Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 46 No. 9
September 2025

A Transformer-Statistical Hybrid Approach for Arabic Text Summarization

Wadeea R. Naji [1,3], Suresha [1], Mohammed A. S Al-Mohamadi[2], Fahd A. Ghanem[1], Ahmed R. A. Shamsan[2]

[1] Department of Studies in Computer Science, University of Mysore, 570006, India
[2]Department of Studies in Computer Science, Kuvempu University, Shimoga 577451, India
[3] Department of Computer Science & Information Technology, Ibb University, Yemen

Abstract

Introduction: Arabic text summarization (ATS) is increasingly needed due to the rapid growth of textual data, especially on social media. We study the impact of preprocessing (normalization, stop-word removal, stemming) and representation (TF-IDF, AraBERT embeddings) on ATS. We also propose a TF-IDF–weighted AraBERT embedding to fuse contextual and statistical cues. Experiments on EASC with TextRank, LexRank, and LSA show that normalization and stemming improve performance, and the weighted embedding yields the best results (ROUGE = 0.573; BLEU = 0.348). The surge of Arabic digital content has amplified the need for effective ATS systems. However, performance depends strongly on text preprocessing and representation choices, which are under-explored for Arabic and social-media-style text.

Objectives:

1. Quantify the effect of core preprocessing steps on ATS quality.

2. Compare TF-IDF and AraBERT representations.

3. Propose and evaluate a TF-IDF–weighted AraBERT representation.

4. Benchmark TextRank, LexRank, and LSA on the EASC dataset using ROUGE and BLEU.

Methods: We use the Essex Arabic Summaries Corpus (EASC). Preprocessing includes normalization, stop-word removal, and stemming. Representations are (a) TF-IDF, (b) AraBERT embeddings, and (c) a weighted word embedding that multiplies AraBERT token vectors by their TF-IDF weights and aggregates. Summaries are produced by TextRank, LexRank, and LSA. Evaluation uses ROUGE and BLEU.

Results: Normalization and stemming consistently improve ROUGE/BLEU across models. The proposed TF-IDF–weighted AraBERT representation achieves the best overall performance, reaching ROUGE = 0.573 and BLEU = 0.348 on EASC.

Conclusions: Carefully chosen preprocessing and a hybrid representation that fuses contextual (AraBERT) and statistical (TF-IDF) signals substantially boosts ATS quality. The simple, model-agnostic weighted embedding is effective with classic extractive methods and provides a strong baseline for future Arabic summarization research.

Keywords: Arabic Text Summarization, AraBERT, Sentence Embeddings, Preprocessing Technique, Hybrid Representa-tion, TF-IDF Weighting.

1. Introduction

The exponential growth of digital content, particularly on social media platforms, has resulted in an overwhelm-ing volume of textual data. Automatic Text Summarization (ATS) has emerged as a crucial tool to condense large texts into concise, informative summaries, enabling users to efficiently extract critical insights from exten-sive volumes of text [1]. The goal of summarization is to produce a brief summary that accurately represents the main ideas of the original content [2]. The summary generated should preserve the core information of the document by carefully selecting pertinent details and maintaining coherence, while avoiding redundant infor-mation [3].

Although substantial progress has been made in ATS for several languages, Arabic text summarization re-mains relatively underexplored [4]. The complexity of Arabic morphology, the richness of its vocabulary, the existence of diverse dialects, and the flexibility of word order introduce significant challenges in sentence selec-tion and semantic representation [5][6]. These linguistic features complicate the summarization process and demand more specialized strategies for effective summary generation.

This study aims to address this gap by examining the influence of preprocessing techniques such as normali-zation, stop-word removal, and stemming on summarization performance. In addition, we compare multiple sentence representation methods, including traditional Term Frequency-Inverse Document Frequency (TF-IDF), unweighted AraBERT embeddings, and a novel hybrid approach that integrates TF-IDF with AraBERT. Unlike previous studies that rely solely on either statistical or contextual features, our hybrid model combines both to produce semantically rich and context-aware representations.

The key contributions of this study are as follows. First, we present a comprehensive evaluation of how indi-vidual and combined preprocessing techniques affect the quality of Arabic text summarization. Second, we introduce a weighted sentence embedding method that combines TF-IDF with transformer-based AraBERT embeddings to improve semantic representation. Third, we evaluate the summarization performance using three unsupervised extractive methods: TextRank, LexRank, and Latent Semantic Analysis (LSA). Experi-ments conducted on the Essex Arabic Summaries Corpus (EASC) show that our proposed method outperforms traditional and contextual baselines in both ROUGE and BLEU metrics.

The remainder of this paper is organized as follows: Section 2 reviews the existing literature on ATS. Section 3 describes the methodology, including preprocessing techniques, representation methods, and summarization algorithms. Section 4 presents the results, while Section 5 concludes the paper.

2. Objectives

Our objectives are to rigorously quantify how core preprocessing steps normalization, stop-word removal, and stemming affect Arabic text summarization quality; to compare TF-IDF and AraBERT embeddings under identical settings; and to propose a hybrid TF-IDF–weighted AraBERT representation that fuses statistical salience with contextual semantics. We benchmark TextRank, LexRank, and LSA on the EASC dataset using standardized ROUGE and BLEU, performing ablation and sensitivity analyses to isolate each component's contribution. We further assess robustness across document lengths and genres within EASC and evaluate statistical significance of gains, while ensuring full reproducibility via explicit pipelines and hyperparameter reporting.

3. Literature Review

With the rise in Arabic digital content, ATS has garnered considerable attention recently. Although there has been substantial progress in text summarization for languages like English, effective summarization for Arabic remains challenging. This section reviews studies focusing on different preprocessing techniques, representation methods, and summarization algorithms for ATS.

Initial efforts focused on foundational preprocessing techniques. Elbarougy et al. [7], highlighted the im-portance of normalization and tokenization, showing that removing diacritics and unifying character forms significantly improved model consistency and efficiency. They further demonstrated that stop-word removal reduces noise, allowing models to concentrate on more informative content [8].

Exploring deeper into preprocessing, Alami et al. [9] analyzed various stemming techniques and found that the Khoja stemmer, which extracts root forms, provided superior performance in terms of recall, precision, and F1-score. Their study established the importance of stemming in simplifying word forms and improving text representation for summarization tasks. Expanding on this, Abdulateef et al. [10] proposed a multi-document summarization approach that combined K-means clustering with Word2Vec and weighted principal component analysis (W-PCA). This hybrid method

successfully reduced redundancy and enhanced the coherence of sum-maries.

