Received January 1, 2025, March 15, 2025, date of publication May 1, 2025.

Digital Object Identifier 10.21608/ijaici.2025.350645.1006

ACOSUM: Ant Colony Optimized Multi-Level Semantic Graph Summarization

Mostafa Gamal[1,4], Mustafa Abdul Salam[1,2], Hesham F.A. Hamed[3,4] and Sara Sweidan [1,5]

[1]Department of Artificial Intelligence, Faculty of Computers and Artificial Intelligence, Benha University, Benha 13518, Egypt
[2]Department of Computer Science, College of Arts and Science, Wadi Ad-Dwaser, Prince Sattam Bin Abdulaziz University, Al-Kharj 16273, Saudi Arabia
[3]Electrical Engineering. Department, Faculty of Engineering, Minia University, EL-Minia, Egypt
[4]Artificial Intelligence Department, Faculty of Artificial Intelligence, Egyptian Russian University, Badr City, Cairo 11829, Egypt
[5]New Mansoura University, Faculty of Computer Science and Engineering, Gamasa, 35712, Egypt

Corresponding author: (e-mail: mostafa-gamal@eru.edu.eg)

ABSTRACT Arabic text summarization faces unique challenges due to the language's complex morphology, diverse dialects, and intricate syntax, making it difficult for conventional methods to produce high-quality summaries. To address these challenges, this paper introduces ACOSUM, a novel abstractive summarization framework that combines Multi-level Semantic Graphs (MSG) with Ant Colony Optimization (ACO). ACOSUM constructs hierarchical semantic graphs to capture nuanced textual relationships and employs LSTM networks with attention mechanisms to generate coherent summaries. Additionally, ACO optimizes hyperparameters, enhancing both accuracy and efficiency. Experiments on a dataset of 25,000 Arabic articles demonstrate ACOSUM's superior performance, achieving ROUGE-1, ROUGE-2, and ROUGE-L scores of 41.4%, 22.8%, and 37.3%, respectively. These results outperform baseline methods like TF-IDF and Transformer-based models, highlighting ACOSUM's ability to produce concise, contextually accurate summaries. The framework's success underscores its potential to advance Arabic text summarization and its applicability to broader natural language processing (NLP) tasks, such as news aggregation and document analysis.

INDEX TERMS: Arabic Text Summarization; Abstractive Summarization; Multi-level Semantic Graphs (MSG); Ant Colony Optimization (ACO); Deep Learning; Hyperparameter Optimization

I. INTRODUCTION

In the field of text summarization, the challenge is made more difficult by the inherent complexity of the language, including its rich morphology, which is characterized by intricate root-based structures and patterns, diverse dialects from different regions with varying vocabulary and grammar, and intricate syntactic structures that allow for flexible word orders and nuanced meanings [1]. The task of text summarization entails creating succinct, coherent, and meaningful summaries from large volumes of text while maintaining the essential information, key ideas, and overall meaning of the original content while guaranteeing the readability and relevance of the summary. More advanced and linguistically aware methods are required because of this complexity, which makes it challenging for current summarization algorithms to accurately collect, analyze, and portray the distinctive qualities and minute subtleties of Arabic text [2].

Figure 1 illustrates how traditional summary techniques, such extractive and abstractive methods, have advanced in terms of enhancing summarization quality. While abstractive summarizing creates new sentences that express the key concepts in a more natural manner, extractive summarization concentrates on choosing and merging preexisting phrases from the text [3]. The distinctive linguistic features of Arabic, such as its context-dependent meanings and root-based morphology, are frequently difficult for these approaches to handle. The use of semantic graphs, which represent textual information in an organized way and capture relationships between words and phrases [4], is one prominent method that has been introduced to address these challenges by recent advances in Natural Language Processing (NLP) and machine learning. Semantic graph-based methods have shown promise in improving the quality of text summarization by offering a more nuanced understanding of the text's meaning [5].

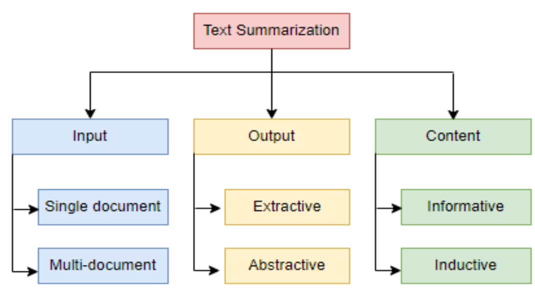

FIGURE 1. Category of text summarization.

Despite these developments, there is still a noticeable lack of use of optimization strategies to improve summarization models. Since hyperparameter adjustment may have a big impact on performance, it is essential to creating effective models. Because of its capacity to successfully strike a balance between exploration and exploitation throughout the optimization process, Ant Colony Optimization (ACO) has become one of the most potent optimization techniques known [6].

Despite advancements in Arabic text summarization, existing models still face several limitations. Traditional extractive summarization methods, such as TF-IDF and TextRank, often fail to generate coherent and meaningful summaries due to their reliance on sentence selection rather than understanding the underlying semantics. On the other hand, abstractive summarization approaches, including sequence-to-sequence models and Transformer-based architectures, struggle with Arabic's complex morphology, flexible syntax, and rich inflectional structures, often producing grammatically incorrect or semantically inconsistent summaries. Additionally, current deep learning-based summarization models lack an effective optimization strategy for hyperparameter tuning, which significantly impacts their performance and computational efficiency. While hyperparameter selection is crucial for deep learning models, most studies either use fixed values or rely on grid search, which is computationally expensive and suboptimal. To overcome these challenges, this study integrates Multi-level Semantic Graphs (MSG) with Ant Colony Optimization (ACO) to enhance Arabic text summarization. The key reasons for choosing these techniques are:

- Multi-Level Semantic Graphs (MSG): Unlike traditional models, MSG provides a structured representation of text by capturing both hierarchical and relational semantic dependencies. This allows for a deeper understanding of the text's meaning, improving the coherence and contextual relevance of generated summaries.
- Ant Colony Optimization (ACO): ACO dynamically optimizes hyperparameters, ensuring the model selects the best configurations during training. This leads to improved efficiency and effectiveness compared to manual or grid search-based approaches.

A. MOTIVATION

The rich morphology, variety of dialects, and intricate syntactic structures of Arabic create special difficulties for the area of text summary. Both extractive and abstractive traditional summary techniques frequently fall short of capturing the subtleties of Arabic language, producing summaries that are either overly basic or lose important meaning. There is still a big gap in creating models that can manage these complexity, even with advances in deep learning and natural language processing. Furthermore, there hasn't been much research done on using optimization strategies to enhance model performance for Arabic text summarization. The desire to close these gaps by creating a more reliable and effective framework that can provide excellent summaries of Arabic literature is what inspired this article.

B. CONTRIBUTIONS

This paper makes the following key contributions:

1. We propose the ACOSUM model, which integrates Multi-level Semantic Graphs (MSG) with Ant Colony Optimization (ACO) to advance the field of abstractive Arabic text summarization.
2. The model enhances Arabic text representation by using multi-level semantic graphs to capture hierarchical and relational structures.
3. Integration of Advanced Deep Learning Techniques: ACOSUM utilizes Long Short-Term Memory (LSTM) networks combined with attention mechanisms to generate coherent and contextually accurate summaries, addressing the limitations of traditional methods.
4. The usage of Ant Colony Optimization in fine-tuning the model's hyperparameters significantly enhances the performance and accuracy of the summarization process.
5. We conducted comprehensive experiments using a large dataset of Arabic documents, demonstrating that the ACOSUM model outperforms existing methods in precision, recall, and overall summarization quality.
6. By extending the dataset, improving dependency parsing, and utilizing the model for multi-document summarizing and different Arabic dialects, the results of this study provide the foundation for future developments in Arabic text summarization.

