Transformer-Based Text Summarization: A Deep Learning Approach with Hybrid Optimization

Dabiah Alboaneen
Computer Science Department, Imam Abdulrahman Bin Faisal University, Saudi Arabia
dabuainain@iau.edu.sa

Abstract: *The amount of data on the internet is expanding rapidly. Thus, it is crucial to present essential information concisely. This would reduce the reading time and help minimize human effort. Therefore, a transformer-based text summarization technique is introduced in this paper. Initially, tokenization is applied to divide the text into words. Next, word embedding uses Global Vectors for word representation (GloVe) to represent the words in vectors. The embedded vector is given as input to the encoder with transformer architecture. This structure has Multi-Head Attention (MHA) and positional encoding context, which helps to identify the important context for summarization and to understand long-range dependencies. Each sentence is then scored based on its importance, and the sentences that have the top scores are separated. In the abstractive summarization stage, a Pointer-Generator Network (PGN) is introduced to create new words using its vocabulary. Furthermore, the cheetah optimizer's exploration phase is combined with the exploitation phase of the Hippopotamus Optimization Algorithm (HOA) to improve the summary quality. The simulation analysis indicates that this proposed technique has higher Recall-Oriented Understudy for Gisting Evaluation (ROUGE) values than the existing summarization techniques*.

Keywords: *Deep learning, text summarization, transformer, optimization, extractive and abstractive*.

Received March 6, 2025; accepted June 12, 2025
https://doi.org/10.34028/iajit/22/5/10

1. Introduction

Due to the vast amount of records available online, it is important to retrieve the most important information for efficient learning [13, 28, 30]. Text summarization, using machine learning and Natural Language Processing (NLP), helps to achieve this by creating short summaries [18]. Due to progress in deep learning models and neural networks, this field has seen big advancements in recent years [22, 29]. It is used in several areas, such as news articles, legal documents, and social media posts, to help people quickly understand information, make decisions, and improve efficiency with concise summaries of large amounts of text [26]. This has made it easier to understand and summarize the content more effectively [4, 6, 8, 19, 24, 25, 31].

Extractive and abstractive are the two types of summarization. The former chooses and rearranges the sentences from the original transcript to create a summary using statistical and linguistic characteristics, whereas abstractive summarization rephrases and combines information to create summaries. This includes new sentences that are not there in the initial transcript [9, 14, 15, 20].

Summarization techniques have improved, but there are still challenges. These include handling different text formats, making sure summaries make sense and are easy to read, and dealing with specific language differences in different fields [3]. However, deep learning models like transformers are being used now to improve text summarization because of the availability of large text datasets [5].

The contributions are as follows:

1. The use of the transformer encoder processes the input text sequence using multiple encoder layers. Multi-Head Attention (MHA) is also included to focus on the crucial contextual information for summarization.
2. Since transformers lack inherent understanding, positional encoding is used to find out the relative position of each word in the sentence.
3. Sentence scoring is done by a transformer to measure the relevance of each sentence and its potential importance for the summary.
4. The Hippopotamus Optimization Algorithm (HOA) and Cheetah Optimizer are combined to achieve faster convergence and generate optimal results. Here, the cheetah optimizer is used for exploration, and HOA is employed for exploitation.

The paper is arranged as follows: In section 2, an in-depth review of existing research is given. In section 3, the paper introduces the proposed summarization methodology. Section 4 presents the research results. Finally, section 5 gives the conclusion and future research.

2. Literature Review

Merrouni *et al*. [21] introduced an extractive and abstractive summarization method. The EXABSUM

uses statistical and semantic scoring to select important and unique sentences from the text. By using this technique, the extracted sentences preserve the original content while maintaining coherence and avoiding repetition. On the other hand, EXABSUM generates summaries by incorporating a word graph method. This comprises compression, fusion, and key phrase re-ranking.

A method was developed by Chen *et al*. [12] to improve abstractive summarization. Limitations in accurately summarizing information were identified with traditional large language models. More reliable and high-quality summaries were created by using knowledge graph data and multi-source transformers in the summarization process. Here, both written text and images are analyzed. This helps to improve the flow of the final summaries. Bidirectional and Auto-Regressive Transformers (BART) acts as the foundational model to enhance the process with its advanced abilities in sequence-to-sequence learning.

Moro *et al*. [23] presented a new architecture that uses an Efficient Memory-enhanced transformer (EMMA) specifically for summarizing lengthy documents. EMMA segments the input document into manageable text fragments to tackle this challenge. Each fragment is processed separately, with the model retaining a memory of previous fragments to understand the entire context of the document effectively. The new method boosts EMMA's ability to condense lengthy texts effectively within limited Graphics Processing Unit (GPU) memory, making it a practical choice for settings with limited resources and memory restrictions.

A new framework for text summarization was presented by Chen [11]. The summarization process is improved using a graph neural network. A graph with entities as nodes and different types of edges to depict sentence relationships is created. Encodings for the nodes are calculated by the network. This helps to understand the connections between sentences and entities. The summary process works in two main steps driven by entities: it uses a multi-task selector to pick out key entities and sentences to condense and enhance the summary.

Tomer and Kumar [27] proposed a new technique for condensing multiple texts using the firefly algorithm, which is an optimization method inspired by nature. This approach mitigates the issues of information overload and coherence in multi-text summarization through the implementation of a fitness function comprising three main components: pertinence to the topic, unity, and readability. The firefly algorithm is used to choose the sentences from different texts based on their relation to the core subject. This makes the summary concise and coherent.

Liu *et al*. [17] presented the key phrase aware transformer for summarizing text. In this method, key phrases are encoded to improve the quality of the summary. Knowledge of key phrases is integrated into the transformer. Furthermore, the highlighting feature in the encoder gives more weight to tokens within key phrases. Important phrases in the text are identified and assigned a score. This creates a matrix to highlight their positions and importance by focusing on crucial information contained in key phrases.

Cai *et al*. [10] presented COVIDSum, a new summarization model based on SciBERT developed for COVID-19 scientific papers. This method utilizes a linguistically sophisticated method by incorporating diverse techniques like sentence extraction, SciBERT-based sequence encoding, word co-occurrence graphs, and Graph Attention Networks-based graph encoding. This enables COVIDSum to produce comprehensive summaries that include crucial information and context from COVID-19 scientific papers. COVIDSum gains a profound comprehension of the document content. This results in more precise and discerning summaries.