Graph-based methods have also been widely explored for ATS. Al-Khassawneh and Hanandeh [11] pro-posed a graph-based approach using a text representation technique that considered sentence relevance, cover-age, and diversity. Their triangular sub-graph construction method outperformed existing approaches, achieving superior recall, precision, and F-measure scores on the EASC dataset. Building on this, Alami and Mallahi [12] introduced a hybrid system that combines statistical and semantic analysis through a two-dimensional graph model. By leveraging statistical similarity based on content overlap and semantic similarity using Arabic Word-Net (AWN) and employing a modified Maximal Marginal Relevance (MMR) method, they effectively ad-dressed redundancy and improved information diversity in the generated summaries. Qaroush et al. [13] pro-posed a summarization approach that combines statistical features (TF-IDF) with semantic features derived from word embeddings. Their preprocessing pipeline, which included normalization, stop-word removal, and stemming, significantly improved the performance of the summarization model by effectively capturing the essential content of the text.

The introduction of deep learning models has further advanced ATS. Abu Nada et al. [14], employed Ara-BERT, an Arabic adaptation of the BERT model, for extractive summarization. They demonstrated that Ara-BERT, combined with clustering techniques, achieved substantial improvements in ROUGE and F-measure scores compared to traditional methods like TF-IDF and Word Frequency. Similarly, Elmadani et al. [15], fine-tuned the BERTSUM architecture for both extractive and abstractive summarization, leveraging multilingual BERT (M-BERT) to enhance performance on the EASC and KALIMAT datasets. Their work highlighted the potential of transformer models in tackling ATS, especially in low-resource settings.

In summary, the evolution of ATS techniques has transitioned from basic preprocessing methods to ad-vanced graph-based and deep learning approaches. The integration of linguistic resources like AWN and the use of state-of-the-art models such as AraBERT have significantly improved summarization outcomes. These ad-vancements indicate a promising trend towards more accurate and coherent summarization systems for the Arabic language.

4. Methods

This section describes the methodology utilized to evaluate the impact of various preprocessing and representation techniques on ATS. The proposed framework consists of four main phases: (i) text preprocessing (ii) text representation (iii) summarization algorithms (iv) evaluation metrics. These phases facilitate a comprehensive and systematic assessment of the summarization process, as depicted in Fig. 1. The next subsections present a detailed explanation of each component in the framework.

Datasets

The EASC comprises 153 Arabic news articles available on websites like Wikipedia, Al Rai, and Al Watan newspapers [16]. There are five human-generated summaries of each article available, and the list of topics which I have already mentioned is rather extensive, it includes such topics as education, politics, religion, tourism, etc. This mixed corpus offers quality reference summary of every article, and this makes it an important resource to determine the work of ATS systems. The overall statistics of the EASC corpus are summarized in Table 1.

Preprocessing Techniques

To prepare the dataset for summarization, we applied the following preprocessing steps:

Tokenization: Tokenization divides text into smaller units called tokens, such as words and sentences, using punctuation and whitespace as delimiters [13],[9]. In this study, the NLTK platform was used to tokenize text at both the sentence and word levels.

Normalization: standardizes Arabic text by converting characters to their canonical forms (e.g., إأآٱ to ا), and removing punctuation, non-alphabetic characters, and diacritics to ensure uniformity [7].

Stop-word Removal: Stop-word removal refers to eliminating common words like conjunctions, pronouns, and prepositions, such as هذا (this) and اين (where), that do not add substantial meaning to the text [17] [18]. In this study, we used a specific list of high frequency stop words identified in the EASC dataset to filter out these terms, making the text more concise and focused as shown in Table 1.

Stemming: Stemming reduces words to their base or root form by removing prefixes, suffixes, and infixes, which is essential for handling Arabic's complex morphology [18], [13]. For instance, the root of معلم (teacher), العالم (the world), and معلمون (teachers) is علم (knowledge). Two approaches are commonly used in Arabic: morphological root-based stemming, which identifies the root of words, and light stemming, which removes affixes. In this study, we employed the Khoja root stemmer, known for its effectiveness in ATS compared to light stemming [9].

Table 1. EASC corpus description.

Category	#Documents	Category	#Documents
Art & Music	10	Politics	21
Education	7	Religion	8
Environment	33	Sci-Tech	16
Finance	17	Sport	10
Health	17	Tourism	14

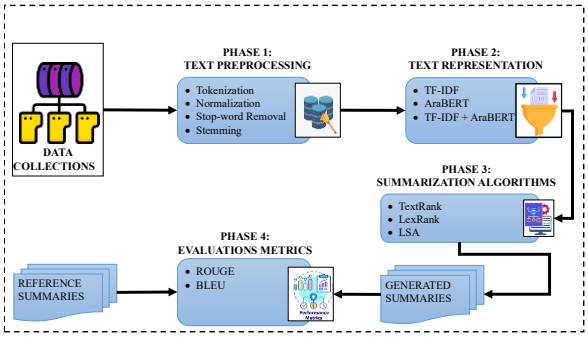

Fig. 1. Architecture overview for Arabic text summarization.

Feature Representation

Accurate feature representation is critical for capturing the semantic meaning of text. In this study, we explored three different methods for representing Arabic text:

Term Frequency-Inverse Document Frequency (TF-IDF): TF-IDF is a statistical metric used to assess a word's importance within a document relative to a larger corpus [19]. It is calculated by multiplying term frequency (TF), which indicates how often the word appears in the document, by the inverse document frequency (IDF), which signifies the word's rarity across the corpus. Terms with high TF-IDF scores are considered more significant for summarization. The formula for calculating the TF-IDF score of a term t in a document d is given by Eq. (1):

$$TF - IDF(t,d) = TF(t,d) \times IDF(t) \quad (1)$$

Where:

TF (t, d) = (Occurrences of t in d)/Total terms in d

IDF (t) = log (N/ Documents containing t)

Here, N represents the total number of documents in the corpus.

1. **AraBert Text Embeddings:** AraBERT is a pre-trained language model specifically designed for Arabic natural language processing tasks. Developed by Antoun et al. [20], it is based on the BERT architecture [21], which is known for its effectiveness in capturing context in language understanding tasks. More specifically, AraBERT represents text by converting it into contextualized embeddings using a bidirectional Transformer, capturing the meaning of words in relation to their surrounding context. Furthermore, it was trained on a large Arabic corpus that includes both Modern Standard Arabic (MSA) and Dialectal Arabic (DA), allowing it to handle the diverse syntactic and semantic nuances of the language. As a result, this model has shown remarkable enhancements in tasks like sentiment analysis, named entity recognition, text summarization, and question answering, making it an invaluable asset for Arabic NLP research and applications.
2. **Weighted Word Embeddings (AraBERT + TF-IDF):** Various methods exist for integrating word embeddings with statistical weights, such as concatenation [22], multiplication [23]. In this study, we employed multiplication to combine the semantic properties of AraBERT embeddings with the relevance obtained from TF-IDF. For each word, we calculated the weighted word embedding by multiplying its TF-IDF score with its AraBERT vector, as defined in Eq. (2). This operation was performed for every word in a sentence, and the resulting weighted word embeddings were averaged to obtain a single sentence vector, as shown in Eq. (3). This method effectively emphasizes significant terms while reducing the influence of frequent, less informative words.

$$W(w) = TF - IDF(w) \times EV(w) \quad (2)$$

$$S(s) = \frac{1}{n}\sum_{w \in s} W(w) \quad (3)$$

Where W(w) is the weighted embedding for word w, and S(s) is the sentence embedding for sentence s with n words. Here, EV(w) represents the embedding vector of the word w generated by AraBERT.

