I.J. Modern Education and Computer Science, **2023, 6, 56-72**
Published Online on December 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2023.06.05

Tangent Search Long Short Term Memory with Aadaptive Reinforcement Transient Learning based Extractive and Abstractive Document Summarization

Reshmi P Rajan*
Research Scholar, Department of Computer Science, Christ (Deemed to be University), S.G. Palya, Bengaluru, Karnataka 560029, India
E-mail: reshmirajan704@gmail.com
ORCID iD: https://orcid.org/ 0000-0003-3059-4199
*Corresponding Author

Deepa V. Jose
Associate Professor, Department of Computer Science, Christ (Deemed to be University), Bangalore, India
E-mail: deepavjose@yahoo.com
ORCID iD: https://orcid.org/ 0000-0001-5022-4924

Roopashree Gurumoorthy
Project Manager, NTT DATA Services, Winnipeg, Canada
Email: roopashree.gurumurthy@nttdata.com
ORCID iD: https://orcid.org/0009-0002-6588-8082

Received: 14 July, 2023; Revised: 13 August, 2023; Accepted: 25 September, 2023; Published: 08 December, 2023

Abstract: Text summarization is the process of creating a shorter version of a longer text document while retaining its most important information. There have been a number of methods proposed for text summarization, but the existing method does not provide better results and has a problem with sequence classification. To overcome these limitations, a tangent search long short term memory with adaptive reinforcement transient learning-based extractive and abstractive document summarization is proposed in this manuscript. In abstractive phase, the features of the extractive summary are extracted and then the optimal features are selected by Adaptive Flamingo Optimization (AFO). With these optimal features, the abstractive summary is generated. The proposed method is implemented in python. For extractive text summarization, the proposed method attains 42.11% ROUGE-1 Score, 23.55% ROUGE-2 score and 41.05% ROUGE-L score using Gigaword. Additionally, 57.13% ROUGE-1 Score, 28.35% ROUGE-2 score and 52.85% ROUGE-L score using DUC-2004 dataset. For abstractive text summarization the proposed method attains 47.05% ROUGE-1 Score, 22.02% ROUGE-2 score and 48.96% ROUGE-L score using Gigaword. Also, 35.13% ROUGE-1 Score, 20.35% ROUGE-2 score and 35.25% ROUGE-L score using DUC-2004 dataset.

Index Terms: Summarization, Knowledge Graph, Feature Selection, Optimization Technique, Improved Reinforcement Learning, ROUGE Scores.

1. Introduction

Text Summarization (TS) is the process of making brief summary. There are numerous techniques suggested to extract the information from the document overall, they are classified as extractive and abstractive [1,2] . In the extraction, the key objects and sentences are extracted without modifying themselves. After understanding the language, the abstraction includes rephrasing the context aware sentences [3,4]. Summarization of text helps to develop a brief version of the actual data. Single and multi-Document Summarization (DS) are two types of DS [5,6]. The abstractive techniques can able to properly rewrite the core ideas of the source document, than extractive summarization. The Abstractive Text Summarization (ATS) plays a major part in natural language processing. Due to the rapid growth of

Deep Learning (DL), many models are proposed to change an input into brief output sequence [7,8]. They have been successful in many tasks or applications like, video captioning, speech recognition and machine translation. Unlike the above tasks, in TS, the output sequence is much shorter than the input sequence [9,10].

The Extractive Text Summarization (ETS) summarize the documents without rephrasing it. It extracts few sentences from the document and provides a meaningful summary as output [11,12]. The extractive summaries can't capture important information spread over the files. Dangling anaphora is a problem faced by the extractive summaries, i.e., the pronouns present in the text lose their meaning when extracted [13,14]. In the extraction-related summarization field, a large number of works has been done, but in abstractive-related summarization, it is very difficult because, the computer can choose only the most valuable data from the text; due to this problem, it is very crucial to combine valuable information and to make compressed content of the text[15,16]. At first, the important information must be identified, and then summarization can be performed to summarise all the documents in a single paragraph [17,18]. The difficulty of generating abstractive summaries is shown in recent papers [19,20]. The approach [28] failed to identify correct summaries of lengthy document of books or chapters. The approach [33] chooses the significant sentences from the summary without rephrasing it, because the rephrasing makes the sentences clearer and much easier to comprehend. To overcome these issues, the proposed method uses TS-LSTM for ETS and ARTL for ATS. The proposed method helps to summarize long documents without losing its quality. The ARTL based abstractive summarization rephrases the sentences without changing its content. The major research objectives of this research work are:

- To find correct summaries of lengthy document of books or chapters, without losing its quality.
- To rephrase the final summary, which makes easier to comprehend the sentences.
- To compare the experimental outcomes of proposed method with various extractive and ATS techniques for demonstrating an efficiency of proposed method.

Remaining manuscript is arranged as: section 2 delineates the comprehensive review. The proposed technique is explained in section 3 followed by the explanation of outcomes and discussions in section 4. This manuscript concludes in section 5.

2. Literature Survey

Several studies were formerly suggested in the literature related to TS. Among these, a few recent studies are expressed below,

In 2021, Shi T et al. [21] had suggested a neural ATS with sequence-to-sequence models. In this approach an open source library was developed, called neural ATS. This approach mainly focused on the challenges that related with the architecture of neural network, summary generation procedures and different algorithmic solutions. The evaluations were done on the CNN/daily mail database. But most of the seq2seq methods depend on beam search algorithm for summary generation. In 2021, Rani R and Lobiyal DK [22] had developed an ETS approach using tagged-LDA based topic modeling. The findings show that this strategy creates concise, well-written, and cohesive summaries. But need to develop some entities in this approach for extracting more informative topics.

In 2022, Mishra SK et al. [23] had introduced a scientific DS in multi-objective clustering framework. The sentences were clustered by the concept of multi-objective clustering. Need to explore some features like marginal relevance to extract the sentence from the cluster. In 2018, S. Song et al. [24] had suggested an ATS using LSTM-CNN based DL. This method consists two phases, initially, from the source sentence, the phrases were extracted and then generates text summaries with the help of DL. This approach has high accuracy but the text pre-processing stage consumes more time, so it is essential to develop a more convenient text preprocessing tool. In 2022, Wazery YM et al [25] has suggested an abstractive Arabic TS based on DL. The goal of this approach was to create a seq-to-seq model with the help of some deep artificial neural networks to attain greatest performance. This approach works through two components which is decoder and encoder. But this approach looking forward for applying Reinforcement Learning (RL) algorithms, and to combine RL methods with DL models for enhancing the quality of the GS. In 2020, Yuan C et al. [26] had suggested an integrating Word Attention by Convolutional Neural Networks (WACNNs) for ATS. The effectiveness of this method is evaluated on the Gigaword, Chinese summarization and DUC-2004 datasets. The combined multilayer convolutional neural network and word attention offers a better-learned depiction of the input. But need to examine the effect of the hierarchical representations of the input documents.

In 2019, Wenbo W et al. [27] have developed a concept pointer network for ATS. This approach provides a concept pointer model (concept pointer with Distant Supervision strategy (concept pointer+DS) and concept pointer with Reinforcement Learning (RL) to generate concept-oriented summaries and to enhance the ATS. But the overarching tendency of the model is still to copy segments of the source text and rearrange the phrases into a summary. In 2020, Gao Y et al. [28] had suggested a neural ATS fusing by global generative topics. The experiments of this approach were conducted on Gigaword and DUC-2004 datasets. Global generative topic fusion with Local sentence generation Encoding for Abstractive text summarization (GLEAM) has some corresponding aids. But need to identify correct summaries of lengthy document of books or chapters. In 2022, Mahalakshmi P et al. [29] had suggested a TS and image captioning in data retrieval using DL methods. This approach was validated with the help of DUC and

Gigaword. The experimental outcomes of DBN was compared with several existing techniques like MAPCoL. Need to improve the performance of Deep Belief Network (DBN) with the use of hyperparameter tuning strategies.

