



OPEN ACCESS

EDITED BY

Arkaitz Zubiaga,
Queen Mary University of London,
United Kingdom

REVIEWED BY

Rakesh Balabantaray,
International Institute of Information
Technology, India
Nadim Rana,
Jazan University, Saudi Arabia

*CORRESPONDENCE

Bharathi Mohan Gurusamy
✉ g_bharathimohan@ch.amrita.edu

RECEIVED 04 April 2024

ACCEPTED 09 August 2024

PUBLISHED 29 August 2024

CITATION

Gurusamy BM, Rangarajan PK and
Altalbe A (2024) Whale-optimized LSTM
networks for enhanced automatic text
summarization.

Front. Artif. Intell. 7:1399168.

doi: 10.3389/frai.2024.1399168

COPYRIGHT

© 2024 Gurusamy, Rangarajan and Altalbe.
This is an open-access article distributed
under the terms of the [Creative Commons
Attribution License \(CC BY\)](#). The use,
distribution or reproduction in other forums is
permitted, provided the original author(s) and
the copyright owner(s) are credited and that
the original publication in this journal is cited,
in accordance with accepted academic
practice. No use, distribution or reproduction
is permitted which does not comply with
these terms.

Whale-optimized LSTM networks for enhanced automatic text summarization

Bharathi Mohan Gurusamy^{1*}, Prasanna Kumar Rangarajan¹ and Ali Altalbe^{2,3}

¹Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Chennai, India, ²Department of Computer Engineering, Prince Sattam bin Abdulaziz University, Alkharj, Saudi Arabia, ³Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

Automatic text summarization is a cornerstone of natural language processing, yet existing methods often struggle to maintain contextual integrity and capture nuanced sentence relationships. Introducing the Optimized Auto Encoded Long Short-Term Memory Network (OAELSTM), enhanced by the Whale Optimization Algorithm (WOA), offers a novel approach to this challenge. Existing summarization models frequently produce summaries that are either too generic or disjointed, failing to preserve the essential content. The OAELSTM model, integrating deep LSTM layers and autoencoder mechanisms, focuses on extracting key phrases and concepts, ensuring that summaries are both informative and coherent. WOA fine-tunes the model's parameters, enhancing its precision and efficiency. Evaluation on datasets like CNN/Daily Mail and Gigaword demonstrates the model's superiority over existing approaches. It achieves a ROUGE Score of 0.456, an accuracy rate of 84.47%, and a specificity score of 0.3244, all within an efficient processing time of 4,341.95 s.

KEYWORDS

LSTM, Whale Optimization Algorithm, summarization, optimization, Auto Encoded

1 Introduction

Over the past decade, text summarization has predominantly been a manual process, often time-consuming and subject to individual biases. The surge in the volume of information available online and from various sources has rendered manual summarization increasingly impractical. In the current era of information overload, automatic summarization has become essential for efficiently managing and distilling extensive text data.

Document summarization involves condensing text while retaining its crucial content, making it vital for effective information consumption. Whether extracting a summary from a single document or synthesizing multiple documents, summarization highlights key content, simplifying overall comprehension. Text summarization encompasses acquiring textual documents, processing the content, and delivering necessary information in a concise, user-friendly format. This essential process in modern information handling is broadly categorized into two types: abstractive and extractive summarization (Yang et al., 2019).

Abstractive summarization leverages Natural Language Processing (NLP) techniques to parse, reduce words, and generate summaries that encapsulate the original text's core ideas in a new form. Conversely, extractive summarization, known for its flexibility and time efficiency, involves analyzing sentences in matrix forms and identifying key sentences through feature vectors—*n*-dimensional numerical representations of text objects

(Mohan and Kumar, 2023). The objective of extractive summarization is to select sentences that align with specific requirements, effectively reducing text content while preserving its main message. Single document summarization focuses on condensing individual texts, whereas multi document summarization aims to synthesize and shorten a collection of similar documents into a cohesive summary. Despite their straightforward objectives, these tasks are complex and challenging to execute to the desired standards.

The advent of deep learning techniques has driven remarkable advancements in NLP. Text summarization, along with other tasks such as text translation and sentiment analysis, has significantly benefited from deep neural network models. These contemporary summarization approaches often utilize a sequence-to-sequence framework, typically an encoder–decoder model comprising neural networks trained on both input and output. Deep neural networks, leveraging large datasets, have demonstrated substantial improvements in summarization results.

Addressing these challenges and opportunities, this research introduces a groundbreaking system: the Optimized Auto Encoded Long Short-Term Memory Network (OAELSTM), enhanced by the Whale Optimization Algorithm (WOA). This innovative approach combines the robust feature extraction capabilities of Long Short-Term Memory (LSTM) networks with the data compression and efficiency benefits of autoencoders. The OAELSTM model is particularly adept at processing and synthesizing complex textual data into concise summaries.

The inclusion of WOA introduces a significant advancement in the approach. WOA is preferred over other benchmark metaheuristic algorithms like particle swarm optimization, gray fox optimization for the following reasons. Inspired by the bubble-net feeding behavior of humpback whales, WOA offers a unique biological basis that mimics natural behaviors not found in other algorithms, potentially providing advantages in handling specific optimization challenges. Moreover, WOA inherently balances exploration and exploitation, crucial for optimizing complex models like neural networks, which allows it to efficiently discover optimal solutions. Its randomized approach to solution updates, akin to the diving behavior of whales, may also lead to better convergence rates and solution quality compared to other swarm-based optimization algorithms. These factors collectively make WOA a robust choice for enhancing the efficiency and effectiveness of text summarization models.

Departing from traditional sentence-level summarization, the OAELSTM model focuses on extracting and rearticulating key phrases and pivotal concepts. This approach enhances the relevance and informativeness of generated summaries while effectively addressing common issues such as repetition and redundancy often encountered in standard summarization techniques. By prioritizing meaningful content extraction and employing advanced neural network optimization, this system sets a new benchmark in automatic text summarization.

The important contribution of this paper as follows:

- Introduction of the Optimized Auto Encoded Long short-term memory for text summarization, blending the robust feature extraction of LSTMs with the efficiency of autoencoders, marking a significant advancement in the field of automatic text summarization.
- Innovative application of the Whale Optimization Algorithm to optimize the OAELSTM model, a pioneering approach in natural language processing that enhances accuracy and efficiency in processing complex text.
- Shift from traditional sentence-level summarization to a focus on extracting and synthesizing key phrases and concepts, leading to more concise and content-rich summaries.
- Demonstrated superior performance of the OAELSTM model through rigorous evaluation on challenging datasets like Daily Mail and Gigaword, outperforming existing models as evidenced by ROUGE metric benchmarks.

The organization of this paper is structured as follows: Section 2 discusses related works, highlighting existing research and approaches in text summarization. Section 3 details the methodology employed, including the implementation of the WOA for feature selection and model optimization for summarization. Section 4 presents the results and discussions, where the effectiveness of the proposed approach is analyzed and compared with existing methods. Section 5 concludes the paper with a summary of findings and outlines avenues for future research and enhancements in text summarization techniques.

2 Related works

The field of text summarization has been predominantly driven by extractive techniques, where the focus has been on identifying and reproducing key sentences and phrases from source documents. This backdrop provides a context for exploration into the OAELSTM with WOA.

Ramachandran et al. (2022) introduced an autonomous approach for document summarization that emphasizes discourse coherence and keyword extraction through clustering algorithms. Their results demonstrate the efficiency of their method in creating clustered summaries. However, their approach may encounter scalability issues when applied to large-scale document collections, and the subjective nature of discourse coherence assessment. Complementing this, Yadav et al. (2022) offer a comprehensive overview of various text summarizing approaches and techniques, providing a broad perspective on the status and future directions of text summarization. In a similar vein, Latha et al. (2022) utilized NLP techniques, particularly Hugging Face transformers, to summarize video transcripts. This approach underlines the efficiency of NLP in extracting key patterns for summarization. On the abstractive front, Dugar et al. (2022) tackled unsupervised text summarization using an adversarial autoencoder model, combining K-Means clustering with language models for summary generation, model limits adaptability to less structured or specialized domains.

