgment and semantic accuracy.

B. LOOKING AHEAD

Our work lays the groundwork for more reproducible, interpretable, and forward-looking Arabic summarization research. Key future directions include:

- Development of neural evaluation metrics for Arabic.
- Expansion of datasets targeting long-context summarization.
- Adoption of the broader community's proposed evaluation splits and metrics.
- Deeper exploration into domain-specific and cross-lingual summarization.

We hope our contributions—Mukhtasar, the evaluation framework, and this analytical synthesis—catalyze advancing Arabic summarization toward greater rigor, inclusivity, and innovation.

APPENDIX
TEXT SUMMARIZATION METHODS

In this section, we select one traditional approach and one transformer-based approach to highlight the contrast between classical methods and modern neural techniques, particularly those based on transformers. The chosen traditional method is the one proposed by Qaroush et al. [21], while the selected transformer-based method is the one introduced by Nagoudi et al. [41]. These methods have been carefully chosen to illustrate the differences in their underlying mechanisms and performance clearly.

C. TRADITIONAL METHOD

We present the traditional approach proposed by Qaroush et al. [21]. This method has three main stages: preprocessing, feature extraction, and modeling. Each of these stages will be described in the following subsections.

TABLE 11. Reported results of methods that have used the XL-Sum dataset over the years, excluding the Large Language Models papers. For each method, the table includes the model name (if unique), author, ROUGE-1, ROUGE-2, ROUGE-L scores, and the publication year.

Paper	Author	ROUGE-1	ROUGE-2	ROUGE-L	Year
MLongt5	Uthus et al. [45]	34.91	14.79	29.16	2023
Modular Mt5	Pfeiffer et al. [44]	37.0	17.0	29.0	2023
Mt5-peft-methods	Qin et al. [39]	35.36	15.11	29.49	2022
mBart-50	Kahla et al. [38]	23.0	6.0	16.0	2022
Arabart	Eddine et al. [83]	34.5	14.0	30.5	2022
Mt5-xlsum	Hasan et al. [37]	33.23	13.74	27.84	2021
ours-SHORT		35.0	18.0	29.0	2024
ours-LONG		35.0	18.0	29.0	2024

TABLE 12. Reported results of methods that used the XL-Sum dataset and Large Language Models. For each method, the table includes the model name (if specified), author, ROUGE-2, ROUGE-L, and ROUGE-Lsum scores, along with the publication year.

Name	Author	ROUGE-2	ROUGE-L	ROUGE-Lsum	Year
Palm(one shot)	Google Team [52]	14.40	_	_	2023
Palm 2 large(one shot)	Google Team [52]	25.00	_	_	2023
LLaMA-3-70B	Khondaker et al. [50]	_	18.83	_	2024
Aya-23B	Üstün et al. [53]	_	_	23.20	2024
EMMA-500 Llama 2 7B	ji et al. [54]	_	(avg)9.00	_	2024
xLLMs-100	Lai et al. [59]	_	(avg)13.57	_	2024
PaLM 2-XXS-LoRA-rank-4	Whitehouse et al. [56]	_	(avg)21.13	_	2024

1) PRE-PROCESSING

We begin with the pre-processing stage, a critical step in most traditional methods. Since these approaches often rely on statistical features, pre-processing aims to minimize data noise to enhance the quality and reliability of the extracted information.

- **Tokenization:** The pre-processing phase begins with tokenization, where the input text is broken down into smaller units at multiple structural levels. Initially, the document is divided into paragraphs using the newline character (`\n`) as a delimiter. Each paragraph is then segmented into individual sentences based on punctuation such as periods (.), question marks (?), and exclamation marks (!). These sentences are further divided into tokens using delimiters like whitespace, commas, semicolons, and quotation marks. The approach also employs morphological segmentation based on punctuation to capture the text's structure better.
- **Letter Normalization:** In Arabic, some letters have variant forms, and certain characters are used interchangeably due to visual similarity. Additionally, the presence of diacritics introduces multiple forms of the same word. A normalization step is applied to reduce this variability to convert these different forms into a consistent representation, ensuring that equivalent characters are treated uniformly throughout the text.
- **Stop-word Removal:** Commonly occurring functional words—such as prepositions, conjunctions, and pronouns—often do not contribute significantly to the semantic content of a sentence. These stop words are removed from the text to prevent them from influencing statistical analyses and feature extraction. Their elimination is significant in preserving the meaningful information within the text.
- **Stemming:** Arabic words frequently undergo inflection and derivation, resulting in multiple surface forms with the same root meaning. This morphological complexity can hinder various natural language processing tasks, such as text classification or similarity measurement. Stemming helps address this by removing affixes by reducing words to their base form—typically a root or stem. This unification simplifies linguistic variation and enhances the effectiveness of the model.

An example of the pre-processing pipeline can be seen in Figure 3, adapted from the original work by Qaroush et al. [21]. It is important to note that many of the resulting tokens produced by this pipeline may no longer carry meaningful semantics in Arabic. Instead, they serve as symbolic representations whose significance is derived from their frequency and patterns, as will be explained later.