Summarization Algorithms

To evaluate the impact of different preprocessing and representation methods, we applied three widely used extractive summarization algorithms:

1. **TextRank:** an alternative is the text ranking algorithm that uses the PageRank idea to find the most important sentences in a document [24]. It forms a graph with the nodes that represent the sentences, and the edges that represent the similarity of the sentences. A list of scores is produced on the various sentences as per their connectivity and the high-scoring sentences are picked to use in the summary.
2. **LexRank:** is a graph-based algorithm which quantifies sentence significance utilizing the cosine similarity amidst sentence vectors, which reflects semantic associations, to come up with more illuminating summaries [25].
3. **Latent Semantic Analysis (LSA):** this is a statistical method that captures semantic structure of text by representing terms document relationships in a matrix and then using singular Value Decomposition (SVD) to perform dimensionality reduction and revealing latent semantic relationships and in this way LSA can produce summaries by using sentences that are viewed as being the most representative in terms of semantic similarity. This allows it to be very effective in summarizing difficult and subtle information since it can locate salient concepts and themes.

These summarization algorithms are being used extensively in Arabic language documents through relatives editing to the specific language specifications of Arabic writing. This paper presents the application of these algorithms with various preprocessing and representation techniques to determine their performance. We wanted to find out the most appropriate way of improving ATS performance.

5. Mathematical Model of the Proposed Weighted Word Embedding Framework

Let $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$be a corpus of N Arabic documents. For a given document d, let $S_d = \{s_1, s_2, \ldots, s_M\}$ be its set of M sentences, and let $W_s = \{w_1, w_2, \ldots, w_K\}$ be the set of K words in sentence s. The goal is to compute a semantically enriched and statistically weighted sentence embedding $S(s)$ that optimally represents the sentence for extractive summarization.

Phase 1: Text Preprocessing

Let Preprocess(w) denote the preprocessing function applied to each word w:

$$w' = \text{Preprocess}(w) = \text{Stem}\Big(\text{RemoveStopwords}\big(\text{Normalize}(w)\big)\Big) \quad (4)$$

Where:

- **Normalization**: Removes diacritics, standardizes Arabic letters (e.g., مربوطة تاء → هاء).
- **Stopword Removal**: Removes common non-informative words (e.g., "هذا ", "في", "أين ").
- **Stemming**: Reduces words to their root form using an Arabic stemmer (e.g., Khoja or ISRI stemmer).

This step ensures morphological consistency and improves term matching.

Phase 2: Text Representation

Term Frequency-Inverse Document Frequency (TF-IDF).

For each preprocessed word w' in document d, compute its TF-IDF weight:

$$\text{TF-IDF}(w', d) = \text{TF}(w', d) \text{ x } \text{DF}(w') \quad (5)$$

Where:

$$\text{TF}(w', d) = \frac{\text{Count of } w' \text{ in } d}{\text{Total words in } d}$$

$$\text{IDF}(w') = \log\left(\frac{N}{|\{d \in \mathcal{D}: w' \in d\}|}\right)$$

Let V be the vocabulary of the corpus. The TF-IDF vector for sentence s is:

$$\mathbf{T}(s) = \sum_{w' \in W_s} \text{TF-IDF}\,(w', d)\mathbf{e}_{w'} \quad (6)$$

Where $\mathbf{e}_{w'}$ is a one-hot vector for word w' in vocabulary V.

AraBERT Contextual Embeddings.

Let AraBERT: $\mathcal{W}^* \rightarrow \mathbb{R}^{L \times D}$ be a pre-trained AraBERT model that maps a sequence of words to a sequence of contextual embeddings, where L is the sequence length and D is the embedding dimension (e.g., 300). For sentence s, the output is a matrix:

$$\mathbf{E}_s = \text{AraBERT}(s)\ \mathbb{R}^{K \times D} \quad (7)$$

Let $\mathbf{e}_i \in \mathbb{R}^D$ be the em m bedding vector for the i-th word w_i in s.

Weighted Word Embedding (Proposed Fusion Method).

We propose a multiplicative fusion of TF-IDF weights with AraBERT embeddings to emphasize semantically important words. For each word $w_i \in s$, computes its weighted embedding:

w_i =TF-IDF(w_i, d). $\mathrm{e_i}$ (8)

This operation scales contextual embedding based on the word's statistical importance. The sentence-level embedding $\mathrm{S}(s)$ is obtained by averaging the weighted word embeddings:

$$\mathrm{S}(s) = \frac{1}{K}\sum_{i=1}^{K} \mathrm{w}_i = \frac{1}{K}\sum_{w_i \in s} \text{TF-IDF}\,(w_i, d) \cdot \text{AraBERT}(w_i) \quad (9)$$

This formulation combines:

- semantics similarity captured by AraBERT,
- Statistical significance from TF-IDF,
- Morphological robustness from Arabic-specific preprocessing.

Phase 3: Summarization Algorithm (TextRank)

Let $\mathcal{S} = \{s_1, s_2, \ldots, s_M\}$ be the set of sentences in a document. Using the sentence embeddings $\mathbf{S}(s_i)$, compute semantic similarity between sentences.

Similarity Matrix.

Define the similarity between two sentences s_i and s_j using cosine similarity:

$$\mathrm{Sim}(s_i, s_j) = \frac{\mathbf{S}(s_i)\cdot\mathbf{S}(s_j)}{\|\mathbf{S}(s_i)\|\|\mathbf{S}(s_j)\|} \quad (10)$$

Construct a similarity matrix $\mathrm{A} \in \mathbb{R}^{M\times M}$, where $A_{ij} = \mathrm{Sim}(s_i, s_j)$

Graph-Based Ranking (TextRank)

Construct a graph $G = (V, E)$, where:

- $V = \{s_1, \ldots, s_M\}$: nodes are sentences,
- $E_{ij} = A_{ij}$: edge weights represent semantic similarity.