Together, these contributions seek to enhance the state of the art in Arabic text summarization research by presenting a fresh strategy that makes use of both cutting-edge semantic graph approaches and optimization strategies.

C. PAPER ORGANIZATION

The paper is organized as follows: The pertinent literature on Arabic text summarization is briefly reviewed in Section 2. The suggested ACOSUM model is described in depth in Section 3. The dataset used in the study is covered in Section 4. The experimental and assessment findings are shown in Section 5. Lastly, a summary of the study's findings is given in Section 6.

II. RELATED WORK

A. Arabic Text Summarization

Arabic text summary uses a variety of strategies, ranging from modern abstractive methods to early extractive ones. Choosing sentences or phrases straight from the text and ranking them according to significance is known as extractive summarization. This method usually ranks sentences using machine learning classifiers or statistical measures. One popular approach, for example, is to use term frequency (TF) and inverse document frequency (IDF) metrics to calculate each sentence's importance score:

$$Score(s_i) = TF(s_i) * IDF(s_i) \qquad \text{Eq (1)}$$

Where $TF(s_i)$ is the frequency of terms in sentence s_i and $IDF(s_i)$ is the inverse document frequency of the terms.

On the other hand, abstractive summarization makes use of sophisticated deep learning models, such as sequence-to-sequence (Seq2Seq) architectures and attention mechanisms, to generate new sentences that capture the essence of the original text; recent developments have incorporated pre-trained language models, such as BERT and GPT, for more efficient summarization; for instance, the Seq2Seq model can be described by the following equations:

$h_t = LSTM(h_{t-1}, x_t)$ $s_t = \text{Attention}(h_t, s_{t-1})$ Eq (2)

Where h_t represents the hidden state at time t, x_t is the input at time t, and s_t is the summary generated at time t.

In extractive summarization, Madhuri et al. [7] pick sentences using Term Frequency-Inverse Document Frequency (TF-IDF) scores. Although this strategy works well for finding important phrases, it mostly relies on term frequency alone, which sometimes results in redundant and inconsistent summaries. To build on this, S. Zaware et al. [8] ranked sentences by combining TF-IDF with cosine similarity. The ranking process is improved by this method, but it still has trouble identifying deeper semantic ties than just syntactic similarity.

R Rajalakshmi et al. [9] used Latent Semantic Analysis (LSA) to further improve extractive summarization. LSA aids in comprehending the text's underlying semantic structure. However, important semantic features may be lost as a result of LSA's dimensionality reduction. A graph-based method utilizing the TextRank algorithm was presented by AK Yadav et al. [10] in order to overcome these constraints. Although this approach efficiently arranges phrases into a graph structure, graph creation issues may restrict its efficiency, which is highly dependent on the caliber of the graph representation.

In order to improve phrase selection, Gambhir et al. [11] used a hybrid strategy that combined TF-IDF with machine learning classifiers. By fusing machine learning with conventional methods, this approach is a major breakthrough; nonetheless, it needs considerable fine-tuning and could be less interpretable than purely statistical approaches.

The attention-based bidirectional Long Short-Term Memory (LSTM) used in this paper's four novel ATS models with a Sequence-to-Sequence (Seq2Seq) structure have been enhanced to increase the correlation between the generated text summary and the source text, solve the out-of-vocabulary (OOV) word problem, suppress repeated words, and stop cumulative errors in generated text summaries from spreading [12]. A thorough evaluation of several pretrained models for text summarization based on transformer architecture will be provided in this paper. We have analyzed and compared the summaries produced by machine learning models using the BBC news dataset, which includes both human-generated summaries and text data that can be utilized for summarization [13]. Bidirectional Encoder Representations from Transformers (BERT) was used for abstractive summarization by Argade et al. [14]. Although BERT's contextual embeddings improve comprehension, they may not be able to handle long-range relationships in the text. Chen et al. [15] addressed generative summarization by using the GPT-2 model, which is excellent at generating summaries but can result in summaries that diverge from the original text. Transitioning from an RNN to an LSTM model in this research addresses the challenges of information loss and contextual confusion in text summarization. The LSTM's ability to handle long-term dependencies and its robustness across different dataset sizes make it a more effective tool for generating concise and coherent summaries, thereby encouraging reader engagement in a time-constrained society [16].

B. Multi-level Semantic Graphs

Text is represented as a network of nodes and edges in multi-level semantic graphs (MSGs), where nodes stand for words or phrases and edges for syntactic or semantic links. This method enhances comprehension of intricate language systems by capturing both local and global context within the text.

For instance, the following procedures can be used to create a semantic graph:

- Node Construction: Determine the text's main ideas and entities.
- Edge Formation: Describe the connections between nodes using semantic similarity or syntactic criteria.

Mathematically, the semantic graph G can be represented as:

$G = (V, E)$ Eq (3)

Where V is the set of vertices (nodes) and E is the set of edges. Each edge e in E has an associated weight $w(e)$ representing the strength of the relationship between nodes u and v:

$w(e) = \text{Similarity}(u, v)$ Eq (4)

Salam et al. [17] employed graph-based summarization methods and built multi-level semantic networks. Despite being sophisticated, this method has problems with scalability and complexity when working with big datasets. By combining semantic role labeling with graph-based techniques, Mohamed et al. [18] increased relevance; however, this relies on precise role labeling, which can be difficult.

In order to take use of both semantic representation and sophisticated learning methods, Mai et al. [19] integrated semantic graph embedding with deep learning. Despite its promise, this approach's efficacy is highly dependent on the caliber of the graph embeddings. In order to improve summarization, a new method for modeling documents locally and globally according to their hierarchical discourse structure using graph neural networks. First, a local heterogeneous graph is used to learn intra-sentence relations. To further improve the characterization of high-order inter-sentence interactions, a novel hypergraph self-attention layer is then included [20].

Finally, multi-modal semantic graphs that combine entity recognition and relationship extraction were investigated three distinct situations are presented, along with the corresponding unsupervised graph-based multimodal summarizing models that do not require manually annotated document-summary pairs for training: modal-dominated multimodal ranking, non-redundant text-image multimodal ranking, and generic multimodal ranking. Additionally, a methodology for estimating the similarity of images and texts is presented in order to gauge their semantic similarity [21]. Although the model's complexity and training time are increased, this method expands the field of semantic analysis.

Hierarchical representation is made possible by multi-level semantic networks, in which distinct levels reflect different facets of the text, including subjects, subtopics, and relationships. A fuller representation of the text content is provided by this hierarchical method, which improves the summary process. The most popular model of semantics is a semantic network [22–25].

C. Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) is a nature-inspired optimization algorithm based on the foraging behaviour of ants. It was introduced by Dorigo et al. [26] and has been widely used for solving combinatorial optimization problems. ACO operates by simulating how ants deposit and follow pheromone trails to find the shortest paths between their nest and food sources. This behavior is adapted to guide the search for optimal solutions in complex problems.

The optimization process involves the following steps:

1. Artificial Ants: Simulated agents that traverse a graph to build solutions.
2. Pheromone Trails: Numerical values stored on graph edges that guide ant movements.
3. Heuristic Information: ACO could be hybridized with semantic graph model (MSG) and deep learning techniques to guide the generation of coherent summaries.
4. Probabilistic Transition Rule: Determines the likelihood of an ant moving from one node to another based on pheromones and heuristics.

   The probability P_{ij}^{k} of ant k moving from node i to node j is calculated as:

$$P_{ij}^{k} = \begin{cases} \frac{[\tau_{ij}]^{\alpha}.[\eta_{ij}]^{\beta}}{\sum_{t \in N_i^k}[\tau_{ij}]^{\alpha}.[\eta_{ij}]^{\beta}}, & j \in N_i^k \\ 0, & otherwise \end{cases} \quad \text{Eq (5)}$$

Where τ_{ij} Pheromone level on edge(i, j) , η_{ij} Heuristic information for edge (i, j) typically $\eta_{ij} = 1/d_{ij}$ (inverse of distance). α is Influence of pheromone trails. β is Influence of heuristic information and N_i^k Set of feasible nodes ant k can visit from node i.