2) FEATURE EXTRACTION

Following the pre-processing phase, a set of handcrafted features is extracted to characterize each sentence in the document. These features capture various linguistic, structural, and semantic properties contributing to sentence importance. Below is a summary of the features used in the method proposed by Qaroush et al. [21]:

- **Key-Phrases:** A set of salient terms that concisely represent the main topics of a document, enhancing summary quality by ensuring the inclusion of diverse and representative concepts. These phrases help capture the core themes while maintaining broad coverage of the document's content.
- **Sentence Location:** The positional context of a sentence within a paragraph or document, which capitalizes on the tendency of important information to appear in certain positions, such as the first or last sentences

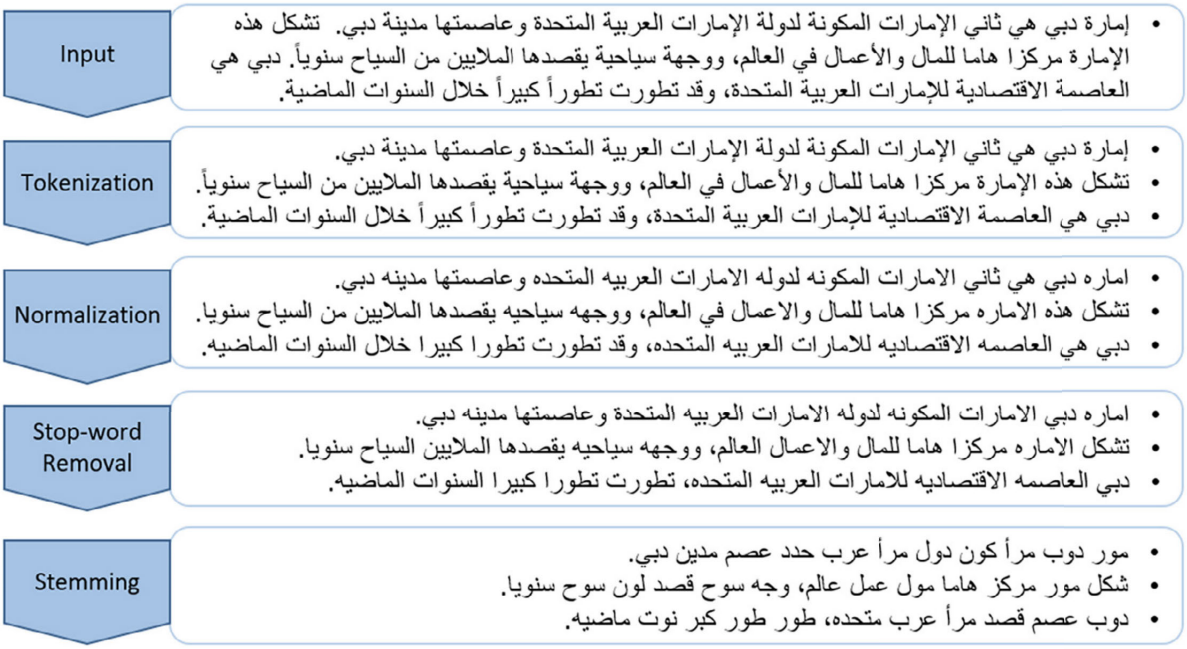

FIGURE 3. An example illustrating the effect of pre-processing, figure taken from [21].

of a paragraph. This feature helps identify structurally significant content.

- **Similarity with Title:** The lexical or semantic overlap between a sentence and the document title, identifying sentences closely aligned with the main theme. Sentences that share keywords or concepts with the title are often more central to the document's primary subject.
- **Sentence Centrality:** The degree of semantic similarity between a sentence and other sentences in the document, promoting those that are central to the overall meaning. This feature improves coverage by selecting sentences representing the document's key ideas.
- **Sentence Length:** The number of words or tokens in a sentence, used to filter out overly short (insignificant) or long (potentially verbose) sentences. This ensures that the summary contains concise yet meaningful information.
- **Cue Words:** Discourse markers (e.g., ''therefore,'' ''in conclusion'') or emphasis phrases that flag sentences likely to contain summaries, conclusions, or key arguments. These words act as natural indicators of important content.
- **Positive Keywords:** Terms that emphasize importance (e.g., ''significant,'' ''notably''), prioritizing sentences with strong affirmative or focal language. These keywords help highlight impactful statements within the text.
- **Sentence Inclusion of Numerical Data:** numbers, statistics, or quantifiable information favors fact-rich sentences, often found in scientific or technical summaries. Numerical data typically signals concrete and noteworthy details.
- **Occurrence of Non-essential Information:** Explanatory phrases (e.g., ''for example,'' ''such as'') that signal supplementary content, allowing the system to filter out tangential or elaborative sentences. This improves conciseness by focusing on core information.

Certainly, the feature generation phase is the most challenging and where much of the engineering effort lies. If a feature does not contribute to the final objective, it becomes meaningless and redundant. In such cases, including it may even reduce the model's performance, though this is not always guaranteed.

3) MODELING

This paper uses two methods: a score-based approach and a binary classification approach, both generating different results.