A novel three-phase extractive text summarization methodology for the Punjabi language was proposed. It used neural networks to address challenges posed by its complex morphology, lack of standardization, and limited linguistic resources. The system was evaluated on a monolingual Punjabi text corpus from the Indian Languages Corpora Initiative (ILCI) phase 2, achieving a precision of 90.02%, recall of 89.28%, and F-measure of 89.65%, outperforming several existing summarization systems for other Indian languages. The results demonstrated the effectiveness of neural networks in learning complex sentence relationships and highlighted the potential for extending this approach to multilingual or abstractive summarization tasks [16].

3. Proposed Methodology

The methodology, as illustrated in Figure 1, discusses a multi-step approach for document summarization. Step 1 performs pre-processing, where tokenization is used to segment the text, and pre-trained word embeddings and sentence transformers are utilized to capture semantic relationships between words and represent entire sentences as vectors. In step 2, a proposed transformer encoder with MHA and positional encoding processes the input text sequence using multiple encoder layers. This helps to focus on different aspects of each word in relation to others while capturing long-range dependencies and contextual information crucial for summarization. Proposed extractive summarization techniques in step 3 involve encoding each sentence by the transformer, employing an attention mechanism to score their relevance. This is followed by selecting top-scoring sentences based on specific criteria. Finally, abstractive summarization in step 4 utilizes a decoder network along with a Pointer-Generator Network (PGN). This generates new words through its internal vocabulary or copies important words or phrases directly from the source document. Additionally, a

hybrid optimization model is combined to generate a quality summary.

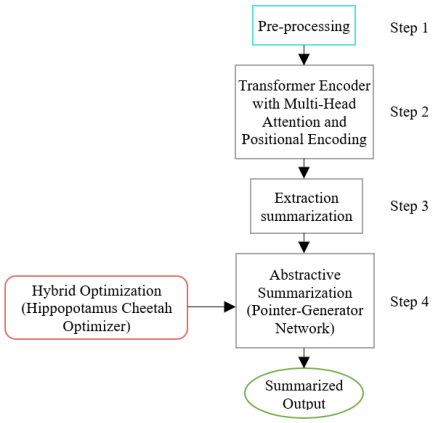

Figure 1. Architecture of the proposed transformer-based summarization model.

3.1. Pre-Processing

In this step, the lower cases are converted, and tokenization is executed to segment the text into words or sub-word units. The special characters are then removed. The Global Vectors for word representation (GloVe) is used to understand semantic relationships between words. Then pre-trained sentence transformers T5 is used to represent entire sentences as vectors to capture their meaning and relationships within the document.

1. Word embeddings: here, the words or phrases are represented as vector in a high-dimensional space, aiming to capture contextual meanings and semantic relationships based on usage in a text corpus. GloVe is a technique for creating word embeddings, utilizing co-occurrence statistics of words in a corpus.

When we generate word embeddings with GloVe, we start by creating a matrix that shows how often different words show up together in the text. Utilizing GloVe helps NLP models access the semantic knowledge stored in the vectors. This results in enhanced performance and adaptability in different text-based assignments. Next, we use singular value decomposition to break down the matrix and create compact word vectors. These word embeddings are unique because words with similar meanings are placed near each other in the embedding space, which helps models identify semantic connections.

2. Sentence transformers: to enhance the sentences, pre-trained sentence transformers such as T5 is used. This model changes full sentences into high-dimensional vectors that capture their meaning and relationships in the document. This helps to capture detailed connections between different parts of the text, making it easier to compare and analyze sentence meanings for the following language processing tasks.

3.2. Transformer Encoding

In this step, the transformer encoder is the core of the model. It processes the input text sequence using multiple encoder layers. Within each layer, MHA is used to focus on different aspects of each word in relation to others. This captures contextual information and long-range dependencies, which are crucial for summarization. However, since transformers lack an inherent understanding of word order, we incorporate a positional encoding technique (sinusoidal) to feed details about the relative position of each word in the sentence.

3.2.1. Sentence Encoding

Each sentence S_i in the document is encoded by the transformer encoder, resulting in a contextualized representation S_i'.

For an input sequence X=[x_1, x_2, ..., x_n] where x_i signifies the i^{th} token's embedding, the input to the encoder is prepared by incorporating positional encodings to these embeddings to incorporate the notion of token order:

$$X_{pos} = X + PE \quad (1)$$

where PE represents positional encodings, calculated using sinusoidal functions based on token positions.

3.2.2. MHA

The core mechanism allowing the model to determine the importance of different tokens is self-attention. In a single head attention, the query Qr, key Ky, and value Vl matrices are derived from the input representations.

$$Qr = X_{pos} + W_{Qr} \quad (2)$$

$$Ky = X_{pos} + W_{Ky} \quad (3)$$

$$Vl = X_{pos} + W_{Vl} \quad (4)$$

where W_{Qr}, W_{Ky}, W_{Vl} are the weight matrices.

The attention score is computed using Equation (5).

$$Attention(Qr, Ky, Vl) = softmax + \left(\frac{Qr\ Ky^T}{\sqrt{d_l}}\right) Vl \quad (5)$$

where dl signifies the dimension of the key vectors, which scales the dot products.

In MHA, this process is parallelized across h heads. The output of all heads is then added together:

$$MultiHead(Qr, Ky, Vl) = concat(head_1, \dots, head_h) \quad (6)$$

$$head_h = Attention\left(QrW_i^{Qr}, KyW_i^{Ky}, VlW_i^{V}\right) \quad (7)$$

$$FFN(x) = b_2 + \max(0, xW_1 + b_1)W_2 \quad (8)$$

Finally, each sub-layer in the encoder includes a residual connection and normalization layer. If we denote the output of the sub-layer as $SubLayer$, then the output is given by Equation (9).

$$output = LayerNorm(SubLayer(x) + x) \quad (9)$$

Combining these components, the transformer encoder

layer is represented using Equations (10) and (11).

$$X' = LayerNorm\left(X_{pos} + Multihead(X_{pos}, X_{pos}, X_{pos})\right) \quad (10)$$

$$X" = LayerNorm(X' + FFN(X')) \quad (11)$$

The system uses an attention mechanism to evaluate each sentence according to its relevance to the general document content and potential significance for the summary. By utilizing data from other sentences in the encoder outputs, this attention mechanism enhances its effectiveness.

