IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 May 2024, accepted 6 August 2024, date of publication 9 August 2024, date of current version 3 September 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3441489*

RESEARCH ARTICLE

# MSG-ATS: Multi-Level Semantic Graph for Arabic Text Summarization

**MUSTAFA ABDUL SALAM[1,2], MOHAMED ALDAWSARI[1], MOSTAFA GAMAL[2,3], HESHAM F. A. HAMED[3], AND SARA SWEIDAN[2,4]**

[1]Department of Computer Engineering and Information, College of Engineering at Wadi Ad Dwaser, Prince Sattam Bin Abdulaziz University, Al-Kharj 16273, Saudi Arabia

[2]Department of Artificial Intelligence, Faculty of Computers and Artificial Intelligence, Benha University, Benha 13518, Egypt

[3]Artificial Intelligence Department, Faulty of Artificial Intelligence, Egyptian Russian University, Badr City, Cairo 11829, Egypt

[4]Faculty of Computer Science and Engineering, New Mansoura University, Gamasa 35712, Egypt

Corresponding author: Mustafa Abdul Salam (mustafa.abdo@fci.bu.edu.eg)

This work was supported by the Deanship of Scientific Research, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia, under Project PSAU/2024/R/1445.

**ABSTRACT** Arabic language processing presents significant challenges due to its complex linguistic patterns and shortage of resources. This study describes MSG-ATS, a new technique to abstractive text summarization in Arabic that aims to overcome these issues. The key challenge is producing coherent and high-quality summaries given the Arabic language's rich syntactic, semantic, and contextual elements. Traditional approaches, such as word2vec, frequently fail to capture these subtleties well. MSG-ATS uses multilevel semantic graphs and deep learning techniques to create a more thorough representation of Arabic text. This approach improves traditional text generation and embedding approaches by collecting syntactic, semantic, and contextual information fully. MSG-ATS uses a deep neural network to create high-quality summaries that are coherent and contextually appropriate. To verify MSG-ATS, we performed rigorous assessments that compared its performance to word2vec, a fundamental word embedding approach. These assessments employed a unique dataset created expressly for this study and included automated assessment using the ROUGE measure. The results are compelling: MSG-ATS outperformed the baseline model by 42.4% in precision, 23.8% in recall, and 38.3% overall. The outcomes of this study highlight MSG-ATS's potential to considerably increase Arabic text summarization by providing a strong framework that solves the constraints of existing models while also laying the groundwork for future developments in the area.

**INDEX TERMS** Automatic text summarization, multi-level semantic graph, semantic graph embedding, graph neural networks, attention mechanisms.

## I. INTRODUCTION

We are often inundated by information from numerous publications and documents. The task of sifting through it can be both time consuming and draining. Arabic, being one of the most widely spoken languages globally, has over 420 million speakers across 26 countries and holds significant cultural, religious, and economic importance. Despite its widespread use, Arabic remains underrepresented in computational linguistics research compared to languages like English, highlighting a critical need for effective NLP tools tailored to Arabic. Text summarization offers a solution by allowing us to condense text and convey its core ideas quickly, thereby saving time and effort. However, automation of this process is not straightforward. As previous research has shown, we require models that can summarize long publications while maintaining their main ideas. These studies highlight challenges such as redundancy, where the final summary may contain repetitive information, the risk of including irrelevant details, and the possibility of omitting crucial information. Additionally, the readability of the summary suffers if it contains disconnected terms. As shown in figure 1 text summarization encompasses a range of categories and characteristics that facilitate the classification [1].

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

- Text summarization falls into categories of single-document and multi-document summaries. Multi-document summarization presents significant challenges, including addressing issues such as content redundancy.
- To categorize text summarization, we distinguish between extractive and abstractive methods based on the output summary. Extractive summarization involves selecting and prioritizing the most significant textual components, such as sentences, to compose a summary. By contrast, abstractive summarization starts after incorporating words and phrases that may not appear in the original text. Consequently, an abstractive summary maintains the original structure while reinterpreting its core ideas using diverse terms and phrases. Extractive summarization is simpler compared to abstractive methods, as it requires fewer Natural Language Processing (NLP) techniques.
- The total number of statements in an output summary determines the two primary classifications of text summarization. If the final summary is limited to one sentence, it is called a single-statement summarization. It is called a multi-statement summary if it isn't.

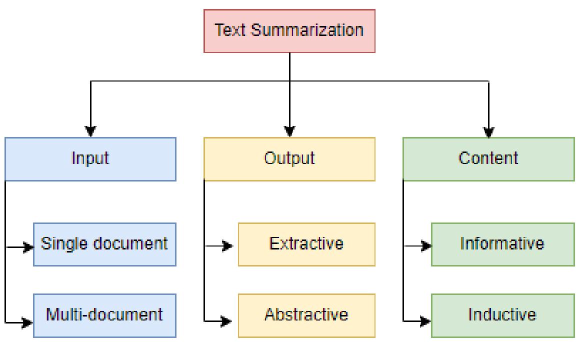


**FIGURE 1.** **Category of text summarization.**

Many efforts in text summarization have focused on summarizing English texts. Nevertheless, difficulties with text summarizing have brought attention to the need for more research to improve the effectiveness of methods currently in use for text summarization in languages other than English. For example, the author in [2] claimed that summarizing Arabic materials is more challenging than summarizing English writing. Because Arabic has unique qualities, text summarizing systems for Arabic have not advanced as much as those for other languages. Methods for producing Arabic text summaries have not been the subject of many studies.

### A. MOTIVATION

The primary reasons for these challenges lie in the Arabic language's inherent traits, including its high levels of ambiguity, highly derivational and inflectional structure, syntactic complexity, and morphological intricacy. Due to the relative simplicity of extractive text summarization compared to abstractive methods, many text summarization algorithms have focused on extraction rather than abstraction [3]. Extractive summaries distill key elements from the original text, which may be extensive, complex, and challenging to comprehend. On the other hand, abstractive summarization is more intricate and demanding, as it involves crafting concise, coherent sentences that encapsulate the core concepts of the original material.

Semantic representation is used in many Natural Language Processing (NLP) applications, including machine translation and question answering, to improve computer linguistic outcomes. Semantic representation is used to produce in-depth annotations of a text that faithfully capture its meaning. Graph standard and formal methods for representing and codifying large and complex data structures. The graph model is more effective than other text representation methods like rule-based representation, frame representation, and predicate logic representation due to the fact that it can record the semantic connections between words in a text [4], [5]. The most popular method of representing semantics is using a semantic network [6], [7], [8], [9]. Semantic graphs are networks that represent the semantic connections between different concepts, such as phrases and sentences. The edges in the network depict the semantic relationships between concepts, while the vertices represent individual concepts. Semantic graphs serve as a means to encode plain text and convey its context visually. Preserving semantics in graph representation poses challenges, as semantic links can vary depending on the language of the text and may be challenging to capture accurately. Several text properties for words, phrases, and paragraphs are used in text summarization. Manually extracting the beneficial properties from older methodologies that necessitate the use of hand-crafted features takes time and effort [10]. On the other hand, deep learning enables the creation of valuable features from training data. Unlike hand-crafted features, which are difficult to employ with vast quantities of data and usually rely on the prior knowledge of designers, deep learning dynamically learns features from the available information.

Given Arabic's intricate linguistic characteristics and the dearth of sophisticated summary systems, it is clear that the language requires an efficient text summarization model. To overcome these obstacles, creative solutions are needed that may make use of the distinctive qualities of the Arabic language to produce summaries of the highest caliber. To address this need, a unique approach is proposed in this research.