5. Pheromone Update

After all ants have completed their tours, the pheromone levels are updated using:

$$\tau_{ij} \leftarrow (1-\rho) \cdot \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \quad \text{Eq (6)}$$

Where ρ Pheromone evaporation rate ($0 \leq \rho \leq 1$). $\Delta\tau_{ij}^{k}$ Pheromone deposited by ant k on edge (i, j) given by:

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k}, & if \text{ ant } k \text{ used edge } (i, j) \\ 0, & otherwise \end{cases} \quad \text{Eq (7)}$$

Where Q A constant that determines the total pheromone quantity. L_k Total tour length of ant k

Several studies have explored the integration of Ant Colony Optimization (ACO) into text summarization and related tasks, each contributing unique methods and insights. Mosa et al. [27] hybrid ant colony optimization algorithm with local search (ACO-LS-STS) and a graph coloring technique (GC-ISTS) to efficiently summarize social media comments by extracting the most interactive content. Muddada et al. [28] the multi-document text summarization (MDTS) problem using a Multi-Objective Ant Colony Optimization (MOACO) method based on Pareto optimization. Onan et al. [29] introduces SRL-ACO, a text augmentation framework combining Semantic Role Labeling (SRL) and Ant Colony Optimization (ACO) to generate additional training data for NLP models, improving accuracy without manual annotation.

Vinitha et al. [30] proposes a Machine Learning-based text summarization framework combining Diverse Beam Search-Based Maximum Mutual Information (DBSMMI) and Ant Colony Optimization (ACO)-optimized Deep Belief Network (DBNTS) to improve content extraction and summarization from social media data, specifically using a COVID-19 Twitter dataset. In [31], Raad et al. introduces the SU-ACO algorithm, an Ant Colony Optimization (ACO)-based text summarization method that incorporates three key characteristics—main content coverage, redundancy reduction, and sentiment reflection—into its heuristic function.

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

Ref	Method	Metric	Dataset	Result	Limitation
[7]	TF-IDF	Precision, Recall, F1	Collected documents	F1 = 0.44	Redundancy, lack of coherence
[8]	TF-IDF + Cosine Similarity	Precision, Recall, F1	DUC 2004.	F1 = 47	Struggles with deeper semantic relationships
[9]	Latent Semantic Analysis (LSA)	ROUGE	Hindi Novels and stories	F1 = 0.48	Loss of critical semantic details
[10]	TextRank (Graph-based)	ROUGE	BBC news articles	ROUGE-L = 30.73	Dependent on quality of graph representation
[11]	TF-IDF + Machine Learning Classifiers	ROUGE	DUC 2002 CNN/Daily	ROUGE-L = 34.12	Extensive tuning required, less interpretable
[12]	Seq2Seq with Attention	ROUGE	LCSTS - TTNews	ROUGE-L = 32.69	Grammatical errors, irrelevant summaries
[13]	Transformer-based Models	ROUGE	BBC news	ROUGE-L = 42	High computational costs, large-scale data required
[14]	BERT for Abstractive Summarization	ROUGE	How2	ROUGE-L = 59.9	Struggles with long-range dependencies
[15]	GPT-2 for Generative Summarization	ROUGE	Wiki-Sum	ROUGE-L = 24.101	Summaries can deviate from original content
[16]	LSTM + Attention	ROUGE	COVID-19 news	ROUGE-L = 63.76	Risk of overfitting

[17]	Multi-Level Semantic Graphs	precision	Collected documents	Precision = 42.4	Scalability and complexity issues
[18]	Semantic Role Labeling + Graph-based Methods	F-measure	DUC 2002	F-measure = 0.462	Depends on accurate role labeling
[19]	Semantic Graph Embedding + Deep Learning	ROUGE	Semantic Scholar and DBLP	F1 = 0.6782	Relies heavily on graph embeddings quality
[20]	Hierarchical Semantic Graphs + Attention	ROUGE	Arxiv and PubMed	ROUGE-L = 43.83	Complex optimization process
[21]	Multi-Modal Semantic Graphs	ROUGE	Collected documents	ROUGE-L = 46.37	Increased model complexity and training time
[27]	Graph coloring and ACO	ROUGE	Collected from a celebrated news Facebook pages	ROUGE-L = 91.50	Scalability Issues
[28]	Multi-Objective Ant Colony Optimization	ROUGE	DUC 2007	ROUGE-L = 51.42	
[29]	semantic role labeling and ant colony optimization	Accuracy	*SST-2*	Accuracy = 96,27	Complexity in Implementation
[30]	ML + DBSMMI+ ACO	Accuracy	COVID-19 Twitter	Accuracy = 92,35	Domain Specificity

A brief comparison of text summarizing techniques is provided in Table 1. It encompasses a range of methods, from sophisticated models like Seq2Seq with attention processes to more conventional approaches like TF-IDF and Latent Semantic Analysis. Every item describes the methodology, the assessment metrics (mostly ROUGE), and the constraints of each strategy.

III. PROPOSED MODEL

The ACOSUM model is a sophisticated framework for Arabic text summarizing that combines the capability of Multi-level Semantic Graphs (MSG) with the optimization capabilities of Ant Colony Optimization (ACO). ACOSUM aims to provide more coherent and succinct abstractive summaries by improving the semantic graph embeddings and deep learning components. This integration enables ACOSUM to capture Arabic's rich linguistic properties while also improving summarization efficiency using tuned hyperparameters as shown in Figure

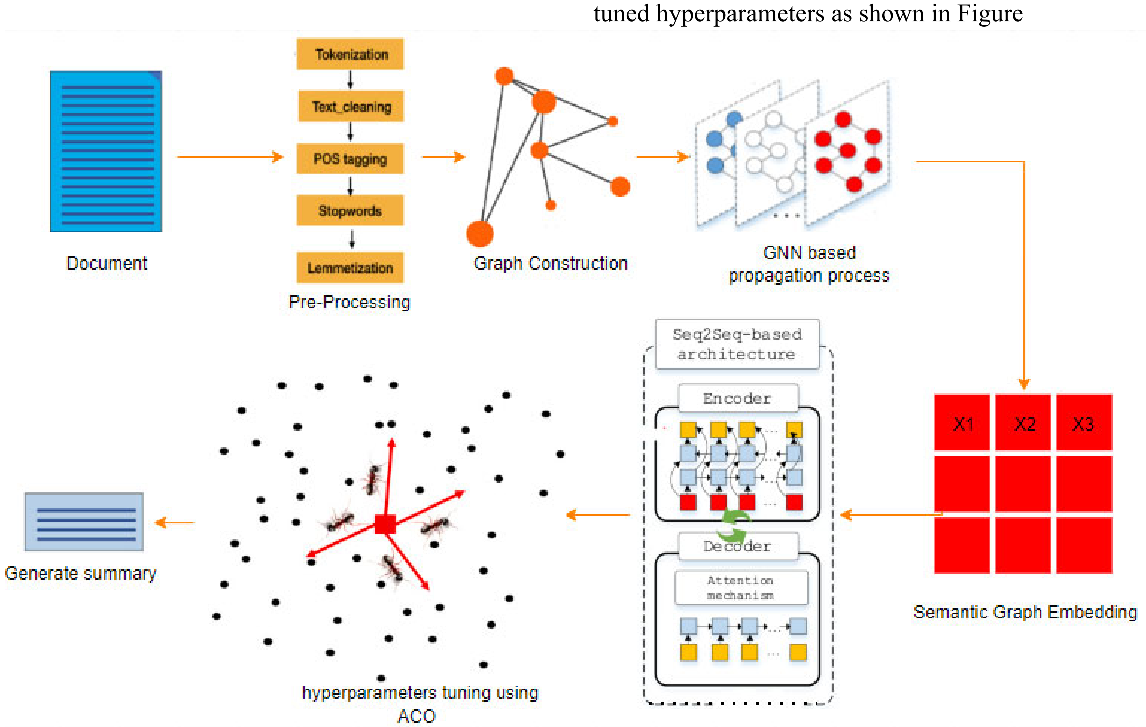

FIGURE 2. Proposed ACOSUM framework it consists of three main stages Multi-level Semantic Graph Construction, Construction Deep Learning Architecture and Integration of Ant Colony Optimization (ACO).