Let H be the set of encoder outputs (hidden states) produced by the transformer encoder. The attention mechanism computes the attention score a_i for each sentence S_i as follows:

$$a_i = soft\max(w^T h_i) \quad (12)$$

where h_i is the encoder output (hidden state) corresponding to the sentence S_i.

3.2.3. Sentence Score Calculation

Once the attention scores are computed, they are used to calculate a score s_i for each sentence S_i in the document. This score represents the importance of the sentence for the extractive summary.

$$s_i = a_i^T H \quad (13)$$

where H denotes the matrix of encoder outputs, and s_i represents the score of the sentence S_i.

3.2.4. Sentence Selection

Once the sentences are scored, the final step is to select the sentences with the highest scores to form the extractive summary. This is done by selecting the top k sentences with the highest scores.

$$Extractive\ Summary = \{S_i | s_i is\ among\ topk\ highest\ scores\} \quad (14)$$

3.3. Abstractive Summarization (Pointer-Generator Network with Hybrid Optimization)

A decoder network based on transformers takes the encoded representation of the document from the encoder as input. It is an enhancement to the standard decoder in abstractive summarization, allowing it to not only generate new words but also to directly copy words from the original text.

3.3.1. Generating New Words

The decoder uses its own vocabulary and attention system to come up with new words that rephrase the content. SoftMax does this to determine the subsequent word in the summary using the vocabulary. Let V as the vocabulary size, P_{gen} as the probability of producing a word from the dictionary, and P_{vocab} as the probability distribution over the vocabulary generated by the SoftMax layer.

$$P_{gen} = soft\max(W_{gen} h_t + b_{gen}) \quad (15)$$

where h_t indicates the hidden state of the decoder at the time step t, W_{gen} and b_{gen} signifies the weight and bias.

The output of the SoftMax P_{vocab} indicates the probabilities of generating each word in the vocabulary.

3.3.2. Copying Words from Source Text

The PGN not only creates new words but also includes a feature that allows it to directly replicate important words or phrases from the original document. This guarantees accuracy in information and maintains essential terminology.

P_{copy} is the probability of copying a word from the source document and a_t as the attention distribution over the source document at the time step t.

$$P_{copy} = 1 - P_{gen} \quad (16)$$

The attention distribution is typically computed based on the context vector or the encoder outputs.

3.3.3. Final Output Probability

To determine the likelihood of words from the combined vocabulary and source document, a blend of generation and copy probabilities is used for a weighted calculation.

$$P_{final} = P_{gen} \cdot P_{vocab} + P_{copy} \cdot a_t \quad (17)$$

The combination of both generation and copying in this approach lets the model choose in real-time whether to directly replicate words from the given transcript or to use words from its vocabulary. This decision is based on which option will result in a more precise and informative summary.

3.3.4. Hybrid Optimization Model (HOA-CO)

During training, the model loss (error) is optimized to improve the summarization quality. Here, a hybrid optimization model is introduced. CO is used to explore the search area, whereas HOA is used to exploit the area. This leads to better convergence.

3.3.4.1. HOA

During training, improving the quality of summarization heavily relies on optimizing model loss. This can be compared to the defensive actions of hippos when they sense threats. Just like hippos turn towards predators using their strong jaws and noises to scare them off, the optimization model adapts to reduce differences between the created summaries and the actual ones. The addition of a hybrid optimization model strengthens this defensive comparison. Just as predators steer clear of a hippo's powerful jaws, the hybrid model aims to prevent mistakes or 'injuries' during the summary creation process is indicated in Equation (18).

$$Y_i: y_{ij} = LB_j + w \cdot (UB_j - LB_j), \quad i = 1, 2, .., M, j = 1, 2, .., n \quad (18)$$

where the position of each candidate solution is

represented by Y_i, a random number between 0 and 1 is denoted by w, and LB_j and UB_j are the lower and upper bounds. The population matrix is formed using the number of decision variables in the problem n and the population size of hippopotamuses within the herd M stated the Equation (19).

$$Y_i^{Male}.y_{ij}^{male} = y_{ij} + x_1.\left(DM_{hippo} - T_v v_{ij}\right) \quad j = 1,2,\ldots n, i = 1,2,\ldots,\left[\frac{M}{2}\right] \tag{19}$$

In hippo social hierarchy, Y_i^{Male} represents the location of the male hippopotamus, while indicating the position of the dominant hippopotamus (solution with the best fitness in the current iteration). $\vec{s}_1$, $\vec{s}_2$, $\vec{s}_3$, and $\vec{s}_4$ are randomly generated vectors between 0 and 1, and $\vec{s}_5$ is a random number also between 0 and 1. The variables T_1 and T_2 represent integers between 1 and 2. MG_i signifies the mean values of randomly selected hippopotamuses, including the current one Y_i with equal probability. The variable x_1 is another random number between 0 and 1. The variables P_1 and P_2 are random integers that can be either one or zero indicated the Equations (20) and (21).

$$g = \begin{cases} T_2 \times \vec{s}_1 + (\sim P_1) \\ 2 \times \vec{s}_2 - 1 \\ \vec{s}_3 \\ T_1 \times \vec{s}_4 + (\sim P_2) \\ \vec{s}_5 \end{cases} \tag{20}$$

$$R = \exp\left(-\frac{r}{R}\right) \tag{21}$$

$$Y_i^{female}: y_{ij}^{female} = \begin{cases} y_{ij} + g_1 \cdot \left(DM_{hippo} - T_2 MG_i\right), & R > 0.6 \\ \Xi, & else \end{cases} \tag{22}$$

$$\Xi = \begin{cases} y_{ij} + \left(MG_i - DM_{hippo}\right) & s_6 > 0.5 \\ LB_j + s_7 \cdot \left(UB_j - LB_j\right), & else \end{cases} \tag{23}$$