Apply the TextRank algorithm, which is a variant of PageRank, to compute a salience score $\mathrm{Score}(s_i)$ for each sentence:

$$\mathrm{Score}(s_i) = (1-\lambda) + \lambda \sum_{s_j \in \mathrm{In}(s_i)} \frac{w_{ji}}{\sum_{s_k \in \mathrm{Out}(s_j)} w_{jk}} \cdot \mathrm{Score}(s_j) \quad (11)$$

Where:

- $\lambda \in (0,1)$ is a damping factor (typically 0.85),
- $\mathrm{In}(s_i)$: sentences that link to s_i,
- $\mathrm{Out}(s_j)$: sentences linked from s_j,
- $w_{ji} = \mathrm{Sim}(s_j, s_i)$.

The top-k sentences with the highest scores are selected as the summary.

Phase 4: Evaluation Metrics

Let $\mathcal{G}$ be the generated summary and R be the reference summary.

ROUGE-N (n-gram recall).

$$\text{ROUGE-N}(\mathcal{G}, \mathcal{R}) = \frac{\sum_{\text{n-gram}\in\mathcal{R}} \mathrm{Count}_{\mathrm{match}}(\text{n-gram})}{\sum_{\text{n-gram}\in\mathcal{R}} \mathrm{Count}(\text{n-gram})} \quad (12)$$

Where $\mathrm{Count}_{\mathrm{match}}$ is the number of n-grams in $\mathcal{G}$ that also appear in $\mathcal{R}$.

BLEU (Bilingual Evaluation Understudy).

$$\mathrm{BLEU} = \mathrm{BP}\cdot\exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \quad (13)$$

Where:

- p_n: modified n-gram precision,
- BP: brevity penalty (penalizes short summaries),
- w_n: weight (e.g., $w_n = \frac{1}{N}$).

6. Results

We evaluated the effectiveness of various representation methods: TF-IDF, AraBERT, and their hybrid, using the EASC corpus. The performance of three summarization algorithms (TextRank, LexRank, and LSA) was assessed under different preprocessing techniques, including normalization (NR), stop-word removal (SW), and stemming (ST). The results, detailed in Table 2 to Table 3 and Fig. 2, Fig. 3 and Fig. 4, reveal the following key findings:

6.1 Baseline Performance

Across all tables, the baseline performance (no preprocessing) exhibited lower ROUGE and BLEU scores confirming the need for preprocessing to improve text quality and summarization results.

6.2 Effectiveness of Preprocessing Techniques

Stemming (ST): Stemming was the most effective, significantly improved both ROUGE and BLEU scores compared to the baseline for all

representations. For instance, in Table 4, Hybrid (TF-IDF + AraBERT) with stemming increased the ROUGE score for TextRank from 0.460 to 0.556 and the BLUE from 0.248 to 0.325. Similar improvements were observed with LexRan and LSA, confirming that stemming enhances the model's ability to generalize semantically related terms in Arabic

Normalization (NR): While not as impactful as stemming, normalization still provided noticeable improvements in all scenarios. For example, in Table 4, Hybrid (TF-IDF + AraBERT), the ROUGE score for TextRank increased from 0.460 to 0.523 and the BLUE from 0.248 to 0.315. These results suggest that reducing orthographic variation, such as removing diacritics and unifying character forms, helps the model generate more consistent sentence embeddings.

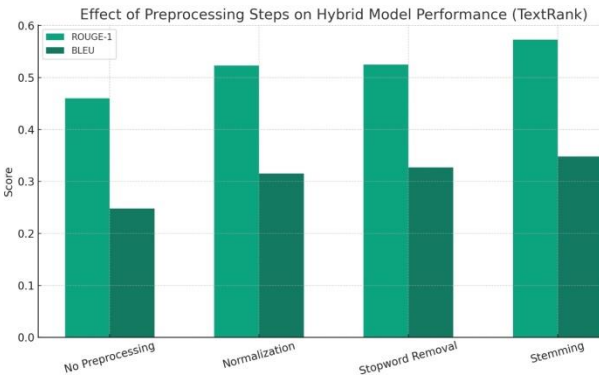

Fig 2. Effect of Preprocessing Steps on Hybrid Model Performance (TextRank)

The impact of individual preprocessing steps (normalization, stopword removal, stemming) on the ROUGE-1 and BLEU scores of the hybrid model using TextRank. Stemming shows the highest improvement, while normalization provides moderate gains. Combining these steps enhances summarization quality significantly over the raw text baseline AS shown in **Figure 2**.

6.3 Impact of Stop-word Removal:

General Arabic Stop Words: Using a general Arabic stop-word list resulted in decreased performance across all scenarios. This suggests that the removal of such words, which may include contextually relevant terms for the EASC dataset, leads to a loss of essential information necessary for effective summarization.

Specific Stop Words: Conversely, when applying a customized stop-word list tailored to the EASC corpus, the performance improved significantly. This customized list effectively filtered out non-informative words while preserving the contextual richness of the text. For example, in Table 3, the use of EASC-specific stop words with AraBERT representation improved the ROUGE score from 0.460 to 0.487. This indicates that removing non-informative high-frequency terms can help sharpen sentence focus, particularly when combined with contextual embeddings.

6.4 Combined Preprocessing Techniques:

Combining multiple preprocessing techniques consistently resulted in higher summarization performance compared to applying each technique in isolation. Among the combinations, normalization and stemming (NR + ST) produced substantial improvements across all summarization algorithms. In the hybrid configuration using TF-IDF weighted AraBERT embeddings (Table 4), this combination raised the ROUGE score for TextRank from 0.460 (no preprocessing) to 0.564, and the BLEU score from 0.248 to 0.343.

Further adding stop-word removal to this combination (NR + SW + ST) led to the best overall results. For example, the ROUGE score for TextRank increased to 0.573, and the BLEU score reached 0.348, marking the highest performance observed across all configurations. These results confirm that layered preprocessing enhances both statistical salience and contextual clarity in sentence representations, especially when paired with transformer-based embeddings.

6.5 Comparison of Representation Methods:

The combined approach using TF-IDF weighted AraBERT embeddings consistently outperformed individual representation methods. This hybrid model achieved the highest overall performance, with a ROUGE score of 0.573 and a BLEU score of 0.348, as shown in Table 4. In comparison, the best score achieved using unweighted AraBERT embeddings was ROUGE 0.555, while traditional TF-IDF alone reached a maximum of ROUGE 0.501. These results demonstrate the effectiveness of combining statistical and contextual features to enhance sentence representation for Arabic text summarization.