In the score-based method, sentences are ranked based on a weighted linear sum of their feature scores. The features are assigned a one-weight, with important features like sentence location, key-phrases, and title similarity receiving higher weights. The top-ranked sentences are selected for the summary, maintaining their original order to ensure coherence.

The summarization is framed as a binary classification problem in the binary classification approach. Each sentence is represented as a feature vector, and a binary classifier is trained on a dataset of documents with known summaries. The classifier predicts which sentences should be included in the summary, and the final summary consists of the sentences predicted as relevant.

To clarify, an input sample is shown in 4. Additionally, you can refer to 6 for the score-based method, and 5 for the binary classification approach.

D. TRANSFORMER BASED METHODS

In transformer-based approaches, many components of traditional NLP pipelines have become obsolete. Previously, the standard pipeline consisted of three main stages: pre-processing, feature extraction, and modeling. However, in modern transformer-based systems, pre-processing and essential steps such as tokenization have become optional. Manual feature extraction has been entirely replaced by automated representation learning. As a result, modeling has emerged as the main focus, significantly improving performance.

This shift in focus has made NLP research increasingly data-centric. A clear example of this can be seen in the work of Nagoudi et al. [41], where the emphasis is placed more on data design than on architectural innovation. Their methodology is essentially divided into two main components: pre-processing and modeling.

1) PRE-PROCESSING

- **Text normalization:** This step includes removing diacritics, URLs (replaced with <URL>), user mentions (replaced with <USER>), HTML tags, elongation, hash signs, and reducing repetitions of characters, emojis, and emoticons.
- **Tokenization:** They utilize SentencePiece to tokenize the text into 110K WordPieces. The vocabulary is constructed from a multilingual corpus comprising 70 million MSA sentences, 200 million Arabic tweets, 15 million English Wikipedia sentences, and 5 million sentences from 10 other languages.

Even within the pre-processing stage, the design reflects a data-centric philosophy. The tokenizer itself is trained on a carefully curated multilingual dataset, emphasizing the importance of data composition in achieving broader modeling goals.

2) MODELING

The modeling phase is divided into two main components: pretraining and fine-tuning, as outlined in the methods section.

- **Pretraining:** Pretraining involves leveraging a large and diverse corpus to learn the general structure and semantics of a language. In this work, the authors constructed a 248GB corpus (comprising 29 billion tokens) that combines 70GB of Modern Standard Arabic (MSA) text with 178GB of Arabic Twitter data. MSA sources include well-known datasets such as AraNews [85], El-Khair [86], OSCAR [87], and Arabic Wikipedia. The Twitter data was filtered to ensure a minimum level of Arabic content and analyzed for dialectal diversity. The resulting distribution was approximately 72% MSA and 28% dialect, with the dialectal content covering a broad geographic range. Additionally, around 4% of tweets exhibited natural code-switching with foreign languages like English and French. This rich, linguistically diverse corpus serves as a robust foundation for training Arabic language models from scratch.
- **Fine-tuning:** Following pretraining, the model undergoes fine-tuning to adapt it to specific downstream tasks. To assess the performance of their pretrained models, the authors introduced ARGEN, a benchmark suite covering seven Arabic generation tasks, including machine translation, code-switched translation, summarization, question generation, news title generation, transliteration, and paraphrasing. For summarization, a dedicated sub-benchmark called ARGENTS was developed, leveraging datasets such as EASC and WikiLingua, both previously described in the dataset section.

  Notably, the modeling phase in modern transformer-based architectures requires minimal manual intervention. Once the data is prepared, it can be directly fed into the model, which handles learning automatically. This reflects the current trend in NLP pipelines, where transformers are the model itself that handles the standard and much of the complexity.

Transformers have become the preferred choice in modern NLP when discussing model architecture due to their ability to handle complex language tasks effectively. The primary reason for their success lies in their capacity to process and understand contextual information in a nuanced manner. At its core is the multi-head attention mechanism, which allows the model to simultaneously evaluate different parts of the input sequence. This parallel processing enables the Transformer to capture various relationships and dependencies across the data, leading to a more comprehensive understanding of context.

Another key feature is the self-attention mechanism, which allows the model to assign different levels of importance to each word based on its relationship with other words in the sequence. This ability to process the entire context of a sentence or document, regardless of the distance between words, ensures that the model captures deep semantic meaning. Unlike older architectures like RNNs [26], [27] or LSTMs [24], which process sequences step-by-step, Transformers handle the entire input at once. This results in better performance, especially on tasks that require sophisticated understanding, such as translation, summarization, and question answering.