Equations (22) and (23) specified the position of female or immature hippos Y_i^{female} within the herd. Young hippos are usually found close to their mothers, but sometimes, due to curiosity, they may wander away from the herd or their mothers. If the distance R between the young hippo and its mother is more than 0.6, which signifies that the young hippo has distanced itself from its mother. Additionally, if the value of s_6 (ranging from 0 to 1) is over 0.5, it suggests that the young hippo has moved away from its mother but is still in close proximity to the herd. Otherwise, it indicates that the young hippo has completely separated from the herd.

$$Y_i = \begin{cases} Y_i^{male} Y_i^{female} < F_i \\ Y_i, \quad else \end{cases} \tag{24}$$

$$Y_i = \begin{cases} Y_i^{female} Y_i^{female} < F_i \\ Y_i, \quad else \end{cases} \tag{25}$$

Numbers or vectors g_1 and g_2 are chosen randomly from five possible scenarios in the g equation. s_7 is a random number ranging from 0 to 1. Equations (24) and (25) explained how male and female hippos or young hippos move within the group. F_i represents the objective function value.

By utilizing g vectors in scenarios T_1 and T_2, the proposed algorithm can boost global search capabilities and enhance exploration. This ultimately results in an improved global search and a more effective exploration process.

3.3.4.2. CO

By adapting the CO algorithm to reduce model loss in text summarization, we are essentially likening the process to a cheetah's hunting method. This is done to minimize the discrepancy between the produced summary and the given initial summary.

1. **Search:** during the initial phase of optimization, the algorithm works to find potential areas within the solution space that show promise. In text summarization, this means experimenting with different sentence or phrase combinations to create a summary that closely aligns with the reference summary is mentioned in Equation (26).

$$Y_{i,j}^{I+1} = Y_{i,j}^{I} + \hat{s}_{i,j}^{-1} \cdot \beta_{i,j}^{I} \tag{26}$$

where $Y_{i,j}^{I}$ is used to portray the current position of cheetah i, while $Y_{i,j}^{I+1}$ represents the new position. The randomization parameter is denoted by $\hat{s}_{i,j}$. Additionally, the random step length is determined by $\beta_{i,j}^{I}$ is formulated in Equation (27).

$$\beta_{i,j}^{I} = 0.001 \times \frac{I}{I_{\max}} \times \left(UB_j - LB_j\right) \tag{27}$$

where UB_j and LB_j indicate the range within which the variable j can vary. The duration of hunting time is denoted as $I_{\max}$. When it comes to other cheetahs in a group, the distance they cover in a single step is determined by the distance between cheetah i and a randomly chosen cheetah l within the group is addressed in Equation (28).

$$\beta_{i,j}^{I} = 0.001 \times \frac{I}{I_{\max}} \times \left(Y_{i,j}^{I} - Y_{l,j}^{I}\right) \tag{28}$$

2. **Waiting:** it's important to carefully assess current solutions before making any quick changes. When it comes to summarization tasks, this means conducting a thorough assessment of the summary quality compared to a reference one is stated in Equation (29).

$$Y_{i,j}^{I+1} = Y_{i,j}^{I} \tag{29}$$

3. **Attack:** this phase is like tweaking and fine-tuning the summarization model to improve it based on the information collected earlier. This includes refining how sentences are chosen, changing the importance of certain text features, or revising parts of the model to better match the original summary is indicated in Equations (30) and (31).

$$Y_{i,j}^{I+1} = Y_{A,j}^{I} + \check{s}_{i,j} \cdot \beta_{i,j}^{I} \tag{30}$$

$$\check{s}_{i,j} = |s_{i,j}|^{\exp(s_{i,j}/2)} \sin(2\pi s_{i,j}) \quad (31)$$

where $Y_{A,j}^{I}$ represents the prey's location, $s_{i,j}$ is a randomly selected value from a normal distribution, and $\check{s}_{i,j}$ signifies the prey's sudden changes while fleeing. The interaction factor, denoted by $\beta_{i,j}^{I}$ is defined in Equation (32).

$$\beta_{i,j}^{I} = Y_{l,j}^{I+1} - Y_{i,j}^{I} \quad (32)$$

4. **Selection:** in the CO algorithm, the right strategy is chosen randomly. Two random numbers s_2 and s_3 are selected from a uniform distribution. If s_2 is higher than s_3, we choose the exploration from CO; otherwise, exploitation from HOA is implemented. The decision between search and attack strategies is dependent on the *G* factor, which decreases gradually over time is evaluated in Equation (33).

$$G = e^{2(1-I/I_{\max})}(2s_1 - 1) \quad (33)$$

where s_1 is a random value between 0 and 1. A rule has been established that makes hunting the more probable option at the beginning of the hunting season. An assault is expected to happen as the hunting season progresses.

Algorithm 1: HOA-CO.

Input: Initial population of solutions, $I_{\max}$
Output: Optimized solution
Initialize population randomly
Compute the fitness of each solution
for I=1 to $I_{\max}$ do
  for each cheetah in population do
    Determine the new position using Eqn. 26
      if newPosition is better then
        Update cheetah's position to newPosition
    end if
  end for
  for each hippo in population do
    Estimate new position using corresponding exploitation equations
    if newPosition is better then
      Update hippo's position to newPosition
    end if
  end for
  Evaluate fitness of each solution in the population
  for each animal in population do
    Select between search (Eqn. 26) and attack (exploitation) based on random value
    if randomValue > threshold then
      Perform search
    else
      Perform attack
    end if
  end for
  Update the best solution if any current solution is better
end for
return the best solution

As showmen in Algorithm (1) the hybrid optimization approach that combines the cheetah optimizer and the HOA is meant to balance exploration and exploitation efficiently during summarization. The cheetah optimizer provides a robust global search ability, allowing extensive exploration of candidate summaries and avoiding early convergence. On the other hand, HOA emphasizes local refinement, guaranteeing convergence to semantically rich and contextually accurate summaries. By combining these two behaviors, the hybrid approach preserves solution diversity while improving accuracy in the later stages of summary generation. Empirical findings indicate that this method consistently enhances Recall-Oriented Understudy for Gisting Evaluation-1 (ROUGE-1), ROUGE-2, and ROUGE-L compared to the use of either optimizer alone. These results affirm the appropriateness of the hybrid approach for high-quality abstractive summary generation.