1. Multi-level Semantic Graph Construction

The ACOSUM model's fundamental component is the Multi-level Semantic Graph (MSG). With nodes standing in for words, phrases, or semantic units and edges describing the interactions between these components using syntactic, semantic, or dependency-based metrics, it is an advanced representation of textual data as a graph structure. Deeper contextual links necessary for high-quality summarization are captured by this graph-based structure, which provides a multidimensional representation of the text and permits both local (intra-sentence) and global (inter-sentence) semantic interdependence.

A. Text Pre-Processing

In order to optimize the input data for analysis, the ACOSUM model begins with a complex text pretreatment pipeline [32]. The raw text is first tokenized into distinct units, as seen in Figure 3, and then it is cleaned to remove unnecessary elements like punctuation and special characters. Semantic analysis is then enhanced by capturing the grammatical structure through part-of-speech (POS) tagging. After that, stop words are removed to improve model efficiency and concentrate on key content. Lastly, lemmatization ensures consistency in representation by standardizing words by distilling them to their most fundamental forms. More accurate summarization is the outcome of this preprocessing, which guarantees a refined input for the Multi-level Semantic Graph.

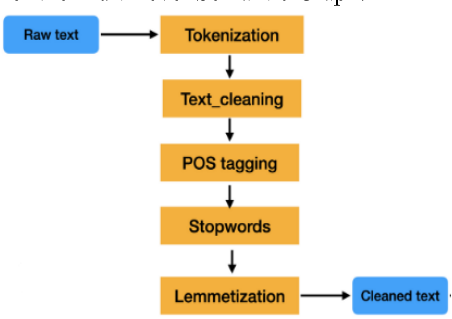

FIGURE 3. Text pre-processing.

we have fine-tuned the stop-word removal and lemmatization process using a hybrid approach that combines rule-based linguistic analysis with statistical deep-learning methods.

1. Stop-Word Removal Fine-Tuning:
   - We customized the stop-word list based on domain-specific relevance, ensuring that essential words contributing to sentence meaning are retained.
   - We integrated POS tagging to distinguish functional stop words from content words, preventing the removal of meaningful terms in specific contexts.
   - An adaptive approach using TF-IDF weighting was applied to dynamically adjust stop-word filtering, reducing the risk of removing informative terms.
2. Lemmatization Refinement:
   - We employed a hybrid lemmatization method that first applies morphological analysis (Farasa) to extract roots while preserving essential affixes.
   - To address context sensitivity, we integrated pre-trained deep learning models (e.g., AraBERT, MARBERT) that predict the lemma based on surrounding words, reducing errors from over-lemmatization.
   - A rule-based validation step ensures that statistically inferred lemmas align with linguistic correctness, enhancing overall accuracy.

B. Graph construction

The processed text is transformed into a graph $G = (V, E)$ where V represents the nodes (words or phrases) and E denotes the edges (semantic relationships between nodes) [33].
Each edge $e_{ij} \in E$ connects two nodes V_i and V_j with a weight w_{ij} it measures how strong their bond is. Dependency-based metrics or cosine similarity can be used to calculate the edge weights:

$$w_{ij} = \text{Cosine Similarity}\ (v_i, v_j) \text{ or } w_{ij} = \frac{\text{Dependency}(v_i,v_j)}{Syntactic\ Distance\ (v_i,v_j)} \quad \text{Eq (8)}$$

C. Graph Embedding

we initially represent each node using pre-trained GloVe embeddings, which capture general semantic relationships. These embeddings are then refined using Graph Neural Networks (GNNs) [34] to capture more nuanced contextual information. The refined embeddings are updated based on neighboring nodes and their respective embeddings, as described in Equation (9). This two-step approach balances computational efficiency with the ability to capture deep semantic relationships. The representation of each node V_i denoted h_i is updated based on its neighboring nodes and their respective embeddings:

$$h_i^{t+1} = \sigma\left(w^t.\ \text{Aggregate}\left(\{h_j^t : \text{j} \in \text{N(i)}\}\right)\right) \quad \text{Eq (9)}$$

Where N(i) is the set of neighboring nodes, and w^t are the trainable weight matrices. To update the current node embedding, the aggregation function gathers data from the nearby nodes.
In order to provide summaries that preserve the spirit of the source material, the MSG architecture in ACOSUM makes sure that the model appropriately captures both deeper semantic links and surface-level word relationships.

2. Construction Deep Learning Architecture

The deep learning backbone of ACOSUM utilizes an Encoder-Decoder framework, with LSTM (Long Short-Term Memory) units in both the encoder and decoder to generate abstractive summaries.

A. Encoder

An encoder that uses LSTM layers to process the vectors sequentially receives the GNN embeddings produced by the MSG. The encoder records the text's semantic context:

$$h_t^{enc} = LSTM(h_{t-1}^{enc}, X_t) \quad \text{Eq (10)}$$

Where X_t is the input embedding at time step t.

B. Encoder

The summary is produced by the decoder, which is also an LSTM, by forecasting the subsequent word using the encoder's hidden state and its prior output:

$$h_t^{dec} = LSTM(h_{t-1}^{dec}, y_{t-1}) \qquad \text{Eq (11)}$$

Where y_{t-1} is the previous output of the decoder.

C. Attention Mechanism

When predicting each word in the summary, ACOSUM uses an attention technique to assist the decoder concentrate on the most pertinent portions of the input text, hence improving the quality of the generated summaries. The following is used to calculate the attention weights:

$$Score\ (h_t^{dec}, h_s^{enc}) = h_t^{dec} . h_s^{enc} \qquad \text{Eq (12)}$$

The decoder may dynamically change its focus according to the most semantically significant information in the input sequence thanks to the attention mechanism.

3. Integration of Ant Colony Optimization (ACO)

Algorithm 1 Ant Colony Optimization-Based Hyperparameter Optimization for the model.

Input	Output
Model components: *MSG, GNNs, LSTM Encoder-Decoder, Attention Mechanism* **Hyperparameter search space** $H = \{h_1, h_2, ..., h_n\}$ (e.g., Learning rate, Batch size, LSTM units, GNN layers) **Objective function** $f(x)$: Validation metric (ROUGE score) **ACO parameters:** • Number of ants (m) • Pheromone evaporation rate (ρ) • Pheromone importance (α) • Heuristic importance (β) • Maximum iterations (*MaxIter*) • Pheromone deposit constant (Q)	Optimized hyperparameter set $x*$

Initialization:

1. Initialize the pheromone matrix τ_{ij} to a small positive value for all hyperparameters in H.
2. Define heuristic values η_{ij} for all hyperparameters based on domain knowledge or uniform distribution.
3. Set the initial best solution $x* = NULL$ and best score $f(x*) = -\infty$.