### B. CONTRIBUTIONS

The challenges outlined above necessitate the development of a novel framework for abstractive Arabic text summarization that utilizes Arabic semantic graph representation, termed MSG-ATS. When constructing a semantic graph and learning Arabic, it is crucial to consider the morphological and syntactic features of the language. Therefore, the Graph Neural Network (GNN) model was employed to preserve the semantic relationships between words in the text graph. Given the promising results of deep learning across various

AI domains and data mining challenges, it is leveraged to produce text summaries [12].

To assess the proposed MSG-ATS, a new text summarization dataset based on well-written and published news items is curated. Additionally, the evaluation comprises both automated and manual assessment methods. The results underscore MSG-ATS's efficacy, revealing significant improvements in precision, recall, and overall performance over baseline models.

### C. PAPER ORGANIZATION

The paper is structured as follows: Section II provides a concise review of relevant literature on Arabic text summarization. Section III outlines the proposed MSG-ATS model. Section IV describes the dataset utilized in the research. Section V presents the experimental results and evaluation findings. Finally, Section VI summarizes the conclusions drawn from the study.

## II. RELATED WORK

Since text summarization has been around for more than 50 years, there has been a lot of activity in the scholarly community [1]. Researchers continuously strive to enhance existing text summarization techniques or introduce novel ones to elevate the quality of generated summaries. However, the accuracy of current text summarization methods remains relatively modest. The morphological and structural intricacies of the Arabic language pose challenges, rendering current methods and approaches for summarizing Arabic texts inadequate and still in their early stages [11]. Techniques for Arabic text summarization can be broadly categorized into three main groups: graph-based approaches, deep learning-based approaches, and genetic algorithm and machine learning-based approaches.

In comparison, text summarization in languages such as English, Chinese, and Spanish has advanced significantly because to the availability of rich linguistic information and more robust models. These languages benefit from larger datasets, more complex language models, and established NLP frameworks. For example, in English, models such as BERT and GPT have considerably improved summarization jobs by comprehending and producing cohesive text. Similarly, Chinese summarization has evolved with models designed expressly for its distinct logographic script and tonal characteristics, using large annotated corpus and powerful pre-trained models.

### A. GRAPH BASED METHODS

AL-Khassawneh and Hanandeh [3] proposed a model that evaluates sentence relevance, coverage, and variety to create summaries using a textual graph approach. The final summary is generated by blending statistical and semantic criteria, condensing the original text into a sub-graph while excluding less significant terms. Their study showed promising results when tested on the Essex Arabic Summary Corpus (EASC) using the ROUGE index and compared against advanced In comparison to advanced methods utilizing the RED metric. In a separate study [12] an automated text summarization (ATS) method based on graphs is introduced. Text documents are depicted as graphs, with sentences as nodes and similarity measured by edge weights. Due to the complexity of Arabic, traditional cosine similarity-based approaches face challenges. To improve summary quality, the study proposes a metric that integrates mutual nouns and cosine similarity between related phrases. They also introduce three morphological analyzers (Safar Alkhalil, Stanford NLP, and BAMA), with Safar Alkhalil outperforming others in evaluations using the EASC corpus.

Another study [13] introduces an extractive approach to summarizing Arabic reviews, integrating sentiment and polarity characteristics. Reviews from Booking.com were grouped by polarity, and TextRank was applied for graph-based ranking to produce summaries. These summaries underwent evaluation by humans and through BLEU and ROUGE criteria, resulting in improved scores. In a distinct study, Azmi and Altmami [14] proposed an extraction-based method for summarizing single Arabic documents. They identified relevant phrases for inclusion in the summary, refining and reducing them using a rule-based approach. However, this method does not qualify as abstractive summarization since it does not introduce new phrases absent from the original text. The methodology was evaluated using a dataset of 150 news stories. In another investigation [15], Scholars explored an extractive approach to summarizing Single Arabic documents. Significant sentences identified using graph-based methods, and various text representation techniques, including TF-IDF, fastText, and Word2Vec, were employed to extract sentences. Evaluation using the ROUGE metric and the EASC corpus revealed that cosine similarity, PageRank ranking, and TF-IDF representation all produced high-quality summaries with satisfactory performance.

### B. DEEP LEARNING BASED METHODS

In this study [16], a sequence-to-sequence model-based abstractive Arabic text summarization system was introduced, comprising encoder and decoder components. Deep artificial neural networks, including Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM), and Bidirectional Long Short-Term Memory (BiLSTM) layers, were explored to enhance performance. The AraBERT pre-processing and global attention mechanism were employed to improve Arabic word comprehension. System evaluation was evaluated using the ROUGE and BLEU metrics.

Another article [17] developed basic neural network algorithms for summarizing Indian court judgment papers, focusing on word and phrase embeddings to capture semantics. These methods offer versatility across different domains without requiring handcrafted features or domain-specific expertise. Sentences in the training set were assigned classes or scores based on their proximity to reference summaries to

**TABLE 1.** Arabic text summarization techniques have been presented recently.

| Ref | Year | Method | Doc | Type | Dataset | Metric |
|---|---|---|---|---|---|---|
| [3] | 2023 | Graph-Based | Single | Extractive | EASC | ROUGE |
| [12] | 2020 | Graph-Based | Single | Extractive | EASC | F-score |
| [13] | 2022 | Graph-Based | Single | Extractive | hotel reviews | BLEU and ROUGE |
| [14] | 2018 | Graph-Based | Single | Extractive and abstractive | 150 document | recall, precision and ROUGE |
| [15] | 2022 | Graph-Based | Single | Extractive | Arabic articles | FI-score |
| [16] | 2022 | Deep Learning | Single | abstractive | Arabic data | ROUGE |
| [17] | 2022 | Deep Learning | Single | Extractive | Legal judgment | ROUGE |
| [18] | 2022 | Deep Learning | Single | Extractive | Arabic data | BLEU and ROUGE |
| [19] | 2020 | Evolutionary | multi-document | Extractive | DUC 2002 | ROUGE |
| [20] | 2021 | Multi-Objective Optimization | multi-document | Extractive | DUC 2005 and 2007 | ROUGE |
| [21] | 2018 | Quantum-Inspired Genetic Algorithm | Single | Extractive | EASC | ROUGE |

address the lack of labeled data. Experimental results demonstrate the efficacy of the proposed methods compared to alternative baselines. In a related work [18], introduce a novel Arabic story generation approach using a fine-tuned davinci-003 LLM for text and the Midjourney model for images, showcasing improved results through fine-tuning on a dataset of 527 Arabic stories. This approach addresses gaps in Arabic story generation by integrating text and image generation in a cohesive pipeline.

### C. MACHINE LEARNING AND EVOLUTIONARY BASED METHODS

This work [19] provided an autonomous, generic, and extractive Arabic multi-document summarization system employing evolutionary multi-objective optimization methodologies and clustering-based techniques. The clustering strategy finds the main themes, while the optimization method improves coverage, diversity/redundancy, and relevance. The TAC 2011 and DUC 2002 datasets are used to assess the system's performance. This paper describes MTSQIGA, a multi-document text summarizing approach that employs a modified quantum-inspired genetic algorithm (QIGA) to extract significant terms for summary production [20]. As the goal function, the approach optimizes a linear combination of redundancy, relevance, and coverage factors using six phrase score metrics. To ensure that summaries adhere to predefined length constraints, we employ a modified quantum measurement and a self-adaptive quantum rotation gate. The effectiveness of this technique is evaluated using ROUGE standard measurements on the DUC 2005 and 2007 benchmark datasets.

In a study by Al-Radaideh et al. [21], a hybrid approach is proposed to extract essential components from political materials. This approach combines statistical properties, evolutionary algorithms, and domain expertise. The methodology involves preprocessing the texts, scoring phrases based on various factors such as word frequency and domain-specific keywords, and synthesizing summaries using genetic algorithms, cosine similarity, and sentence scores. Experiments were conducted on the KALIMAT and EASC corpora to compare the proposed technique with three state-of-the-art methods, utilizing ROUGE measures.

Table 1 highlights that a significant portion of the reviewed literature focused on extractive text summarization rather than abstractive text summarization. Extractive summarization, which involves identifying the most important sentences from the original text, is generally more challenging than abstractive summarization, which involves producing a revised version of the original text. In the context of Arabic text summarization, there have been fewer proposals for abstractive methods. Furthermore, most of the examined articles focused on single-document summarization rather than multi-document summarization. The Essex Arabic Summaries Corpus emerged as the most commonly used dataset for evaluating Arabic text summarization techniques. Finally, the ROUGE metric was the predominant tool for assessing the effectiveness of text summarization techniques.

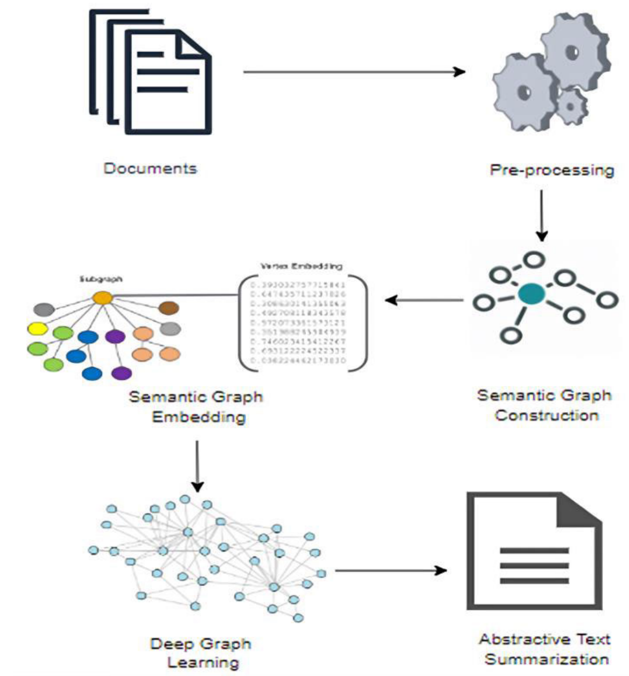


**FIGURE 2.** Proposed MSG-ATS framework.

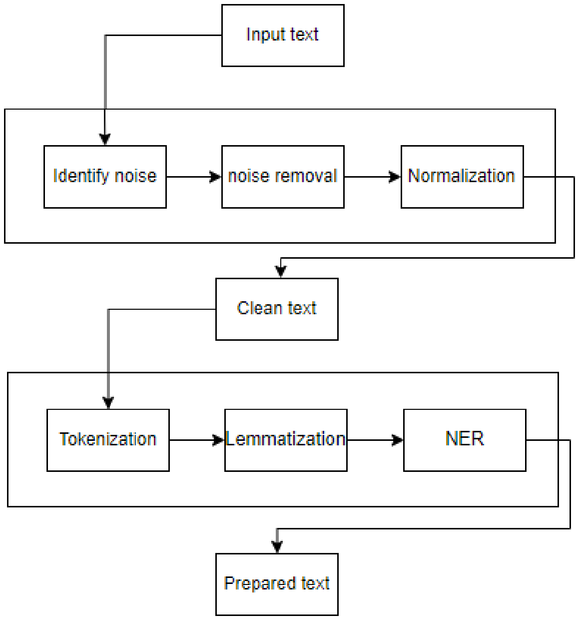


**FIGURE 3.** **Summarized steps of pre-processing steps.**

## III. MSG-ATS: MULTI-LEVEL SEMANTIC GRAPH ARABIC TEXT SUMMARIZATION

Figure 2 depicts the proposed approach for abstractive single-statement Arabic text summarization. The process begins with preprocessing input Arabic texts, involving tasks such as stop word removal, tokenization, and normalization. Subsequently, features such as part-of-speech tags, syntactic dependencies, and named entities are extracted from the text to capture syntactic and semantic information. To represent semantic connections, a semantic graph of the text is constructed, where words serve as nodes and relationships as edges. Graph neural network (GNN) embedding is then utilized to generate low-dimensional representations of nodes while preserving structural and semantic information. Semantic walks, guided by elements specific to the Arabic language, are employed to produce language-specific embeddings. Next, a deep neural network (NN) is employed, leveraging attention and sequence-to-sequence learning, to process the output embeddings. This NN produces concise, logically coherent abstractive summaries that capture the main ideas of the original text. The framework seamlessly integrates preprocessing, feature extraction, semantic graph construction, GNN embedding, and NN processing, ensuring a systematic workflow. The primary goal is to generate high-quality abstractive summaries while retaining the essential concepts of the source Arabic text.

The following stages make up the MSG-ATS:

1. Pre-processing: During the preparation phase, the source texts are cleaned, the sentences are divided, annotated, and tokenized, and the characteristics are extracted. Figure 3 illustrates all steps from input the text to get prepared text.

a. Tokenization: Tokenization is the process of dividing the text into discrete words or units. Arabic words are normally separated by spaces, however because Arabic diacritical marks and ligatures are used, tokenization may need to take complicated tokenization rules into consideration.

b. Normalization: Normalization seeks to normalize the representation of Arabic language by eliminating diacritics, standardizing letter shapes, and managing Arabic-specific punctuation signs. The word "ذَهَبتُ" (I went) would be normalized to "ذهبت".

c. Stopword Removal: Stopwords are popular words with no substantial semantic significance that are frequently eliminated from texts to decrease noise. Stopwords in Arabic can comprise articles, prepositions, and conjunctions. In the sentence "إلى المكتب أنا ذهبت", the stopwords "أنا" (I), "إلى" (to), and "المكتب" (the office) might be removed, leaving the meaningful words "ذهبت" (went).

d. Lemmatization/Stemming: Lemmatization reduces words to their base or dictionary forms, whereas stemming eliminates affixes to reveal the root form of words. This reduces variance in word forms and increases processing efficiency. The word "كتبت" (I wrote) would be lemmatized to "كتب" (write). The word "كتابات" (writings) would be stemmed to "كتاب" (book).

e. Named Entity Recognition: Named Entity Recognition (NER) recognizes and categorizes named entities in text, including people, organizations, locations, and dates. In the sentence "يوم الثلاثاء زار مصطفى القاهرة", the named entities "مصطفى" (Mostafa) and "القاهرة" (Cairo) would be recognized as a person and a location, respectively.

2. Data Representation Enhancement: The improvement of data representation entails using a range of linguistic elements to construct a complete multi-level representation of the text.

a. Syntactic Analysis: Syntactic analysis is the process of studying a text's syntactic structure to detect word relationships such as subject-verb-object dependencies, modifier relationships, and other syntactic dependencies. This stage may include applying part-of-speech tagging and dependency parsing algorithms. In the sentence "الكتاب يقع على الطاولة" (The book is on the table), syntactic analysis identifies that "الكتاب" (the book) is the subject, "يقع" (is) is the verb, and "الطاولة" (the table) is the object.

b. Semantic Analysis: Semantic analysis is the process of extracting semantic information from text in order to understand the meaning and relationships between words. This might involve recognizing word meanings, named things, semantic similarities, and other semantic linkages. In the sentence "زار مصطفى القاهرة" (Mostafa visited Cairo), semantic analysis identifies "مصطفى" (Mostafa) as a person entity and "القاهرة" (Cairo) as a location entity.

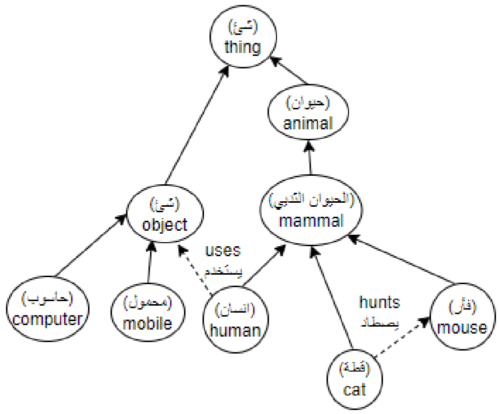


**FIGURE 4.** Simple example on semantic graph.

c. Contextual features: Contextual features play a crucial role in enriching the semantic graph embeddings with additional contextual information extracted from the text. For instance, contextual embeddings, such as those generated by pre-trained models like BERT or RoBERTa, capture the meaning of each word based on its surrounding context in the text. For example, in the sentence 'الكتاب جميل ومثير للاهتمام' (The book is beautiful and interesting), the contextual embeddings of 'جميل' (beautiful) and 'مثير للاهتمام' (interesting) would reflect their contextual meanings within the sentence. Additionally, word frequency provides valuable insights into the importance or relevance of words within the text. Words with higher frequencies, such as 'كتاب' (book) in a literary review, may carry more significance in summarization tasks. Furthermore, positional information, such as the position of words within a sentence or their relative positions to other words, offers valuable context for understanding the text. For example, in the phrase 'بدأ الكتاب بفصل مثير' (The book began with an exciting chapter), the positional with an exciting chapter), the positional information of 'بدأ' (began) indicates its role as the starting point of an action, influencing the semantic context of the sentence. By integrating these contextual features into the semantic graph embeddings, the model gains a deeper understanding of the text, enabling more accurate and informative summarization results.

d. Integration of linguistic features: The integration of linguistic features enhances the multi-level representation of the text by combining syntactic, semantic, and contextual information. For instance, syntactic features such as part-of-speech tags and dependency relationships provide insights into the grammatical structure of the text. In the sentence 'جميل ومثير للاهتمام الكتاب' (The book is beautiful and interesting), syntactic analysis reveals the relationship between words, with 'جميل' (beautiful) and 'مثير للاهتمام' (interesting) identified as adjectives modifying 'الكتاب' (the book). Semantic features, on the other hand, capture the meaning and relationships between words. For example, in the phrase 'بدأ الكتاب بفصل مثير' (The book began with an exciting chapter), semantic analysis identifies 'بدأ' (began) as the verb indicating the start of an action, while 'مثير' (exciting) conveys the semantic attribute of the chapter. Additionally, contextual features, such as word embeddings generated by pre-trained language models like BERT, incorporate information from the surrounding context to enrich the representation. By integrating these linguistic features, the model creates a comprehensive multi-level representation of the text, capturing both its structural and semantic complexities.

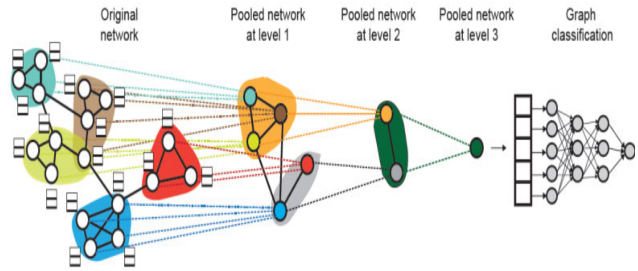


**FIGURE 5.** An example of the coarsening GNN technique for graphs.

3. Semantic Graph Construction: Semantic graph creation is the foundation of many text analysis tasks, offering a systematic representation of the relationships between words in a text. Semantic networks, which represent words as nodes and relationships as edges, provide a strong framework for capturing syntactic dependencies and semantic linkages inside a document. This introduction lays the setting for a more in-depth look at the procedures involved in semantic graph creation, emphasizing its importance in uncovering the complexity of language interpretation Figure 4 illustrates simple example on sematic graph.

a. Graph Construction: Graph creation is the process of creating a semantic graph representation of the text, with each word serving as a node and links between words represented as edges. These linkages include both syntactic dependencies and semantic links between words. Using the syntactic and semantic analyses from the previous steps, the semantic graph would represent "الكتاب" (the book) as a node connected to "يقع" (is) with an edge labeled "subject-verb" and to "الطاولة" (the table) with an edge labeled "verb-object". Additionally, "مصطفى" (Mostafa) would be connected to "القاهرة" (Cairo) with an edge labeled "visited-location".

b. Integration with GNN: To capture the structural and semantic context, the semantic graph is processed using a Graph Neural Network (GNN). GNNs update node representations using message-passing algorithms that aggregate information from surrounding nodes while retaining the graph's intricate linkages. Attention mechanisms focus on important nodes, whereas multimodal fusion strategies use different language aspects to improve embeddings. This integration allows the GNN to produce comprehensive, low-dimensional vector representations of the semantic graph, which aids in successful text summarization.

4. Semantic Graph Embedding: Semantic graph embedding converts the intricate linkages and semantic nuances depicted in the text's semantic graph representation into compact, low-dimensional vector representations. This transformation is facilitated by Graph Neural Networks (GNNs), specialized neural networks designed to operate on graph-structured data. GNNs employ message-passing algorithms to iteratively update node representations by aggregating information from neighboring nodes, thus capturing the structural and semantic context of each word in the graph. Attention mechanisms and multimodal fusion techniques enhance the embedding process further. Attention methods enable the model to focus on pertinent nodes throughout the embedding process, while multimodal fusion strategies integrate information from various modalities, such as syntactic and semantic properties, to enhance the embeddings. The refined semantic graph embeddings, incorporating attention mechanisms and multimodal fusion, offer concise yet comprehensive representations of words/nodes in the semantic graph, capturing semantic connections and contextual information. Embedding the semantic graph in a continuous vector space facilitates downstream operations like text summarization and information retrieval, leveraging the rich semantic information encapsulated within the embeddings. As show in Figure 5.

5. Abstractive Text Summarization: The selection of the deep neural network (NN) model at this stage is informed by the superior performance of deep learning over other machine learning techniques across various natural language processing (NLP) tasks, including named entity identification and machine translation [22], [23], [24], [25]. The primary objective at this stage is to utilize the low-dimensional vectors generated in the previous phase for constructing the text summary. Deep learning exhibits enhanced efficiency, particularly when employing sequence-to-sequence learning and attention mechanisms. To assess the effectiveness of statistical learning and local representations of words and phrases in machine translation, deep learning leverages distributed word embedding [26]. These techniques find applications in diverse NLP tasks such as text summarization and question answering. Consequently, the proposed model incorporates an attention mechanism and sequence-to-sequence learning.

## IV. DATASET

There are no excellent datasets available for Arabic that would allow the researchers' efforts to be compared at the same time. The majority of Arabic-speaking researchers translated English datasets into Arabic in order to verify their findings, as seen in [27]. A dearth of Arabic datasets suitable for text summarization persists, illustrated by EASC [28], Arabic Gigaword [29], and KALIMAT [30]. However, each of these resources is plagued by distinct limitations. For instance, the EASC dataset, with only 153 documents, suffers from restricted dataset size. Conversely, the KALIMAT dataset, primarily intended for extractive text summarization, is constrained by its limited abstraction. Furthermore, Arabic Gigaword faces challenges related to dataset orientation. Compounded by academia's preference for independent data collection, the absence of standardized Arabic datasets exacerbates the challenges inherent in the evaluation process, fostering subjectivity [2]. Consequently, the assessment of summaries remains intricate due to the absence of a universally applicable ideal summary for any given text.

For Arabic text summarizing, a sizable dataset is required because of the aforementioned constraints. Reputable websites like CNN-Arabic news and AlJazeera.net frequently have high-quality databases with little grammatical problems. These websites host numerous high-quality Arabic articles accompanied by one-line summaries (titles and highlights). Leveraging articles from the AlJazeera.net website, we curated a dataset for our experiments in abstractive text summarization. The effectiveness of the proposed abstractive single-statement Arabic text summarizer was evaluated using a dataset comprising 11,641 documents. This dataset encompassed a total of 4,102,085 words, with an average word length of 6.1 characters. Additionally, it consisted of 17,320 paragraphs with an average length of 217 words. It's worth noting that each article's summary could often be found within its title. Articles were meticulously chosen based on a predefined list of keywords, with a preference for those published within the last five years. Table 2 demonstrates the careful selection of keywords, ensuring coverage of a diverse range of article topics, including sports, politics, economics, and the arts.

**TABLE 2.** Terms employed in construction of the dataset.

| Keyword | No. of articles collected |
|---|---|
| "الولايات المتحده" (United States) | 3206 |
| "مصر" (Egypt) | 2741 |
| "مباراه" (Match) | 1650 |
| "معاهده" (treaty) | 704 |
| "سلام" (Peace) | 942 |
| "فيلم" (Movie) | 2147 |
| "'طبيعه" (Nature) | 251 |
| Total | 11641 |

## V. RESULTS

The tests were divided into three main parts: generating the semantic network, embedding the semantic graph, and evaluating the suggested MSG-ATS model with deep learning. The first two portions made use of a Tower Workstation equipped with an Intel Xeon Silver 4114 up to 3.0GHz (10-Core), 96GB RAM, 1TB SSD + 3TB, and a Quadro P2000 5GB. In the third portion, a standalone workstation with a twin 2.10 GHz Intel Xeon E5-2620v4 CPU, 128 GB RAM, and an Nvidia GTX 3060 was used.

**TABLE 3.** Set the parameters for the sensitivity analysis of learning rates.

| Parameter | Value |
|---|---|
| Units | 300 |
| Hidden layers | 2 |
| Beam width | 8 |
| Embedding size | 128 |
| Epochs | 50 |
| Batch size | 32 |
| Probability | 0.85 |

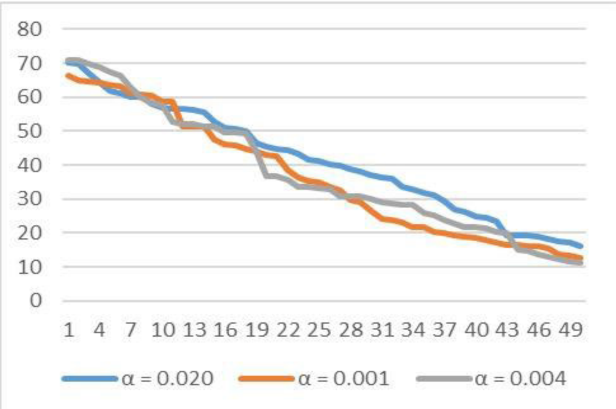


**FIGURE 6.** Loss value for learning rate.

**TABLE 4.** Setup the settings for the sensitivity analysis of beam width.

| Parameter | Value |
|---|---|
| Units | 300 |
| Hidden layers | 2 |
| Learning rate | 0.004 |
| Embedding size | 128 |
| Epochs | 50 |
| Batch size | 32 |
| Probability | 0.85 |

- To construct a semantic graph from the dataset, we follow a two-step process: identifying word relationships and applying predefined rules.

Identifying Relationships between Words: We utilize existing tools like NLTK and spaCy for linguistic analysis. Firstly, we segment the text into words and categorize them grammatically using POS tagging. Named entities are recognized, and stemming or lemmatization is applied for pattern recognition.

Applying Predefined Rules: We establish rules based on linguistic principles and apply them to infer semantic connections between words. These principles serve as a guide for our investigation, allowing us to construct the semantic network.

- In the following stage of the research, we sought to encapsulate the semantic network in low-dimensional vectors by treating each vertex as a vector. We employed Graph Neural Networks (GNNs) for this project. GNNs offer a robust framework for learning representations of graphed data. We employed GNNs to explore the semantic network, generating semantic walks prioritized by semantic linkages.

**TABLE 5.** Loss value for beam width.

| Epoch No | Beam width = 4 | Beam width = 8 | Beam width = 12 |
|---|---|---|---|
| 1 | 70.43 | 68.79 | 70.17 |
| 2 | 68.86 | 63.48 | 66.98 |
| 3 | 57.87 | 51.63 | 51.96 |
| ...... | ...... | ...... | ...... |
| ...... | ...... | ...... | ...... |
| 49 | 25.21 | 22.55 | 23.28 |
| 50 | 24.31 | 18.58 | 20.47 |

**TABLE 6.** Setup the settings for the sensitivity analysis of batch size.

| Parameter | Value |
|---|---|
| Units | 300 |
| Hidden layers | 2 |
| Learning rate | 0.004 |
| Embedding size | 128 |
| Epochs | 50 |
| Beam width | 8 |
| Probability | 0.85 |

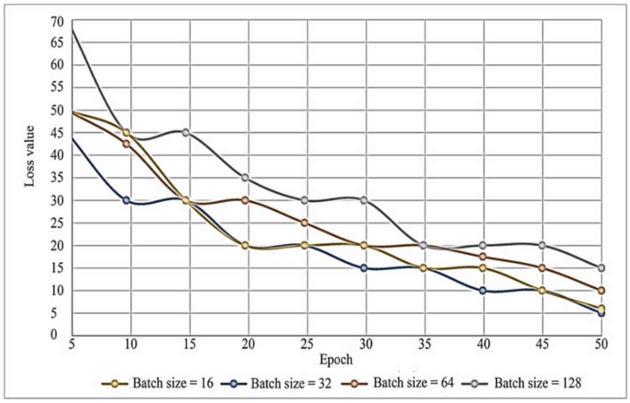


**FIGURE 7.** Study of batch size on testing dataset.

**TABLE 7.** Setup the settings for the sensitivity analysis of number of units.

| Parameter | Value |
|---|---|
| Hidden layers | 2 |
| Learning rate | 0.004 |
| Embedding size | 128 |
| Epochs | 50 |
| Beam width | 8 |
| Batch size | 32 |
| Probability | 0.85 |

We next trained the vertices' representations by maximizing the semantic neighborhood objective with techniques such as Stochastic Gradient Descent (SGD) and negative sampling. This strategy allowed us to successfully combine the semantic network into low-dimensional vectors, allowing for further analysis and processing.

- Deep learning played a pivotal role in conducting abstractive Arabic text summarization across three distinct studies. We trained independent models for abstractive summarization of Arabic text using the provided dataset. To gauge

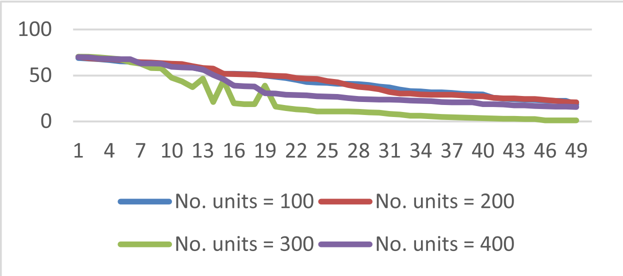


**FIGURE 8.** Results for the testing dataset's unit count.

**TABLE 8.** Setup the settings for the sensitivity analysis of batch size.

| Parameter | Value |
|---|---|
| Number of units | 300 |
| Hidden layers | 2 |
| Learning rate | 0.004 |
| Embedding size | 128 |
| Beam width | 8 |
| Batch size | 32 |
| Probability | 0.85 |

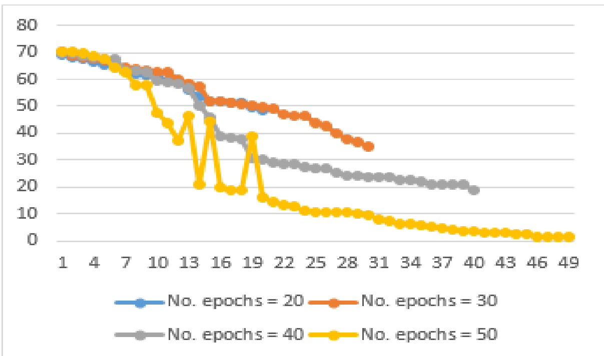


**FIGURE 9.** Results for the testing dataset's No. of epochs.

the efficacy of summarization, we employed eight-fold cross-validation, wherein the dataset was randomly divided into eight equal-sized segments. During training, we utilized 7/8 of the dataset, reserving the remaining portion for testing. Evaluation metrics were averaged over the eight-fold validation process. Our deep learning network featured Long Short-Term Memory (LSTM) units in both the encoder and decoder components. For training, we employed a BasicDecoder, while a BeamSearchDecoder was utilized for inference. To enhance our model's performance, we integrated the LuongAttention method with a global alignment score as our attention mechanism. The architecture of our network comprised two hidden layers, each housing 300 hidden units. Additionally, we fine-tuned parameters such as a beam width of 8, an embedding size of 128, 50 total epochs, a learning rate of 0.004, a batch size of 32, and a keep probability of 0.85 through sensitivity analysis.

### A. EXPERIMENT ANALYSIS

In a subsequent series of experiments, we conducted a sensitivity analysis to evaluate the impact of various parameters on the performance of the deep learning network. The objective of this analysis was to assess how changes in individual parameters affected the overall performance of the deep neural network (NN). Throughout these experiments, we divided the dataset randomly into two parts: a training set comprising 80% of the data and a testing set comprising the remaining 20%. We utilized the loss value as the primary assessment metric. In each experiment, we varied one parameter while keeping the others constant, allowing us to isolate the effect of each parameter on the network's performance. The parameters examined in the sensitivity analysis included:

- Learning Rate: Three distinct tests were carried out to establish the optimal learning rate. Other parameters were defined, as indicated in Table 3. Three alternative learning rates were tested: 0.020, 0.004, and 0.001. The loss values in Figure 6 demonstrate that the lowest loss value was attained with a learning rate of 0.004.
- Beam width: To find the most efficient beam width, three different tests were carried out. Additional settings were made, as Table 4 illustrates. We looked at four, eight, and twelve beam width values. The loss statistics in Table 5 show that with a beam width of 8, the loss value was minimized.
- Batch size: To ascertain the ideal batch size, four different tests were carried out. Table 6 displays the other experiment parameters that were established. We looked at four distinct batch size values: 128; 64; 32; and 16. According to the loss values shown in Figure 7, the batch size of 32 resulted in the lowest loss value.
- Units: To ascertain the optimal number of units in the deep neural network, we conducted four separate experiments. Table 7 presents the additional experimental parameters that were set for these experiments. We explored four different numbers of units: 100, 200, 300, and 400. The loss values depicted in Figure 8 revealed that 300 units resulted in the lowest loss value achievable.
- Epochs: We conducted four distinct experiments to determine the optimal number of epochs in a deep neural network. Table 8 outlines the additional experimental parameters established for these trials. The number of epochs was investigated at four different values: 20, 30, 40, and 50. As depicted in Figure 9, the loss values continued to decrease after epochs 20, 30, and 40 but plateaued after epoch 50. This indicates that the deep neural network ceased learning after 50 epochs, with further epochs failing to improve learning outcomes. The diminished loss value remained consistent thereafter. Consequently, as the network continued to learn until the 47th epoch, conducting 50 test epochs ultimately yielded superior results.

### B. EVALUATION

The results of all the experiments are compiled in this section. Additionally, the collected data is examined and used to assess how well the suggested MSG-ATS (which makes use of semantic graph neural networks) can enhance the quality of abstractive Arabic text summarization in comparison to the word2vec baseline word embedding model.

The presented summary was automatically assessed using the ROUGE assessment measure. Known as ROUGE, an automated assessment metric for summaries was first presented by Hovy et al. [31]. This metric contrasts a system summary, which is a summary produced by a model, with a set of human-provided reference summaries. Using n-grams, which are subsequences of n words, ROUGE calculates similarity [32], [33]. Numerous ROUGE versions, such as ROUGE-N, ROUGE-L, and ROUGE-S, are used to assess text summarization models. ROUGE-1, for instance, evaluates the degree to which the system and reference summaries overlap with respect to unigrams (single words), whereas ROUGE-2 evaluates the degree to which bigrams (two-word sequences) overlap. ROUGE-N compares higher-order n-grams, bigrams, trigrams, and unigrams between the reference summary and the system to perform a comprehensive evaluation. Let's identify the reference summary (R) as well as the system summary (S). The definitions of recall R and precision P are as follows:

$$R = \frac{\text{number of overlapping words between } S \text{ and } R}{\text{total number of words in } R} \quad (1)$$

$$P = \frac{\text{number of overlapping words between } S \text{ and } R}{\text{total number of words in S}} \quad (2)$$

Let's consider another example:

System summary: ''قفز الثعلب البني السريع فوق الكلب الكسول'' ''The quick brown fox jumps over the lazy dog''

Reference summary: ''الثعلب البني السريع قفز فوق الكلب الكسول'' ''A quick brown fox jumps over a lazy dog''

In this case, ROUGE-1 would calculate recall and precision as 6/8 = 0.75 and 6/9 = 0.67 respectively.

The efficiency of the proposed semantic graph embedding in generating high-quality and relevant summaries is compared with the state-of-the-art word2vec embedding paradigm. Introduced by the Google Research Team in 2013, Word2vec is a two-layer neural network that ''vectorizes'' words and comprehends text. It takes a corpus of text as input and outputs a collection of vectors representing the words in the corpus. Word2vec aims to group vectors with similar words in vector space, essentially seeking word similarities using mathematical methods. Vector-based word representation is a fundamental component of most NLP systems, with Word2vec finding applications in various domains, including sentiment analysis and plagiarism detection. In our study, the performance of the proposed MSG-ATS model is automatically compared with two versions of Word2vec on the testing dataset detailed in the dataset section. The MSG-ATS model's starting vectors for text summarization were generated in the first iteration using the specified dataset as training data. In the second iteration, a uniform random initial vector was assigned to each word. We employed ROUGE, an evaluation metric, to assess the performance of the three models using the aforementioned dataset. The results are summarized in Table 9. Notably, the MSG-ATS model emerged as the best performer for text summarization, boasting an outstanding F-measure of 0.0579. Compared to the Pre-trained Word2vec model, the MSG-ATS model demonstrated significant improvements in accuracy (42.4%), recall (23.8%), and overall performance (38.3%).

The baseline models utilized in this investigation, pre-trained Word2vec and random Word2vec, are well-established methodologies in natural language processing. Word2vec has become extensively used for its ability to construct word embeddings that reflect semantic similarity. It does not, however, directly represent sentence syntactic structure and instead depends on the co-occurrence of terms inside a certain timeframe. This shortcoming frequently results in less accurate performance in jobs requiring deeper semantic knowledge, such as abstractive summarization.

**TABLE 9. Evaluation ROUGE result for different models.**

| | Pre-trained word2vec | Random word2vec | MSG-ATS |
|---|---|---|---|
| Precision | 0.0326 | 0.0492 | 0.0554 |
| Recall | 0.0342 | 0.0492 | 0.0608 |
| F-measure | 0.0334 | 0.0490 | 0.0579 |

In contrast, the MSG-ATS model makes use of semantic graph neural networks, which can capture more complicated word associations by taking into account both syntactic and semantic contexts. This method allows for creating more cohesive and contextually relevant summaries, as indicated by higher ROUGE ratings. MSG-ATS achieved the primary objective of the trials, despite the relatively low ROUGE performance of all three models. Notably, it outperformed word2vec, the baseline word embedding model, in both versions. A number of approaches might be investigated to improve these outcomes, such as: (1) Increasing the volume of training data, since the quantity of data available usually affects the performance of any deep learning model [34]. (2) Improving the dependency parser's precision to identify more exact word dependency links. (3) Expanding the scope of semantic linkages to improve the models' semantic representation.

Furthermore, a manual evaluation was carried out to examine the summaries' quality even more. The selection of human reviewers was based on their proficiency in Arabic and prior experience with text summarizing assignments. Experts in computational linguistics, linguists, and NLP specialists with graduate degrees in Arabic were among the assessors. The following were the criteria used in the manual evaluation:

Relevance: how well the major concepts of the original text are reflected in the summary.

Coherence: The readability and logical flow of the summary.

Conciseness: The summary's conciseness while keeping important information.

Fluency: The summary's grammatical soundness and smooth flow.

Each summary was graded on a 1–5 scale for each category, with 1 being the lowest and 5 being the greatest. The manual assessment findings supported the automated ROUGE scores, with the MSG-ATS model regularly surpassing the Word2vec models in creating more relevant, coherent, succinct, and fluent summaries.

## VI. CONCLUSION AND FUTURE WORK

The proposed Multi-level Semantic Graph Arabic Text Summarization (MSG-ATS) model presents a comprehensive framework for abstractive single-statement Arabic text summarization. By seamlessly integrating pre-processing methodologies, feature extraction mechanisms, semantic graph construction, Graph Neural Network (GNN) embedding techniques, and deep neural network (NN) processing algorithms, MSG-ATS aims to produce concise and coherent summaries that faithfully capture the essence of the original Arabic text. A notable highlight of MSG-ATS is its significant outperformance compared to the baseline word embedding model (word2vec), evident in its impressive F-measure of 0.0579. This accomplishment marks a substantial advancement, showcasing a remarkable 42.4% enhancement in precision, 23.8% in recall, and an impressive 38.3% overall improvement over the baseline model. While modest ROUGE scores were observed across all models, these results underscore MSG-ATS's efficacy in surpassing the baseline, highlighting its potential for further enhancement through strategies such as increasing training data volume, refining dependency parsing accuracy, and expanding semantic linkage incorporation. These findings hold promise for advancing the field of Arabic text summarization, contributing to ongoing efforts aimed at enhancing the quality and efficiency of summarization techniques in Arabic language processing.

Despite these hopeful findings, the study had significant limitations. One important restriction is that all models got relatively low ROUGE ratings, indicating space for improvement in the quality of the generated summaries. Furthermore, the training dataset was limited in size and variety, which may have hampered the model's performance and generalizability. The precision of dependency parsing and the extent of semantic links integrated into the model are further opportunities for development. Furthermore, the present approach is built for single-document summarization and does not account for variation between Arabic dialects, which may restrict its usefulness in a variety of real-world contexts.

In future work, further exploration into expanding the training dataset with more diverse and extensive Arabic text sources could significantly enhance the model's performance. Additionally, refining the accuracy of dependency parsing and improving semantic linkage algorithms may offer substantial gains in summary coherence and relevance. Exploring advanced deep learning architectures and hybrid approaches combining extractive and abstractive techniques could also provide new avenues for improving the MSG-ATS framework. Lastly, extending the model to handle multi-document summarization and other Arabic dialects would broaden its applicability and robustness, further solidifying its utility in real-world Arabic language processing tasks.

## CONFLICTS OF INTEREST

The author declares no conflicts of interest.

## ACKNOWLEDGEMENT

The authors would like to thank the Deanship of Scientific Research, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia, for supporting this research via funding from Prince Sattam Bin Abdulaziz University project number (PSAU/2024/R/1445).

## REFERENCES

[1] A. P. Widyassari, S. Rustad, G. F. Shidik, E. Noersasongko, A. Syukur, and A. Affandy, "Review of automatic text summarization techniques & methods," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 4, pp. 1029–1046, 2022.

[2] K. J. Abdelqader, A. Mohamed, and K. Shaalan, "Systematic review of automatic Arabic text summarization techniques," in *Proc. Int. Conf. Variability Sun Sun-Like Stars: From Asteroseismol. Space Weather*. Singapore: Springer, 2023, pp. 783–796.

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

[4] J. Wang and Y. Dong, "Measurement of text similarity: A survey," *Information*, vol. 11, no. 9, p. 421, Aug. 2020.

[5] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang, and Y. Chang, "Structure-augmented text representation learning for efficient knowledge graph completion," in *Proc. Web Conf.*, Apr. 2021, pp. 1737–1748.

[6] P. Kapanipathi et al., "Leveraging abstract meaning representation for knowledge base question answering," 2020, *arXiv:2012.01707*.

[7] N. Bölücü and B. Can, "Turkish universal conceptual cognitive annotation," in *Proc. 13th Lang. Resour. Eval. Conf.*, Jun. 2022, pp. 89–99.

[8] M. N. Y. Ali, M. L. Rahman, J. Chaki, N. Dey, and K. C. Santosh, "Machine translation using deep learning for universal networking language based on their structure," *Int. J. Mach. Learn. Cybern.*, vol. 12, no. 8, pp. 2365–2376, Aug. 2021.

[9] W. Etaiwi and A. Awajan, "Graph-based Arabic text semantic representation," *Inf. Process. Manage.*, vol. 57, no. 3, May 2020, Art. no. 102183.

[10] H. Liang, X. Sun, Y. Sun, and Y. Gao, "Text feature extraction based on deep learning: A review," *EURASIP J. Wireless Commun. Netw.*, vol. 2017, no. 1, pp. 1–12, Dec. 2017.

[11] L. M. Al Qassem, D. Wang, Z. Al Mahmoud, H. Barada, A. Al-Rubaie, and N. I. Almoosa, "Automatic Arabic summarization: A survey of methodologies and systems," *Proc. Comput. Sci.*, vol. 117, pp. 10–18, Jan. 2017.

[12] R. Elbarougy, G. Behery, and A. E. Khatib, "Graph-based extractive Arabic text summarization using multiple morphological analyzers," *J. Inf. Sci. Eng.*, vol. 36, no. 2, p. 347, 2020.