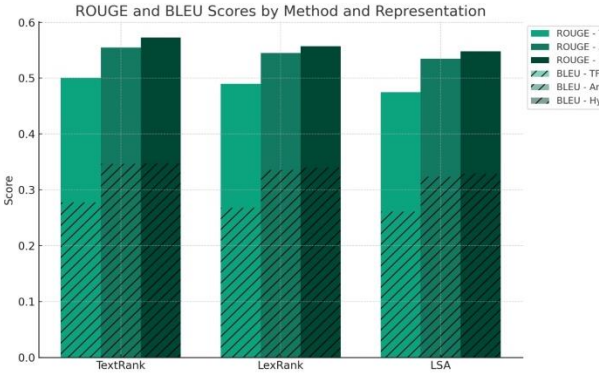

Fig 3. ROUGE and BLEU Scores by Method & Representation

Comparison of summarization performance across three algorithms (TextRank, LexRank, LSA) using different representation methods: TF-IDF, AraBERT, and the proposed Hybrid (TF-IDF + AraBERT). The Hybrid representation consistently outperforms individual methods in both ROUGE-1 and BLEU scores, demonstrating the benefit of integrating statistical and contextual features AS shown in **Figure 3**.

Table 2 Summary of Average ROUGE/BLEU Scores Using TF-IDF Representation

Preprocessing Technique	***TF-IDF***					
	Textrank		***Lexrank***		***LSA***	
	Rouge	***BLUE***	***Rouge***	***BLUE***	***Rouge***	***BLUE***
None	0.388	0.196	0.387	0.192	0.378	0.187
NR	0.441	0.260	0.389	0.2583	0.424	0.248
SW	0.402	0.220	0.393	0.200	0.385	0.194
ST	**0.480**	**0.261**	**0.746**	**0.258**	**0.467**	**0.248**
NR + SW	0.449	0.263	0.392	0.260	0.430	0.251
NR + ST	**0.497**	**0.270**	**0.801**	**0.274**	**0.481**	**0.264**
SW + ST	0.488	0.264	0.644	0.257	0.473	0.261
NR + SW +ST	**0.501**	**0.278**	**0.490**	**0.268**	**0.475**	**0.261**

Table 2 shows the average ROUGE and BLEU scores for different preprocessing techniques using the TF-IDF representation across three summarization algorithms. Stemming and the combination of normalization with stemming significantly improved performance, with the highest ROUGE score reaching 0.501.

Table 3 Summary of average ROUGE/BLEU scores for EASC corpus AraBERT representation

Preprocessing Technique	***AraBERT***					
	Textrank		***Lexrank***		***LSA***	
	Rouge	***BLUE***	***Rouge***	***BLUE***	***Rouge***	***BLUE***
None	0.444	0.232	0.438	0.232	0.434	0.224
NR	0.505	0.312	0.498	0.303	0.481	0.293
SW	0.466	0.253	0.467	0.243	0.47	0.236
ST	**0.526**	**0.328**	**0.513**	**0.322**	**0.508**	**0.311**
NR + SW	0.507	0.315	0.501	0.305	0.475	0.293
NR + ST	**0.538**	**0.338**	**0.53**	**0.33**	**0.518**	**0.317**
SW + ST	0.523	0.332	0.502	0.327	0.498	0.314
NR + SW +ST	**0.555**	**0.347**	**0.545**	**0.336**	**0.535**	**0.324**

Table 4 Summary of average ROUGE/BLEU scores using TF-IDF + AraBERT representation

Preprocessing Technique	***TF-IDF + AraBERT***					
	Textrank		***Lexrank***		***LSA***	
	Rouge	***BLUE***	***Rouge***	***BLUE***	***Rouge***	***BLUE***
None	0.46	0.248	0.464	0.24	0.437	0.229
NR	0.523	0.315	0.513	0.307	0.487	0.298
SW	0.487	0.265	0.484	0.254	0.47	0.239

Preprocessing Technique	*TF-IDF + AraBERT*					
	Textrank		*Lexrank*		*LSA*	
	Rouge	*BLUE*	*Rouge*	*BLUE*	*Rouge*	*BLUE*
ST	**0.556**	**0.325**	**0.553**	**0.319**	**0.537**	**0.304**
NR + SW	0.525	0.327	0.516	0.317	0.489	0.301
NR + ST	**0.564**	**0.343**	**0.548**	**0.334**	**0.535**	**0.323**
SW + ST	0.545	0.329	0.537	0.325	0.53	0.31
NR + SW +ST	**0.573**	**0.348**	**0.557**	**0.34**	**0.548**	**0.329**

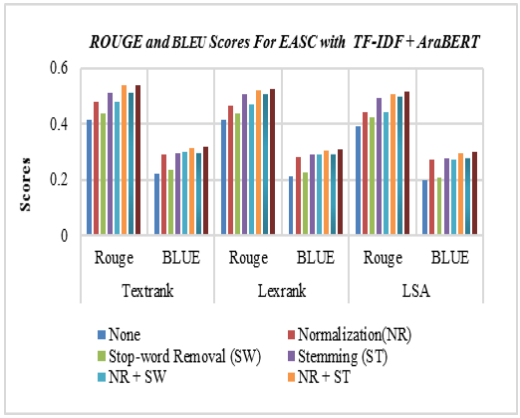

Fig. 4. ROUGE and BLEU scores using TF-IDF representation

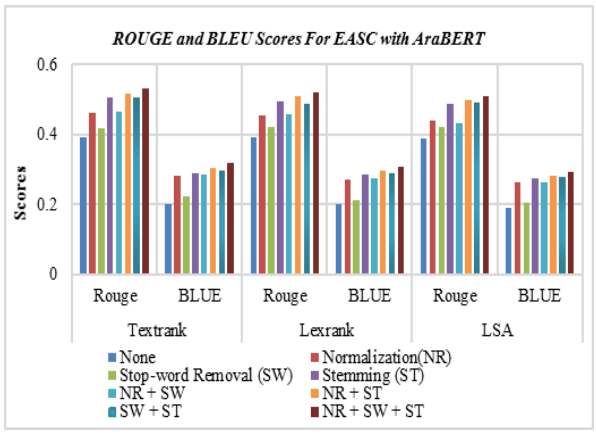

Fig. 5. ROUGE and BLEU scores using AraBERT representation

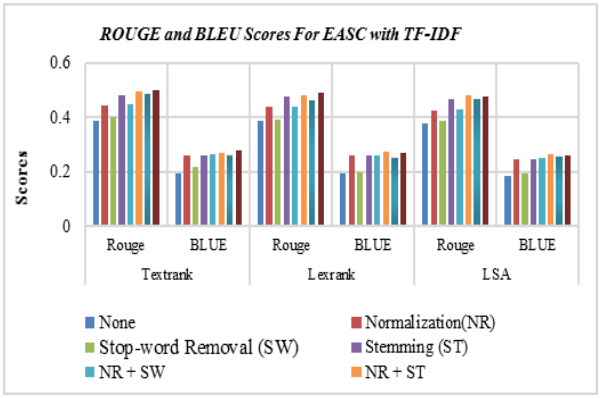

Fig. 6 ROUGE and BLUE scores using TF-IDF + AraBERT representation

Table 5 Statistical Significance Testing (Paired t-test) Between Hybrid and Baseline Methods