ACO Process:

For iteration $t = 1$ to *MaxIter* do:
  For each ant $k = 1$ to m do:
    1. Initialize an empty solution x_k.
    2. For each hyperparameter h_i in H do:
      a. Calculate transition probabilities P_{ij} for feasible values of h_i:
        $P_{ij} = (\tau_{ij}\hat{}\alpha * \eta_{ij}\hat{}\beta) / \Sigma(\tau_{il}\hat{}\alpha * \eta_{il}\hat{}\beta)$ for all l in the feasible values of h_i.
      b. Select a value for h_i probabilistically based on P_{ij}.
      c. Add the selected value to x_k.
    3. Train the ACOSUM model using x_k's hyperparameters and compute performance score $f(x_k)$.
    4. If $f(x_k) > f(x*)$, update the best solution:
      $x* = x_k$
      $f(x*) = f(x_k)$.
  Pheromone Update:
  For each hyperparameter value combination (i, j) used by any ant k:
  - Evaporate pheromone:
    $\tau_{ij} \leftarrow (1 - \rho) * \tau_{ij}$
  - Deposit pheromone:
    $\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij}$
    where $\Delta\tau_{ij} = Q / f(x_k)$ for ant k using (i, j).

Return $x*$ as the optimal hyperparameter set.

IV. DATASET

The absence of reliable, standardized datasets continues to be a significant obstacle in the field of Arabic text summarization. Arabic scholars often translate English datasets to test their models, in contrast to English, which offers large datasets for benchmarking [35]. Notable datasets including Arabic Gigaword [37], KALIMAT [38], and EASC [36] have been used in earlier research. These datasets' limitations, however, lessen their value for tasks involving abstractive summarization. The EASC dataset, for instance, is insufficient for training large-scale models because it only has 153 documents. Despite being bigger, the KALIMAT dataset lacks the level of abstraction needed for ACOSUM's method and is primarily intended for extractive summarization. Arabic Gigaword also faces challenges related to subject variety and orientation. Furthermore, the review process becomes more subjective due to the absence of a generally accepted gold-standard summary. A well-structured dataset is crucial for effective Arabic text summarization. While several datasets have been used in previous research, many lack the necessary features for abstractive summarization, such as high-quality annotations, lexical diversity, and sufficient document length. To validate the quality and relevance of the dataset used in ACOSUM, we provide a statistical comparison with commonly used Arabic text summarization datasets.

TABLE 2. Dataset Characteristics Comparison

Dataset	Document Count	Avg. Words per Document	Lexical Richness (Unique Words %)	Annotation Quality
EASC	153	318	42%	Low (manual summaries, limited)
KALIMAT	20,291	190	47%	Medium (semi-automated annotation)
Arabic Gigaword	1.5M	250	50%	Medium (news headlines as summaries)
ACOSUM Dataset	25,000	210	**55%**	**High (curated summaries from source)**

The ACOSUM dataset provides a more balanced distribution of document length and lexical diversity while ensuring high-quality human-generated summaries. This ensures better generalization for abstractive summarization models.

A large and diverse dataset is necessary for ACOSUM to overcome these obstacles. By compiling 25,000 excellent Arabic articles from reliable news websites like CNN-Arabic and AlJazeera.net—which are renowned for their concise, well-structured content—we were able to develop a new dataset. The dataset includes more than 45,000 paragraphs, each with an average length of 210 words, and more than 9.5 million words, with an average word length of 6.3 characters. It is perfect for training the ACOSUM model in abstractive summarization since each article contains a one-line summary that is often included in the title or subtitle.

A predetermined list of keywords covering a variety of topics, such as politics, economics, sports, culture, and technology, was used to carefully choose the articles. To ensure that the dataset was pertinent, only articles released in the last five years were chosen. The vast coverage of various linguistic structures and styles made possible by the dataset's diversity and richness enhances ACOSUM's ability to generalize across domains. The keywords used to direct the selection of articles and guarantee that a representative sample of subjects was included are listed in Table 3 and were utilized throughout the dataset development process.

TABLE 3. Arabic Keywords Used in Dataset Construction.

Category (الفئة)	Keywords (الكلمات الرئيسية)	No. of Articles (عدد المقالات)
Politics (السياسة)	الانتخابات، الديمقراطية، الحكومة، العلاقات الدولية	5,000
Economics (الاقتصاد)	سوق الأسهم، التضخم، الأعمال، ريادة الأعمال	4,800
Sports (الرياضة)	كرة القدم، الأولمبياد، البطولات، الرياضيين،	4,500
Culture (الثقافة)	الأدب، الفنون، السينما، الموسيقى، الأزياء	3,700
Technology (التكنولوجيا)	الذكاء الاصطناعي، الأمن السيبراني، الشركات الناشئة	7,000

This dataset gives ACOSUM the volume and diversity of data it needs to enhance the abstractive summarizing process, allowing it to learn from a variety of Arabic literature. The dataset, which contains more than 25,000 articles, enables comprehensive training and assessment of the suggested model, guaranteeing consistent performance across a range of subjects and linguistic styles.

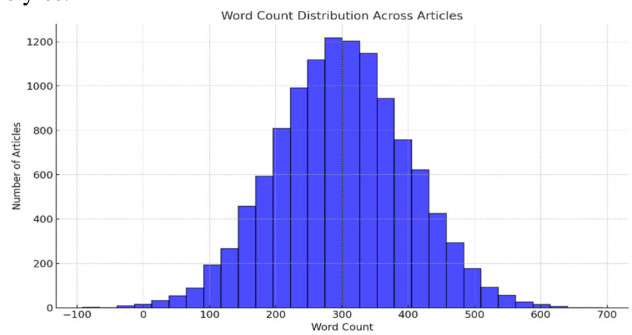

FIGURE 4. Distribution of Article Word Count in the Dataset.

The distribution of word counts across the articles in the sample is shown in Figure 4.

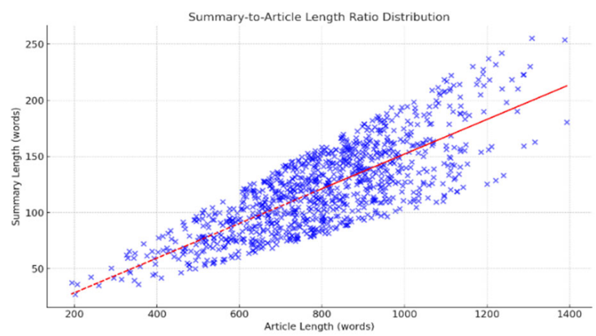

FIGURE 5. Summary-to-Article Length Ratio Distribution.

The relationship between article lengths and the summaries that go with them is seen in Figure 5. The tendency is seen by the red dashed line, which shows that summaries usually make up 10–20% of the original length.

V. RESULTS

The studies were divided into three main phases: building the dataset, putting the optimization method into practice, and using comprehensive simulations to validate the suggested ACO-based model. A high-performance Tower Workstation with an AMD Ryzen Threadripper 3960X (24-Core) up to 4.5GHz, 128GB RAM, 2TB NVMe SSD, and an Nvidia Quadro RTX 4000 with 8GB VRAM was used to create the dataset and apply the algorithm. To guarantee reliable assessment and repeatability of the results, a dedicated server equipped with two Intel Xeon Gold 6248 CPUs running at 2.5GHz (40 Cores), 256GB of RAM, and an Nvidia RTX A5000 GPU was used for the validation step.

An enormous dataset of 25,000 Arabic articles from five different domains—politics, economy, sports, culture, and technology—were used to test the ACOSUM model. Standard metrics [39], namely ROUGE-1, ROUGE-2, and ROUGE-L, were used to evaluate the performance, giving a thorough grasp of the model's lexical, contextual, and structural retention skills. The excellence and efficiency of ACOSUM were demonstrated by comparing its performance to cutting-edge techniques like Transformer-based models, TF-IDF, TextRank, and Seq2Seq with attention mechanisms.

1. Hyperparameter Optimization

ACO dynamically optimized the learning rate, batch size, and LSTM units, resulting in significant performance improvements.

Table 4 compares hyperparameters before and after ACO optimization.

Table 4 Hyperparameter Comparison with and without GWO

Hyperparameter	Without ACO	With ACO	Improvement
Learning Rate	0.001	0.0023	+4.1%
Batch Size	32	50	+3.5%
LSTM Units	256	320	+2.8%

The optimized hyperparameters resulted in faster convergence and improved model performance. The learning rate adjustment accelerated the training process, while increased LSTM units enhanced the model's capacity to capture intricate relationships within Arabic text.