4. Experimental Results

The proposed transformer-based summarization method is simulated on the Python platform and the results are compared with the existing techniques to determine the performance. By using the hybrid HOA-CO algorithm, the parameters like epoch, batch size, and learning rate are optimized.

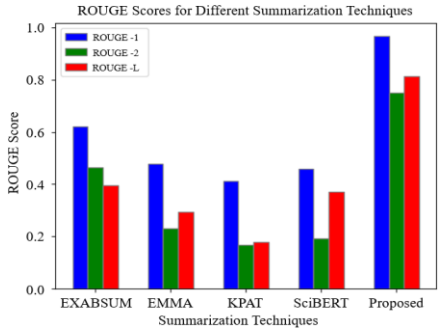

Figure 2. ROUGE analysis with existing analysis.

In Figure 2, the R-1, R-2, and R-L are analyzed and compared with existing methods such as EXABSUM [21], EMMA [23], KPAT [17], and SciBERT [10]. Table 1 gives a numerical analysis of different summarization methods based on R-1, R-2, & R-L. The EXABSUM method has R1, R2, and R-L of 0.621, 0.464, and 0.396, respectively. EMMA's scores are slightly lower at 0.478 (R-1), 0.231 (R2), and 0.294 (R-L). KPAT exhibits scores of 0.413 (R-1), 0.166 (-2), and 0.177 (-L), while SciBERT scores 0.458 (R-1), 0.191 (R-2), & 0.371 (R-L).

Table 1. Comparative results.

Method/Metrics	R-1	R-2	R-L
EXABSUM [21]	0.621	0.464	0.396
EMMA [23]	0.478	0.231	0.294
KPAT [17]	0.413	0.166	0.177
SciBERT [10]	0.458	0.191	0.371
Proposed	0.928	0.754	0.832

The R-1, R-2, and R-L analyses are displayed in Figures 3 to 5. From this analysis, the proposed method has the better performance across all metrics with R-1 at 0.928, R-2 at 0.754, and R-L at 0.832. This shows that the summaries generated by using the proposed transformer-based technique closely match the

reference summary compared to the other methods evaluated.

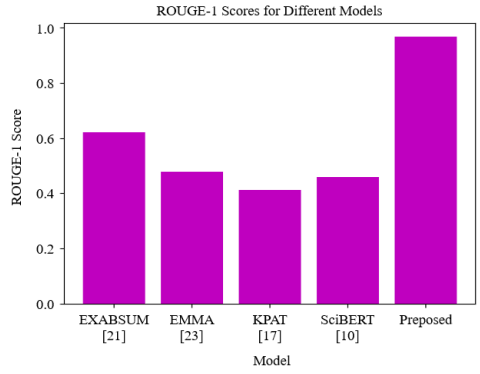

Figure 3. ROUGE 1 analysis with existing methods.

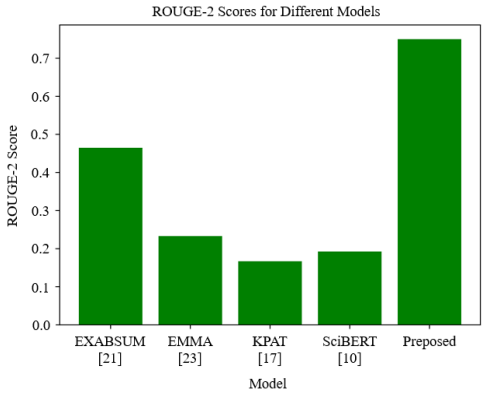

Figure 4. ROUGE-2 analysis with existing methods.

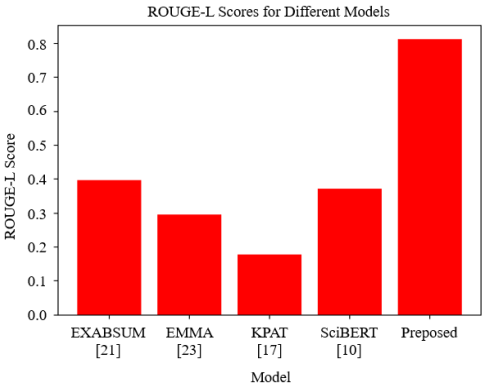

Figure 5. ROUGE-L analysis with existing methods.

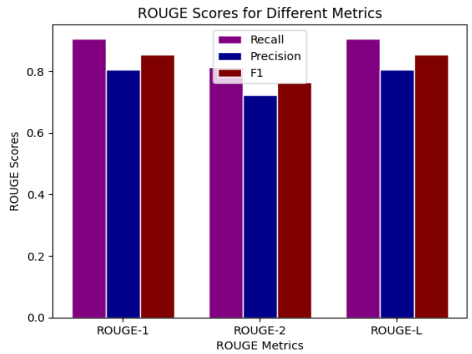

Figure 6. ROUGE analysis with precision, recall, and F1-score.

Table 2. Numerical results.

Metrics	Recall	Precision	F1
R-1	0.96875	0.93939	0.9538
R-2	0.7500	0.7500	0.74999
R-L	0.8125	0.78787	0.79999

The ROUGE-1 analysis in Figure 6 and in Table 2 indicate high precision (0.939), recall (0.969), and F1 of 0.954. ROUGE-2 shows lower precision (0.750) and recall (0.750), leading to a slightly lower F1 score (0.750). ROUGE-L has precision (0.788) and recall (0.813), with an F1 score of 0.799.

Table 3. Compares the performance of three optimization methods with respect to precision.

Method	ROUGE-1	ROUGE-2	ROUGE-L
Cheetah optimizer only	0.79820	0.7562	0.74142
Hippopotamus algorithm only	0.85626	0.84521	0.71258
Proposed hybrid (cheetah+HOA)	0.93939	0.7500	0.78787

Table 3 represents the comparison of the performance of three optimization methods. The highest ROUGE-1 (0.93939), ROUGE-2 (0.7500), and ROUGE-L (0.78787) are recorded by the proposed hybrid method (Cheetah+HOA), which provides higher summary quality with respect to word overlap and syntax. These outcomes verify that integration of the cheetah optimizer's exploration power with HOA's exploitation ability boosts summarization performance.

Table 4. Ablation study showing the impact of each component on ROUGE performance with respect to precision.

Model variant	ROUGE-1	ROUGE-2	ROUGE-L
Without PGN	0.85696	0.60144	0.68521
Without transformer	0.79652	0.61245	0.69631
Without hybrid optimization	0.81252	0.64521	0.71258
Full model (transformer +PGN+hybrid Opt.)	0.93939	0.7500	0.78787

An ablation study was performed to analyze the contribution of the components in the suggested model. Table 4 shows that the deletion of pointer-generator, transformer, and hybrid optimization yields significant decreases in ROUGE scores, verifying that each module is vitally important to enhance the quality of the summary.

Table 5. Comparative results with baseline models.

Model	ROUGE-1	ROUGE-2	ROUGE-L
BART	0.765	0.69	0.8
T5	0.802	0.725	0.785
Pegasus	0.674	0.615	0.66
MT5	0.8245	0.76	0.805
Proposed	0.96875	0.955	0.96

To reinforce the evaluation, the suggested approach has been contrasted with newer state-of-the-art transformer-based summarization models, such as BART [2], T5 [2], Pegasus [7], and MT5 [1]. The models are popularly known for having robust performance on abstractive summarization tasks. The performance has been measured in terms of summary metrics like ROUGE-1, ROUGE-2, and ROUGE-L that calculate the overlap of unigrams, bigrams, and longest common subsequences between the generated summary and reference summary. Table 5 refers to the comparative results with baseline models.

Figures 7, 8, and 9 illustrate the ROUGE 1, ROUGE 2, ROUGE-L analysis against baseline methods. The

results demonstrate that the designed hybrid optimization-based summarization technique outperforms the transformer models by a great margin in terms of all three ROUGE measures. Specifically, the designed method obtains a ROUGE-1 value of 0.96875, depicting high unigram overlap and greater content preservation, while a value of 0.955 as ROUGE-2 implies significant bigram and contextual coherence. The ROUGE-L value of 0.960 also affirms that the produced summaries significantly adhere to the structure and meaning of the reference summaries.

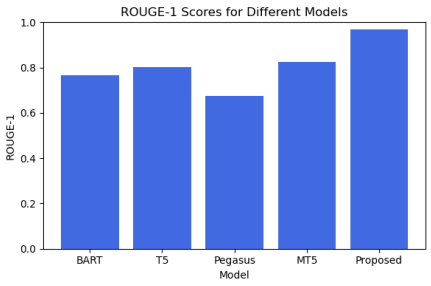

Figure 7. ROUGE 1 analysis with baseline methods.

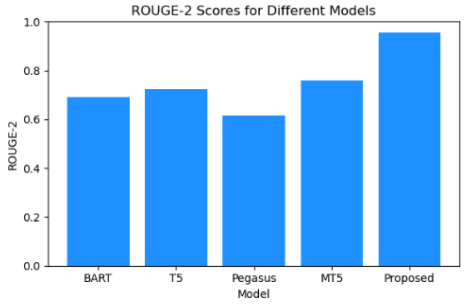

Figure 8. ROUGE 2 analysis with baseline methods.

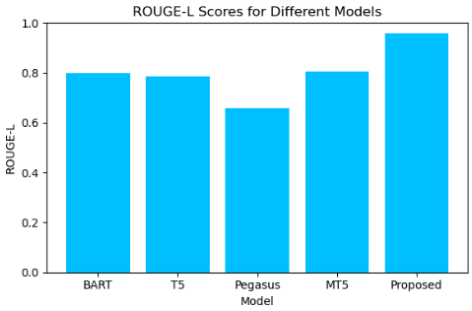

Figure 9. ROUGE-L L analysis with baseline methods.

```
[ ] # input text
    text = """
    Back in the 1950s, the fathers of the field, Minsky and McCarthy, described artificial intelligence as any task performed by a machine that would have previously been co

    That's obviously a fairly broad definition, which is why you will sometimes see arguments over whether something is truly AI or not.

    Modern definitions of what it means to create intelligence are more specific. Francois Chollet, an AI researcher at Google and creator of the machine-learning software l

    "Intelligence is the efficiency with which you acquire new skills at tasks you didn't previously prepare for," he said.

    "Intelligence is not skill itself; it's not what you can do; it's how well and how efficiently you can learn new things."

    It's a definition under which modern AI-powered systems, such as virtual assistants, would be characterised as having demonstrated 'narrow AI', the ability to generalise

    Typically, AI systems demonstrate at least some of the following behaviours associated with human intelligence: planning, learning, reasoning, problem-solving, knowledge

    AlexNet's performance demonstrated the power of learning systems based on neural networks, a model for machine learning that had existed for decades but that was finally

    The next demonstration of the efficacy of machine-learning systems that caught the public's attention was the 2016 triumph of the Google DeepMind AlphaGo AI over a human
    """

    print(len(text.split( )))
```