What makes the Transformer architecture particularly appealing is its language-agnostic nature. Since the core component of the model relies on tokens, the architecture doesn't require any special modifications for different languages. Arabic, for example, doesn't need any specific adjustments; the model processes Arabic text using the same tokenization and attention mecha-

إمارة دبي هي ثاني الإمارات المكونة لدولة الإمارات العربية المتحدة وعاصمتها مدينة دبي. تشكل هذه الإمارة مركزا هاما للمال والأعمال في العالم، ووجهة سياحية يقصدها الملايين من السياح سنوياً. دبي هي العاصمة الاقتصادية للإمارات العربية المتحدة، وقد تطورت تطوراً كبيراً خلال السنوات الماضية. الاقتصاد الحر والنشط في الإمارة وعدم وجود نظام ضريبي لعب دورا كبيرا في جذب المستثمرين من جميع أنحاء العالم.
وتقع إمارة دبي بين إمارتي أبو ظبي والشارقة. وأهل إمارة دبي ينحدرون من قبائل عربية متنوعة، على رأسها قبيلة آل بو فلاسه التي تنحدر منها أسرة آل مكتوم الحاكمة. وتقطنها قبائل بني كعب وآل بو فلاح وآل بو مهير والسودان والشوامس والبلوش والمناصير والرميثات والشحوح وغيرهم. وبها عوائل كثيرة من أصول أفريقية وفارسية. ودين أهالي دبي هو الإسلام على نهج أهل السنة والجماعة، والمذهب الرسمي في دبي هو المذهب المالكي.
آل مكتوم هم حكام دبي. وهم من آل بو فلاسه من بني ياس. حاكمها الأن هو الشيخ محمد بن راشد آل مكتوم. وهو أيضا نائب لرئيس الدولة ورئيس لمجلس الوزراء في الحكومة الاتحادية. ونائبيه في الحكم هما: شقيقه الشيخ حمدان بن راشد آل مكتوم وزير المالية والصناعة والشيخ مكتوم بن محمد بن راشد آل مكتوم. بينما يتولى منصب ولاية العهد بالإمارة الشيخ حمدان بن محمد بن راشد آل مكتوم رئيس المجلس التنفيذي للإمارة.
يرأس المجلس التنفيذي لحكومة دبي الشيخ حمدان بن محمد بن راشد آل مكتوم. ويجمع هذا المجلس في عضويته جميع مدراء الدوائر في حكومة دبي حيث يعقدون اجتماعاتهم الدورية لتسيير شؤون الإمارة.

The Emirate of Dubai is the second Emirate of the United Arab Emirates and Dubai its capital. This Emirate is an important financial and business center in the world and a tourist destination for millions of tourists annually. Dubai is the economic capital of the United Arab Emirates and has developed significantly over the past years. The free and active economy of this emirate and the absence of a tax system played a major role in attracting investors from all over the world.
Emirate of Dubai is located between Abu Dhabi and Sharjah. The people of the emirate of Dubai come from various Arab tribes, led by the tribe of the Al-Felisha, which descends from the ruling Al-Actium family. The tribes of Bani Ka'ab, Al Bo Falah, Al Bu Muheir, Sudan, Shawams, Baloch, Manasir, Rumaythat, Al Shahouh and others, inhabit it. With many families of African and Persian origin. The religion of the people of Dubai is Islam on the approach of the Sunnis and the community, and the official doctrine in Dubai is the Maliki school.
Al Actium is the rulers of Dubai. They are from Al Bu Falsa from Bani Yes. The ruler now is Sheikh Mohammed bin Rashid Al Actium. He is also Vice President and Prime Minister of the Federal Government. His two deputies in the government are his brother Sheikh Hamdan bin Rashid Al Actium, Minister of Finance and Industry and Sheikh Actium bin Mohammed bin Rashid Al Actium. While Sheikh Hamdan bin Mohammed bin Rashid Al Actium, the Chairman of the Executive Council of the Emirate, is the Crown Prince. The Dubai Executive Council is headed by Sheikh Hamdan bin Mohammed bin Rashid Al Actium. This council brings together all the directors of the departments in the Government of Dubai, where they hold regular meetings to manage the affairs of the Emirate.

FIGURE 4. Sample input document along with its English translation. Figure taken from [21].

No	إمارة دبي هي ثاني الإمارات المكونة لدولة الإمارات العربية المتحدة وعاصمتها مدينة دبي.
Yes	تشكل هذه الإمارة مركزا هاما للمال والأعمال في العالم، ووجهة سياحية يقصدها الملايين من السياح سنوياً.
No	دبي هي العاصمة الاقتصادية للإمارات العربية المتحدة، وقد تطورت تطوراً كبيراً خلال السنوات الماضية.
Yes	الاقتصاد الحر والنشط في الإمارة وعدم وجود نظام ضريبي لعب دورا كبيرا في جذب المستثمرين من جميع أنحاء العالم.
No	وتقع إمارة دبي بين إمارتي أبو ظبي والشارقة.
No	وأهل إمارة دبي ينحدرون من قبائل عربية متنوعة، على رأسها قبيلة آل بو فلاسه التي تنحدر منها أسرة آل مكتوم الحاكمة.
Yes	وتقطنها قبائل بني كعب وآل بو فلاح وآل بو مهير والسودان والشوامس والبلوش والمناصير والرميثات والشحوح وغيرهم.
Yes	وبها عوائل كثيرة من أصول أفريقية وفارسية.
Yes	ودين أهالي دبي هو الإسلام على نهج أهل السنة والجماعة، والمذهب الرسمي في دبي هو المذهب المالكي.
Yes	آل مكتوم هم حكام دبي.
Yes	وهم من آل بو فلاسه من بني ياس.
Yes	ونائبيه في الحكم هما: شقيقه الشيخ حمدان بن راشد آل مكتوم وزير المالية والصناعة والشيخ مكتوم بن محمد بن راشد آل مكتوم.
Yes	بينما يتولى منصب ولاية العهد بالإمارة الشيخ حمدان بن محمد بن راشد آل مكتوم رئيس المجلس التنفيذي للإمارة.
Yes	يرأس المجلس التنفيذي لحكومة دبي الشيخ حمدان بن محمد بن راشد آل مكتوم.
No	ويجمع هذا المجلس في عضويته جميع مدراء الدوائر في حكومة دبي حيث يعقدون اجتماعاتهم الدورية لتسيير شؤون الإمارة.