In 2018, Yao K et al. [30] had suggested a deep RL for ETS. This approach depends on Deep Q-Network (DQN). The abstract attributes like, salience, data content and idleness of the sentences was taken in the Q-value approximation. This approach was evaluated on DUC-2002, CNN/Daily corpus and DUC-2004 datasets. But need to discover a word-level based ETS via deep RL. In 2020, Jiang Z et al. [31] had suggested a n-grams and combining word embeddings for unsupervised DS. For sentence compression a transformer-based model was suggested to aid in DS. This method can outperform the TF-IDF based method. In 2023, Sakhare DY [32] had suggested a sequence-to-TS with the help of LSTM based neural approach. The non-differential valuation metrics were employed in this technique and it stores significant attributes for active TS. Need to propose many patterns of linguistic TS. In 2020, Tomer M and Kumar M [33] has suggested an improved TS with ensembled approach based on Fuzzy with LSTM (FLSTM). The experiment of this approach was performed on DUC and CNN/daily mail datasets. This approach utilized interference and fuzzy measures for textual information extraction. The restriction of this approach is that, it chooses the significant sentences from the summary without rephrasing it.

In 2020, Hark C and Karcı A [34] had suggested a Karcı summarization an effective and simple method for automatic TS with Karcı entropy. This approach was evaluated on DUC-2002, DUC-2004 and document understanding conference datasets. This approach selects the generic, effective and informational sentences within a unit of text or paragraph. Further efforts are needed to improve the ROUGE scores of this approach.

2.1. Background

Reducing number of words and sentences of a document without changing its meaning is known as TS. The ETS involves picking up the most important lines and phrases from the ATS is the purpose of generating concise summary which contains the document's prominent ideas.

2.1.1. Problem formulation

There are several methods suggested for TS. But the ROUGE scores of existing methods are low and it is difficult to summarize long documents with high quality. The existing methods in the above section 2 has several drawbacks in TS such as, choosing the significant sentences from the summary without rephrasing it, failed to find correct summaries of lengthy document of books or chapters, etc. To overcome these limitations this work is motivated.

3. Proposed Methodology

In, this section, both ETS and ATS is proposed. In the extractive phase, the input text is preprocessed, it contains of four steps such as tokenization, segmentation, stemming and stop words removal and given to TS-LSTM for generating extractive summary. In the abstractive, the important features are withdrawn from the extractive summary. Then the optimal features are selected with the help of AFO. With these optimal features, the abstractive summary is generated using ARTL. Fig. 1 displays the workflow of proposed scheme.

3.1. Input Acquisition

For TS, the input data is taken from Gigaword and DUC-2004 datasets [35,36].

3.2. Pre-processing of Text

Pre-processing of text is the primary phase in text summarizing, which includes identifying the sentences that make up a section (segmentation); tokenizing the streaming text; removing unnecessary information and stop-words such as essays, pronouns, and adverbs; and identifying the material that contributes meaning. In this pre-processing section, the literature survey part is dropped (i.e. initially, the articles are read). The read articles are written in a text file (csv) with a corresponding section name.

- **Tokenization & segmentation:** Various operations are included in the pre-processing modules. It begins with discovering various component sentences and segmenting them. Furthermore, tokens are separated from the continuous set of characters. (tokenization).
- **Stemming and stop words removal:** Stop words and other unnecessary words are eliminated after tokenization. These popular words can be disregarded namely, articles, prepositions, adverbs, and pronouns (have no meaning). Stop word removal refers to removing certain words from these texts. To reduce words to their word stem, linguistic normalization is carried out in the end (stemming) [37].

3.3. Extractive Text Summarization using TS-LSTM

The preprocessed text is given to TS-LSTM for ETS. The extractive process entails choosing the document's key words and passages. The summary is then produced by merging all the important lines. An artificial neural network called LSTM is employed in DL and artificial intelligence. LSTM is influenced by feedback. It is employed in the processing, prediction, and classification of time-series data.

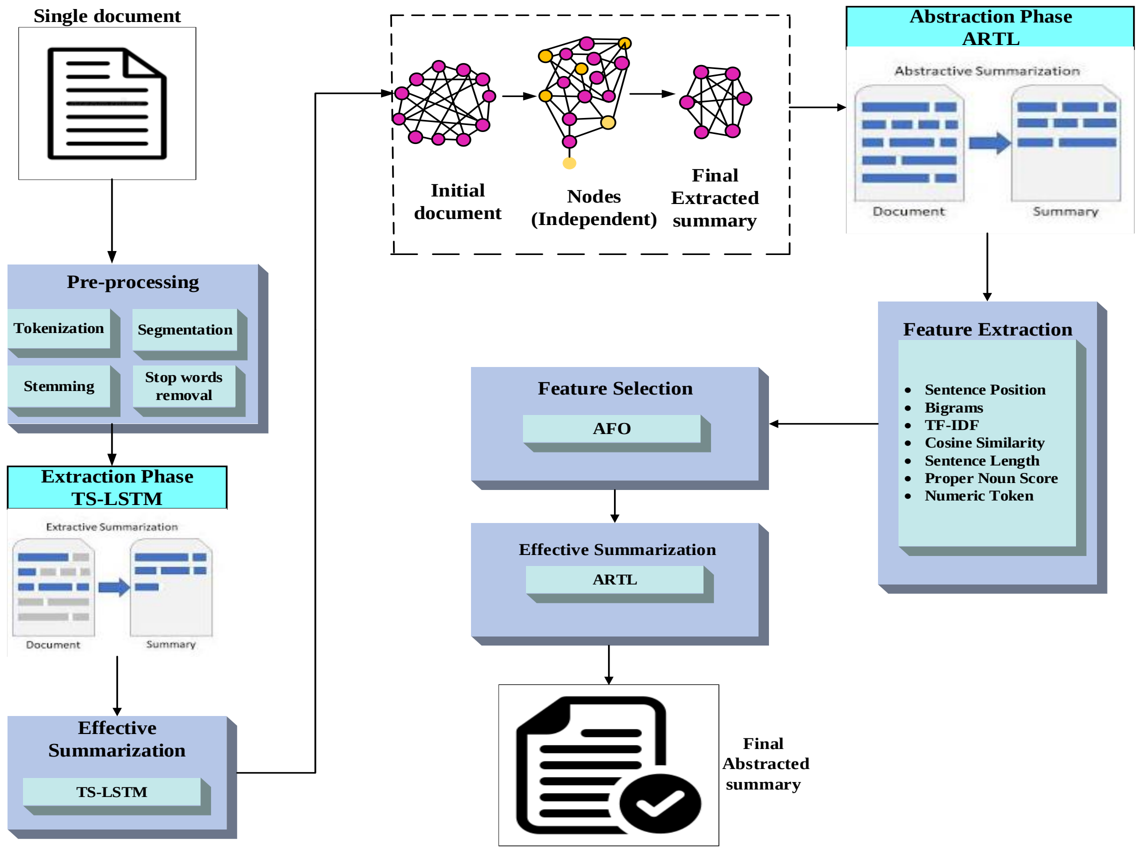


Fig. 1. Block diagram of proposed extractive and abstractive text summarization

LSTM: The basic units of LSTM are cell, input, output and forget gate. Equations (1)-(6) display the weight matrix and transfer vector formulas.

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i) \tag{1}$$

$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f) \tag{2}$$

$$\tilde{c}_t = \tanh(W_c h_{t-1} + U_c x_t + b_c) \tag{3}$$

$$Z_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \tag{4}$$

$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o) \tag{5}$$

$$h_t = o_t \otimes \tanh(z_t) \tag{6}$$

where σ is the sigmoid role and $\otimes$ is element- wise product, the input at time t is x_t, f_t is the forget gate, H_t is the hidden state vector storing sequence information, bias vectors are represented as b_i, b_f, b_c, b_o and the matrices of weights for hidden state h_t are W_i, W_f, W_c, W_o are matrices of weights for hidden state. The weight vector W_i is optimized with the help of TS for a better extracted summary [38].

3.4. Tangent Search (TS) Algorithm

The TS depends on Tangent Function (TF) which is a mathematical function. This function provides an excellent capability for thoroughly exploring the search space, and its disparity between $-\infty$ and $+\infty$. The periodicity aid in maintaining good stability between exploitation and exploration. In the TS method, every motion equation is controlled by an overall step of the form "$step * \tan(\theta)$", where the levy flight function is replaced with the tangent flight function. The flowchart of TS is given in Fig. 2.

Step 1: Initialization

The TSA starts by random initial population generation. The following equation (7) calculates the initial solution equally dispersed over the Search Space (SS).

$$Y = L_B + (U_B - L_B)^* Rand(D) \tag{7}$$

where, L_B signifies the Lower Bound, U_B signifies the Upper Bound, $Rand$ is a function to create evenly distributed arbitrary numbers (i.e. between [0, 1]) and D is the problem's dimension.

Step 2: Random generation

After initialization, the input constraints of TS are generated randomly.

Step 3: Fitness Function (FF)

FF is used to derive the objective function.

$$FitnessFunction = [\min(W_i)] \tag{8}$$

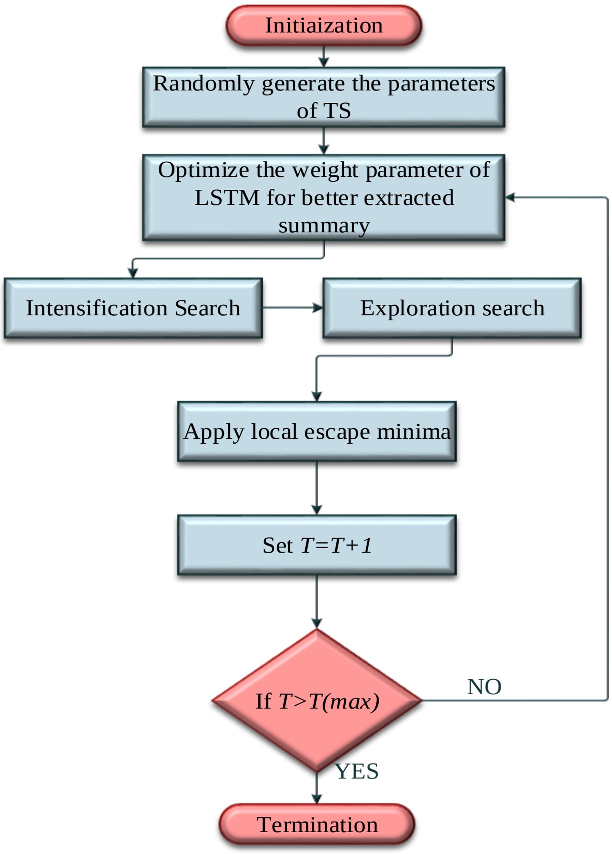


Fig. 2. Flowchart of tangent search algorithm

Step 4: Search for intensification

In the Intensification Search (IS), the TS initially makes a random local walk assisted by (9) and then few variables (i)of Obtained Solution (OS) is substituted by the values of i in the ideal Current Solution (CS) with the help of (10).

$$Y_i^{t+1} = Y_i^t + step * \tan(\theta) * (Y_i^t - optS_i^t) \tag{9}$$

$$Y_i^{t+1} = optS_i^t \text{ if variable } i \text{ is selected} \tag{10}$$

Each OS Y is repaired by (11) and (12), if its values overflow, U_B and L_B (problem bounds).

$$Y(Y < L_B) = Rand * (U_B - L_B) + L_B \tag{11}$$

$$Y(Y > U_B) = Rand * (U_B - L_B) + L_B \tag{12}$$

Step 5: Exploration search (ES)

The TS creates a global accidental walk with a creation of flexible step size and tangent flight. Tangent function facilitates effective SS exploration. The TF aids to explore the SS, indeed θ close to $pi/2$ make the parameter of tangent bigger and the OS would be near to the existing solution.

$$Y_i^{t+1} = Y_i^t + step * \tan(\theta) \tag{13}$$

The ES equation (13) is applied on i with a probability equal to $1/D$, where the dimension of the problem is D.

Step 6: Escape local minima approach

TSA combines a strategy to deal with the local minima stagnation problem by a precise procedure. This procedure has two sections with some probability. Additionally, a new random solution with a probability of 0.01 can replace the worst solution. And it is given in (14)-(15).

$$Y = Y + R * (optS - random * (optS - Y)) \tag{14}$$

$$Y = Y + \tan(\theta) * (U_B - L_B) \tag{15}$$

Step 7: Explanation about parameters

TSA utilizes two variants of the step-size, in IS the first step-size variant is used and it is calculated with the help of (16) and (17),

$$step\ 1 = 10 * sign(random - 0.5) * Norm(opts) * \log(1 + 10 * \dim/t) \tag{16}$$

$$step\ 2 = 1 * sign(random - 0.5) * Norm(opts - X) / \log(20 + t) \tag{17}$$

where $Norm$ is a mathematical form, TS can also use other norms; the present solution is Y, and which guides the exploration procedure to the best outcome, the best CS $optS$ is utilized to guide the search process towards the ideal solution. The exploration and exploitation search direction are controlled by the component sign (-, +). With the help of (16) and (17) the weight parameter of LSTM W_i is optimized

In the convergence of TSA, the angle θ plays an important role.After an experimental study, θ is selected randomly in the range of $[0, pi/2.1]$ in the IS and in the range of $[0, pi/3]$ in the ES. As a result, tangent function value range is 0 to 13.34 $(\tan(pi/2.1))$. The weight vector W_i is optimized with the help of TS for a better extracted summary.

Step 8: Termination

Here the TS is used to improve the weight parameter of LSTM for an effective extracted summary. In this proposed method, the DS for both extractive and abstractive phase established accurately. The weight parameter of LSTM is optimized. Thus, the rate of recall, F1-score, precision and ROUGE scores for both extractive and abstractive phases are enhanced. Until met the termination condition, the algorithm iterates the phase 7 to 3 till the criteria time iteration $T = T + 1$ is reached. After optimizing the weight parameter of LSTM, an extracted summary is generated. Then the extracted summary is given to abstraction phase as input [39].

3.5. Abstractive Summarization using ARTL

The features such as numeric token, trigrams, bigrams, sentence position, TF-IDF, proper noun score, thematic score, sentence length, and cosine similarity of the extractive summary is extracted.

3.5.1. Feature Extraction

Extract the most significant features in the dataset by feature extraction.

- *Sentence position*

In sentence position the sentence appeared in the end or beginning of input sentence is considered to be most significant. Equation (18) is utilized to calculate the sentence position.

$$Sentence_{position} = \frac{Sentence_{total\ number} - Sentence_{index}}{Sentence_{total\ number}} \tag{18}$$

- *Bigrams*

Bigram is the order of two adjacent words exist in the input. To get the count of two adjoining words in the document bigram is used.

- *Trigrams*

Trigram is any order of three adjacent words exist in the input. The trigrams help to obtain the value of three adjoining word sets.

- *TF-IDF*

In topic description TF is used for estimating the sentence's relative frequency. IDF is a factor that helps to minimize the weight of the frequent words in the document. $TF-IDF$ is calculated with the help of (19)-(21).

$$TF(t) = \frac{no\, of\ times\ term\ t\ appeared}{Entire\ no\ of\ terms} \tag{19}$$

$$IDF(t) = \log \frac{Total\, number\ of\ documents}{Number\ of\ documents\ containing\ the\ term\ t} \tag{20}$$

$$TF - IDF = TF(t) * IDF(t) \tag{21}$$

- *Cosine similarity*

This function helps to identify the similarity between various sentences with centroid.

- *Thematic score*

The document keywords are thematic words. From the document these thematic words are withdrawn with the help of RAKE (Rapid Automatic Keyword Extraction)

- *Sentence length*

It is utilized to calculate the sentence length. It is calculated with the help of (22).

$$Sentence_{length}(L) = \frac{Sentence_{length}}{Longest\ Sentence_{length}} \tag{22}$$

- *Proper Noun (PN) score*

The sentence containing PN is considered to carry most important information. Equation (23) helps to calculate the proper noun score.

$$\Pr oper\ Noun Score(N) = \frac{Total\, number\ of\ proper\ nouns}{Total\, number\ of\ words} \tag{23}$$

- *Numeric token*

It is estimated by dividing the entire number of arithmetical information in the sentence by the input sentence. The extracted features are then given to AFO.

3.5.2. Adaptive Flamingo Search Algorithm for Feature Selection

A revolutionary swarm intelligence optimization system called AFO was motivated by flamingos' migratory and searching behaviour.

Step 1: Initialization

Initially the parameters of AFO are generated.

Step 2: Random generation

The initialized parameters of AFO are generated randomly.

Step 3: FF

In this, the FF of the AFO is utilized for selecting the optimal features. The FF is calculated with the help of (24).

$$FF(X) = \text{w1Rouge-1(X) + w2Rouge-2(X) + w3Rouge-L(X) + w4Coverage(X) + w5*Diversity(X)} \tag{24}$$

where, X signifies the binary vector, the elements in X signifies whether the optimal features are selected or not. Rouge-1, Rouge-2 and Rouge-L are the ROUGE scores for numeric token, trigrams, bigrams, sentence position, TF-IDF, thematic score, proper noun score, sentence length, and cosine similarity. These scores help to estimate the relevance of the selected features to the original document. w1, w2, w3, w4, and w5 are weights that can be adjusted to prioritize different aspects of the summary. The foraging behaviour of flamingo depends upon three features. They are communicative, beak scanning and bipedal mobile behaviour.

Feature 1: Communicative behaviour

AFO is a technique that simulates flamingos to seek the ideal result in the exploration region depend on the limited available data. Consider that the flamingo has more food in the dimension j is xb_j.

Feature 2: Beak Scanning (BS) behaviour

The flamingos foraging behaviour encounters an error with the transmitted information. A Standard Normal Distribution (SND) is presented for simulating the error. Flamingo's behaviour in an aspect of maximum distance can be quantified as $\left|g_1 \times xb_j + \varepsilon_2 \times x_{ij}\right|$. Here, ε_2 represents a random number in the range of -1 *or* 1. g_1 is a random number which follows SND. During BS behavior, to simulate the flamingos scanning range the normal distribution is again introduced, and its variation curve approximates the variation of the flamingo's BS range as $g_2 \times \left|g_1 \times xb_j + \varepsilon_2 \times x_{ij}\right|$, where g_2 denotes random number followed by SND.

Feature 3: Bipedal mobile behaviour

The distance between the food and flamingo can be quantified as $\varepsilon_1 \times xb_j$, Here ε is a random number of -1 *or* 1. An input variable is split into large, small, and medium, depending on the triangular membership function, and the range of this triangular function is between -1 *and* 1. The following equation (25) is used for finding the iteration value.

$$Iteration = \frac{current\ iteration}{Total\ iteration} \tag{25}$$

Fuzzy structure rules are designed to reach the accurate triangular value. The following conditions are described by the following fuzzy system.

- If R is -1, then the iteration is low.
- If R is 0, then the iteration is medium.
- If R is 1, then the iteration is high.

To examine flamingos foraging in the t^{th} iteration of moving step is given in (26).

$$b_{ij}^t = \varepsilon_1 \times xb_j^t + g_2 \times \left|g_1 \times xb_j^t + \varepsilon_2 \times x_{ij}^t\right| \tag{26}$$

The formula for foraging the flamingo by updating the image location is given in (27).

$$x_{ij}^{t+1} = \frac{\left(x_{ij}^t + \varepsilon_1 \times xb_j^t + g_2 \times |\, g_1 \times xb_j^t + \varepsilon_2 \times x_{ij}^t \,|\right)}{K} \tag{27}$$

where x_{ij}^{t+1} and x_{ij}^t signifies the flamingo's position i, in dimension j of the population in the iteration $t+1$ and t. A n degree of freedom that can be chosen as an arbitrary number followed by chi-square, is a diffusion factor denoted by $K = k(n)$. SND followed by random numbers are represented as $g_1 = N(0,1)$ and $g_2 = N(0,1)$. $\varepsilon_1, \varepsilon_2$ are randomize by -1 *or* 1.

Step 4: Shifting behaviour

When food is infrequent in the present foraging area, the flamingo moves closer to the area where food is plentiful. The migration of the flamingo's population is given in (28).

$$x_{ij}^{t+1} = x_{ij}^{t} + \omega + \left(xb_{j}^{t} - x_{ij}^{t}\right) \tag{28}$$

where, $\omega = N(0, n)$ is the gaussian random number with n degrees of freedom.

Step 5: Termination

Here the AFO is used to select the significant features from the extracted summary. If the optimal solution is not achieved then repeat phase 4 to 3 till met the halting conditions $i = i+1$. After selecting the important features, the data is given to ARTL for effective ATS [40].

3.5.3. Abstractive text summarization Using ARTL

After feature selection the important features are given to ARTL for ATS. The process of making a brief summary that encapsulates the important contents of the actual text is known as ATS. The generated summaries might include new words and phrases that aren't in the original text. ARTL is the hybridization of TSO and RL.

TSO: The TSO is modelled as, initialization of the search agents among upper and lower bounds of the search area, searching for optimal solution and reaching the optimal solution.

1. First and foremost, the search agent's initialization is created randomly, as shown in (29).

$$Y = LB + Rand \times (UB - LB) \tag{29}$$

where Y mentioned as search agent position
Rand is the evenly distributed random number
UB - Upper Bound
LB - Lower Bound

2. Searching for optimal solution.
3. Reaching the optimal solution.

The r_1 is the random number utilized to balance among the exploitation $(r_1 < 0.5)$ and exploration $(r_1 \geq 0.5)$ of the Transient search algorithm (TSO). The best position $(Y_L)^*$ of former search agents can be established from the new position Y_L. The whole method is explained in following (30)-(33):

$$f(x) = \left\{ (Y_L)^* \left(Y_L - C_1 \times (Y_L)^* \right) e^{-T} \;\; for\ r_1 < 0.5 (Y_L)^* + e^{-T} [\cos(2\pi T) + \sin(2\pi T)] \left| Y_L - C_1 (Y_L)^* \right| \,\&\, for\, r_1 \geq 0.5 \right. \tag{30}$$

$$T = 2 \times M \times r_2 - M \tag{31}$$

$$C_1 = k \times M \times r_3 + 1 \tag{32}$$

$$M = 2 - 2 \times \left(\frac{L}{l_{\max}} \right) \tag{33}$$

In this, M is the variable that changes from 2 to 0, c_1 and T are random coefficients $(r_i)_i^3 = 1$ are random numbers distributed evenly $\in [0,1]$, L is the iteration number, the constant number is k as $(0,1,2,\ldots..)$, and the maximal number of iterations is $l_{\max}$. By the coefficient T the exploitation and exploration process is realized which is varies among $-2 to 2$, The exploration and exploitation of TSO is achieved (when $T < 0$, for exploration phase, and when $T > 0$ for exploitation phase).

It is clear that the TSO is not complex and used only one equation for balancing and updating between the exploitation and exploration. The TSO's computational complexity is represented as O. The complexity of all search agent function is then reported as $O(N * l_{\max})$. The TSO method's computational complexity is thus denoted as $O\left(Nx\left(l_{\max} D + l_{\max} + 1\right)\right)$.

Reinforcement learning:

The final abstracted summary is differed from the input summary but it has same meaning. Equation (34) helps to derive the iterative updates.

$$Q_{i+1}(s,a) = e\left[r + \beta \max_{a'} Q_i\left(s', a' \mid s, a\right)\right] \tag{34}$$

where
r - reward
e - Expectation (that include what action agent takes in states)
β - Reduced factor for upcoming rewards where $(\beta \in [0,1])$

Set $\beta = 0$ indicates that present strategy is short-sighted and merely takes the benefits of the current course of action into account. Conversely, a higher score β indicates the best future optimism. A successful policy should strike a balance between the current and future rewards. A smart policy should maximise the existing reward while maximizing the existing reward with the upcoming reward.

The reward function calculates the comparison among the gold summary $\{y_1, \ldots\ldots, y_j\}$ and the final abstracted summary $\{x_1, \ldots\ldots, x_j\}$ depends on the Longest Common Subsequence (LCS) recall metric contained with ROUGE and n-gram concurrence score, which is given in (35).

$$\mathrm{Re}ward(Sum) = \{-1 \quad if\ \ length(sum) > k, score(sum) \quad otherwise \tag{35}$$

The difference of score among the previous and current iteration is measured, hence $Score(sum)$ is formulated as in (36),

$$Score(sum) = \{score(s_c) - score(s_p) \quad if\ score(s_n) - score(s_p) > threshold - 1\ otherwise \tag{36}$$

where $threshold$ is a constant; s_c and s_p illustrate the summary abstracted in the current and previous iteration steps, respectively, which means for the current summary to be more successful and effective than the previous summary. s_n is the score function. The summary generated by proposed method composed of natural sentences and it is clear that this GS is same to the reference summary in semantics [41].

4. Results and Discussion

This section explains the summarization of text using extractive and abstractive using improved techniques. PYTHON is used to assess the outcomes. The TS-LSTM is compared with existing approaches such as DQNrnn-rnn [30] and N-grams [31] for ETS. And the effectiveness of the ARTL is compared with the present methods, such as WACNNs [16], concept pointer+DS, concept pointer+DL [27], and GLEAM [28] for ATS.

4.1 Experimental Setup

In the Gigaword and DUC-2004 datasets the experiment was conducted. There are 20% testing samples and 80% training samples in it. The training and testing for both extractive and abstractive text summarization were done. Google Colaboratory, or Google Colab for short, served as the process' Integrated Development Environment (IDE). Google colab requires no set-up and it is cloud-based Jupyter notebook environment that is free of cost to use. With Google Colab, it can use powerful computer resources, store and share our studies, and create and run code from the browser. Using Google Drive, both datasets were mounted to Google Colab. Python is the programming language utilized, together with the machine learning packages Keras and Tensorflow. Table 1 includes additional setup information.

Table 1. Additional setup information

Configuration parameters	Value
Device name	DESKTOP-G518UGB
System type	64-bit operating system, x64-based processor
OS build	19044.1889
Processor	Intel (R) Core (TM) i7-6700 CPU @3.40 GHz 3.40GHz
Installed Ram	16.0 GB (15.8 GB usable)

4.2. Dataset Description

To estimate the efficiency of proposed technique, the training and testing are conducted on DUC-2004 and Gigaword. The DUC-2004 dataset contains of 500 novel articles, while Gigaword is a dataset for sentence summarising. Four human-written reference descriptions are paired with every article in the collection. The comparison of actual and the GS produced by proposed technique is shown in Table 2.

4.3. Evaluation Metrics

The introduced technique is estimated with the help of precision, recall, F1-score and ROUGE scores.

Precision: The precision is calculated using (37),

$$\Pr ecision = \frac{Sentences_{relevant} \cap Sentences_{retrieved}}{Sentences_{retrieved}} \tag{37}$$

where relevant sentences are represented as $Sentences_{relevant}$ and retrieved sentences are mentioned as $Sentences_{retrieved}$.

Recall: The recall is evaluated by (38),

$$\text{Re}call = \frac{Sentences_{relevant} \cap Sentences_{retrieved}}{Sentences_{relevant}} \tag{38}$$

where relevant sentences are represented as $Sentences_{relevant}$ and retrieved sentences are represented as $Sentences_{retrieved}$.

F1-score: The F1-Score can be calculated using (39).

$$F1 - score = \frac{2 * precision * recall}{precision + recall} \tag{39}$$

ROUGE: It is determined whether the produced summary and reference summary overlaps for lexical units like unigrams, bigrams, word sequences, and LCS. It is a measure that is often used for text summarization.

It is calculated by following (40):

$$ROUGE - N = \frac{\sum s \in \{reference_{\ summaries}\} gram_n \in s \sum count_{match}(gram_n)}{\sum s \in \{reference_{\ summaries}\} gram_n \in s \sum count(gram_n)} \tag{40}$$

From (40), N is n-gram distance, $reference_{\ summaries}$ is the reference summaries, $count_{match}(gram_n)$ is the supreme quantity of n-grams and $count_{match}$ is number of n-grams in reference summary.

Table 2. Comparison among the actual and the GS

Reference Article 1: “In the conflict-torn Southern Philippines (SP), a supposed bomb attack on a traveler bus resulted in at least two fatalities.” Actual Summary (AS): “Philippines explosion leaves at least two people dead” GS: “Bombing in SP results in two death”
Reference Article 2: “Spanish real estate company Colonial, which is drowning in debt, reported losses of 1 billion euros for the first half of the year, which it attributed to asset depreciation.” AS:“Spain ’s colonial posts #.## billion euro loss” GS: “Spain ’s colonial posts #.## billion billion billion”
Reference Article 3: “Kim Sook, the nuclear envoy for South Korea, urged North Korea on Monday to resume efforts to shut down its nuclear facilities and to cease its "usual" brinkmanship in negotiations.” AS: “Envoy requests that North Korea resume nuclear disarmament” GS: “North Korea is advised to continue work by the nuclear envoy for South Korea.”

4.4. ROUGE Scores for Extractive Model

Fig. 3 displays the performance comparison of summarization models on ROUGE Score for ETS using DUC-2004 and Gigaword dataset. Fig. 3(a) shows the ROUGE-1 score analysis of introduced ETS using DUC-2004 dataset. The comparison is made with DQNrnn-rnn and N-grams. The analysis shows that the proposed method attains higher ROUGE-1 score than the existing approaches. The TS-LSTM attains higher ROUGE-1 score as 57.13%. Fig. 3(b) shows the ROUGE-2 score analysis of proposed ETS using DUC-2004 dataset. The proposed method attains 28.35% ROUGE-2 score which is higher than the existing approaches. The N-grams attains lower ROUGE-2 score as 10.89%. The Fig. 3(c) shows the ROUGE-L analysis of proposed ETS using DUC-2004 dataset. The proposed TS-LSTM attains higher ROUGE-L score than the existing approaches. The ROUGE-L score of proposed method is 52.83%. The N-grams has lower ROUGE-L score as 26.4%.

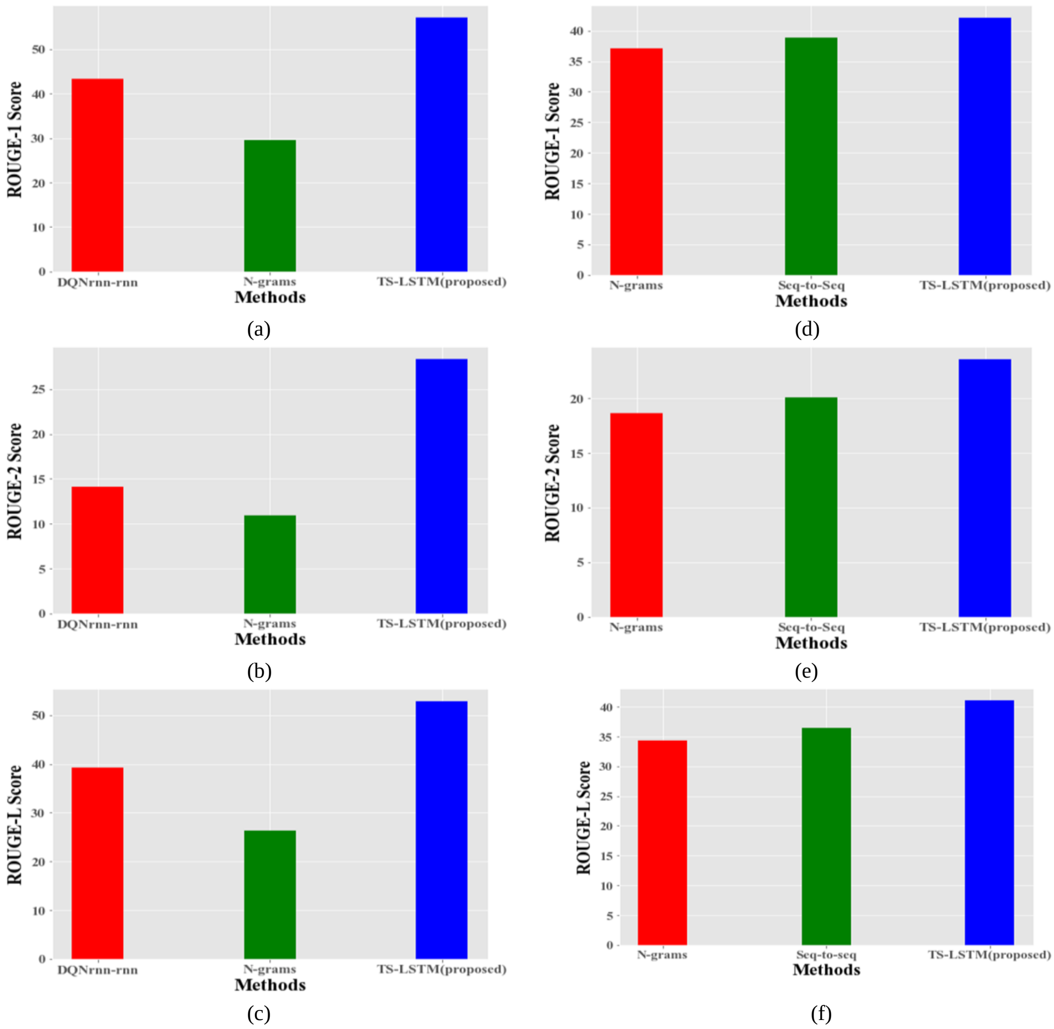
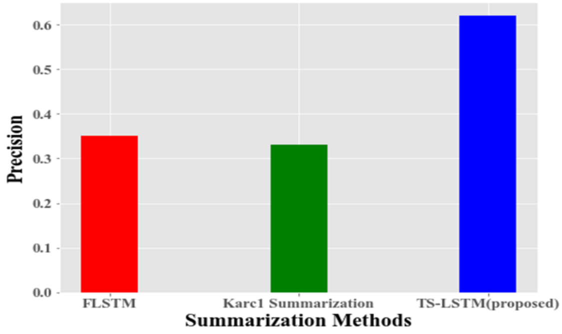


Fig. 3. Performance comparison of summarization models on ROUGE Score for extractive method using DUC-2004 and Gigaword dataset

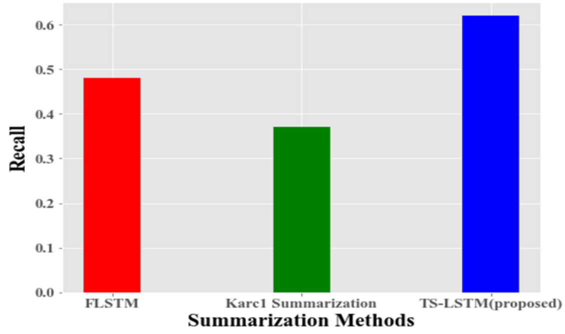
Fig. 3(d) shows the ROUGE-1 score analysis of TS-LSTM using Gigaword dataset. The comparison is made with N-grams and seq-to-seq. While comparing with other approaches the N-gram has lower ROUGE-1 score. The ROUGE -1 score of TS-LTSM is 42.11% which is higher than other approaches. Fig. 3(e) shows the ROUGE-2 score analysis of TS-LSTM using Gigaword dataset. The proposed method attains 23.55% ROUGE-2 score which is higher than the existing approaches. The N-grams attains lower ROUGE-2 score as 18.66%. The Fig. 3(f) shows the ROUGE-L analysis of proposed ETS using Gigaword dataset. The proposed TS-LSTM attains higher ROUGE-L score than the existing approaches. The ROUGE-L score of proposed method is 41.05%. The N-grams has lower ROUGE-L score as 34.38%.

4.5. Precision, Recall and F1-score for the Extractive Summarization of Text

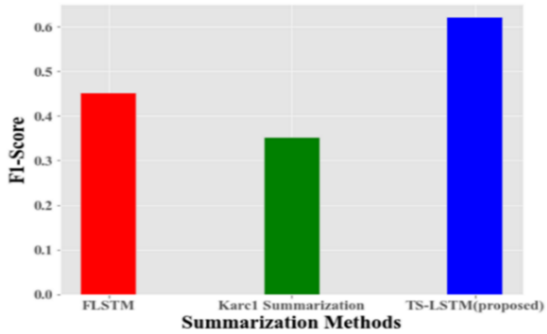
This section describes the precision, recall and F1-score analysis of TS-LSTM. Fig. 4(a) shows the precision analysis of proposed TS-LSTM. TS-LSTM attains 62.5% of precision which is higher than the existing approaches. The comparison is made with FLSTM [33] and Karcı Summarization [34]. Karcı Summarization attains lower precision as 33.3%. Fig. 4(b) shows the recall analysis of proposed method. The Karcı Summarization attains lower recall value as 37.3%. The recall value of TS-LSTM is 63% which is higher than the existing approaches. Fig. 4(c) shows the F1-score analysis of proposed method. The proposed method attains higher F1-score as 62%, which is higher than the existing approaches. The F1-score of Karcı Summarization is 35.1% which is lower than other approaches.



(a)



(b)



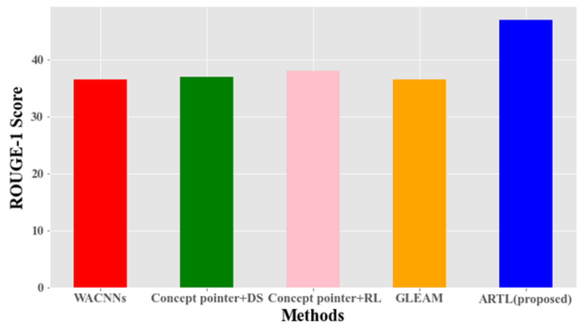
(c)

Fig. 4. Precision, recall and F1-score analysis of proposed extractive text summarization

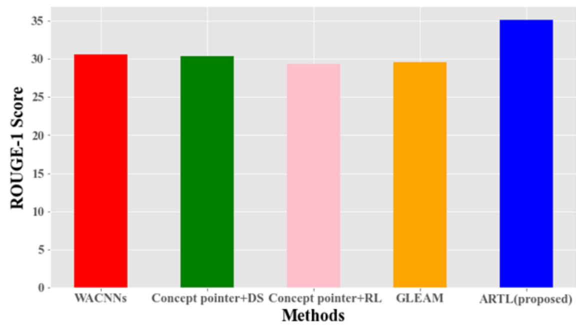
4.6. ROUGE Scores for Abstractive Model

Fig. 5 shows the performance comparison of summarization models on ROUGE Score for abstractive method using Gigaword dataset and DUC-2004 datasets. Fig. 5(a) displays the ROUGE-1 score analysis of introduced ATS using Gigaword dataset. The performance of ARTL is compared with the existing approaches such as; WACNNs, concept pointer+DS, concept pointer+DL, and GLEAM respectively. The ROUGE-1score of GLEAM is lower than other approaches. The proposed method attains higher ROUGE-1 score as 47.05%, which is higher than the existing approaches. Fig. 5(b) shows the ROUGE-2 score analysis of proposed ATS using Gigaword dataset. The ROUGE-1 score of GLEAM is lower than other approaches. The ROUGE-2 score of concept pointer+DS and concept pointer+RL are 17.10% and 16.97%. The ROUGE-2 score of WACNNs is 17.74% which is higher than other existing approaches but it is lower than the proposed method. The ROUGE-2 score of ARTL is 22.02%. Fig. 5(c) shows the ROUGE-L score analysis of ARTL using Gigaword dataset. The ARTL's ROUGE-L is higher than the existing approaches. The ROUGE-L score of WACNNs is lower than other approaches. The proposed method attains higher ROUGE-L score as 48.96% using Gigaword dataset.

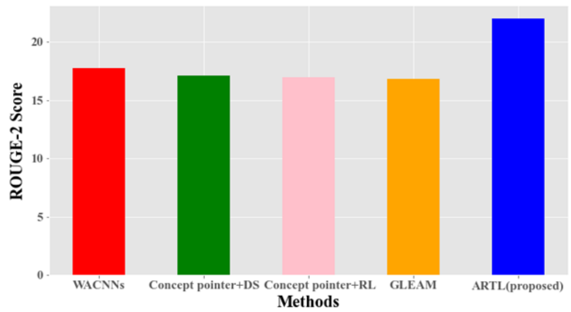
Fig. 5(d) shows the ROUGE-1 score analysis of ARTL using DUC-2004 dataset. The performance of ARTL is compared with the existing approaches such as; WACNNs, concept pointer+DS, concept pointer+DL, and GLEAM respectively. The proposed ARTL attains higher ROUGE-1 score than the existing approaches. While comparing to other approaches the Concept pointer+RL approach attains lower ROUGE-1 score. The Concept pointer+DS's ROUGE-1 score is almost near to the WACNNs approach. The proposed method attains higher ROUGE-1 score as 35.13%. Fig. 5(e) shows the ROUGE-2 score analysis of proposed ATS using DUC-2004 dataset. The analysis shows that the ARTL attains higher ROUGE-2 score than other approaches. The ROUGE-2 score of GLEAM is lower than other approaches. The proposed method attains higher ROUGE-2 score as 20.35%. Fig.5(f) shows the ROUGE-L score analysis of proposed ATS using DUC-2004 dataset. The analysis shows that the GLEAM attains lower ROUGE-L score than other approaches and the proposed ARTL attains higher ROUGE-L score as 32.25%.



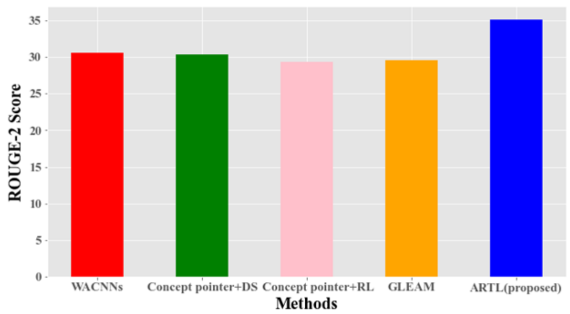
(a)



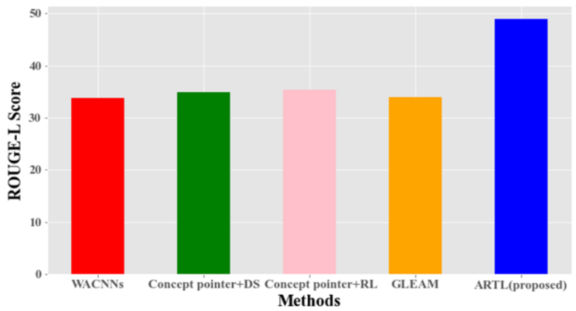
(d)



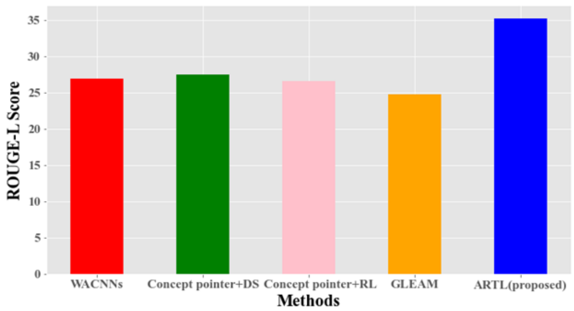
(b)



(e)



(c)



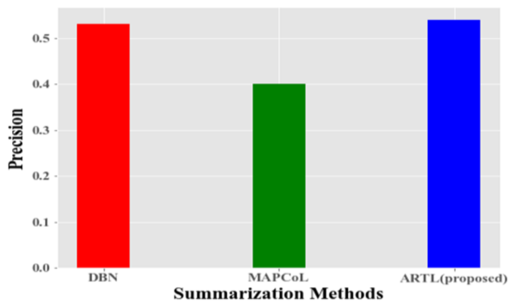
(f)

Fig. 5. Performance comparison of summarization models on ROUGE Score for abstractive method using Gigaword and DUC-2004 dataset

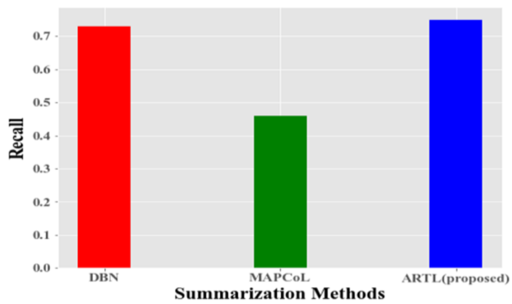
4.7 Precision, Recall and F1-score for the Abstractive Summarization of Text

Fig. 6 shows the precision, recall and F1-score analysis for ATS. Fig. 6(a) displays the precision analysis of ARTL. The analysis shows that the proposed method attains higher precision than the existing approaches. The comparison is made with DBN and MapCoL [29] approaches. The MAPCoL has lower precision than other approaches. The precision of proposed method is 54% which is higher than the existing approaches. Fig. 6(b) displays the recall analysis of ARTL. The recall value of ARTL is higher than the existing approaches. The recall value of ARTL is 75%. Fig. 6(c) displays the F1-score analysis of ARTL. The ARTL’s F1-score s higher than the existing approaches. The F1-score value of ARTL is 60%. The MAPCoL has lower F1-score as 42%.

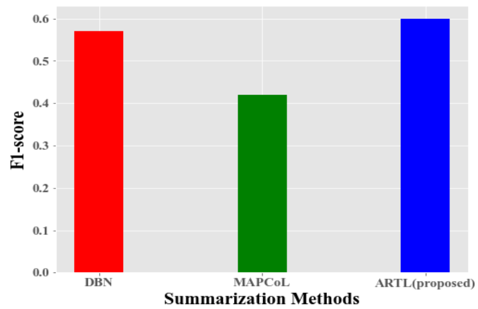
Fig. 7(a), displays the training and testing loss for ETS. The red line mentions the testing and the blue line mentions the training. It shows that the TS-LSTM achieved minimum loss for ETS. Fig 7(b), displays the training and testing loss for ATS. The analysis shows that the ARTL attains minimum loss. From the analysis it is clear that the TS-LSTM and ARTL attains high accuracy with minimum loss rate.



(a)



(b)



(c)

Fig. 6. Precision, recall and F1-score analysis of proposed abstracted text summarization

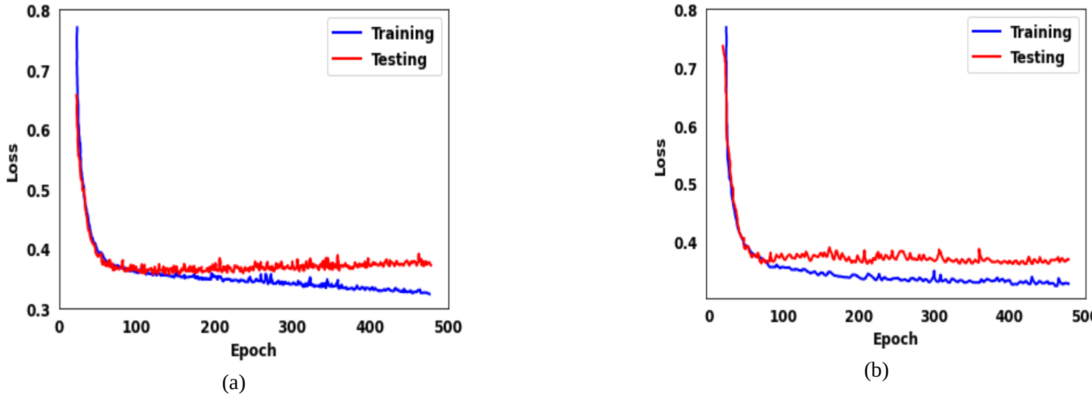


Fig. 7. Training and testing loss for extractive and abstractive text summarization

5. Conclusion

The proposed method effectively summarizes the document with high quality. The classification process uses specific keywords to select the sections to summarize. Knowledge graph-based extraction of summaries is extracted for the ETS. In the abstraction phase, a certain feature selection process is done to expand the summary's quality. After feature extraction, AFO is used to select the optimal features. The output is sent to the Adaptive Transient Reinforcement Learning technique for effective summarization in the abstraction phase. The performance metrics like F1-score, precision, recall and ROUGE scores for both extraction and abstraction phases are evaluated. The proposed method attains 0.625% precision, 0.63% recall and 0.62% F1-score for ETS and 0.54% precision, 0.75% recall and 0.60% F1-score for ATS.

References

[1] R. Bhargava and Y. Sharma, "Deep Extractive Text summarization," *Proc. Comp. Sci.*, vol. 167, pp. 138–146, 2020.
[2] A. K. Yadav, A. Singh, M. Dhiman, Vineet, R. Kaundal, A. Verma, and D. Yadav, "Extractive text summarization using Deep Learning Approach," *Inter J. Inf Technol.*, vol. 14, no. 5, pp. 2407–2415, 2022.
[3] H. Aliakbarpour, M. T. Manzuri, A. M. Rahmani, "Improving the readability and saliency of abstractive text summarization using combination of deep neural networks equipped with auxiliary attention mechanism," *J. Supercomput.* vol. 78, no. 2, pp. 2528–2555, 2021.
[4] D. Debnath, R. Das, and S. Rafi, "Sentiment-based abstractive text summarization using attention oriented LSTM model," *Intell. Data Eng. Anal.*, pp. 199–208, 2022.
[5] H. Kumar, G. Kumar, S. Singh, and S. Paul, "Text summarization of articles using LSTM and attention-based LSTM," *Mach. Learn. Auto. Syst.*, pp. 133–145, 2022.
[6] Q. Wang, P. Liu, Z. Zhu, H. Yin, Q. Zhang, and L. Zhang, "A text abstraction summary model based on Bert word embedding and reinforcement learning," *Appl. Sci.*, vol. 9, no. 21, p. 4701, 2019.
[7] F. Ertam and G. Aydin, "Abstractive text summarization using Deep Learning with a new Turkish summarization benchmark dataset," *Concur. Comput: Pract. Exp.*, vol. 34, no. 9, 2021.
[8] D. Suleiman and A. Awajan, "Multilayer encoder and single-layer decoder for abstractive Arabic text summarization," *Knowledge-Based Syst.* vol. 237, p. 107791, 2022.
[9] A. Kumar, S. Seth, S. Gupta, and S. Maini, "Sentic computing for aspect-based opinion summarization using multi-head attention with feature pooled pointer generator network," *Cogn. Comput.*, vol. 14, no. 1, pp. 130–148, 2021.
[10] R. C. Belwal, S. Rai, and A. Gupta, "Text summarization using topic-based vector space model and semantic measure," *Inf. Process. Manag.*, vol. 58, no. 3, p. 102536, 2021.
[11] H. P. Chan and I. King, "A condense-then-select strategy for text summarization," *Knowl. Based Syst.*, vol. 227, p. 107235, 2021.
[12] M. Sultana, P. Chakraborty, and T. Choudhury, "Bengali abstractive news summarization using Seq2Seq Learning with attention," *Lect. Notes Netw. Syst.* pp. 279–289, 2021.
[13] T. Vo, "SE4ExSum: An integrated semantic-aware neural approach with graph convolutional network for Extractive Text summarization," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 20, no. 6, pp. 1–22, 2021.
[14] W. Liao, Y. Ma, Y. Yin, G. Ye, and D. Zuo, "Improving abstractive summarization based on dynamic residual network with reinforce dependency," *Neurocomp.*, vol. 448, pp. 228–237, 2021.
[15] B. Mohan Bharath, B. Aravindh Gowtham, and M. Akhil, "Neural Abstractive Text Summarizer for Telugu Language," in Soft Computing and Signal Processing. *Advances in Intelligent Systems and Computing*, V.S. Reddy, V.K. Prasad, J. Wang, K.T.V. Reddy, Eds. Singapore: Springer, 2022, vol. 1340.
[16] T. Ma, Q. Pan, H. Rong, Y. Qian, Y. Tian, and N. Al-Nabhan, "T-bertsum: Topic-aware text summarization based on bert," *IEEE Trans. Comput. Soc.*, vol. 9, no. 3, pp. 879–890, 2022.
[17] A. Abdi, S. Hasan, S. M. Shamsuddin, N. Idris, and J. Piran, "A hybrid deep learning architecture for opinion-oriented multi-document summarization based on multi-feature fusion," *Knowl. Based Syst.*, vol. 213, p. 106658, 2021.

[18] I. Cachola, K. Lo, A. Cohan, and D. S. Weld, "TLDR: Extreme summarization of scientific documents," 2020. arXiv preprint arXiv:2004.15011.
[19] Ankita, S. Rani, A. K. Bashir, A. Alhudhaif, D. Koundal, and E. S. Gunduz, "An efficient CNN-LSTM model for sentiment detection in #BlackLivesMatter," *Expert Syst. Appl.*, vol. 193, p. 116256, 2022.
[20] V. Vaissnave, and P. Deepalakshmi, "A Keyword-Based Multi-label Text Categorization in the Indian Legal Domain Using Bi-LSTM," in Soft Computing: Theories and Applications. *Advances in Intelligent Systems and Computing*, T.K.Sharma, C.W. Ahn, O.P. Verma, B.K. Panigrahi, Eds. Singapore: Springer, vol. 1380, 2022.
[21] T. Shi, Y. Keneshloo, N. Ramakrishnan, and C. K. Reddy, "Neural abstractive text summarization with sequence-to-sequence models," *ACM Transact Data Sci.*, vol. 2, no. 1, pp. 1-37, 2021.
[22] R. Rani, and D. K. Lobiyal, "An extractive text summarization approach using tagged-LDA based topic modeling," *Multimed. Tools Appl.*, vol. 80, pp. 3275-3305, 2021.
[23] S.K. Mishra, N. Saini, and S. Saha S, "Bhattacharyya P. Scientific document summarization in multi-objective clustering framework," *Appl. Intell.*, vol. 52, no. 2, pp. 1520-1543, 2022.
[24] S. Song, H. Huang, and T. Ruan, "Abstractive text summarization using LSTM-CNN based Deep Learning," *Multi. Tools Appl.*, vol. 78, no. 1, pp. 857–875, 2018.
[25] Y. M. Wazery, M. E. Saleh, A. Alharbi, and A. A. Ali, "Abstractive Arabic text summarization based on Deep Learning," *Comp. Intell. Neuro.*, vol. 2022, pp. 1–14, 2022.
[26] C. Yuan, Z. Bao, M. Sanderson, and Y. Tang, "Incorporating word attention with convolutional neural networks for abstractive summarization," *World Wide Web*, pp. 267-287, 2020.
[27] W. Wenbo, G. Yang, H. Heyan, Z. Yuxiang, "Concept pointer network for abstractive summarization," arXiv preprint arXiv:1910.08486. 2019.
[28] Y. Gao, Y. Wang, L. Liu, Y. Guo, and H. Huang, "Neural abstractive summarization fusing by global generative topics," *Neural Comput. Appl.*, vol. 32, pp. 5049-5058, 2020.
[29] P. Mahalakshmi, and N. S. Fatima, "Summarization of Text and Image Captioning in Information Retrieval Using Deep Learning Techniques," *IEEE Access*, vol. 10, pp. 18289-18297, 2022.
[30] K. Yao, L. Zhang, T. Luo, and Y. Wu, "Deep reinforcement learning for extractive document summarization," *Neurocomput.*, vol. 284, pp. 52-62, 2018.
[31] Z. Jiang, M. Srivastava, S. Krishna, D. Akodes, and R. Schwartz, "Combining word embeddings and n-grams for unsupervised document summarization," 2020. arXiv preprint arXiv:2004.14119.
[32] D. Y. Sakhare, "A Sequence-to-Sequence Text Summarization Using Long Short-Term Memory Based Neural Approach," *Int. J. Intell. Syst.*, vol. 16, no. 2, pp. 142-151, 2023.
[33] M. Tomer, and M. Kumar, "Improving text summarization using ensembled approach based on fuzzy with LSTM," *Arab. J. Sci. Eng.*, vol. 45, pp. 10743-10754, 2020.
[34] C. Hark, and A. Karcı, "Karcı summarization: A simple and effective approach for automatic text summarization using Karcı entropy," *Inf Process & Manag*., vol. 57, no. 3, p. 102187, 2020.
[35] https://huggingface.co/datasets/gigaword
[36] https://paperswithcode.com/dataset/duc-2004
[37] A. Y. Muaad, M. A. Al-antari, S. Lee, and H. J. Davanagere, "A novel deep learning arcar system for Arabic text recognition with character-level representation," *IOCA 2021*, 2021.
[38] J. Jiang, H. Zhang, C. Dai, Q. Zhao, H. Feng, Z. Ji, and I. Ganchev, "Enhancements of attention-based bidirectional LSTM for hybrid automatic text summarization," *IEEE Access*, vol. 9, pp. 123660–123671, 2021.
[39] Z. Liu, P. Jiang, J. Wang, and L. Zhang, "Ensemble system for short term carbon dioxide emissions forecasting based on multi-objective tangent search algorithm," *J. Environ. Manag*., vol. 302, p. 113951, 2022.
[40] L. Raamesh, S. Radhika, and S. Jothi, "Generating optimal test case generation using shuffled Shepherd Flamingo Search Model," *Neural Process. Lett.*, vol. 54, no. 6, pp. 5393–5413, 2022.
[41] K. Miao, X. Wang, M. Zhu, S. Yang, X. Pei, and Z. Jiang, "Transient controller design based on Reinforcement Learning for a turbofan engine with Actuator Dynamics," *Sym*., vol. 14, no. 4, p. 684, 2022.

Authors' Profiles

Reshmi P. Rajan, MCA, Mtech has around 16 years of R&D experience in IT industry. She is currently engaged as a Research Scholar in the Department of Computer Science, Christ (Deemed to be University). Her research interests include Natural Language Processing, Computer Vision, and Quantum Machine Learning. She has authored and published numerous Research articles in Journals and Conference proceedings.

Deepa V. Jose, MCA, MTech, MPhil, PhD works as an Associate Professor in the Department of Computer Science of Christ (Deemed to be) University as an Associate professor. She is an active researcher in the areas of Natural Language Processing, Cybersecurity, and Artificial Intelligence. She has authored and published numerous research articles and Book chapters.

Roopashree Gurumurthy is a Senior Project Program Manager at NTT DATA Services. She is PMP certified, Prince 2 and a Scrum Master offering 15 years of experience and expertise in handling end-to-end development of projects from inception, requirement specs, planning, designing, implementation, configuration management, documentation and closure with cross-functional teams.

How to cite this paper: Reshmi P Rajan, Deepa V. Jose, Roopashree Gurumoorthy, "Tangent Search Long Short Term Memory with Aadaptive Reinforcement Transient Learning based Extractive and Abstractive Document Summarization", International Journal of Modern Education and Computer Science(IJMECS), Vol.15, No.6, pp. 56-72, 2023. DOI:10.5815/ijmecs.2023.06.05