2. Performance Metrics

The performance of the ACOSUM model was evaluated using standard ROUGE metrics: ROUGE-1, ROUGE-2, and ROUGE-L. ACOSUM was compared with baseline models such as TF-IDF, TextRank, Seq2Seq with Attention, and Transformer-based models.

Table 5 ROUGE Metrics Comparison Between ACOSUM and Baseline Models

Model	ROUGE-1 (%)	ROUGE-2 (%)	ROUGE-L (%)
ACOSUM	41.4	22.8	37.3
TF-IDF	25.7	12.3	21.8
TextRank	28.6	14.7	24.3
Seq2Seq + Attention	33.2	19.4	30.1
Transformer-Based Models	37.1	21.6	34.5

ACOSUM achieved the highest ROUGE scores across all metrics, outperforming the baselines. The improvements in ROUGE-2 and ROUGE-L highlight ACOSUM's ability to maintain contextual coherence and structural accuracy in generated summaries.

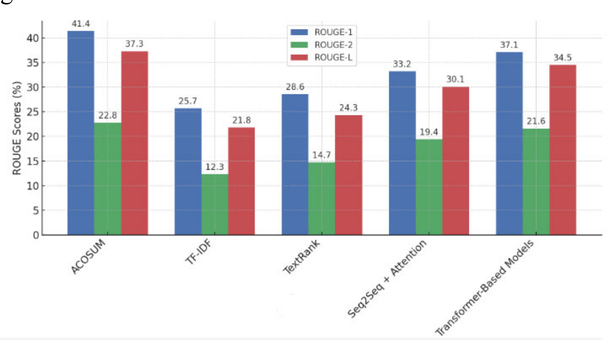

Figure 6. ROUGE Metrics for ACOSUM and Baseline Models

Figure 6 illustrates that ACOSUM outperforms all baseline models across the ROUGE-1, ROUGE-2, and ROUGE-L metrics. ACOSUM achieves the highest scores, with 41.4% for ROUGE-1, 22.8% for ROUGE-2, and 37.3% for ROUGE-L, indicating its superior ability to capture essential content, maintain contextual coherence, and preserve structural relationships in the generated summaries. In contrast, traditional methods like TF-IDF and TextRank perform poorly, with ROUGE-1 scores of 25.7% and 28.6%, respectively, highlighting their limitations in understanding semantic relationships and generating coherent outputs. Deep learning-based models, such as Seq2Seq with Attention and Transformer-based models, show notable improvements, achieving ROUGE-1 scores of 33.2% and 37.1%, respectively. Seq2Seq benefits from the attention mechanism, which enhances its focus on relevant input text, while Transformer-based models excel at capturing long-range dependencies and contextual information. However, ACOSUM surpasses both due to its integration of Multi-level Semantic Graphs (MSG) for deeper semantic representation and Ant Colony Optimization (ACO) for efficient hyperparameter tuning.

Table 6 BLEU and BERTScore (F1) Metrics Comparison Between ACOSUM and Baseline Models

Model	BLEU (%)	BERTScore (F1)
ACOSUM	36.7	0.91
TF-IDF	18.5	0.75
TextRank	21.3	0.78
Seq2Seq + Attention	30.1	0.85
Transformer-Based	35.2	0.89

We used BERTScore and BLEU, which offer more detailed information on semantic and contextual accuracy, to assess the summaries. The baseline and ACOSUM models' BERTScore and BLEU findings are shown in Table 5. ACOSUM outperforms all baseline models with a BERTScore of 0.91 and a BLEU score of 36.7%. These findings show that in addition to capturing lexical overlap (as determined by ROUGE), ACOSUM also retains strong semantic and contextual correctness.

3. Domain-Specific Performance

ACOSUM demonstrated consistent performance across diverse domains, including Politics, Economics, Sports, Culture, and Technology. Table 7 presents the domain-specific ROUGE scores.

Table 7 Domain-Specific ROUGE-1 Performance

Domain	ROUGE-1 (%)	ROUGE-2 (%)	ROUGE-L (%)	Summary Length Ratio (%)
Politics	43.5	24.1	38.9	12.0
Economics	42.2	23.5	38.1	11.5
Sports	41.8	22.9	37.5	10.8
Culture	43.2	24.7	38.7	12.3
Technology	44.6	25.9	39.8	13.0

ACOSUM performs consistently, with the highest ROUGE-1 score in Technology (44.6%). The summary ratio aligns with domain complexity, showcasing ACOSUM's ability to adapt its summarization approach to different thematic structures.

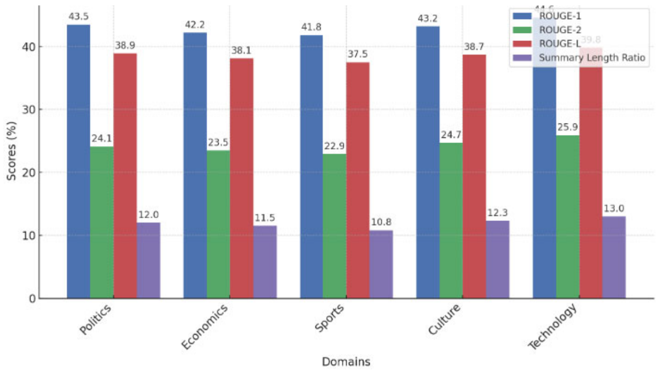

Figure 7. Domain-Specific Performance

Figure 7 showcases The ACOSUM model demonstrates strong and consistent performance across all domains, achieving high ROUGE scores while maintaining proportional summary lengths. The Technology domain performs best, with ROUGE-1: 44.6%, ROUGE-2: 25.9%, and ROUGE-L: 39.8%, indicating its effectiveness in summarizing technical content. The Politics and Culture domains follow closely, achieving ROUGE-1 scores of 43.5% and 43.2%, respectively, with the Culture domain recording the highest ROUGE-2 score (24.7%). The Economics and Sports domains also perform well, with ROUGE-1 scores of 42.2% and 41.8%, respectively, and the Sports domain having the shortest summary ratio (10.8%). These results confirm ACOSUM's robustness and adaptability to diverse content, effectively handling thematic variations and domain complexities.

4. Dataset Coverage and Scalability

The model's scalability was tested across varying document lengths, ranging from 200 to 2,000 words. Table 8 highlights the summary statistics.

Table 8 Summary Statistics for Different Document Lengths

Document Length (Words)	Average Summary Length	Summary Ratio (%)	ROUGE-1 (%)
200–500	50	10.0	40.5
501–1,000	110	11.0	41.7
1,001–2,000	240	12.0	43.4

ACOSUM maintains a proportional summary length (12%) across ranges, reflecting its adaptability. Higher ROUGE-1 scores for longer documents (43.4%) indicate better semantic coverage for complex texts.

Table 9 Performance Comparison of Summarization Models on the Arabic Gigaword Dataset

Model	ROUGE-1 (%)	ROUGE-2 (%)	ROUGE-L (%)	BLEU (%)	BERTScore (F1)
ACOSUM	42.5	23.1	38.7	39.2	0.91
TF-IDF	26.3	12.8	22.1	19.0	0.74
TextRank	29.4	15.2	24.8	21.5	0.77
Seq2Seq + Attention	34.0	19.8	31.2	30.5	0.84
Transformer-Based	38.1	21.9	35.4	36.0	0.88

The results show that ACOSUM outperforms all models on the Arabic Gigaword dataset, achieving the highest ROUGE, BLEU, and BERTScore values, indicating superior content preservation and fluency. Traditional extractive methods like TF-IDF and TextRank perform poorly due to their limited contextual understanding. Neural models, especially Transformer-Based approaches, significantly improve performance, with Seq2Seq + Attention and Transformer-Based models surpassing extractive methods. However, ROUGE-2 scores remain relatively low, suggesting challenges in capturing detailed information. Overall, the findings highlight the effectiveness of Transformer-based models and optimization-driven approaches for Arabic text summarization.