FIGURE 5. Predicted labels generated by the trained machine learning model. Figure taken from [21].

nisms applied to other languages. This universality makes Transformers highly effective across various linguistic contexts.

In summary, while high-quality data is the cornerstone of any successful model, the Transformer architecture enables models to leverage that data fully. Its capacity to capture

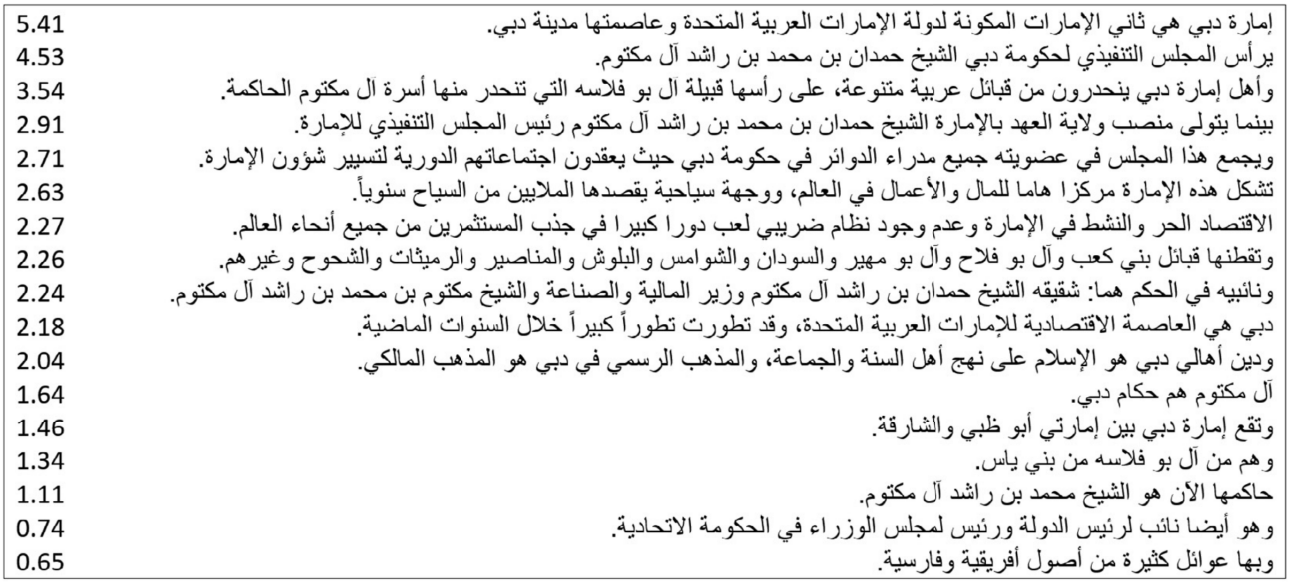

FIGURE 6. Sentence ranking based on scoring Model. Figure taken from [21].

and process contextual relationships through mechanisms like multi-head attention and self-attention has made it the architecture of choice for state-of-the-art NLP tasks. This is why Transformers are widely regarded as the standard in modern NLP.

3) EXTRA NOTES

In recent research, it has become a standard practice to include the parameters used for model training to enhance transparency and facilitate reproducibility. This allows others to replicate the experimental results more accurately. For example, in [41], the authors specify the following parameters for their three pre-trained models: a learning rate of 0.01, a batch size of 128 sequences, and a maximum sequence length 512.

REFERENCES

[1] N. Y. Habash, *Introduction to Arabic Natural Language Processing*. San Rafael, CA, USA: Morgan & Claypool, 2010.

[2] W. Zaghouani, "Critical survey of the freely available Arabic corpora," 2017, *arXiv:1702.07835*.

[3] H. Bouamor, N. Habash, M. Salameh, W. Zaghouani, O. Rambow, D. Abdulrahim, O. Obeid, S. Khalifa, F. Eryani, A. Erdmann, and K. Oflazer, "The madar Arabic dialect corpus and lexicon," in *Proc. 11th Int. Conf. Language Resour. Eval. (LREC)*, May 2018, pp. 3387–3396.

[4] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," 2017, *arXiv:1704.04368*.

[5] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," 2019, *arXiv:1908.08345*.

[6] M. El-Haj, U. Kruschwitz, and C. Fox, "Multi-document Arabic text summarisation," in *Proc. 3rd Comput. Sci. Electron. Eng. Conf. (CEEC)*, Jul. 2011, pp. 40–44.