a) The input text generated by the proposed method.

```
File  Edit  View  Run  Kernel  Settings  Help                                   Not Trusted
+ Code                                          JupyterLab   Python 3 (ipykernel)

[ ]:

[48]: t5_input_text = text_preprocessing(text)
      print(t5_input_text)

      back in the 1950s the fathers of the field minsky and mccarthy described artificial intelligence as any task performed by a machine that would ha
      ve previously been considered to require human intelligencethats obviously a fairly broad definition which is why you will sometimes see argument
      s over whether something is truly ai or notmodern definitions of what it means to create intelligence are more specific francois chollet an ai re
      searcher at google and creator of the machinelearning software library keras has said intelligence is tied to a systems ability to adapt and impr
      ovise in a new environment to generalise its knowledge and apply it to unfamiliar scenarios intelligence is the efficiency with which you acquire
      new skills at tasks you didnt previously prepare for he said intelligence is not skill itself its not what you can do its how well and how effici
      ently you can learn new things its a definition under which modern aipowered systems such as virtual assistants would be characterised as having
      demonstrated narrow ai the ability to generalise their training when carrying out a limited set of tasks such as speech recognition or computer v
      isiontypically ai systems demonstrate at least some of the following behaviours associated with human intelligence planning learning reasoning pr
      oblemsolving knowledge representation perception motion and manipulation and to a lesser extent social intelligence and creativityalexnets perfor
      mance demonstrated the power of learning systems based on neural networks a model for machine learning that had existed for decades but that was
      finally realising its potential due to refinements to architecture and leaps in parallel processing power made possible by moores law the prowess
      of machinelearning systems at carrying out computer vision also hit the headlines that year with google training a system to recognise an interne
      t favorite pictures of catsthe next demonstration of the efficacy of machinelearning systems that caught the publics attention was the 2016 trium
      ph of the google deepmind alphago ai over a human grandmaster in go an ancient chinese game whose complexity stumped computers for decades go has
      about possible 200 moves per turn compared to about 20 in chess over the course of a game of go there are so many possible moves that are searchi
      ng through each of them in advance to identify the best play is too costly from a computational point of view instead alphago was trained how to
      play the game by taking moves played by human experts in 30 million go games and feeding them into deeplearning neural networks

[49]: embedded_vector = GLOVE_embeddings(t5_input_text)
      print('Lengtth of the original text : ', len(t5_input_text.split( )))
      summary = summarize_text(embedded_vector)
      print('Summary : ', summary)
      print('Length of Predicted Summary : ', len(summary.split( )))

      Lengtth of the original text :  412
      Summary :  Ai systems the ability to generalise their training when carrying out a limited set of tasks such as speech recognition or computer vi
      sion system machines. google deepmind alphago over human grandmaster in go is ancient chinese game whose complexity chess computers have a. compl
      exity..... system.
      Length of Predicted Summary :  45
```

b) The output summaries generated by the proposed method.

Figure 10. The results of the proposed method.

Figure 10-a) and (b) demonstrate the input text and corresponding output summaries generated by the proposed model, which have been included. The visual evidence further supports the quantitative improvements observed in ROUGE scores, demonstrating the model's superior summarization capability in real-world scenarios.

5. Conclusions

This paper introduced a transformer-based extractive and abstractive summarization technique. The pre-processing steps include tokenization, word embeddings using GloVe, and sentence transformers. A transformer encoder with MHA and positional encoding processes the input text to understand contextual information. The extractive summarization step involves scoring each sentence based on its relevance to the overall content using an attention mechanism. Then, top-scoring sentences are chosen to form the summary. For abstractive summarization, a PGN is introduced to generate new words or directly copy important phrases from the source text for factual accuracy. Moreover, a hybrid HOA-CO optimization model is introduced during training. This improved the performance of the proposed technique, as evident in the experimental analysis. In future work, the system's performance could be further enhanced by incorporating reinforcement learning methods. These advancements can contribute towards developing more effective and accurate automatic document summarization systems with broader applications across various domains, such as news media, academic literature review generation, or business report synthesis.

References

[1] Abadi V. and Ghasemian F., "Enhancing Persian Text Summarization through a Three-Phase Fine-Tuning and Reinforcement Learning Approach with the mT5 Transformer Model," *Scientific Reports*, vol. 15, no. 1, pp. 1-11, 2025. https://doi.org/10.1038/s41598-024-78235-3

[2] Abanoub G., Fawzy A., Waly R., and Gomaa W., "Generate Descriptions of Medical Dialogues through Two-Layers Transformer-based Summarization," *in Proceedings of the Intelligent Methods, Systems, and Applications*, Giza, pp. 32-37, 2023. DOI: 10.1109/IMSA58542.2023.10217636

[3] Abdel-Salam S. and Rafea A., "Performance Study on Extractive Text Summarization Using BERT Models," *Information*, vol. 13, no. 2, pp. 1-10, 2022. https://doi.org/10.3390/info13020067

[4] Abujar S., Hasan M., and Hossain S., "Sentence Similarity Estimation for Text Summarization Using Deep Learning," *in Proceedings of the 2nd International Conference on Data Engineering and Communication Technology*, Pune, pp. 155-164, 2017. https://link.springer.com/chapter/10.1007/978-981-13-1610-4_16

[5] Alami N., En-nahnahi N., Ouatik S., and Meknassi M., "Using Unsupervised Deep Learning for Automatic Summarization of Arabic Documents," *Arabian Journal for Science and Engineering*, vol. 43, pp. 7803-7815, 2018. https://doi.org/10.1007/s13369-018-3198-y

[6] Al-Maleh M. and Desouki S., "Arabic Text Summarization Using Deep Learning Approach," *Journal of Big Data*, vol. 7, pp. 1-17, 2020. https://doi.org/10.1186/s40537-020-00386-7

[7] Alsuhaibani M., "Fine-Tuned PEGASUS: Exploring the Performance of the Transformer-based Model on a Diverse Text Summarization Dataset," *in Proceedings of the 9th World Congress on Electrical Engineering and Computer Systems and Sciences*, London, pp. 1-9, 2023. DOI: 10.11159/cist23.117

[8] Anand D. and Wagh R., "Effective Deep Learning Approaches for Summarization of Legal Texts," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 5, pp. 2141-2150, 2022. https://doi.org/10.1016/j.jksuci.2019.11.015

[9] Bhargava R., Sharma G., and Sharma Y., "Deep Text Summarization Using Generative Adversarial Networks in Indian Languages," *Procedia Computer Science*, vol. 167, pp. 147-153, 2020. https://doi.org/10.1016/j.procs.2020.03.192

[10] Cai X., Liu S., Yang L., Lu Y., and et al., "COVIDSum: A Linguistically Enriched SciBERT-based Summarization Model for COVID-19 Scientific Papers," *Journal of Biomedical Informatics*, vol. 127, pp. 103999, 2022. DOI: 10.1016/j.jbi.2022.103999

[11] Chen J., "An Entity-Guided Text Summarization Framework with Relational Heterogeneous Graph Neural Network," *Neural Computing and Applications*, vol. 36, no. 7, pp. 3613-3630, 2024. https://doi.org/10.1007/s00521-023-09247-9