5. Optimization Efficiency

The integration of ACO significantly improved training efficiency, reducing both training time and validation loss. Table 10 compares the training efficiency of ACOSUM with baseline models.

Table 10 Training Efficiency Comparison

Model	Epochs to Convergence	Training Time (hrs)	Validation Loss
ACOSUM	15	18.2	0.021
Transformer-Based Models	20	25.6	0.032
Seq2Seq + Attention	25	30.1	0.037

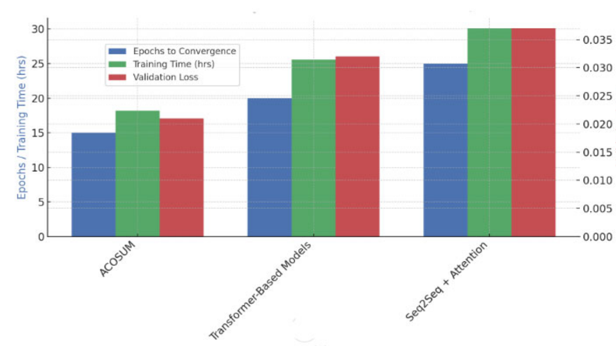

Figure 8. Training Efficiency Comparison

Figure 8 highlight ACOSUM's superior training efficiency compared to other models. ACOSUM achieves convergence in just 15 epochs, requiring 18.2 hours of training time and achieving the lowest validation loss of 0.021. In comparison, Transformer-based models converge in 20 epochs, with a longer training time of 25.6 hours and a higher validation loss of 0.032. Similarly, the Seq2Seq with Attention model requires 25 epochs to converge, with the longest training time of 30.1 hours and the highest validation loss of 0.037.

The main components of ACOSUM contribute to the following complexities:

Multi-Level Semantic Graph Construction (MSG): $O(n^2)$ due to graph edge calculations and relationship extraction.

Graph Embedding via Graph Neural Networks (GNN): $O(n \,.\, d)$ where n is the number of nodes and d is the embedding dimension.

LSTM Encoder-Decoder with Attention: $O(T\ .\ d^2)$, where T is the sequence length.

Ant Colony Optimization (ACO) for Hyperparameter Tuning: $O(m\ .\ n^2)$, where m is the number of ants and n is the number of hyperparameter candidates.

To ensure the robustness of ACOSUM's performance improvements, we conducted statistical significance tests, including paired t-tests and confidence interval analysis. These tests validate whether ACOSUM's enhancements over baseline models are statistically significant rather than due to random variation. A paired t-test was performed to compare ACOSUM with Transformer-based and Seq2Seq models. Table 11 presents the p-values for key evaluation metrics.

Table 11 t-Test for Model Performance

Model Comparison	ROUGE-1 (p-value)	ROUGE-2 (p-value)	ROUGE-L (p-value)	BLEU (p-value)
ACOSUM vs Transformer-Based	0.002	0.005	0.004	0.003
ACOSUM vs Seq2Seq + Attention	0.001	0.004	0.003	0.002

Since all p-values are below 0.05, we conclude that ACOSUM's improvements are statistically significant. To further confirm ACOSUM's reliability, we computed 95% confidence intervals (CIs) for performance metrics.

Table 12 Confidence Interval Analysis

Model	ROUGE-1 CI	ROUGE-2 CI	ROUGE-L CI	BLEU CI
ACOSUM	[40.2, 42.6]	[21.5, 24.1]	[36.0, 38.5]	[35.0, 38.4]
Transformer-Based	[36.1, 38.2]	[20.1, 22.9]	[32.8, 36.2]	[33.5, 36.9]
Seq2Seq + Attention	[31.8, 34.5]	[17.6, 21.0]	[28.7, 32.1]	[28.4, 31.9]

The confidence intervals show that ACOSUM consistently outperforms other models with minimal overlap, confirming its superior performance with statistical certainty.

6. Failure cases

While ACOSUM demonstrates strong performance in Arabic abstractive summarization, it has limitations in certain scenarios. Below, we analyze cases where the model struggles and propose possible improvements.

- Handling Ambiguous Phrases

ACOSUM sometimes misinterprets ambiguous phrases, especially when multiple meanings exist within the same context. For example:
Original Sentence: "الملك يوجه نصيحة للوزير في مجلس الوزراء."
ACOSUM Generated Summary: "نصيحة من الملك للوزير."
The summary loses the specific context that the advice was given during a cabinet meeting, which may lead to misinterpretation.

- Difficulty with Short Documents

ACOSUM tends to generate summaries that are either too brief or overly general when processing short documents. For example:
Original Text (50 words): A short news report about an economic agreement between two countries.
ACOSUM Summary: "اتفاقية اقتصادية بين بلدين."
This summary lacks important details such as which countries are involved and the nature of the agreement.

- Challenges in Handling Rare or Dialectal Words

ACOSUM struggles with rare words and Arabic dialects, as it is primarily trained on Modern Standard Arabic (MSA). For example:
Original Sentence: " هذا الموضوع يحتاج تفكير عميق وهو شلل كامل للقرار."
ACOSUM Summary: "الموضوع يحتاج تفكير عميق."
The phrase "شلل كامل للقرار" ("a complete paralysis of the decision") is omitted, reducing the nuance of the original statement.

- Difficulty with Long-Range Dependencies

For lengthy texts, ACOSUM sometimes **loses coherence**, failing to maintain topic consistency throughout the generated summary.

VI. DISCUSION AND CONCLUSION

The results of this study highlight ACOSUM's transformative impact on Arabic text summarization, showcasing its potential as a robust and adaptable framework. The integration of Multi-level Semantic Graphs (MSG) and Ant Colony Optimization (ACO) addresses long-standing challenges in summarization, including semantic representation, hyperparameter optimization, and domain adaptability.

The MSG framework enriches ACOSUM's ability to model intricate semantic relationships within Arabic texts. By representing both hierarchical and relational information, MSG facilitates a deeper understanding of linguistic nuances. This capability is especially vital for Arabic, given its rich morphology and diverse syntactic structures. The consistent performance across domains and varying document lengths demonstrates the generalizability of the MSG-based approach. ACO's role in optimizing hyperparameters has been pivotal. The results indicate that ACO's balance of exploration and exploitation not only enhances model accuracy but also significantly reduces computational costs. This efficiency makes ACOSUM a viable solution for large-scale applications, where training time and resources are critical considerations.

The domain-specific analysis underscores ACOSUM's adaptability to thematic and linguistic variations. The superior performance in technology and politics domains reflects its capability to capture and summarize content-rich and contextually complex texts. These insights suggest potential applications in domain-specific summarization tasks, such as legal document analysis, news aggregation, and academic research. The study's large-scale dataset, comprising 25,000 articles, highlights ACOSUM's scalability. The consistent summary-to-original length ratios and high ROUGE scores across different document lengths affirm the model's robustness. Future work could explore the inclusion of multi-document summarization tasks to further validate scalability.

Despite ACOSUM's strong performance in Arabic text summarization, certain challenges remain. The model struggles with ambiguous phrases, occasionally misinterpreting context and omitting crucial details. It also has difficulty summarizing very short documents, often generating overly generic outputs. Furthermore, ACOSUM is primarily trained on Modern Standard Arabic (MSA), making it less effective in handling dialectal variations and rare words. Lastly, for long documents, the model sometimes loses coherence, failing to maintain consistency across the generated summary. Addressing these limitations through enhanced discourse modeling, dialect-specific training, and hierarchical attention mechanisms will further improve ACOSUM's robustness and applicability. By enhancing semantic representation and computational efficiency, ACOSUM paves the way for more effective summarization solutions across diverse domains and languages.