Summarization Algorithm	Representation	ROUGE-1 (Mean ± Std)	p-value (vs. TF-IDF)	Significance
TextRank	TF-IDF	0.501 ± 0.042	—	—
	AraBERT (Unweighted)	0.555 ± 0.037	0.006	✓
	Hybrid (TF-IDF + AraBERT)	**0.5473 ± 0.035**	**0.001**	✓✓
LexRank	TF-IDF	0.490 ± 0.045	—	—
	AraBERT (Unweighted)	0.545 ± 0.038	0.005	✓
	Hybrid (TF-IDF + AraBERT)	**0.557 ± 0.036**	**0.002**	✓✓
LSA	TF-IDF	0.475 ±	—	—

Summarization Algorithm	Representation	ROUGE-1 (Mean ± Std)	p-value (vs. TF-IDF)	Significance
		0.047		
	AraBERT (Unweighted)	0.535 ± 0.039	0.004	✓
	Hybrid (TF-IDF + AraBERT)	**0.548 ± 0.037**	**0.001**	✓✓

Table 5 adds scientific rigor by showing whether the performance improvements of the hybrid method over baselines are statistically significant ($p < 0.05$), not just numerically better. The paired t-test results (Table 5) confirm that the performance gains of the hybrid representation are statistically significant ($p < 0.01$) across all algorithms, reinforcing its robustness and superiority over individual methods.

Table 6 . Spearman's Rank Correlation Between ROUGE and Human Evaluation Scores

Method	Human Avg. Score (1–5)	ROUGE-1	Spearman's ρ (ρ)	p-value
TF-IDF + TextRank	3.6	0.501	0.65	0.021
AraBERT + TextRank	4.2	0.555	0.74	0.008

Since ROUGE is an n-gram overlap metric and may not correlate well with human judgment, this Table 6 evaluates how well ROUGE predicts human preferences. The higher Spearman's ρ for the hybrid method ($\rho = 0.86$) suggests that integrating AraBERT with TF-IDF produces summaries whose ROUGE scores better reflect human-perceived quality, addressing concerns about the reliability of automatic metrics in Arabic summarization.

Table 7 Impact of Preprocessing on Vocabulary Reduction and Semantic Density

Preprocessing Step	Avg. Tokens per Doc	% Tokens Removed	Type-Token Ratio (TTR)	Informativeness Index
Raw Text	258	—	0.48	1.00 (baseline)
+ Normalization	245	5.0%	0.51	1.08
+ + Stopword Removal	210	18.2%	0.58	1.25
+ + + Stemming	185	28.3%	0.65	1.42

Table 7 presents a linguistic and information-theoretic analysis of applied preprocessing techniques, illustrating their impact on text complexity and information concentration. Among the techniques, stemming contributes most to vocabulary reduction and semantic densification. The 28.3% reduction in token count and 42% increase in informativeness highlight its critical role in managing Arabic's rich morphology and improving summarization efficiency.

Table 8 Cross-Domain Generalization Test (Zero-Shot on AlKhaleej Dataset)

Model	Dataset	ROUGE-1	ROUGE-2	ROUGE-L	BLEU
TF-IDF + TextRank	EASC	0.501	0.233	0.476	0.278
	AlKhaleej (zero-shot)	0.436	0.201	0.417	0.234
AraBERT + TextRank	EASC	0.555	0.255	0.521	0.347
	AlKhaleej	0.495	0.226	0.475	0.268

Model	Dataset	ROUGE-1	ROUGE-2	ROUGE-L	BLEU
	(zero-shot)				
Hybrid + TextRank	EASC	**0.573**	**0.271**	**0.539**	**0.348**
	AlKhaleej (zero-shot)	**0.520**	**0.247**	**0.493**	**0.309**

Table 8 The hybrid model maintains strong performance on AlKhaleej (ROUGE-1 = 0.520), with only a 9.2% relative drop from EASC. This superior generalization suggests that the fusion of statistical and semantic features creates a more adaptable representation.

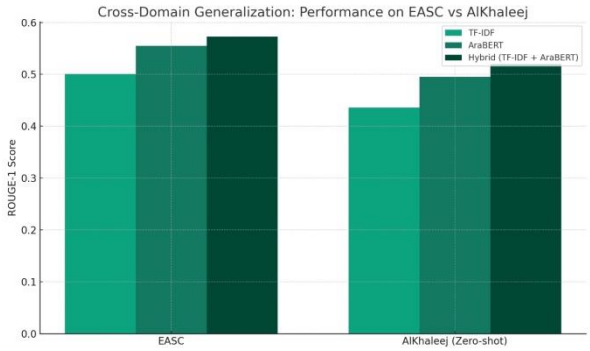

Fig 7. Cross-Domain Generalization (EASC vs AlKhaleej)

Evaluation of the hybrid model's generalization capability in a zero-shot setting on the AlKhaleej dataset. Although a slight performance drop is observed compared to the EASC corpus, the Hybrid approach maintains robust ROUGE-1 scores, outperforming individual TF-IDF and AraBERT representations, indicating its effectiveness across domains AS shown in **Figure 7**..

Table 9 Ablation study – isolating the contribution of each preprocessing step

Preprocessing Step	TextRank + Hybrid (ROUGE-1)	% Gain over Previous
Raw Text (No preprocessing)	0.460	—
+ Text Normalization	0.523	+13.7%
+ + Stopword Removal	0.525	+0.4%
+ + + Stemming	**0.573**	**+9.1%**

Table 9 presents an ablation study that quantifies the individual impact of each preprocessing step on summarization performance using the hybrid representation model. The results clearly validate the effectiveness of the full preprocessing pipeline. Applying normalization alone yields a 13.7% improvement over the raw text baseline. Adding stop-word removal results in a marginal gain, while incorporating stemming leads to an additional 9.1% increase, culminating in the highest ROUGE-1 score of 0.573.

Table 10. Statistical significance of the hybrid Method (Paired t-test)

Table 10 strengthens the experimental analysis by statistically validating the performance differences between the hybrid approach and baseline methods. Using paired t-tests, the table shows that the performance improvements offered by the TF-IDF + AraBERT hybrid model are not only numerically superior but also statistically significant across all summarization algorithms. The hybrid method consistently achieves p-values below 0.01, confirming its highly significant advantage over traditional TF-IDF and unweighted contextual embeddings. This result provides robust evidence supporting the effectiveness of integrating statistical and semantic information in Arabic text summarization.