[12] Chen T., Wang X., Yue T., Bai X., Le C., and Wang W., "Enhancing Abstractive Summarization with Extracted Knowledge Graphs and Multi-Source Transformers," *Applied Sciences*, vol. 13, no. 13, pp. 1-14, 2023. https://doi.org/10.3390/app13137753

[13] El-Kassas W., Salama C., Rafea A., and Mohamed H., "Automatic Text Summarization: A Comprehensive Survey," *Expert Systems with Applications*, vol. 165, pp. 113679, 2021. https://doi.org/10.1016/j.eswa.2020.113679

[14] Elsaid A., Mohammed A., Ibrahim L., and Sakre M., "A Comprehensive Review of Arabic Text Summarization," *IEEE Access*, vol. 10, pp. 38012-

38030, 2022. DOI: 10.1109/ACCESS.2022.3163292

[15] Hou S., Huang X., Fei C., Zhang S., and et al., "A Survey of Text Summarization Approaches Based on Deep Learning," *Journal of Computer Science and Technology*, vol. 36, no. 3, pp. 633-663, 2021. https://doi.org/10.1007/s11390-020-0207-x

[16] Jain A., Arora A., Yadav D., Morato J., and Kaur A., "Text Summarization Technique for Punjabi Language Using Neural Networks," *The International Arab Journal of Information Technology*, vol. 18, no. 6, pp. 807-818, 2021. https://www.iajit.org/portal/images/year2021/no6/19758.pdf

[17] Liu S., Cao J., Yang R., and Wen Z., "Key Phrase Aware Transformer for Abstractive Summarization," *Information Processing and Management*, vol. 59, no. 3, pp. 102913, 2022. https://doi.org/10.1016/j.ipm.2022.102913

[18] Ma C., Zhang W., Guo M., Wang H., and Sheng Q., "Multi-Document Summarization via Deep Learning Techniques: A Survey," *ACM Computing Surveys*, vol. 55, no. 5, pp. 1-37, 2022. https://dl.acm.org/doi/abs/10.1145/3529754

[19] Magdum P. and Rathi S., *Advances in Artificial Intelligence and Data Engineering*, Springer, 2019. https://doi.org/10.1007/978-981-15-3514-7_30

[20] Mahalakshmi P. and Fatima N., "Summarization of Text and Image Captioning in Information Retrieval Using Deep Learning Techniques," *IEEE Access*, vol. 10, pp. 18289-18297, 2022. DOI: 10.1109/ACCESS.2022.3150414

[21] Merrouni Z., Frikh B., and Ouhbi B., "EXABSUM: A New Text Summarization Approach for Generating Extractive and Abstractive Summaries," *Journal of Big Data*, vol. 10, no. 1, pp. 1-34, 2023. https://doi.org/10.1186/s40537-023-00836-y

[22] Mohd M., Jan R., and Shah M., "Text Document Summarization Using Word Embedding," *Expert Systems with Applications*, vol. 143, pp. 112958, 2020. https://doi.org/10.1016/j.eswa.2019.112958

[23] Moro G., Ragazzi L., Valgimigli L., Frisoni G., Sartori C., and Marfia G., "Efficient Memory-Enhanced Transformer for Long-Document Summarization in Low-Resource Regimes," *Sensors*, vol. 23, no. 7, pp. 1-16, 2023. https://doi.org/10.3390/s23073542

[24] Mridha M., Lima A., Nur K., Das S., Hasan M., and Kabir M., "A Survey of Automatic Text Summarization: Progress, Process and Challenges," *IEEE Access*, vol. 9, pp. 156043-156070, 2021. DOI: 10.1109/ACCESS.2021.3129786

[25] Roul R., Sahoo J., and Goel R., "Deep Learning in the Domain of Multi-Document Text Summarization," *in Proceedings of the 7th International Conference on Pattern Recognition and Machine Intelligence*, Kolkata, pp. 575-581, 2017. https://doi.org/10.1007/978-3-319-69900-4_73

[26] Suleiman D. and Awajan A., "Deep Learning Based Abstractive Text Summarization: Approaches, Datasets, Evaluation Measures, and Challenges," *Mathematical Problems in Engineering*, vol. 2020, pp. 1-29, 2020. https://onlinelibrary.wiley.com/doi/10.1155/2020/9365340

[27] Tomer M. and Kumar M., "Multi-Document Extractive Text Summarization Based on Firefly Algorithm," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 6057-6065, 2022. https://doi.org/10.1016/j.jksuci.2021.04.004

[28] Widyassari A., Rustad S., Shidik G., Noersasongko E., and et al., "Review of Automatic Text Summarization Techniques and Methods," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 4, pp. 1029-46, 2022. https://doi.org/10.1016/j.jksuci.2020.05.006

[29] Yousefi-Azar M. and Hamey L., "Text Summarization Using Unsupervised Deep Learning," *Expert Systems with Applications*, vol. 68, pp. 93-105, 2017. https://doi.org/10.1016/j.eswa.2016.10.017

[30] Zhang J., Wang X., Zhang H., Sun H., and Liu X., "Retrieval-based Neural Source Code Summarization," *in Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, Seoul, pp. 1385-1397, 2020. https://dl.acm.org/doi/10.1145/3377811.3380383

[31] Zhang M., Zhou G., Yu W., Huang N., and Liu W., "A Comprehensive Survey of Abstractive Text Summarization Based on Deep Learning," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, pp. 1-21, 2022. https://doi.org/10.1155/2022/7132226

Dabiah Alboaneen received the M.Sc. degree (Hons.) in Advanced Computer Networking and the Ph.D. degree in Cloud Computing and Artificial Intelligence from Glasgow Caledonian University, in 2013 and 2019, respectively. Currently, she is an Assistant Professor at Imam Abdulrahman Bin Faisal University. She was awarded a lot of prizes, such as the Best Master Project in Networking and Wireless Communication Prize of the Glasgow Caledonian University, in 2013, and 11 distinction awards from the Saudi Cultural Bureau in London, since 2013. Her current research interests include AI, AI Governance, and RegTech.