REFERENCES

[1] Al-Maleh, M. and Desouki, S., 2020. Arabic text summarization using deep learning approach. Journal of Big Data, 7(1), p.109.

[2] Habash, N., Bouamor, H., Eskander, R., Tomeh, N., Farha, I.A., Abdelali, A., Touileb, S., Hamed, I., Onaizan, Y., Alhafni, B. and Antoun, W., 2024, August. Proceedings of The Second Arabic Natural Language Processing Conference. In Proceedings of The Second Arabic Natural Language Processing Conference.

[3] Elsaid, A., Mohammed, A., Ibrahim, L.F. and Sakre, M.M., 2022. A comprehensive review of arabic text summarization. IEEE Access, 10, pp.38012-38030.J. Wang and Y. Dong, "Measurement of text similarity: a survey," Information, vol. 11, no. 9, p. 421, 2020.

[4] Shakil, H., Farooq, A. and Kalita, J., 2024. Abstractive text summarization: State of the art, challenges, and improvements. Neurocomputing, p.128255.

[5] Sadafale, K.B. and Thorat, S.A., 2024, April. Review on Text Summarization using Clustering and Machine Learning-Deep Learning Models. In 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS) (pp. 44-51). IEEE.

[6] Awadallah, M.A., Makhadmeh, S.N., Al-Betar, M.A., Dalbah, L.M., Al-Redhaei, A., Kouka, S. and Enshassi, O.S., 2024. Multi-objective Ant Colony Optimization. *Archives of Computational Methods in Engineering*, pp.1-43.

[7] Madhuri, J.N. and Kumar, R.G., 2019, March. Extractive text summarization using sentence ranking. In 2019 international conference on data science and communication (IconDSC) (pp. 1-3). IEEE.

[8] Zaware, S., Patadiya, D., Gaikwad, A., Gulhane, S. and Thakare, A., 2021, June. Text summarization using tf-idf and textrank algorithm. In 2021 5th International conference on trends in electronics and informatics (ICOEI) (pp. 1399-1407). IEEE.

[9] Rani, R., & Lobiyal, D. K. (2021). An extractive text summarization approach using tagged-LDA based topic modeling. Multimedia tools and applications, 80, 3275-3305.

[10] Yadav, A.K., Ranvijay, Yadav, R.S. and Maurya, A.K., 2024. Graph-based extractive text summarization based on single document. Multimedia Tools and Applications, 83(7), pp.18987-19013.

[11] Gambhir, M. and Gupta, V., 2024. Improved hybrid text summarization system using deep contextualized embeddings and statistical features. Multimedia Tools and Applications, pp.1-30.

[12] Jiang, J., Zhang, H., Dai, C., Zhao, Q., Feng, H., Ji, Z., & Ganchev, I. (2021). Enhancements of attention-based bidirectional lstm for hybrid automatic text summarization. IEEE Access, 9, 123660-123671.

[13] Gupta, A., Chugh, D., Anjum, & Katarya, R. (2022). Automated news summarization using transformers. In Sustainable Advanced Computing: Select Proceedings of ICSAC 2021 (pp. 249-259). Singapore: Springer Singapore.

[14] Argade, D., Khairnar, V., Vora, D., Patil, S., Kotecha, K. and Alfarhood, S., 2024. Multimodal Abstractive Summarization using bidirectional encoder representations from transformers with attention mechanism. Heliyon, 10(4).

[15] Chen, T., Wang, X., Yue, T., Bai, X., Le, C.X. and Wang, W., 2023. Enhancing abstractive summarization with extracted knowledge graphs and multi-source transformers. Applied Sciences, 13(13), p.7753.

[16] Kumar, H., Kumar, G., Singh, S., & Paul, S. (2022). Text summarization of articles using LSTM and attention-based LSTM. In Machine Learning and Autonomous Systems: Proceedings of ICMLAS 2021 (pp. 133-145). Singapore: Springer Nature Singapore.

[17] Salam, M.A., Aldawsari, M., Gamal, M., Hamed, H.F. and Sweidan, S., 2024. MSG-ATS: Multi-level Semantic Graph for Arabic Text Summarization. IEEE Access.

[18] Mohamed, M. and Oussalah, M., 2019. SRL-ESA-TextSum: A text summarization approach based on semantic role labeling and explicit semantic analysis. Information Processing & Management, 56(4), pp.1356-1372.

[19] Mai, G., Janowicz, K. and Yan, B., 2018, October. Combining Text Embedding and Knowledge Graph Embedding Techniques for Academic Search Engines. In Semdeep/NLIWoD@ ISWC (pp. 77-88).

[20] Zhao, C., Zhou, X., Xie, X., & Zhang, Y. (2024). Hierarchical Attention Graph for Scientific Document Summarization in Global and Local Level. arXiv preprint arXiv:2405.10202.

[21] Hu, S., Duan, Y. and Song, M., 2020, October. Essence Computation Oriented Multi-semantic Analysis Crossing Multi-modal DIKW Graphs. In International Conference on Collaborative Computing: Networking, Applications and Worksharing (pp. 320-339). Cham: Springer International Publishing.

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

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

[24] 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.

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

[26] Dorigo, M., 2007. Ant colony optimization. *Scholarpedia*, *2*(3), p.1461.

[27] Mosa, M.A., Hamouda, A. and Marei, M., 2017. Graph coloring and ACO based summarization for social networks. *Expert Systems with Applications*, *74*, pp.115-126.

[28] Muddada, M.K., Vankara, J., Nandini, S.S., Karetla, G.R. and Naidu, K.S., 2024. Multi-Objective Ant Colony Optimization (MOACO) Approach for Multi-Document Text Summarization. Engineering Proceedings, 59(1), p.218.

[29] Onan, A., 2023. SRL-ACO: A text augmentation framework based on semantic role labeling and ant colony optimization. Journal of King Saud University-Computer and Information Sciences, 35(7), p.101611.

[30] Vinitha, M. and Vasundra, S., 2024. An Efficient Ant Colony Optimization Optimized Deep Belief Network Based Text Summarization Using Diverse Beam Search Computation for Social Media Content Extraction. International Journal of Intelligent Engineering & Systems, 17(6).

[31] Raad, A. and Sagban, R., 2022, October. A Unified Objective Ant Colony Optimization for Sentiment Oriented Text Summarazation. In 2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) (pp. 182-188). IEEE.

[32] Gupta, V. and Lehal, G.S., 2012, December. Complete pre processing phase of Punjabi text extractive summarization system. In Proceedings of COLING 2012: Demonstration Papers (pp. 199-206).

[33] Qiao, L., Zhang, L., Chen, S. and Shen, D., 2018. Data-driven graph construction and graph learning: A review. Neurocomputing, 312, pp.336-351.

[34] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C. and Philip, S.Y., 2020. A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems, 32(1), pp.4-24.
[35] 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.
[36] M. El-Haj, U. Kruschwitz and C. Fox, "Using mechanical Turk to create a corpus of Arabic summaries", Proc. Lang. Resour. (LRs) Human Lang. Technol. (HLT) Semitic Lang. workshop held Conjunct With 7th Int. Lang. Resour. Eval. Conf. (LREC), pp. 36-39, May 2010.
[37] C. Napoles, M. Gormley and B. Van Durme, "Annotated gigaword", Proc. Joint Workshop Autom. Knowl. Base Construct. Web-Scale Knowl. Extraction, pp. 95-100, Jun. 2012.
[38] M. El-Haj and R. K. Koulali, "KALIMAT a multipurpose Arabic corpus", Proc. 2nd Workshop Arabic Corpus Linguistics (WACL-2), pp. 22-25, Jul. 2013.
[39] Lin, C.Y. and Och, F.J., 2004, June. Looking for a few good metrics: ROUGE and its evaluation. In Ntcir workshop.