[7] H. Oufaida, O. Nouali, and P. Blache, "Minimum redundancy and maximum relevance for single and multi-document Arabic text summarization," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 26, no. 4, pp. 450–461, Dec. 2014.

[8] K. Al-Sabahi, Z. Zhang, J. Long, and K. Alwesabi, "An enhanced latent semantic analysis approach for Arabic document summarization," *Arabian J. Sci. Eng.*, vol. 43, no. 12, pp. 8079–8094, Dec. 2018.

[9] 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.

[10] Q. A. Al-Radaideh and D. Q. Bataineh, "A hybrid approach for Arabic text summarization using domain knowledge and genetic algorithms," *Cognit. Comput.*, vol. 10, no. 4, pp. 651–669, Aug. 2018.

[11] R. Z. Al-Abdallah and A. T. Al-Taani, "Arabic text summarization using firefly algorithm," in *Proc. Amity Int. Conf. Artif. Intell. (AICAI)*, Feb. 2019, pp. 61–65.

[12] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bring order to the Web," Stanford Univ., Stanford, CA, USA, Tech. Rep., 1998.

[13] R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in *Proc. Conf. Empirical Methods Natural Language Process.*, Jul. 2004, pp. 404–411.

[14] 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.

[15] Y. A. Al-Khassawneh and E. S. Hanandeh, "Extractive Arabic text summarization-graph-based approach," *Electronics*, vol. 12, no. 2, p. 437, Jan. 2023.

[16] B. Bahloul, H. Aliane, and M. Benmohammed, "ArA*summarizer: An Arabic text summarization system based on subtopic segmentation and using an A* algorithm for reduction," *Expert Syst.*, vol. 37, no. 2, Apr. 2020, Art. no. e12476.

[17] 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.

[18] N. Alami, Y. E. Adlouni, N. En-Nahnahi, and M. Meknassi, "Using statistical and semantic analysis for Arabic text summarization," in *Proc. Int. Conf. Inf. Technol. Commun. Syst.*, Dec. 2017, pp. 35–50.

[19] N. Alami, M. E. Mallahi, H. Amakdouf, and H. Qjidaa, "Hybrid method for text summarization based on statistical and semantic treatment," *Multimedia Tools Appl.*, vol. 80, no. 13, pp. 19567–19600, May 2021.

[20] 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.

[21] 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.

[22] N. Alami, N. En-Nahnahi, S. A. Ouatik, and M. Meknassi, "Using unsupervised deep learning for automatic summarization of Arabic documents," *Arabian J. Sci. Eng.*, vol. 43, no. 12, pp. 7803–7815, Dec. 2018.

[23] 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.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[25] A. Fadel and G. B. Esmer, "A hybrid long Arabic text summarization system based on integrated approach between abstractive and extractive," in *Proc. 6th Int. Conf. Comput. Technol. Appl.*, Apr. 2020, pp. 109–114.

[26] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[27] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[28] M. Helmy, R. M. Vigneshram, G. Serra, and C. Tasso, "Applying deep learning for Arabic keyphrase extraction," *Proc. Comput. Sci.*, vol. 142, pp. 254–261, Jan. 2018.

[29] M. Al-Maleh and S. Desouki, "Arabic text summarization using deep learning approach," *J. Big Data*, vol. 7, no. 1, pp. 1–17, Dec. 2020.

[30] D. Suleiman and A. Awajan, "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. 1–10, 2020.

[31] 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.

[32] T. Goyal, J. Jessy Li, and G. Durrett, "News summarization and evaluation in the era of GPT-3," 2022, *arXiv:2209.12356*.

[33] A. Reda, N. Salah, J. Adel, M. Ehab, I. Ahmed, M. Magdy, G. Khoriba, and E. H. Mohamed, "A hybrid Arabic text summarization approach based on transformers," in *Proc. 2nd Int. Mobile, Intell., Ubiquitous Comput. Conf. (MIUCC)*, May 2022, pp. 56–62.

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

[35] 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," Tech. Rep., 2020.

[36] F. Ladhak, E. Durmus, C. Cardie, and K. McKeown, "WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization," 2020, *arXiv:2010.03093*.

[37] T. Hasan, A. Bhattacharjee, M. Saiful Islam, K. Samin, Y.-F. Li, Y.-B. Kang, M. Sohel Rahman, and R. Shahriyar, "XL-sum: Large-scale multilingual abstractive summarization for 44 languages," 2021, *arXiv:2106.13822*.

[38] M. Kahla, A. Novák, and Z. G. Yang, "Fine-tuning and multilingual pre-training for abstractive summarization task for the Arabic language," *Ann. Math. Et Inf.*, vol. 57, pp. 24–35, Feb. 2023.

[39] Y. Qin, G. Neubig, and P. Liu, "Searching for effective multilingual fine-tuning methods: A case study in summarization," 2022, *arXiv:2212.05740*.

[40] S. Ma, L. Dong, S. Huang, D. Zhang, A. Muzio, S. Singhal, H. H. Awadalla, X. Song, and F. Wei, "DeltaLM: Encoder–decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders," 2021, *arXiv:2106.13736*.

[41] E. Moatez Billah Nagoudi, A. Elmadany, and M. Abdul-Mageed, "AraT5: Text-to-text transformers for Arabic language generation," 2021, *arXiv:2109.12068*.

[42] A. Ghaddar, Y. Wu, S. Bagga, A. Rashid, K. Bibi, M. Rezagholizadeh, C. Xing, Y. Wang, D. Xinyu, Z. Wang, B. Huai, X. Jiang, Q. Liu, and P. Langlais, "Revisiting pre-trained language models and their evaluation for Arabic natural language understanding," 2022, *arXiv:2205.10687*.

[43] A. Alghamdi, X. Duan, W. Jiang, Z. Wang, Y. Wu, Q. Xia, Z. Wang, Y. Zheng, M. Rezagholizadeh, B. Huai, P. Cheng, and A. Ghaddar, "AraMUS: Pushing the limits of data and model scale for Arabic natural language processing," 2023, *arXiv:2306.06800*.

[44] J. Pfeiffer, F. Piccinno, M. Nicosia, X. Wang, M. Reid, and S. Ruder, "MmT5: Modular multilingual pre-training solves source language Hallucinations," 2023, *arXiv:2305.14224*.

[45] D. Uthus, S. Ontañón, J. Ainslie, and M. Guo, "MLongT5: A multilingual and efficient text-to-text transformer for longer sequences," 2023, *arXiv:2305.11129*.

[46] M. Guo, J. Ainslie, D. Uthus, S. Ontanon, J. Ni, Y.-H. Sung, and Y. Yang, "LongT5: Efficient text-to-text transformer for long sequences," 2021, *arXiv:2112.07916*.

[47] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and efficient foundation language models," 2023, *arXiv:2302.13971*.

[48] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," 2023, *arXiv:2307.09288*.

[49] A. Grattafiori et al., "The llama 3 herd of models," 2024, *arXiv:2407.21783*.

[50] M. T. I. Khondaker, N. Naeem, F. Khan, A. Elmadany, and M. Abdul-Mageed, "Benchmarking LLaMA-3 on Arabic language generation tasks," in *Proc. 2nd Arabic Natural Language Process. Conf.*, 2024, pp. 283–297.

[51] B. Workshop et al., "Bloom: A 176B-parameter open-access multilingual language model," 2022, *arXiv:2211.05100*.

[52] R. Anil et al., "PaLM 2 technical report," 2023, *arXiv:2305.10403*.

[53] A. Üstün, V. Aryabumi, Z.-X. Yong, W.-Y. Ko, D. D'souza, G. Onilude, N. Bhandari, S. Singh, H.-L. Ooi, A. Kayid, F. Vargus, P. Blunsom, S. Longpre, N. Muennighoff, M. Fadaee, J. Kreutzer, and S. Hooker, "Aya model: An instruction finetuned open-access multilingual language model," 2024, *arXiv:2402.07827*.

[54] S. Ji, Z. Li, I. Paul, J. Paavola, P. Lin, P. Chen, D. O'Brien, H. Luo, H. Schütze, J. Tiedemann, and B. Haddow, "EMMA-500: Enhancing massively multilingual adaptation of large language models," 2024, *arXiv:2409.17892*.

[55] A. Bhattacharjee, T. Hasan, W. Uddin Ahmad, Y.-F. Li, Y.-B. Kang, and R. Shahriyar, "CrossSum: Beyond english-centric cross-lingual summarization for 1,500+ language pairs," 2021, *arXiv:2112.08804*.

[56] C. Whitehouse, F. Huot, J. Bastings, M. Dehghani, C.-C. Lin, and M. Lapata, "Low-rank adaptation for multilingual summarization: An empirical study," in *Proc. Findings Assoc. Comput. Linguistics, NAACL*, 2024, pp. 1202–1228.

[57] E. M. B. Nagoudi, M. Abdul-Mageed, A. Elmadany, A. A. Inciarte, and M. T. I. Khondaker, "JASMINE: Arabic GPT models for few-shot learning," 2022, *arXiv:2212.10755*.

[58] G. Team et al., "Gemini: A family of highly capable multimodal models," 2023, *arXiv:2312.11805*.

[59] W. Lai, M. Mesgar, and A. Fraser, "LLMs beyond english: Scaling the multilingual capability of LLMs with cross-lingual feedback," 2024, *arXiv:2406.01771*.

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

[61] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," 2019, *arXiv:1904.09675*.

[62] T. Sellam, D. Das, and A. P. Parikh, "BLEURT: Learning robust metrics for text generation," 2020, *arXiv:2004.04696*.

[63] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 311–318.

[64] 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*, Jun. 2005, pp. 65–72.

[65] R. Bommasani and C. Cardie, "Intrinsic evaluation of summarization datasets," in *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, 2020, pp. 8075–8096.

[66] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 3, Mar. 2003, pp. 993–1022.

[67] M. Grusky, M. Naaman, and Y. Artzi, "Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies," 2018, *arXiv:1804.11283*.

[68] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[69] *Mawdoo3 Website*. [Online]. Available: https://mawdoo3.com/
[70] 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.
[71] M. El-Haj, U. Kruschwitz, and C. Fox, "Using mechanical Turk to create a corpus of Arabic summaries," Tech. Rep., 2010.
[72] *Mechanical Turk*. [Online]. Available: https://www.mturk.com/
[73] M. El-Haj and R. Koulali, "KALIMAT a multipurpose Arabic corpus," in *Proc. 2nd Workshop Arabic Corpus Linguistics (WACL)*, Jan. 2013, pp. 22–25.
[74] D. Varab and N. Schluter, "MassiveSumm: A very large-scale, very multilingual, news summarisation dataset," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2021, pp. 10150–10161.
[75] A. Alhamadani, X. Zhang, J. He, and C.-T. Lu, "LANS: Large-scale Arabic news summarization corpus," 2022, *arXiv:2210.13600*.
[76] P. Tikhonov and V. Malykh, "WikiMulti: A corpus for cross-lingual summarization," in *Proc. 11th Conf. Artif. Intell. Natural Language*, Jan. 2022, pp. 60–69.
[77] Z. Ezzat, A. Khalfallah, and G. Khoriba, "Fused transformers: Fused information of Arabic long article for summarization," *Proc. Comput. Sci.*, vol. 244, pp. 96–104, Jan. 2024.
[78] L. M. A. Qassem, D. Wang, H. Barada, A. Alrubaie, and N. Almoosa, "Automatic Arabic text summarization based on fuzzy logic," in *Proc. 3rd Int. Conf. Natural Language Speech Process.*, Sep. 2019, pp. 42–48.
[79] A. Alshanqiti, A. Namoun, A. Alsughayyir, A. M. Mashraqi, A. R. Gilal, and S. S. Albouq, "Leveraging DistilBERT for summarizing Arabic text: An extractive dual-stage approach," *IEEE Access*, vol. 9, pp. 135594–135607, 2021.
[80] A. Elmadany, M. Abdul-Mageed, and M. Abdul-Mageed, "Arat5: Text-to-text transformers for Arabic language generation," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 628–647.
[81] I. Tanfouri and F. Jarray, "Genetic algorithm and latent semantic analysis based documents summarization technique," in *Proc. 14th Int. Joint Conf. Knowl. Discovery, Knowl. Eng. Knowl. Manage.*, 2022, pp. 223–227.
[82] W. Etaiwi and A. Awajan, "SemG-TS: Abstractive Arabic text summarization using semantic graph embedding," *Mathematics*, vol. 10, no. 18, p. 3225, Sep. 2022.
[83] M. Kamal Eddine, N. Tomeh, N. Habash, J. Le Roux, and M. Vazirgiannis, "AraBART: A pretrained Arabic sequence-to-sequence model for abstractive summarization," 2022, *arXiv:2203.10945*.
[84] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," 2018, *arXiv: 1801.02143*.
[85] E. M. B. Nagoudi, A. Elmadany, M. Abdul-Mageed, T. Alhindi, and H. Cavusoglu, "Machine generation and detection of Arabic manipulated and fake news," 2020, *arXiv: 2011.03092*.
[86] I. A. El-Khair, "1.5 billion words arabic corpus," 2016, *arXiv: 1611.04033*.
[87] P. J. O. Suárez, B. Sagot, and L. Romary, "Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures," in *Proc. 7th Workshop Challenges Manage. Large Corpora (CMLC-7)*. Leibniz-Institut für Deutsche Sprache, 2019.

ZEYAD EZZAT received the B.S. degree in computer and communication engineering from Alexandria University, Egypt, in 2020. He is currently pursuing the master's degree in informatics with the School of Information Technology and Computer Science, Nile University.

In the first two years after graduation, he was a full-time Teaching Assistant, focusing primarily on machine learning and deep learning-related courses. Concurrently, with his role as an NLP Research and Development Engineer with the Applied Innovation Center, where he focuses on diverse applications of NLP, including text summarization and large language model (LLM) pretraining, he was also a part-time Teaching Assistant with Alexandria University.

GHADA KHORIBA (Member, IEEE) received the bachelor's and M.Sc. degrees from Helwan University, in 2000 and 2004, respectively, and the Ph.D. degree in computer science from the University of Tsukuba, Japan, in 2010.

She was promoted to an Associate Professor, in 2020. She is currently an Associate Professor with the School of Information Technology and Computer Science, Nile University. She is also an Associate Professor with the Department of Computer Science, Helwan University, where she has been, since 2000. Her research interests include medical image analysis, machine learning techniques and optimization problems, deep learning models, swarm algorithms, computer vision, natural language processing, LLMs, and knowledge graphs.

Dr. Khoriba is a member of ACM.

AYMAN KHALAFALLAH received the bachelor's and master's degrees in computer engineering from Alexandria University and the Ph.D. degree in computer science from Rutgers University.

He is currently a Principal Research and Development Team Lead with the Applied Innovation Center, Ministry of Communications and Information Technology (MCIT), Alexandria, Egypt, where he leads projects in natural language processing, machine translation, and vision-language models. Additionally, he is an Assistant Professor with the Faculty of Engineering, Alexandria University, and has held previous academic positions with King Saud University and Beirut Arab University. His research interests include problem-solving, theoretical computer science, and applying mathematical and statistical methods to computational challenges.