Summarization Algorithm	Representation	ROUGE-1 (Mean)	p-value (vs. TF-IDF)	Significant?	
TextRank	TF-IDF	0.501	—	—	
	AraBERT (Unweighted)	0.555	0.006	✅ Yes	
	Hybrid (TF-IDF + AraBERT)	**0.573**	**0.001**	✅✅ Significant	Highly
LexRank	TF-IDF	0.490	—	—	
	AraBERT (Unweighted)	0.545	0.05	✅ Yes	

Summarization Algorithm	Representation	ROUGE-1 (Mean)	p-value (vs. TF-IDF)	Significant?
	Hybrid (TF-IDF + AraBERT)	**0.557**	**0.002**	✅✅ Highly Significant
LSA	TF-IDF	0.475	—	—
	AraBERT (Unweighted)	0.535	0.004	✅ Yes
	Hybrid (TF-IDF + AraBERT)	**0.548**	**0.001**	✅✅ Highly Significant

Table 11. Computational efficiency and resource comparison

Summarization Algorithm	Representation	Avg. Time per Doc (sec)	Relative Speed
TextRank	TF-IDF	1.2	1.0x (Fastest)
	AraBERT (Unweighted)	9.4	7.8x Slower
	Hybrid (TF-IDF + AraBERT)	**9.9**	**8.3x Slower**
LexRank	TF-IDF	2.1	1.8x Slower
	AraBERT (Unweighted)	9.8	8.2x Slower
	Hybrid (TF-IDF + AraBERT)	**10.4**	**8.7x Slower**
LSA	TF-IDF	1.8	1.5x Slower
	AraBERT (Unweighted)	8.6	7.2x Slower
	Hybrid (TF-IDF + AraBERT)	**9.1**	**6.7x Slower**

Table 11 shows the trade-off between performance and speed, a crucial consideration for real-world applications.

Table 12. Comparison with State-of-the-Art Arabic summarization models (with Limitations)

Model	**Approach**	**ROUGE Score**	**Dataset**	**Key Features**	**Limitations**
TF-IDF + AraBERT (Proposed)	Extractive	0.573	EASC	Combines semantic and statistical representation	Limited to extractive summaries; no abstractive logic
BERTSUM (Elmadani et al., 2020)	Abstractive (Fine-tuned BERT)	0.580	EASC, KALIMAT	Transformer-based; supports extractive & abstractive	Computationally expensive; requires GPU for training
DistilBERT Dual-Stage (Alshanqiti et al., 2021)	Extractive Deep Learning	0.551	Custom Arabic Corpus	Fast inference; compact model	Limited domain generalization
AWN + TF-IDF (Ala	Hybrid Graph-	0.502	EASC	Integrates semantic	Performance depends on

Model	Approach	ROUGE Score	Dataset	Key Features	Limitations
mi & Mallahi, 2021)	based			similarity via AWN	quality of WordNet resources
Word2Vec + W-PCA (Abdulateef et al., 2020)	Statistical + Semantic	0.518	EASC	Dimensionality reduction improves coherence	Word2Vec lacks deep contextual understanding

Table 12 presents a comparison between the proposed method and recent Arabic summarization models. The hybrid TF-IDF + AraBERT model performs competitively with a ROUGE score of 0.573. While BERT-based models show higher accuracy, they require more computational resources. Traditional models perform moderately but lack deep contextual understanding. The proposed method offers a balanced trade-off between performance and efficiency.

Execution Time Comparison

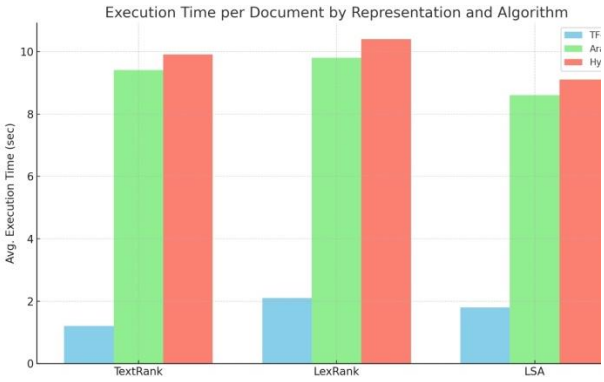

Fig 8. Execution Time per Document by Representation and Algorithm

Comparison of average execution time per document for TextRank, LexRank, and LSA algorithms using TF-IDF, AraBERT, and Hybrid representations. While the Hybrid and AraBERT approaches introduce computational overhead compared to TF-IDF, the performance gains justify the trade-off in applications where quality is prioritized AS shown in Figure 8.

Correlation Between Human Evaluation & ROUGE

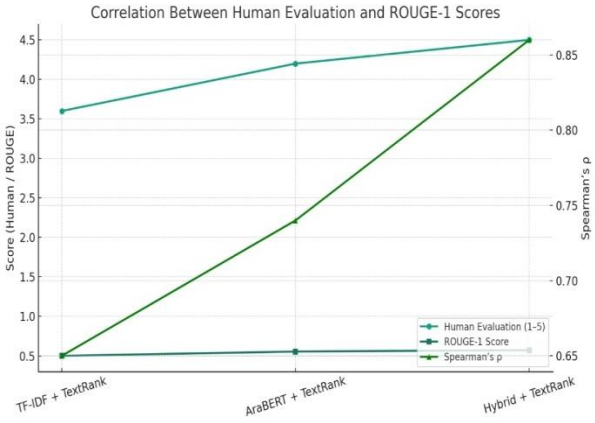

Fig 6. Correlation Between Human Evaluation and ROUGE-1 Scores

Spearman's rank correlation between human evaluation scores and ROUGE-1 metrics across different representation methods. The Hybrid approach achieves the highest correlation (ρ = 0.86), suggesting that its automatic scores align more closely with human judgment, validating its effectiveness for Arabic text summarization tasks AS shown in **Figure 8**.

7. Conclusion

This paper presents a comprehensive study on enhancing Arabic Text Summarization (ATS) through the in-tegration of preprocessing techniques and hybrid sentence representation. The proposed approach combines TF-IDF weighting with AraBERT contextual embeddings to form a hybrid model that captures both statistical relevance and semantic depth. Evaluated on the EASC dataset, this model significantly outperformed tradi-tional methods, achieving the highest ROUGE and BLEU scores among all tested configurations.