[13] G. Amoudi, A. Almansour, and H. S. Alghamdi, "Improved graph-based Arabic hotel review summarization using polarity classification," *Appl. Sci.*, vol. 12, no. 21, p. 10980, Oct. 2022.

[14] A. M. Azmi and N. I. Altmami, "An abstractive Arabic text summarizer with user controlled granularity," *Inf. Process. Manage.*, vol. 54, no. 6, pp. 903–921, Nov. 2018.

[15] N. Burmani, H. Alami, S. Lafkiar, M. Zouitni, M. Taleb, and N. E. Nahnahi, "Graph based method for Arabic text summarization," in *Proc. Int. Conf. Intell. Syst. Comput. Vis. (ISCV)*, May 2022, pp. 1–8.

[16] 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.

[17] D. Anand and R. Wagh, "Effective deep learning approaches for summarization of legal texts," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 5, pp. 2141–2150, May 2022.

[18] M. Alabdulrahman, R. Khayyat, K. Almowallad, and Z. Alharz, "Sarid: Arabic storyteller using a fine-tuned LLM and text-to-image generation," in *Proc. 16th Int. Conf. Comput. Autom. Eng. (ICCAE)*, Mar. 2024, pp. 1–5.

[19] R. Alqaisi, W. Ghanem, and A. Qaroush, "Extractive multi-document Arabic text summarization using evolutionary multi-objective optimization with K-medoid clustering," *IEEE Access*, vol. 8, pp. 228206–228224, 2020.

[20] M. Mojrian and S. A. Mirroshandel, "A novel extractive multi-document text summarization system using quantum-inspired genetic algorithm: MTSQIGA," *Expert Syst. Appl.*, vol. 171, Jun. 2021, Art. no. 114555.

[21] 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.

[22] M. Belguith, C. Aloulou, and B. Gargouri, "Aspect level sentiment analysis based on deep learning and ontologies," *Social Netw. Comput. Sci.*, vol. 5, no. 1, p. 58, Dec. 2023.

[23] W. H. Park, I. F. Siddiqui, D. R. Shin, and N. M. F. Qureshi, "NLP-based subject with emotions joint analytics for epidemic articles," *Comput., Mater. Continua*, vol. 73, no. 2, pp. 2985–3001, 2022.

[24] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, 2018.

[25] M. Gamal, H. F. Hamed, M. A. Salam, and S. Sweidan, "Novel transformer-based approach enhanced by reinforcement learning and attention mechanisms," *J. Southwest Jiaotong Univ.*, vol. 58, no. 6, pp. 46–59, 2023.

[26] L. Deng and Y. Liu, Eds., *Deep Learning in Natural Language Processing*. Singapore: Springer, 2018.

[27] H. N. Fejer and N. Omar, "Automatic multi-document Arabic text summarization using clustering and keyphrase extraction," *J. Artif. Intell.*, vol. 8, no. 1, pp. 1–9, Dec. 2014.

[28] M. El-Haj, U. Kruschwitz, and C. Fox, "Using mechanical Turk to create a corpus of Arabic summaries," in *Proc. Lang. Resour. (LRs) Human Lang. Technol. (HLT) Semitic Lang. workshop held Conjunct With 7th Int. Lang. Resour. Eval. Conf. (LREC)*, Valletta, Malta, May 2010, pp. 36–39.

[29] C. Napoles, M. Gormley, and B. Van Durme, "Annotated gigaword," in *Proc. Joint Workshop Autom. Knowl. Base Construct. Web-Scale Knowl. Extraction*, Montreal, QC, Canada, Jun. 2012, pp. 95–100.

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

[31] E. Hovy, C. Y. Lin, and L. Zhou, "Evaluating DUC 2005 using basic elements," in *Proc. DUC*, 2005, pp. 1–6.

[32] J. Steinberger and K. Ježek, "Evaluation measures for text summarization," *Comput. Informat.*, vol. 28, no. 2, pp. 251–275, 2009.

[33] T. Mikolov, K. Chen, G. S. Corrado, and J. A. Dean, "Computing numeric representations of words in a high-dimensional space," U.S. Patent 9 464 037, May 19, 2015.

[34] J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification," *Comput. Electron. Agricult.*, vol. 153, pp. 46–53, Oct. 2018.

**MUSTAFA ABDUL SALAM** was born in Sharkia, Egypt, in November 1981. He received the B.Sc. degree (Hons.) from the Faculty of Computers and Informatics, Zigzag University, Egypt, in 2003, the master's degree in information systems from the Faculty of Computers and Information, Menoufia University, Egypt, in 2009, with a focus on hybrid machine learning and bio-inspired optimization algorithms, and the Ph.D. degree in information systems from the Faculty of Computers and Information, Cairo University, Egypt. He is currently an Assistant Professor with the AI Department, Faculty of Computers and AI, Benha University, Egypt. He has worked on several research topics. He has contributed more than 25 technical articles in the areas of neural networks, support vector machines, optimization, time series prediction, extreme learning machines, and hybrid AI models in international journals, international conferences, local journals, and local conferences. His research interests include machine learning, big data, stream data mining, and deep learning.

**MOHAMED ALDAWSARI** received the Ph.D. degree in computer science from Florida International University. He is currently the Head of the Computer Engineering and Information Department, Prince Sattam Bin Abdulaziz University. He has published several papers in prestigious conferences and journals in the NLP field. His research interests include AI, with a specific focus on natural language processing (NLP).

**MOSTAFA GAMAL** received the B.Sc. degree from the Faculty of Computers and Artificial Intelligence, Benha University, Egypt, in 2018, and the master's degree in computer science, in 2021, with a focus on natural language processing and swarm optimization techniques. He is currently pursuing the Ph.D. degree in artificial intelligence with Benha University. He is an Assistant Lecturer with the Artificial Intelligence Department, Egyptian Russian University. His research interests include natural language processing, machine learning, deep learning, and optimization.

**HESHAM F. A. HAMED** was born in Giza, Egypt, in 1966. He received the B.Sc. degree in electrical engineering and the M.Sc. and Ph.D. degrees in electronics and communications engineering from Minia University, El Minia, Egypt, in 1989, 1993, and 1997, respectively. He was the Dean of the Faculty of Engineering, Minia University. He was a Visiting Researcher with Ohio University, Athens, OH, USA. From 1989 to 1993, he was a Teacher Assistant with the Electrical Engineering Department, Minia University. From 1993 to 1995, he was a Visiting Scholar with Cairo University, Cairo, Egypt. From 1995 to 1997, he was a Visiting Scholar with Texas A&M University, College Station, TX (with the Group of VLSI), USA. From 1997 to 2003, he was an Assistant Professor with the Electrical Engineering Department, Minia University, where he was an Associate Professor, from 2003 to 2005. He is currently the Dean of the Faculty of Artificial Intelligence, Russian University, Cairo. He has published more than 250 articles. His research interests include artificial intelligence applications, machine learning, deep learning, analog and mixed-mode circuit design, low voltage low power analog circuits, current mode circuits, nano-scale circuits design, and FPGA.

**SARA SWEIDAN** received the B.Sc., M.Sc., and Ph.D. degrees in computer science and artificial intelligence from Mansoura University, Mansoura, Egypt, in 2002, 2011, and 2019, respectively. She is currently an Assistant Professor of artificial intelligence with the Department of Artificial Intelligence, Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt. Her current research interests include machine learning, computer vision, optimization algorithms, data science, quantum computing and biomedical informatics, sentiment analysis, and large language models.

• • •