Pawar et al. (2022) explored an extractive text summarization technique using sentence clustering, employing Jaccard and Cosine similarity methods and the model depends on the similarity measure alone. While their approach shares similarities with the presented work in terms of clustering, the summarization technique differs. Ranganathan and Abuka (2022) leveraged the Text-to-Text Transfer Transformer (T5) model for summarization, focusing on the University of California, Irvine's drug reviews dataset, but the potential bias and capturing drug specific details are limitations. Pan et al. (2022) proposed a context-aware BERT ranking framework using abstractive

summarization to enhance text semantics, while Jain and Saini (2023) evaluated the effects of Extractive and Abstractive Summarization on text document classification. They utilized Gensim and Pegasus summarizers for extractive and abstractive summaries, respectively, highlighting the nuanced effects of NLP preprocessing in deep learning.

Further, Latief et al. (2023) critically evaluated the impact of preprocessing techniques like tokenization and case folding on sentence interpretability in deep learning models. Kuyumcu et al. (2019) challenged the norm by assessing the performance of a fastText classifier on Turkish text without preprocessing, demonstrating robust classification capabilities even with raw text. Shakil and Alam (2022) have made notable strides in the classification of hate speech by combining Convolutional Neural Networks (CNN) with NLP. Their research delves into the nuances of content moderation, focusing on creating deep learning models that are not only effective but also interpretable and explainable. Arora and Kansal (2019) contribute to this field by focusing on sentiment analysis using deep learning techniques to process raw Twitter data. Their work highlights the importance of text normalization in handling unstructured data, a challenge that is also central to proposed research.

Nadimi-Shahraki et al. (2022) propose an enhanced version of the WOA, designed to tackle global optimization problems. The potential of this enhanced WOA in the realm of feature selection for text summarization is noteworthy. The adaptation of such advanced optimization techniques could significantly improve the efficiency and accuracy of summarization models. Kundu et al. (2022) presents a study on an Altruistic WOA, demonstrating its superior accuracy and feature selection capabilities. The insights from this study could be particularly insightful for text summarization tasks, where selecting the most relevant features from the text is crucial for generating concise and coherent summaries.

Riyahi et al. (2022) demonstrates a WOA that effectively reduces the number of features while maintaining or even enhancing classification accuracy. This approach is directly relevant to the presented work, as it aligns with the objective of optimizing feature selection in text summarization, ensuring that our model captures the essence of the text without unnecessary complexity. Sun et al. (2023) developed a two-stage feature selection model using a binary WOA. The methodology proposed in this study could be applied to the feature selection process in text summarization, potentially enhancing the capability of proposed model to distill and summarize large volumes of textual data efficiently. Too et al. (2021) propose a new variant of WOA for high-dimensional feature selection. This variant could be particularly beneficial in summarizing complex text data, where dealing with high-dimensional feature spaces is a common challenge.

Malarselvi and Pandian (2023) introduce a multi-layered CNN model designed for feature representation in text summarization. This model is adept at extracting both linear and non-linear information, demonstrating the growing sophistication in using CNNs for complex text analysis tasks. Yu et al. (2021) present LenAtten, a novel model that notably improves length controllability and ROUGE scores in summarization tasks. Tested on the CNN/Daily Mail dataset, their approach provides insights into enhancing the precision and effectiveness of summary generation. Zhang et al. (2019) propose a novel pretraining-based encoder–decoder framework, generating output sequences for summarization. Their model achieves state-of-the-art results on CNN/Daily Mail and New York Times datasets, highlighting the potential of advanced encoder–decoder structures in text summarization.

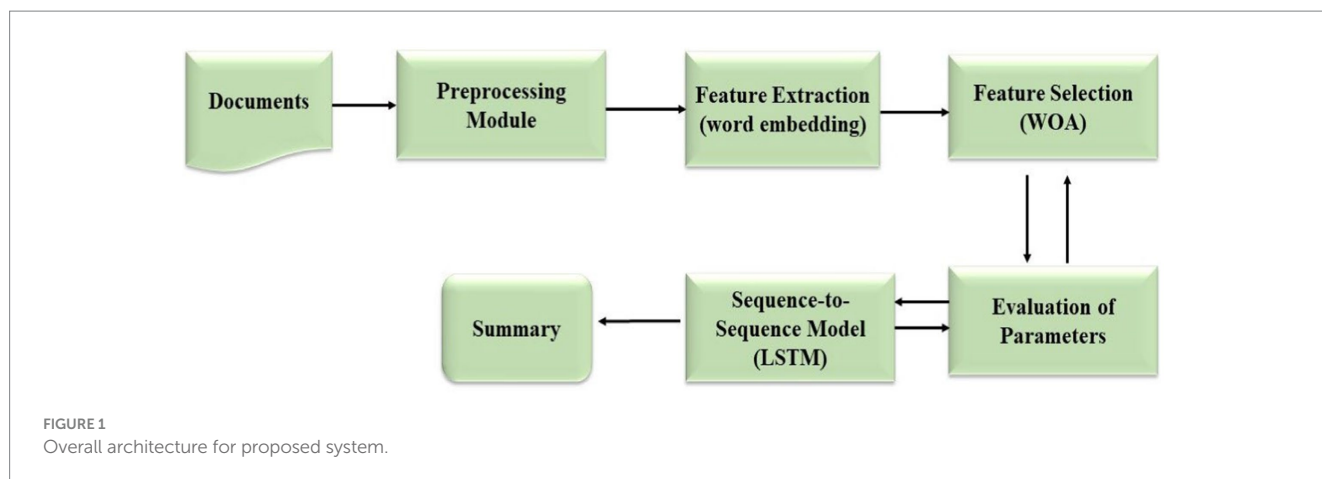
Song et al. (2018) report a framework that outperforms existing state-of-the-art models in terms of semantic and syntactic structure. The effectiveness of their model on datasets including CNN/Daily Mail underscores the importance of syntactic and semantic understanding in creating accurate and coherent summaries. Aliakbarpour et al. (2021) propose a novel abstractive summarization model that utilizes a combination of CNN and LSTM with an auxiliary attention mechanism. In their research, Gurusamy et al. (2023) have made significant strides in automatic text summarization by developing a hybrid model. This model integrates extractive methods, notably Semantic Latent Dirichlet Allocation (Semantic LDA) and Sentence Concept Mapping, with a transformer-based abstractive approach. Bharathi Mohan et al. (2023) delves into the comparison between BERT and GPT-2 in the context of text summarization, focusing on their performance in processing and summarizing multiple documents.

Existing text summarization methods face numerous challenges. Extractive techniques produce summaries lacking deeper context. Clustering-based approaches struggle with scalability and coherence. Graph-based methods are computationally heavy and struggle with informativeness versus redundancy. Adversarial models may not adapt to specialized domains and miss nuanced features. Similarity-focused techniques overlook semantic relationships. Abstractive models are resource-intensive and can introduce biases. Preprocessing techniques reduce interpretability with noisy data. Optimization methods need tuning for text complexity. CNN-based models and frameworks may not generalize well. Semantic methods are computationally demanding and prone to overfitting. WOA addresses these issues by efficiently selecting key features, reducing text dimensionality, and maintaining essential content. It is scalable, handles large tasks efficiently, adapts to various text types, and integrates well with other methods. WOA ensures coherent summaries by balancing informativeness and redundancy, making it a robust and adaptable solution for improving text summarization quality.

3 Methodology

3.1 Overall architecture

Figure 1 illustrates the architecture of the proposed model of text summarization system OAELSTM, employs a sophisticated architecture that progresses from the initial input of raw text to the generation of concise summaries. The system first preprocesses the text, where it undergoes tokenization and normalization to structure the data for feature extraction. Subsequently, meaningful features are distilled from the text, encapsulating its linguistic nuances. These features are then refined through the Whale Optimization Algorithm (WOA), which selects the most salient attributes essential for summarization. The core of the system is the LSTM network, designed to understand and preserve the context within the text. Integral to this network is an iterative optimization loop that fine-tunes the model parameters, enhancing the quality of the summarization. The culmination of this process is the output summary, which is meticulously evaluated for its fidelity to the original text using the ROUGE scoring metric, accuracy, and specificity. This end-to-end architecture is crafted to ensure that the final summary is not only succinct but also retains the integral meaning of the source text.



3.2 Data collection and preprocessing

The study utilizes a comprehensive dataset, CNN/Daily Mail¹ and Gigaword,² comprising a collection of document-summary pairs. The CNN/Daily Mail dataset, comprising news articles and human-written summaries, includes over 287,000 samples. The Gigaword dataset, derived from various news sources, consists of around 3.8 million training sets, alongside substantial testing, and validation sets.

The dataset serves as the foundational corpus for training and evaluating the proposed OAELSTM model. Initially, a table is created with two columns: Description and Category, represents the document content and its associated summary, respectively. The tabular format facilitates easier manipulation and processing of the data. To provide an overview of the training data, generate a word cloud visualization, which helps in identifying the most frequent terms within the dataset shown in Figure 2.

Effective preprocessing is paramount in preparing of the dataset for the OAELSTM model. These steps ensure that the input data is uniform and optimized for subsequent analysis:

- **Tokenization:** Each document is broken down into smaller units, or tokens, typically words or phrases, for systematic analysis.
- **Case Normalization:** All tokens are converted to lowercase to ensure consistency and avoid treating the same words with different capitalizations as distinct.
- **Punctuation Removal:** Punctuation marks are removed to streamline the text and focus on the core content.
- **Stemming:** Tokens are reduced to their root forms, allowing the model to process and recognize the fundamental meanings of words.

3.3 Feature representation through word embedding

In the methodology, MATLAB's `trainWordEmbedding` function is utilized for word embedding training with the Continuous Bag of

Words (CBOW) approach. Parameters include a 30-dimensional embedding dimension to balance semantic richness and computational efficiency, training conducted over 50 epochs to capture intricate semantic patterns, a context window size of 5 to consider word associations, an initial learning rate set at 0.05 for effective parameter updates, and a minimum word frequency count of 5 to filter out rare terms and noise from the vocabulary.

3.3.1 Word embedding training

The training of the word embedding is a crucial aspect of the proposed methodology. We set the embedding dimension to 30, aiming to balance the semantic richness of the word representations with computational efficiency. The embedding dimension directly impacts the semantic representation of words. A higher dimensionality can capture more nuanced relationships between words but requires more computational resources. Conversely, a lower dimensionality sacrifices some semantic granularity but improves computational efficiency. The training is conducted over 50 epochs to ensure comprehensive learning of the relationships between words in the dataset. More epochs generally allow for deeper learning and refinement of word embeddings. The choice of 50 epochs ensures comprehensive learning of word relationships without overfitting to the training data.

Utilizing MATLAB's `trainWordEmbedding` function, the approach can effectively train the word embeddings. This function is specifically designed for creating word embeddings, providing us with the tools to capture the nuanced semantic relationships inherent in the text. During this process, we maintain streamlined execution by disabling verbosity, which not only enhances the efficiency of the training but also minimizes distractions, a key consideration when dealing with large datasets.

3.3.2 Preparing document sequences

Post-training, the focus shifts to standardizing the length of each document to 40 tokens. This uniformity in sequence length, achieved using a custom document function, is crucial for ensuring consistent input lengths for neural network processing. Following the truncation, these documents are converted into sequences of word vectors using the trained embedding (`emb`). This conversion is essential to translate the text into a numerical format, rendering it suitable for analysis by neural network models.

1 https://huggingface.co/datasets/CNN_dailymail (Accessed January 2, 2024).

2 <https://huggingface.co/datasets/gigaword> (Accessed January 2, 2024).



3.3.3 Document sequence processing

A significant aspect of the methodology involves processing each document to transform it into a sequence of word vectors. This process is essential to align the textual data with the trained word embedding, ensuring that every word in a document is represented in its vector form.

a. Word Selection

The process begins by filtering words in a document, retaining only those found in the trained word embedding. This ensures the inclusion of words that carry meaningful semantic information.

b. Conversion to Vectors

Selected words are then transformed into their vector representations, converting textual data into a numerical format suitable for neural network analysis.

c. Optimization for Efficiency

To handle large datasets efficiently, the process is optimized for parallel processing, allowing multiple documents to be prepared simultaneously, thus enhancing speed and efficiency.

d. Resulting Data Format

The outcome is a sequence for each document where words are represented by their vector embeddings, ensuring consistent data representation for deep learning models.

e. Data Padding

Finally, sequences are padded to a uniform length, essential for batch processing in neural networks, ensuring stable and reliable model performance.

3.3.4 Dataset partitioning and parameter setting

3.3.4.1 Data preparation for model training and validation

In this phase of the research process, focus shifts to preparing the dataset for both training and validation. This step is crucial for

evaluating the performance of the model and ensuring its generalizability.

a. Loading the Training Data

The process begins by loading the training data (XTrain), which has been previously processed and prepared for the model.

b. Partitioning the Dataset

To assess the model's performance accurately, we allocate a portion of the data for validation. Using the hold-out method, we set aside 20% of the data as a validation set. This partitioning is achieved using MATLAB's cvpartition function with a hold-out ratio (ho) of 0.2, ensuring that the model is tested on unseen data, a critical factor for evaluating its robustness and predictive power.

3.3.4.2 Parameter setting for the model

The process of setting parameters is essential for the configuration and optimization of the proposed model.

a. Parameter Initialization

We initialize several key parameters, including the number of search agents (N) set to 10 and the maximum number of iterations (max_iter) set to 100. These parameters are integral to the operation of the model, impacting its learning and optimization capabilities.

b. Summary Extraction for Demonstration

To showcase the functionality of the model, we extract a summary from the first category in the dataset. We use the function for summary extraction, specifying the size of the summary as 6 units and ordering it by position. This extracted summary, obtained from the model's vocabulary, serves to demonstrate the summarization capability of the system.

c. Preparation for Optimization

The summary extracted is then used in conjunction with the training data (XTrain) for the subsequent steps in the model's training and optimization process.

3.4 Feature selection using WOA

In the study, an algorithm denoted as WOA_Opt_Feature) was implemented to optimize feature selection using the Whale Optimization Algorithm (WOA) (Algorithm 1). This algorithm is integral to identifying the most effective features for the text summarization model. The algorithm is designed to perform feature optimization through the principles of WOA. It operates by simulating the social and hunting behavior of humpback whales, a process that involves iteratively updating the positions of search agents (potential solutions) within the search space.

ALGORITHM 1: WOA_Opt_Feature

Input:

- o Features (feat)
- o Lower and Upper Bounds (lb, ub)
- o Number of Agents (N)
- o Maximum Iterations (max_Iter)

Output:

- o Selected Features (sFeat)
- o Indices of Selected Features (Sf)
- o Number of Selected Features (Nf)

Initialization:

- o Initialize a population of 'N' agents with random positions in the search space.
- o The dimension (dim) of each agent's position corresponds to the number of features in feat.
- o Define the search space with lower (lb) and upper (ub) bounds.

Fitness Evaluation:

- For each agent, evaluate fitness using WOA_fitness, which measures the effectiveness of the feature set represented by the agent's position.

Iterative Position Update:

- For $t = 1$ to max_Iter:
- Update the position of each agent based on simulated whale behaviors, adjusting parameters A , C , p , and l .
- Simulate encircling prey and spiral movement to adjust the agent's position in the search space.

Feature Selection and Performance Tracking:

- o At the end of iterations, identify the best solution (agent position) based on fitness evaluations.
- o Extract the set of selected features (sFeat) from the best solution.

- o Record the indices (Sf) and count (Nf) of the selected features.
- o Track the performance of the algorithm over iterations for analysis.

3.5 Evaluation of feature set effectiveness

In the study, a two-part algorithm is employed to evaluate the effectiveness of feature sets selected by the Whale Optimization Algorithm for the text summarization model. This evaluation is crucial in identifying the most relevant and impactful features for summarization tasks. The effectiveness of a feature selection discussed in Section 4.2.

a. Fitness Function: Evaluating Feature Sets

The first part of the algorithm is the fitness function, WOA_fitness (Algorithm 1.1). This function plays a pivotal role in the Whale Optimization Algorithm (WOA) by assessing the quality of each feature set.

ALGORITHM 1.1: Fitness Evaluation in WOA_Opt_Feature

Objective: To evaluate the effectiveness of feature sets selected by the Whale Optimization Algorithm (WOA) using a k-Nearest Neighbors (KNN) classifier.

Input:

- o Features (feat)
- o Labels (label)
- o Binary vector representing selected features (X)
- o Hold-out partition object (HO)

Output:

- Cost of the feature set (cost)

Process:

Check for Feature Selection:

- If no features are selected ($\text{sum}(X == 1) == 0$), set cost to 1, indicating the lowest possible effectiveness.
 - If features are selected, proceed to evaluate their effectiveness.
- Feature Set Effectiveness Evaluation:*
- Call jwrapperKNN to evaluate the accuracy of the selected features using the KNN classifier.

Calculate the cost as the inverse of the accuracy obtained from jwrapperKNN.

b. Evaluate the fitness function

The second part of the algorithm is evaluating the fitness function using KNN classifier. To evaluate the effectiveness of a feature set, the fitness function utilizes a wrapper function for the k-Nearest Neighbors (KNN) classifier, named jwrapperKNN and Algorithm 1.2 outlines the fitness function. The process involves using selected features (sFeat) and corresponding labels (label) with a hold-out partition object (HO) to evaluate a KNN classifier. Parameters are set with $k=5$ for the number of nearest neighbors. Data is split into training (xtrain, ytrain) and validation (xvalid, yvalid) sets using HO. The KNN model is trained on xtrain and ytrain, then used to predict labels for xvalid. Accuracy is calculated as the proportion of correct predictions, from which the error rate ($\text{error} = 1 - \text{Acc}$) is derived to assess classifier performance.

ALGORITHM 1.2: Wrapper Function for KNN Classifier: jwrapperKNN

Input:

- Selected features (*sFeat*)
- Labels (*label*)
- Hold-out partition object (*HO*)

Output:

- Error rate of the classifier (*error*)

Process:

Parameter Setting for KNN:

- Set the number of nearest neighbors (*k*) to 5.
- Data Split for Training and Validation:
- Split the data into training (*xtrain*, *ytrain*) and validation (*xvalid*, *yvalid*) sets based on *HO*.

KNN Model Training and Prediction:

- Train the KNN model using the training data.
- Predict labels for the validation data and compare them with actual labels to calculate accuracy.

Accuracy and Error Calculation:

- Calculate the accuracy (*Acc*) as the proportion of correct predictions.
- Compute the error rate ($\text{error} = 1 - \text{Acc}$), representing the inverse of the classifier's effectiveness.

3.5.1 Parameter and objective space in WOA optimization

To illustrate the optimization process, we plot the parameter space and the objective space. The parameter space plot visualizes the function landscape that the WOA navigates, providing insights into the complexity of the optimization task. The objective space plot, drawn in a semilogarithmic scale, shows the progression of the

algorithm over iterations, highlighting the improvement in the solution with each step. The outputs of this process, including the best solution and the optimal value of the objective function, are displayed, providing clear evidence of the effectiveness of WOA in optimizing the model. The performance of the algorithm over iterations is tracked using a convergence curve shown in Figure 3 and detailed analysis done in Section 4.1.

3.6 Summarization using sequence to sequence model LSTM

3.6.1 Data preparation for training and testing

Optimized Training Data (*XTrain1*): This dataset, which has undergone feature selection via WOA, provides a refined set of features for the model, ensuring that only the most relevant information is used for training and testing. This enhances the model's ability to focus on significant textual patterns.

Training and Test Set Partitioning: Splitting the data into training (*XTrain*) and test (*XTest*) sets ensures that the model is evaluated on unseen data, a key aspect for assessing its generalizability.

3.6.2 Model architecture for text summarization

In developing the text summarization model, a carefully designed architecture leverages the strengths of neural networks in processing and interpreting natural language data.

a. Sequence Input Layer

The foundation of the model is the sequence input layer, specifically tailored to handle sequences of data. In this application, this involves processing sequences of word vectors derived from trained word embeddings. This layer effectively captures the sequential nature of text data, setting the stage for deeper analysis.

b. Intermediate Layers

Following the input layer, the model comprises a series of intermediate layers, each serving a distinct purpose:

RELU Activation Layers: These layers introduce non-linearity into the model, enabling it to learn complex patterns in the data.

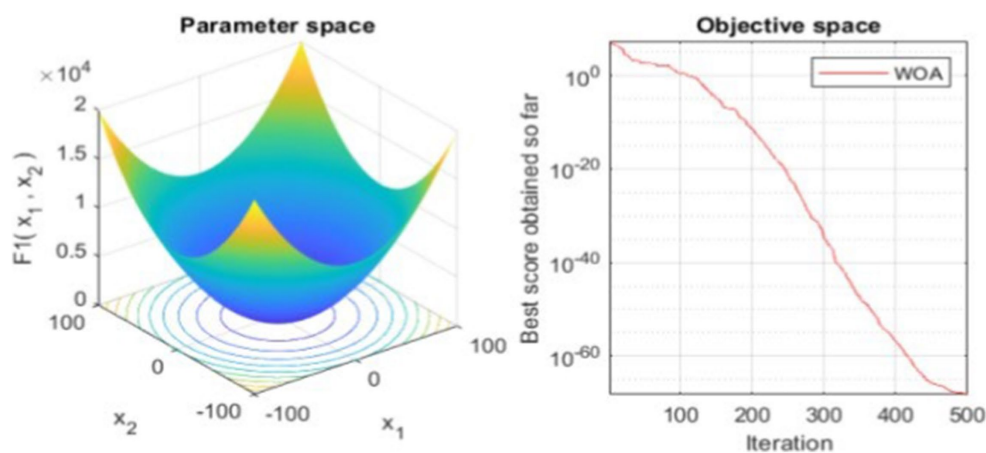


FIGURE 3
Parameter space vs. objective space in WOA optimization.

Flattening Layer: This layer flattens the output from the previous layers into a format suitable for input into the LSTM layers, ensuring seamless data flow.

LSTM Layers with Dropout: We incorporate Long Short-Term Memory (LSTM) layers, each with 200 hidden units. These layers are pivotal in capturing long-range dependencies within the text, a critical aspect of understanding and summarizing content. The inclusion of dropout layers helps in preventing overfitting, a common challenge in deep learning models.

c. Output Layers

The culmination of the model is in its output layers, which consist of:

Softmax Layer: This layer computes the probability distribution across 6,973 classes, corresponding to the potential summaries. It transforms the output of the LSTM layers into a probabilistic format, indicating the likelihood of each class being the correct summary.

Classification Layer: Serving as the final layer, it is responsible for the actual classification task, determining the most probable summary based on the probabilities provided by the softmax layer.

3.6.3 Training setup for the text summarization model

In developing the text summarization model, the training setup has been meticulously configured to optimize the learning process and ensure the model effectively captures the nuances of the data.

a. Adam Optimizer

The 'adam' optimizer used for training the model. Adam, short for Adaptive Moment Estimation, is a popular optimization algorithm in deep learning because it combines the advantages of two other extensions of stochastic gradient descent: Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). Adam is known for its effectiveness in handling sparse gradients and its adaptive learning rate capabilities, making it particularly suited for complex tasks like text summarization.

b. Maximum Epochs

The training process is set to run for a substantial number of epochs, specifically 1,500. This high epoch count is chosen to ensure that the model undergoes comprehensive learning, thoroughly adjusting and refining its weights in response to the intricacies of the training data. Training is crucial for deep learning models, as it allows for the gradual and detailed extraction of features and patterns from the text.

c. Initial Learning Rate from WOA

A novel aspect of the training setup is the use of the best score obtained from the Whale Optimization Algorithm (WOA) as the initial learning rate. This approach integrates the insights from the feature selection process directly into the model's training. By adapting the learning rate based on the performance of feature sets optimized by WOA, we align the model's learning process more closely with the characteristics of the most effective features. This synergy enhances the overall efficacy of the model.

d. Training Data Subset

The model is trained on a carefully selected subset of the processed dataset (XTrain). This subset comprises sequences of text data, pre-processed and structured for efficient learning. The corresponding labels (Feature selected or not), which categorize the data, are used to guide the supervised learning process. These labels play a critical role in shaping the model's understanding and generation of summaries.

Each piece of data within this subset is associated with specific labels that guide the supervised learning process. These labels serve crucial roles in shaping how the model understands and generates summaries. The process of creating this subset involves filtering XTrain based on these labels to construct a balanced and representative training set.

3.6.4 Model evaluation

After the completion of the training phase, we conducted a comprehensive evaluation of the text summarization model. This evaluation aimed to assess the model's performance in accurately summarizing text, both quantitatively and qualitatively.

a. Classification Process

The proposed system employed the trained model (to perform classification tasks on both the training dataset (XTrain) and the test dataset (XTest). This step is crucial to understand how well the model performs on data it has seen (training data) and on new, unseen data (test data).

b. Accuracy and Recall Assessment

The model's predictions were systematically compared against the original labels of the datasets. This comparison allowed us to evaluate two key metrics: accuracy, which measures the proportion of correctly predicted summaries, and recall, which assesses how many relevant summaries were correctly identified by the model.

4 Results and discussion

4.1 Parameter space and object space analysis

In the results, [Figure 1](#) play a crucial role in visualizing the efficacy of the proposed WOA based algorithm WOA_Opt_Feature applied to the text summarization model.

[Figure 3](#) illustrates the complex optimization landscape that the WOA navigates. The three-dimensional surface plot, with axes representing the parameters being optimized, depicts the cost function landscape. The multi-modal nature of this landscape, with its numerous local minima, is evident and underscores the robustness of the WOA in seeking out the global minimum. The valleys and contours on this plot provide a visual representation of the algorithm's search strategy, reflecting the intricate exploration and exploitation balance maintained throughout the optimization process.

[Figure 3](#) also presents the objective space through a semilogarithmic plot, capturing the proposed WOA's iterative performance. The convergence curve showcases the algorithm's efficiency in progressively refining the solution. Notably, the sharp decline in the curve during the initial iterations indicates rapid improvements in solution quality, which is a testament to the WOA's capability for quick convergence. The flattening of the curve in later iterations suggests that the algorithm is approaching the optimal solution, with incremental gains as it fine-tunes the parameters.

The convergence curve, as detailed in [Figure 3](#), is indicative of the optimization depth achieved by the WOA. The best score obtained and the corresponding position in the parameter space are significant outcomes that demonstrate the potential of the WOA in enhancing the model's summarization performance. The plots collectively offer a comprehensive overview of the optimization journey, from the initial parameter setting to the final selection of an optimal feature set.

4.2 Feature selection impact on training data

In the process of refining the model, we applied the Whale Optimization Algorithm (WOA) to select the most relevant features from the training data. This section compares the training data before (XTrain) and after (XTrain1) feature selection to demonstrate the algorithm's effectiveness.

a. Before Feature Selection: XTrain

The initial training data, visualized in Figure 4, contains a full set of features, represented by a multidimensional array where each column corresponds to a specific feature of the data. The diversity in the values reflects the comprehensive nature of the raw data set, which includes a wide range of information encapsulated in the various dimensions.

b. After Feature Selection: XTrain1

Figure 5, which represents the training data after feature selection, shows a refined set of features. The reduction in dimensions is indicative of the WOA's ability to identify and retain the most informative features, discarding those that contribute less to the summarization task's performance. The resulting dataset is expected to be less noisy and more focused on the key aspects that contribute to generating accurate text summaries.

c. Interpretation of Results

By comparing the two figures, we can observe the dimensionality reduction achieved through the optimization process. The selective feature set in XTrain1 is anticipated to enhance the model's learning efficiency by reducing the complexity of the input space and focusing on the quality rather than the quantity of the information provided during training.

The distilled feature set not only streamlines the training process but also is likely to improve the model's generalizability by mitigating the risk of overfitting. With fewer, more impactful features, the model can better capture the essence of the text, resulting in more

precise and coherent summaries. The results suggest that the proposed WOA has successfully optimized the feature space, potentially leading to improved summarization accuracy and a more robust model.

4.3 Quantitative analysis of proposed framework

4.3.1 Rouge scores

The Rouge-N score evaluates the similarity between n-grams in the generated summary and a reference summary, with Rouge-1 specifically focusing on the overlap of individual words. This metric, crucial for assessing summarization quality, quantifies the extent of shared content, providing a clear measure of a summary's accuracy and relevance. The Rouge-N score measures the overlap between the n-grams in the generated summary and the reference summary. For Rouge-1 (which measures the overlap of 1-gram or each word), the formula is shown in Equation 1:

$$Rouge_N = \frac{\sum_{s \in \{Reference\ Summaries\}} \sum_{gramn \in s} Count(gramn)}{\sum_{s \in \{Reference\ Summaries\}} \sum_{gramn \in s} Countmatch(gramn)} \quad (1)$$

Where $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries. $Count(gram_n)$ is the count of n-grams in the reference summaries.

Figure 6 presents a bar chart comparing the Rouge scores of several text summarization algorithms such as Token Level Fact Correction (TLFC), Two Stage Network (TSN), Pointer Generator Algorithm (PGA) and Seq-Seq+Attention Baseline (S-S+AB)

	30	31	32	33	34	35	36	37	38	39	40	41
1	1354	0.1871	0.0463	-0.3305	-0.1326	0.2564	0.1206	0.0463	0.7114	-0.1326	0.1237	0.4040
2	4078	0.0489	-0.1150	-0.2050	-0.2709	0.0034	-0.2666	-0.1150	-0.2638	-0.2709	-0.1087	-0.0730
3	4042	0.1295	-0.1380	-0.0933	0.0140	0.0429	-0.0275	-0.1380	-0.1646	0.0140	0.1665	0.2172
4	4172	-0.1265	-0.0181	-1.0970	0.0058	-0.2061	-0.1815	-0.0181	0.0122	0.0058	0.1562	-0.1298
5	1043	-0.3189	-0.2344	-0.2990	-0.1428	-0.1484	-0.1792	-0.2344	0.1775	-0.1428	-0.0570	-0.2161
6	2328	-0.3746	-0.3548	0.0050	-0.2721	-0.1544	-0.2652	-0.3548	-0.4646	-0.2721	-0.5113	-0.4678
7	4004	0.2663	0.3327	0.2352	0.6102	0.4984	0.4098	0.3327	0.5358	0.6102	0.3108	0.1519
8	3363	-0.2511	0.0208	0.9153	0.0116	-0.0783	0.1667	0.0208	0.7350	0.0116	0.0932	-0.3295
9	2884	0.0641	-0.0386	0.5968	0.1862	0.0115	0.2240	-0.0386	0.0280	0.1862	-0.2954	-0.0757
10	5654	0.0477	-0.1470	-0.5663	0.1998	0.2138	-0.2293	-0.1470	0.2344	0.1998	0.0143	0.0309
11	3322	-0.2279	-0.2661	-0.8007	-0.3496	-0.1959	-0.4565	-0.2661	-0.5606	-0.3496	-0.3464	-0.2762
12	2674	-0.2770	0.0037	-0.0910	-0.1655	-0.3119	-0.2662	0.0037	0.4446	-0.1655	0.1050	-0.3856
13	1827	-0.4479	-0.5685	-0.4951	-0.5362	-0.5551	-0.6518	-0.5685	-0.7166	-0.5362	-0.5994	-0.3189
14	2421	0.1060	0.2060	0.2141	0.0229	0.1292	0.2435	0.2060	0.2816	0.0229	0.2744	0.2059
15	2121	0.3243	0.2790	0.5328	0.3516	0.3148	0.1672	0.2790	-0.3055	0.3516	0.2358	0.3080
16	3011	0.2725	0.2289	-0.1079	0.2918	0.1925	0.2003	0.2289	0.4724	0.2918	0.3371	0.4170

FIGURE 4 Feature matrix XTrain derived from dataset before feature selection.

XTrain1(1, 1)		8	9	10	11	12	13	14	15	16	17	18	19
1784		-0.0484	-0.2570	-0.0260	0.1404	-0.1944	0.1354	0.0463	-0.3305	0.2564	0.0463	0.4040	
2444		0.1100	-0.0543	-0.1451	-0.1130	-0.1630	0.4078	-0.1150	-0.2050	0.0034	-0.1150	-0.0730	
3470		0.1063	-0.2773	0.0714	0.1449	0.3049	0.4042	-0.1380	-0.0933	0.0429	-0.1380	0.2172	
4488		-0.5407	-0.6525	0.1250	-0.8662	0.4365	-0.4172	-0.0181	-1.0970	-0.2061	-0.0181	-0.1298	
5057		-0.4146	-0.1135	-0.4645	-0.4089	-0.3387	-0.1043	-0.2344	-0.2990	-0.1484	-0.2344	-0.2161	
6971		-0.3391	-0.3428	-0.5832	-0.4412	-0.5880	-1.2328	-0.3548	0.0050	-0.1544	-0.3548	-0.4678	
7025		0.3614	0.3271	0.3155	0.2099	0.0616	0.4004	0.3327	0.2352	0.4984	0.3327	0.1519	
8482		0.0089	0.5634	0.1733	1.0584	0.1713	-0.0363	0.0208	0.9153	-0.0783	0.0208	-0.3295	
9731		-0.2223	0.5530	0.4694	-0.3623	0.2432	0.2884	-0.0386	0.5968	0.0115	-0.0386	-0.0757	
10590		-0.3346	-0.6775	-0.0011	-0.1386	-0.1069	-0.5654	-0.1470	-0.5663	0.2138	-0.1470	0.0309	
11790		0.0628	-0.2561	-0.6027	-1.0422	-0.5077	-0.8322	-0.2661	-0.8007	-0.1959	-0.2661	-0.2762	
12470		-0.1540	-0.1354	0.1212	0.0585	-0.2102	-0.0674	0.0037	-0.0910	-0.3119	0.0037	-0.3856	
13299		-0.3883	-0.6917	-0.3236	-0.0390	-0.4690	0.1827	-0.5685	-0.4951	-0.5551	-0.5685	-0.3189	
14981		0.3265	-0.0242	0.2716	0.5293	-0.0377	0.2421	0.2060	0.2141	0.1292	0.2060	0.2059	
15058		0.2321	0.1491	0.3446	0.0775	0.2033	-0.2121	0.2790	0.5328	0.3148	0.2790	0.3080	
16929		0.3657	0.0912	0.1572	0.0746	0.7556	-0.0011	0.2289	-0.1079	0.1925	0.2289	0.4170	

FIGURE 5 Feature matrix XTrain1 derived from dataset after feature selection.

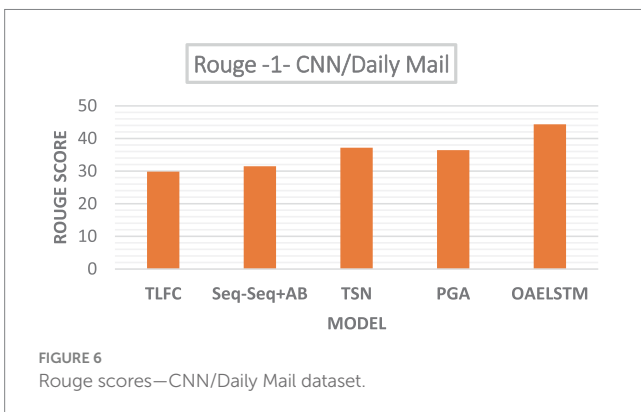


FIGURE 6 Rouge scores—CNN/Daily Mail dataset.

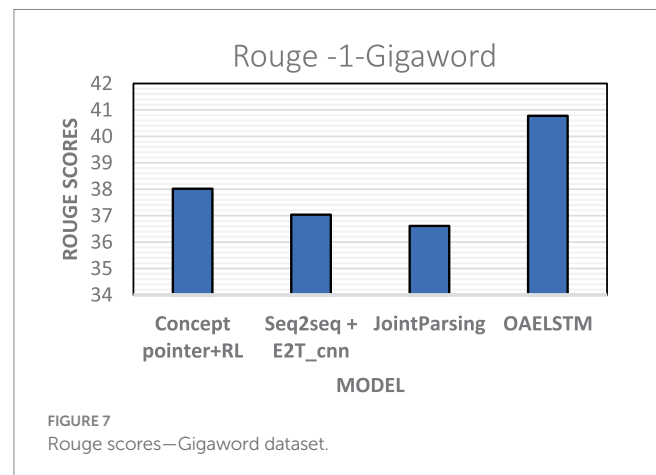


FIGURE 7 Rouge scores—Gigaword dataset.

highlighting the performance of the proposed system. Our study’s comparative analysis on the CNN/Daily Mail dataset, as illustrated, reveals the OAE LSTM system’s superior performance in automated text summarization. With a ROUGE-1 score of 44.34, the OAE LSTM system outperforms other notable algorithms—TLFC (29.86), Seq-Seq + AB (31.48), TSN (37.16), and PGA (36.44)—underscoring its advanced capability in producing summaries that closely align with human reference standards.

The evaluation conducted on the Gigaword dataset demonstrated the effectiveness of proposed OAE LSTM model, which achieved a ROUGE-1 score of 40.78 shown in Figure 7. This outperforms other summarization approaches, including Concept Pointer + RL (38.02), Seq2seq + E2T_cnn (37.04), and JointParsing (36.61). The higher score of OAE LSTM indicates its superior capability in generating summaries that more closely mirror the reference texts.

4.3.2 Specificity

Especially within the context of the approach using the OAE LSTM model, Specificity is important because it measures the model’s ability to correctly identify and reject non-essential or

irrelevant information when creating a summary. In the case of the OAE LSTM model, which likely includes an optimization component (WOA) to refine feature selection, specificity helps to gauge how well the model has been tuned to prioritize the most significant features for summarization. A high specificity score would suggest that the model, after optimization, is adept at pinpointing and leveraging the most informative features while disregarding the less important ones.

Specificity: Also known as the True Negative Rate, it measures the proportion of actual negatives that are correctly identified given in Equation 2

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}} \quad (2)$$

Where True Negatives are the correctly identified negative cases and False Positives are the cases that were incorrectly predicted as positive but are negative.

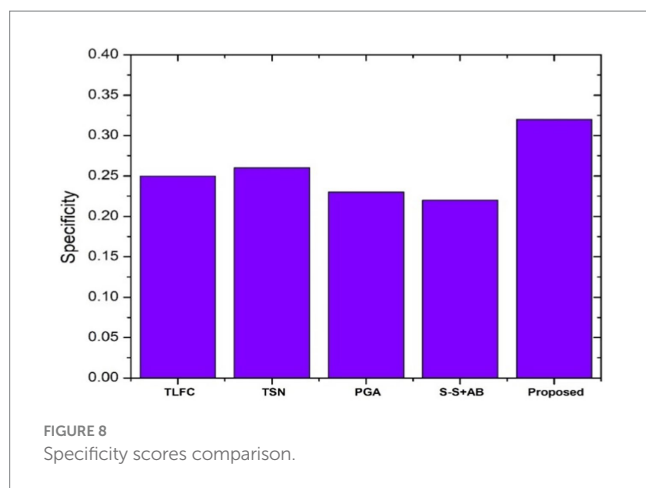


Figure 8 exhibits the specificity scores of different summarization algorithms, emphasizing the performance of the proposed model on CNN/Daily Mail Dataset. The proposed system exhibits a notably higher specificity score, outshining its counterparts. This metric is indicative of the model's ability to correctly include pertinent details in the summaries while excluding irrelevant information. Compared to other models like the Token-level fact correction and the Pointer generator, the proposed system's advanced feature selection process likely contributes to its enhanced specificity, enabling it to focus more accurately on the essential content of the text.

4.3.3 Accuracy

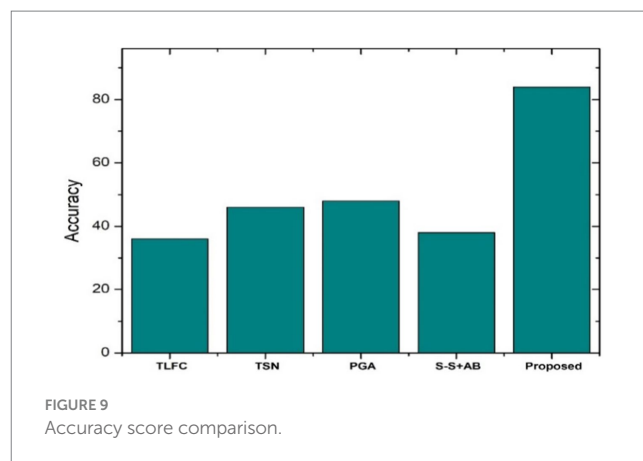
Accuracy measurement in text summarization is essential for evaluating how closely the model's generated summaries match the expected outputs. It serves as a key metric for assessing the effectiveness of the summarization model, like OAE LSTM, and is crucial for fine-tuning the model's parameters and for benchmarking its performance against other models. High accuracy ensures that the summaries are reliable and reflective of the original text, thereby enhancing user trust and experience.

Accuracy is the ratio of correctly predicted observations to the total observations and is used for classification tasks given in Equation 3.

$$\text{Accuracy} = \frac{\text{Total Number of Predictions} / \text{Number of Correct Predictions}}{\quad} \quad (3)$$

Figure 9 depicts the accuracy scores for various summarization algorithms, including the proposed OAE LSTM model on CNN/Daily Mail. The proposed model registers the highest accuracy score, substantially surpassing other algorithms such as the Token-level fact correction, Two-stage network, and Pointer generator. The accuracy score is critical as it measures the proportion of correct predictions made by the summarization model, which in this case, reflects the model's ability to generate summaries that are well-aligned with the expected outputs.

The Token-level fact correction model shows the lowest accuracy, suggesting limitations in its summarization capabilities. The Two-stage network and Pointer generator models exhibit moderate accuracy scores, indicating room for improvement in summary generation. The chart clearly demonstrates the effectiveness of the proposed OAE LSTM model in producing accurate summaries, which



is a testament to the model's robustness and the efficacy of the optimization techniques employed.

4.4 Comparative analysis and discussion

Our study's findings, as illustrated in the ROUGE score comparison table, highlight the enhanced performance of the OAE LSTM model over a range of established summarization approaches. Table 1 encapsulates ROUGE-1, ROUGE-2, and ROUGE-L scores, providing a comprehensive view of each model's capabilities in generating summaries that align with human references.

The OAE LSTM model demonstrates a notable advantage with the highest ROUGE-1 score of 44.34, indicating its strength in capturing key unigrams from the source text. Furthermore, it leads in ROUGE-2 with a score of 20.24, suggesting superior capture of bigrams or phrase-level information. In terms of ROUGE-L, which assesses the longest common subsequence, the OAE LSTM model achieves a score of 35.89, underscoring its ability to maintain sentence-level structure and coherence.

The findings of the study, as illustrated by the ROUGE score comparison in Table 2, highlight the enhanced performance of the OAE LSTM model over a range of established summarization approaches. The table encapsulates ROUGE-1, ROUGE-2, and ROUGE-L scores, providing a comprehensive view of each model's capabilities in generating summaries that align with human references.

In particular, the OAE LSTM model stands out with the highest ROUGE-1 score of 44.34, showcasing its effectiveness in capturing key unigrams from the source text. Additionally, it leads in ROUGE-2 with a score of 20.24, indicating superior capture of bigrams or phrase-level information. When it comes to ROUGE-L, which evaluates the longest common subsequence, the OAE LSTM model achieves a score of 35.89, highlighting its ability to preserve sentence-level structure and coherence in the generated summaries.

4.5 Qualitative analysis

Table 3 showcases examples of summaries generated by the OAE LSTM framework for articles from the CNN/Daily Mail and Gigaword datasets. The summaries provide a concise encapsulation of the primary content and context of the original articles. These summaries illustrate the effectiveness of the OAE LSTM framework in

TABLE 1 Comparison of the existing model with proposed with OAELSTM for ROUGE scores using CNN/Daily Mail dataset.

Model	Rouge-1	Rouge-2	Rouge-L
TLFC (Shin et al., 2023)	29.86	11.28	27.78
Seq-Seq + AB (Nallapati et al., 2016)	31.48	11.17	28.08
TSN (Dilawari et al., 2023)	37.16	17.81	33.83
PGA (See et al., 2017)	36.44	15.66	33.42
OAELSTM	44.34	20.24	35.89

TABLE 2 Comparison of the existing model with proposed with OAELSTM for ROUGE scores with Gigaword dataset.

Model	Rouge-1	Rouge-2	Rouge-L
Concept pointer + RL (Wang et al., 2019)	38.02	16.97	35.43
Seq2seq + E2T_cnn (Amplayo et al., 2018)	37.04	16.66	34.93
JointParsing (Song et al., 2020)	36.61	18.85	34.33
OAELSTM	40.78	20.26	35.67

TABLE 3 Sample summary generated from OAELSTM framework.

CNN/Daily Mail—'article'	Aluko nets winner with 10 min remaining at KC Stadium. Tomas Marek put visitors into shock lead after 2 min. Ahmed Elmohamady equalized for the hosts. Steve Bruce's side now await Europa League play-off.
Gigaword—document	Australia's current account deficit narrows sharply in the June quarter due to soaring commodity prices. Figures released Monday showed that the deficit was \$1.2 billion US.

distilling the essence of complex news articles into succinct, coherent summaries. The model's ability to maintain the key points and factual accuracy of the original texts while condensing the information highlights its potential utility in diverse applications such as news aggregation, content curation, and information retrieval.

5 Conclusion and future work

The proposed study established the Optimized Auto Encoded Convolutional Neural Network (OAELSTM) with the Whale Optimization Algorithm (WOA) as a potent framework for text summarization. The OAELSTM model demonstrated a notable enhancement in summarizing texts, outperforming standard models in terms of Rouge score, accuracy, and specificity. The key innovation lies in its ability to effectively condense complex textual content into succinct, coherent summaries without losing the essence of the original texts. This breakthrough addresses critical challenges in natural language processing and opens new avenues in automated text summarization.

For future research, the focus will be on refining and expanding the capabilities of the OAELSTM framework. One area of interest is the incorporation of more advanced neural network structures to deepen the summarization context and accuracy. Exploring optimization techniques beyond WOA could further enhance feature selection and model tuning. Additionally, adapting the OAELSTM model for specific domains such as legal or medical texts can make it a more versatile tool. Assessing the model's scalability for larger datasets and its practical application in various industries will also be crucial. These enhancements will not only bolster the OAELSTM framework's effectiveness but also broaden its applicability in diverse real-world scenarios.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

BG: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Validation, Visualization, Writing – original draft, Writing – review & editing. PR: Conceptualization, Investigation, Supervision, Writing – review & editing. AA: Funding acquisition, Resources, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. The authors extend their appreciation to Prince Sattam Bin Abdulaziz University for funding this research work through the project number (PSAU/2024/01/921063).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations,

or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Aliakbarpour, H., Manzuri, M. T., and Rahmani, A. M. (2021). Improving the readability and saliency of abstractive text summarization using combination of deep neural networks equipped with auxiliary attention mechanism. *J. Supercomput.* 78, 2528–2555. doi: 10.1007/s11227-021-03950-x
- Amplayo, R. K., Lim, S., and Hwang, S. (2018). Entity commonsense representation for neural abstractive summarization. In Proceedings of the 2018 conference of the North American chapter of the Association for Computational Linguistics, pp. 697–707, Human Language Technologies, New Orleans, Louisiana.
- Arora, M., and Kansal, V. (2019). Character level embedding with deep convolutional neural network for text normalization of unstructured data for twitter sentiment analysis. *Soc. Netw. Anal. Min.* 9, 1–14. doi: 10.1007/s13278-019-0557-y
- Bharathi Mohan, G., Prasanna Kumar, R., Parathasarathy, S., Aravind, S., Hanish, K. B., and Pavithria, G. (2023). "Text summarization for big data analytics: a comprehensive review of GPT 2 and BERT approaches" in Data analytics for internet of things infrastructure. Internet of things. eds. R. Sharma, G. Jeon and Y. Zhang (Cham: Springer).
- Dilawari, A., Khan, M. U. G., Saleem, S., and Shaikh, F. S. (2023). Neural attention model for abstractive text summarization using linguistic feature space. *IEEE Access* 11, 23557–23564. doi: 10.1109/ACCESS.2023.3249783
- Dugar, A., Singh, G., B. N., and M. A. K. (2022). "Unsupervised abstractive text summarization with length controlled autoencoder," 2022 IEEE 19th India council international conference (INDICON), Kochi, India, pp. 1–6.
- Gurusamy, B. M., Rengarajan, P. K., and Srinivasan, P. (2023). A hybrid approach for text summarization using semantic latent Dirichlet allocation and sentence concept mapping with transformer. *Int. J. Electric. Comput. Eng.* 13, 6663–6672. doi: 10.11591/ijece.v13i6.pp6663-6672
- Jain, C., and Saini, J. R. (2023). "A novel approach using extractive and abstractive summarization for the genre classification of short text," 2023 third international conference on advances in electrical, computing, communication and sustainable technologies (ICAECT), Bhilai, India, pp. 1–7.
- Kundu, R., Chattopadhyay, S., Cuevas, E., and Sarkar, R. (2022). AltWOA: altruistic whale optimization algorithm for feature selection on microarray datasets. *Comput. Biol. Med.* 144:105349. doi: 10.1016/j.compbiomed.2022.105349
- Kuyumcu, Birol, Aksakalli, Cuneyt, and Delil, Selman. (2019). "An automated new approach in fast text classification (fastText) a case study for Turkish text classification without pre-processing." In Proceedings of the 2019 3rd international conference on natural language processing and information retrieval, pp. 1–4.
- Latha, B., Nivedha, B., and Ranjana, Y. (2022). "Visual audio summarization based on NLP models," 2022 1st international conference on computational science and technology (ICCST), Chennai, India, pp. 63–66.
- Latief, A.D., Sampurno, T., and Arisha, A.O. (2023). Next sentence prediction: the impact of preprocessing techniques in deep learning. 2023 international conference on computer, control, informatics and its applications. (Bandung, Indonesia: IC3INA), 274–278.
- Malarselvi, G., and Pandian, A. (2023). Multi-layered network model for text summarization using feature representation. *Soft. Comput.* 27, 311–322. doi: 10.1007/s00500-022-07617-4
- Mohan, G. B., and Kumar, R. P. (2023). Lattice abstraction-based content summarization using baseline abstractive lexical chaining progress. *Int. J. Inf. Technol.* 15, 369–378. doi: 10.1007/s41870-022-01080-y
- Nadimi-Shahraki, M. H., Zamani, H., and Mirjalili, S. (2022). Enhanced whale optimization algorithm for medical feature selection: a COVID-19 case study. *Comput. Biol. Med.* 148:105858. doi: 10.1016/j.compbiomed.2022.105858
- Nallapati, R., Zhou, B., dos Santos, C., Gulcehre, C., and Xiang, B. (2016). "Abstractive text summarization using sequence-to-sequence RNNs and beyond," in Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn. (Berlin, Germany: Association for Computational Linguistics), 280–290.
- Pan, M., Li, T., Yang, C., Zhou, S., Feng, S., Fang, Y., et al. (2022). "A context-aware BERT retrieval framework utilizing abstractive summarization," 2022 IEEE/WIC/ACM international joint conference on web intelligence and intelligent agent technology (WI-IAT), Niagara Falls, ON, Canada, pp. 873–878.
- Pawar, S., Rao, M. G., and Chiplunkar, N. N. (2022). "A novel approach for multi-document summarization using Jaccard and cosine similarity," 2022 second international conference on advanced technologies in intelligent control, environment, computing & communication engineering (ICATIECE), Bangalore, India, pp. 1–6.
- Ramachandran, R., Jayachandran, S., and Das, V. (2022). "A novel method for text summarization and clustering of documents," 2022 IEEE 3rd global conference for advancement in technology (GCAT), Bangalore, India, pp. 1–6.
- Ranganathan, J., and Abuka, G. (2022). "Text summarization using transformer model," 2022 ninth international conference on social networks analysis, management and security (SNAMS), Milan, Italy, pp. 1–5.
- Riyahi, M., Rafsanjani, M. K., Gupta, B. B., and Alhalabi, W. (2022). Multiobjective whale optimization algorithm-based feature selection for intelligent systems. *Int. J. Intell. Syst.* 37, 9037–9054. doi: 10.1002/int.22979
- See, P. J., Liu, and Manning, C. D. (2017). "Get to the point: summarization with pointer-generator networks," In Proc. 55th Annu. Meeting Assoc. Comput. Linguistics, vol. 1, pp. 1073–1083.
- Shakil, M.H., and Alam, M.G.R. (2022) Hate speech classification implementing NLP and CNN with machine learning algorithm through interpretable explainable AI. In 2022 IEEE region 10 symposium. (Mumbai, India: TENSYPMP) 1–6.
- Shin, J., Park, S. B., and Song, H. J. (2023). Token-level fact correction in abstractive summarization. *IEEE Access* 11, 1934–1943. doi: 10.1109/ACCESS.2022.3233854
- Song, S., Huang, H., and Ruan, T. (2018). Abstractive text summarization using CNN-LSTM based deep learning. *Multimed. Tools Appl.* 78, 857–875. doi: 10.1007/s11042-018-5749-3
- Song, K., Lebanoff, L., Guo, Q., et al. (2020), "Joint parsing and generation for abstractive summarization," In Proceedings of the AAAI conference on artificial intelligence, New York, USA, vol. 34, pp. 8894–8901.
- Sun, L., Wang, X., Ding, W., Xu, J., and Meng, H. (2023). TSFNFS: two-stage-fuzzy-neighborhood feature selection with binary whale optimization algorithm. *Int. J. Mach. Learn. Cybern.* 14, 609–631. doi: 10.1007/s13042-022-01653-0
- Too, J., Mafarja, M., and Mirjalili, S. (2021). Spatial bound whale optimization algorithm: an efficient high-dimensional feature selection approach. *Neural Comput. Appl.* 33, 16229–16250. doi: 10.1007/s00521-021-06224-y
- Wang, W., Gao, Y., and Huang, H. Y. (2019). "Concept pointer network for abstractive summarization," in Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing, pp. 3076–3085, Hong Kong, China.
- Yadav, D., Katna, R., Yadav, A. K., and Morato, J. (2022). Feature based automatic text summarization methods: a comprehensive state-of-the-art survey. *IEEE Access* 10, 133981–134003. doi: 10.1109/ACCESS.2022.3231016
- Yang, P., Li, W., and Zhao, G. (2019). Language model-driven topic clustering and summarization for news articles. *IEEE Access* 7, 185506–185519. doi: 10.1109/ACCESS.2019.2960538
- Yu, Z., Wu, Z., Zheng, H., Xuan Yuan, Z., Fong, J., and Su, W. (2021), "LenAtten: an effective length controlling unit for text summarization," C. Zong, F. Xia, W. Li and R. Navigli, ACL/IJCNLP (Findings), Association for Computational Linguistics. (Stroudsburg, USA: ACL-IJCNLP), 363–370.
- Zhang, H., Cai, J., Xu, J., and Wang, J. (2019). "Pretraining-based natural language generation for text summarization," Proceedings of the 23rd conference on computational natural language learning (CoNLL). (Hong Kong, China: Association for Computational Linguistics), 789–797.