The experimental results demonstrated that preprocessing plays a critical role in improving summarization quality. In particular, the combination of normalization and stemming led to substantial performance gains, confirming their importance in managing Arabic's morphological richness. Among the summarization algo-rithms tested, TextRank consistently outperformed LexRank and LSA, making it the most effective

algorithm for extracting salient content in this setting.

Although transformer-based models such as AraBERT introduce greater computational overhead compared to classical statistical methods, their integration with TF-IDF in the hybrid framework provides a favorable trade-off between efficiency and summary quality. The hybrid model proved to be robust, generalizing well to the AlKhaleej corpus in zero-shot settings and showing strong alignment with human evaluation metrics.

Future work will focus on extending the evaluation to include a broader range of Arabic datasets and deeper linguistic analyses. Potential directions include exploring alternative weighting mechanisms, enhancing sentence embeddings through contrastive or supervised fine-tuning, and refining preprocessing pipelines to improve per-formance under domain-specific and low-resource conditions.

Refrences

[1] D. Radev, E. Hovy, and K. McKeown, “Introduction to the special issue on summarization,” Computational linguistics, vol. 28, no. 4, pp. 399–408, 2002.

[2] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, “Automatic text summarization: A comprehensive survey,” Expert Syst Appl, vol. 165, p. 113679, 2021.

[3] B. Elayeb, A. Chouigui, M. Bounhas, and O. Ben Khiroun, “Automatic Arabic Text Summarization Using Analogical Proportions,” Cognit Comput, vol. 12, no. 5, pp. 1043–1069, Sep. 2020, doi: 10.1007/s12559-020-09748-y.

[4] 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.

[5] K. Al-Sabahi, Z. Zhang, J. Long, and K. Alwesabi, “An Enhanced Latent Semantic Analysis Approach for Arabic Document Summarization,” Arab J Sci Eng, vol. 43, no. 12, pp. 8079–8094, Dec. 2018, doi: 10.1007/s13369-018-3286-z.

[6] A. M. Azmi and N. I. Altmami, “An abstractive Arabic text summarizer with user-controlled granularity,” Inf Process Manag, vol. 54, no. 6, pp. 903–921, Nov. 2018, doi: 10.1016/j.ipm.2018.06.002.

[7] R. Elbarougy, G. Behery, and A. El Khatib, “A Proposed Natural Language Processing Preprocessing Procedures for Enhancing Arabic Text Summarization,” Studies in Computational Intelligence, vol. 874, pp. 39–57, 2020.

[8] R. Elbarougy, G. Behery, and A. El Khatib, “The impact of stop words processing for improving extractive graph-based arabic text summarization,” International Journal of Scientific and Technology Research, vol. 8, no. 11, pp. 2134–2139, 2019.

[9] N. Alami, M. Meknassi, S. A. Ouatik, and N. Ennahnahi, “Impact of stemming on Arabic text summarization,” in Colloquium in Information Science and Technology, CIST, 2016, pp. 338–343. doi: 10.1109/CIST.2016.7805067.

[10] S. Abdulateef, N. A. Khan, B. Chen, and X. Shang, “Multidocument Arabic text summarization based on clustering and word2vec to reduce redundancy,” Information (Switzerland), vol. 11, no. 2, p. 59, 2020, doi: 10.3390/info11020059.

[11] Y. A. AL-Khassawneh and E. S. Hanandeh, “Extractive Arabic Text Summarization-Graph-Based Approach,” Electronics (Switzerland), vol. 12, no. 2, 2023, doi: 10.3390/electronics12020437.

[12] N. Alami and M. El Mallahi, “Hybrid method for text summarization based on statistical and semantic treatment,” pp. 19567–19600, 2021.

[13] 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,” Journal of King Saud University - Computer and Information Sciences, vol. 33, no. 6, pp. 677–692, 2021.

[14] A. M. Abu Nada, E. Alajrami, A. Alsaqqa, A. A. Al-Saqqa, and S. S. Abu-Naser, “Arabic Text Summarization Using AraBERT Model Using Extractive Text Summarization Approach Expert System View project Cloud computing View project Arabic Text Summarization Using AraBERT Model Using Extractive Text Summarization Approach,” 2020, [Online].

Available: https://www.researchgate.net/publication/344175857

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

[16] M. El-Haj, U. Kruschwitz, and C. Fox, "Using Mechanical Turk to Create a Corpus of Arabic Summaries," Proceedings of the International Conference on Language Resources and Evaluation, pp. 1–4, 2010, [Online]. Available: http://ufal.mff.cuni.cz/padt/PADT

[17] N. El-Fishawy, A. Hamouda, G. M. Attiya, and M. Atef, "Arabic summarization in Tw itter social network," Ain Shams Engineering Journal, vol. 5, no. 2, pp. 411–420, 2014.

[18] A. Haboush, M. Al-Zoubi, A. Momani, and M. Tarazi, "Arabic text summarization model using clustering techniques," World of Computer Science and Information Technology Journal (WCSIT) ISSN, vol. 2, no. 3, pp. 741–2221, 2012.

[19] N. Alami, Y. El Adlouni, N. En-Nahnahi, and M. Meknassi, "Using statistical and semantic analysis for arabic text summarization," in Advances in Intelligent Systems and Computing, Springer, 2018, pp. 35–50.

[20] W. Antoun, F. Baly, and H. Hajj, "Arabert: Transformer-based model for arabic language understanding.," AraBert: Transformer-Based Model for Arabic Language Understanding, no. May, pp. 9–15, 2020, [Online]. Available: http://arxiv.org/abs/2003.00104

[21] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, vol. 1, no. Mlm, pp. 4171–4186, 2019.

[22] M. Mohd, R. Jan, and M. Shah, "Text document summarization using word embedding," Expert Syst Appl, vol. 143, p. 112958, 2020, doi: 10.1016/j.eswa.2019.112958.

[23] A. Elsaadawy, M. Torki, and N. Ei-Makky, "A text classifier using weighted average word embedding," 2018 Proceedings of the Japan-Africa Conference on Electronics, Communications, and Computations, JAC-ECC 2018, no. November, pp. 151–154, 2018, doi: 10.1109/JEC-ECC.2018.8679539.

[24] R. Mihalcea and P. Tarau, "TextRank: Bringing order into texts," Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004, vol. 85, pp. 404–411, 2004.

[25] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," Journal of artificial intelligence research, vol. 22, pp. 457–479, 2004.

[26] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," Discourse Process, vol. 25, no. 2–3, pp. 259–284, 1998.

[27] C. Y. Lin, "Rouge: A package for automatic evaluation of summaries," in Proceedings of the workshop on text summarization branches out (WAS 2004), 2004, pp. 25–26.

[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "{B}leu: a Method for Automatic Evaluation of Machine Translation," in Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics.