Computer Science • 26(3) 2025 https://doi.org/10.7494/csci.2025.26.3.6352

Sunil Upadhyay
Hemant Kumar Soni

ANAPHORA SOLVED AD-DL-BERT MODEL FOR TEXT SUMMARIZATION WITH AUTO ENCODING USING THE TOPIC DESCRIPTION AND SEVERAL PRIORS (ATDS) APPROACH

Abstract *Although several models for automatic text summarization exist, there are still limitations – like the anaphora problem that occurs during summarizations. To overcome such limitations, this paper proposes the Added dropout-Deleted Layer norm-Bidirectional Encoder Representations from Transformers (Ad-DL-BERT)-based extractive text summarization (ETS). Primarily, the input document's sentences are prepared for accurate summarization by pre-processing; then, the unwanted sentences are removed. With the Auto encoding using the Topic Description and Several priors (ATDS) approach, any sentences under the same topic are clustered afterwards. Moreover, keywords for summarization are extracted with an AnaphoraPOS (An-POS) extractor. For removing the redundant sentences, the rankings with Exponential Linear Unit-Generative Adversarial Network (ELU-GAN) and saliency score assignment processes are performed thereafter. Also, assignments for sentences are performed to enhance the coherency, sorting, and cosine-similarity score. Lastly, the Ad-DL-BERT-generated summary and the proposed technique's performance are evaluated on the document understanding conference (DUC2002) data set. Regarding the clustering time, execution time, recall-oriented understudy for the gisting evaluation (ROUGE-1) scores of recall, F-measure, and precision, the experimental outcomes exhibited the proposed technique's dominance over the conventional approaches.*

Keywords Automatic Text Summarization (ATS), Co-reference Resolution (CR), Anaphora, Extractive Text Summarization (ETS), Bidirectional Encoder Representations from Transformers (BERT), Deep Learning (DL), Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

Citation Computer Science 26(3) 2025: 1–24

1. Introduction

Recently, the process of going through entire texts on major platforms for getting relevant information has taken more time. To understand an overall digital files in less time, researchers are developing technological approaches that can summarize the text data automatically [29]. The process of diminishing the amount of text to get the most significant parts from the actual text and providing it to users is called text summarization (TS). The summarization process is automatically carried out by the ATS method [10] [12]. ATS is grounded on three types; namely, input-centric summarization, context-centric summarization, and output-centric summarization. Input-centric TS is categorized as single-document or multi-document summarization (DS) [20]. A single document is given for summarization in single-DS, while multiple documents are given at a time to get summarized texts in multi-DS.

Grounded on the output, the summaries are classified as extractive and abstractive. The ETS technique is successful in delivering a summary by utilizing the most significant words that are present in the actual text, which delivers the most relevant information accurately [15]. In an abstractive TS system, the tokens are extracted; also, the summary is formed by utilizing some natural language-generation mechanisms [6]. Since abstractive TS must deal with issues such as natural language generation, semantic representation, vocabulary building, et cetera, the ETS system produces better summaries in contrast to abstractive TS [7]. For the term "frequency identification," different similarity measures (namely, cosine similarity) are wielded in the term frequency in ATS [23].

To summarize the text, most approaches like neural networks (NNs), sequence-to-sequence modeling, machine learning (ML), reinforcement learning, and fuzzy logic [2] are utilized; however, none of these are perfect. Thus, an opportunity is provided to continue finding breakthroughs in this area of automation. One of the significant limitations is that sentences are discarded for lack of context, which drastically affects the quality of the produced summary. To overcome this issue, BERT-based summarization models have been developed [25]. Owing to certain limitations like anaphora problems, however, such models cannot perform well; to resolve these limitations, this paper proposes a novel Ad-DL-BERT-based technique for automatic single DS with semi-supervised ELU-GAN and anaphora feature extraction.

To summarize texts automatically, much research was conducted, and several models [19] have been developed. However, some limitations in the previous models [5] that have failed to solve some problems have been detected; these are stated as follows:

(a) TS approaches used several features to extract the exact keyword and the relevant sentences for a summary. In the case of summarization, these approaches faced a problem in sorting sentences and ordering them accurately.

(b) The sentences that are selected for a summary may go verbatim due to the anaphora problem; this can result in incoherent summary results. Previous methods have attempted to resolve this problem and have been left owing to mismatches between anaphors with their antecedents.

(c) While scoring, semantically and syntactically incorrect sentences are ignored; this created an issue, as these metrics fail in evaluating grammatically incorrect sentences.

By analyzing these problems, the proposed framework aims to deploy a multi-domain anaphora-solved ATS model to generate summaries for texts more accurately.

The remaining paper is structured as follows: related works are discussed in Section II; the proposed mechanism is displayed in Section III; the outcomes are delineated in Section IV; and, finally, the paper is wrapped up in Section V.

2. Literature survey

An extractive multi-document TS system that was grounded on graph-independent sets was established by [28]. The presented system removed nodes from summaries, which were the set of sentences that were analogous to the nodes in the independent set on the graph model. Better ROUGE performance outcomes were attained by the developed model. Owing to limited ROUGE-2 score values, however, a summary for 400 words could not achieve better performance than the existing works could.

An ETS technique with tagged LDA (latent Dirichlet allocation)-centered topic modeling was employed by [21]. Extractive lexical knowledge-rich topic modeling for Hindi novels and stories was developed by the implemented approach, where different sentence-weighting schemes were available for independent variants. As per the results, the implemented approach had confessed optimal outcomes. Nevertheless, the essence of the summary deteriorated without semantic feature extraction and a co-reference resolution (CR).

Extractive document summarization (EDS) that is grounded on dynamic feature-space mapping was unveiled by [11]. Here, both supervised and unsupervised approaches were leveraged in a single framework for EDS purposes. The outcomes exhibited such behavior that, when contrasted with conventional techniques, the deployed model obtained high ROUGE scores; yet, some of the significant sentences were missed in the summaries (owing to the unresolved anaphora problem).

Extractive multi-DScentered on multi-objective optimization with K-Medoid clustering was deployed by [4]. The main topics in the text were discovered by the clustering-centric technique, while the three objectives were optimized by the evolutionary multi-objective optimization mechanism. For the ROUGE-1 evaluation, the summarized results achieved an F-measure score of 38.9%; however, the clustering process that was wielded in the model took more time, which caused high time complexity when the model was trained.

An enhanced TS approach with an ensemble approach was discussed in [26]; it was grounded on fuzzy and long short-term memory (FLSTM). To produce an abstractive summary, fuzzy logic and bidirectional LSTM (Bi-LSTM) were hybridized. As per the empirical outcomes, FLSTM outperformed all of the other techniques. Nevertheless, certain sentences that contribute to the summary might be omitted in FLSTM since it requires more memory bandwidth.

A Persian ATS that was grounded on neural named entity recognition (NER) was established by [16]. The model was comprised of three stages: supervised NER model training, named entities of the text recognition, and a summary generation. The NER model attained an overall improvement of 10.2% on the ROUGE-2 recall score without the use of any handcrafted features. However, reliable performance could not be attained by the ROUGE-2 scores on the pasokh single document.

A combined topic modeling with classification techniques for extractive multi-document TS was implemented by [22]. Primarily, the corpus of sentences was minimized with the latent Dirichlet allocation (LDA) technique. Then, important sentences were decided by various classifiers; among these, one representative sentence was chosen. For generating the summary, all of the representative sentences were organized in descending order. The prevailing models were outperformed by the experimental results on the DUC2002 and DUC2006 data sets; on DUC2006, however, the compared model overpowered the ROUGE2 performance of the developed model.

A single-document Arabic TS scheme was investigated by [1]. A formulation was generated for efficient summarization that was centered on diversity, coverage, and sentence importance. The experiential outcomes exhibited the introduced scheme's strength regarding the precision, recall, and F-score metrics when analogized with other prevailing works. The total score value's contribution was affected by the random selection of the weight values of the extracted features (regardless of of the fact that is was a better model for Arabic TS).

Unsupervised NNs for Arabic TS that were based on document clustering and topic modeling were developed by [3]. With extreme-learning machines, a new document was clustered; then, topic modeling was applied to the documents to identify the topics. Afterwards, every single document was signified by a matrix and was trained by utilizing various unsupervised NNs in the topic space. As per the outcomes, the models that was trained on topic representation enhanced the summarization performance. Nevertheless, the technique failed in obtaining a satisfactory F-measure value.

An unsupervised technique for extractive multi-DScentered on the centroid system and sentence embedding was recommended by [17]. Grounded on the three scores (namely, sentence position-, sentence novelty-, and sentence content-relevance scores), the relevant sentences were selected by the suggested technique. The outcomes exhibited that, when contrasted with the conventional techniques, the suggested technique attained more-promising and -outstanding outcomes. However, the centroid approach was sensitive to centroid initialization.

A multi-document ETS framework that was grounded on the firefly approach was developed by [27]. For effective summarization, the fitness function of the topic-relation factor, readability factor, and cohesion factor were wielded in the firefly algorithm. As per the outcomes, the developed model outperformed the other adopted algorithms. Nevertheless, the performance of the presented system was affected by the redundancy in the summarized sentences.

A single-document TS technique that was addressed with a cat swarm optimization (CSO) algorithm was demonstrated by [8]. Generating good summaries concerning information, readability, content coverage, and anti-redundancy was the CSO's objective. The presented approach attained a 25% enhancement on the ROUGE-1 score and a 5% improvement on the ROUGE-2 score; however, more time was taken to generate good summaries (owing to the complexity of the CSO algorithm).

In this work, the authors have proposed a model for both extractive and abstractive summarization that is called an automatic feature-rich model architecture that is comprised of hierarchical bidirectional LSTM (long short-term memory) [9]. Observation of the results showed that the model outperformed the existing techniques having ROUGE score equal to 37.76% and retaining high generality.

Here, the authors designed a fitness function that was based on genetic programming to generate the automatic text summarization. The experimental results displayed that the grouping of lexical and semantic information (LDA+Doc2Vec+TF-IDF) achieved excellent outcomes in finding key ideas to form a summary [14].

In this paper, the authors have demonstrated an approach for an extractive text summarization using NLP with an optimal DL (ETS-NLPODL) model [13]. A research analysis of the various parameters reflected that the ETS-NLPODL approach achieved excellent performance as compared to other models regarding a diverse set of performance measures.

3. Proposed methodology

Since ATS makes it easier to understand a large amount of information and reduces the time for understanding the content in a large document, it is becoming a popular approach. When summarizing the document automatically, however, numerous complexities are faced during the sentence arranging and accurate summarizing. To overcome this issue, this paper proposes a novel Ad-DL-BERT approach-based ATS framework (which is displayed in Figure 1).

3.1. Input data

Primarily, the text document from the DUC2002 data set was taken as the input data of the proposed ATS framework. The input data set (I) was mathematically expressed as follows:

$$I = \{S_1, S_2, ..., S_n\} \, or S_t, t = 1, 2, ..., n, \tag{1}$$

where S_n illustrates the n^{th} document in I.

3.2. Preprocessing

To obtain efficient summarized results, the input document(s_t) is preprocessed. Here, I is preprocessed by stemming, part-of-speech (POS) tagging and CR processes.

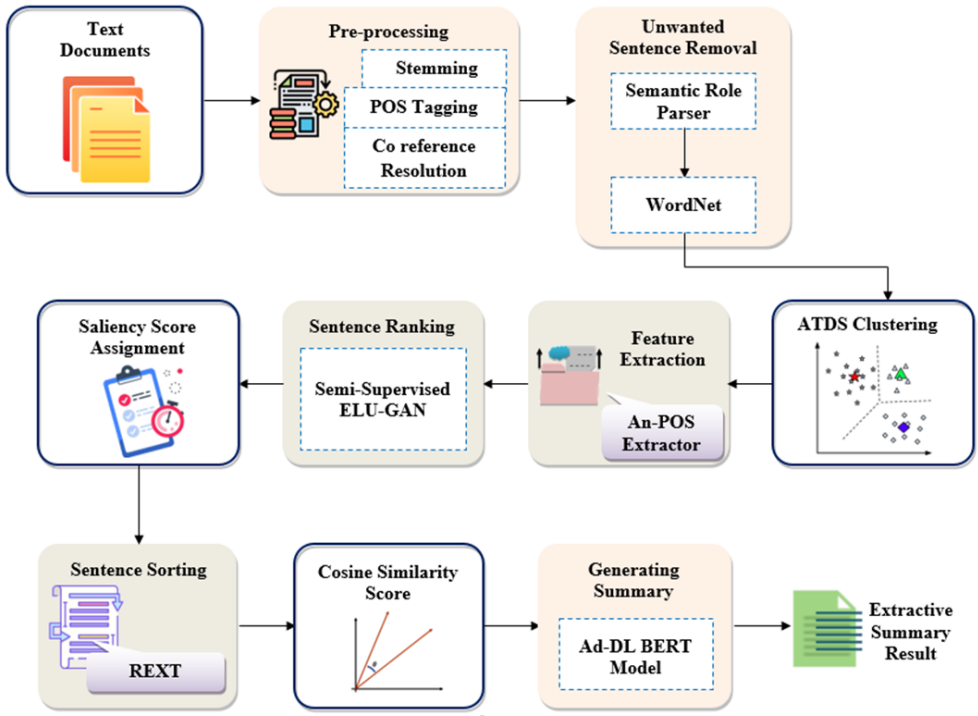

Figure 1. Schematic outline of proposed ATS framework

3.2.1. Stemming

Helping to reduce text dimensionality, stemming is the process of stripping suffixes. By neglecting the suffixes, the stems of the texts are wielded. Stemming is signified as follows:

$$chang \leftarrow change, changed, changes, changing, changer. \tag{2}$$

Here, words such as $change, changed, changes, changing, changer$ are stemmed as $chang$. To reduce the text dimensionality, the other texts are likewise stemmed.

3.2.2. POS tagging

The process of tagging POS texts is called POS tagging, which will help the summarization model recognize the POSs of a text. The POS-tagged texts tag POSs such as $article, adjective, noun, pronoun, verb, adverb$, and so on. These tagged POSs are wielded for training the summarization model, which identifies the POS during the testing process.

3.2.3. CR process

During the CR process, the linguistic expression that refers to the same entity in a document is identified without any error. This aids in increasing the text quality with minimum redundancy; thus, grammatically incorrect sentences cannot get entities. Afterwards, such sentences without entities are removed from the document. The preprocessed document (P) is given as follows:

$$P = \{p_1, p_2, ..., p_i\} \ \ or \ p_\alpha, \alpha = 1, 2, ..., i, \tag{3}$$

where p_i signifies the i^{th} preprocessed sentence.

3.3. Unwanted-sentence removal

By evaluating the WordNet similarity matrix with the utilization of semantic role analysis, the unwanted (repeated) sentences are removed after the document is preprocessed.

3.3.1. Semantic role analysis

From the preprocessed document, a word's semantic role in the sentence is labeled using a semantic role parser. This is to make the machines understand the role of a word within the corresponding sentence, which aids in summarizing the sentence effectively. The semantic role analysis process is explicated with an example as follows:

$$\begin{aligned} Anasoldherpropertytojohnfortwogoldcoins \rightarrow \quad & [agent(or)source]\,[Theme]\,[Goal] \\ & fortwogoldcoins \end{aligned} \tag{4}$$

Here, $[Ana]$ is considered to be the $agent(or)source$, $[soldherproperty]$ is labeled the $[Theme]$ of the sentence, and $[tojohn]$ is labeled as the goal of the sentence. Similarly, the entire document P is labeled for semantic role analysis. After the semantic role labeling, the document (D) is epitomized as follows:

$$D = \{d_1, d_2, ..., d_\Gamma\}, d_\epsilon, \epsilon = 1, 2, ...\Gamma, \tag{5}$$

where d_Γ symbolizes the Γ^{th}semantically role-labeled sentence.

3.3.2. Similarity matrix construction

After the semantic roles are labeled, the WordNet tool is utilized for evaluating the similarity scores between the sentences. With the predicted similarity scores by WordNet, a similarity matrix is constructed, and those sentences with the same similarity scores are eliminated. The similarity score (δ) evaluation is signified as follows:

$$\delta = 2 \times \frac{\Theta\left(lcs\left(d_\epsilon, d_{\epsilon+1}\right)\right)}{\Theta\left(d_\epsilon\right) + \Theta\left(d_{\epsilon+1}\right)}. \tag{6}$$

Here, $\Theta\left(\right)$ displays the depth function of synset in the WordNet (which is a special kind of interface for looking up words). lcs exemplifies the least-common subsumer.

Afterwards, the similarity score matrix is constructed with the similarity score δ for the entire document. Similarity score matrix (M) is represented as follows:

$$M = \begin{bmatrix} \delta_{1,1}\delta_{1,1} \cdots \delta_{1,rr} \\ \delta_{2,1}\delta_{2,2} \cdots \delta_{2,rr} \\ \vdots \cdots \vdots \\ \delta_{rr,1}\delta_{rr,2} \cdots \delta_{rr,rr} \end{bmatrix}. \tag{7}$$

Here, $\delta_{1,rr}$ represents the similarity score that is predicted between sentences 1 and rr in document D. The same similarity scores were removed from the similarity matrix to avoid repetitions of sentences. After the sentence removal, document (O) is exemplified as follows:

$$O = \{sc_1, sc_2, ..., sc_o\} \, or \, sc_{sr}, sr = 1, 2, ..., o, \tag{8}$$

where sc_o implies sentence o after the unwanted sentence removal.

3.4. ATDS clustering

Document (O) is given for topic modeling and further clustered utilizing the ATDS model. During the topic modeling, the topic for the word is determined as the most-often-repeated word in the document. Thereafter, the topic-assigned document is given for clustering by utilizing auto-encoders. During the clustering, those sentences under a single topic are categorized in a single cluster. For the main topic and the sub-topics, clusters are created; hence, a large number of clusters are generated. The topic modeling for O is given as follows:

$$\beta_q(O) = \left[\phi_l \left\{\rho\left(\frac{w_\gamma}{sc_{sr}}\right)\right\}\right], \tag{9}$$

where $\beta_q(O)$ exemplifies the q^{th} topic in document O, ϕ_l specifies the l^{th}-most-repetitive word, and $\rho\left(\frac{w_\gamma}{O}\right)$ implies the likelihood of the occurrence of word w_γ in document O. By doing this throughout the document, topics β_q are obtained. After the topics are identified, the words are clustered under the corresponding topics by using the auto-encoder model.

In the auto-encoder model, input sentence sc_{sr} is mapped to hidden unit $\hbar$ with non-linear mapping function $\Delta()$ as follows:

$$h_\epsilon = \Delta(sc_{sr}) = \frac{e^{sc_{sr}d_{sr}+B}}{1+e^{\omega_{sr}sc_{sr}+B}}, \tag{10}$$

where ω_{sr} signifies the weight value for input sc_{sr}, and B notates the bias value. After the hidden unit performs the mapping, the decoder reconstructs the input from h_{sr} as follows:

$$C_{sr} = \frac{e^{\omega_h h_{sr}+B_h}}{1+e^{\omega_h h_{sr}+B_h}}, \tag{11}$$

where C_{out} elucidates the cluster output, and ω_h, B_h epitomize the decoding weight and bias vectors, correspondingly. For minimizing the reconstruction error, the weight and bias parameters are resolved by the following optimization problem:

$$L = min\frac{1}{o}\sum_{sr=1}^{o}\left\|sc_{sr} - C_{sr}\right\|^{2} - \chi.\sum_{sr=1}^{o}\left\|\Delta^{I}\left(sc_{sr}\right) - \lambda_{k}^{*}\right\|^{2}, \tag{12}$$

$$\lambda_{k}^{*} = argmin\left\|\Delta^{I}\left(d_{\epsilon}\right) - \lambda_{k}^{I-1}\right\|^{2}, \tag{13}$$

where $\Delta^{I}()$ implies the non-linear mapping function at iteration I, and λ_{k}^{I-1} is the cluster center k at iteration $I-1$. λ_{k}^{*} is the nearest cluster center to the sentence sc_{sr} in the encoder layer. By doing this, a set of samples are clustered under a single cluster. To obtain optimal clusters, $\Delta^{I}()$ is optimized first; then, the cluster centers are updated as follows:

$$\lambda_{k}^{I} = \frac{\sum_{d_{\epsilon\in R_{k}^{I-1}}}\Delta^{I}\left(sc_{sr}\right)}{\left|R_{k}^{I-1}\right|}, \tag{14}$$

where R_{k}^{I-1} exemplifies a set of samples that belong to cluster k at iteration $I-1$. Therefore, the final topic clusters are represented as follows:

$$C = \{c_1, c_2, ..., c_m\}\ or\ c_{\eta},\ \eta = 1, 2, ..., m. \tag{15}$$

Here, the obtained cluster set is notated as C, and the clusters of the m^{th} topic is symbolized as c_m.

The pseudo code of the ATDS clustering is given as follows:

Input: Document O

Output: Clusters

Begin

Initialize $\boldsymbol{w_{\gamma}}$, sentencessc_{sr}, bias values (B, B_h)

For document O **do**

Perform topic modeling $\beta_q\left(O\right)$

End For

For each topic **do**

Perform clustering of sentences

For input sentences$\{sc_{sr}\}$**do**

Perform non-linear mapping $\Delta\left(sc_{sr}\right)$

Perform decoder reconstruction $C_{sr} = \frac{e^{\omega_h h_{sr}+B_h}}{1+e^{\omega_h h_{sr}+B_h}}$

If $(\boldsymbol{L == Threshold})\{$

Update clusters C_{sr}

$\}$ **Else** $\{$

Optimize $\boldsymbol{\Delta^{I}}()$and update cluster centers using $\boldsymbol{\lambda_{k}^{I}}$

}
End If
End For
End For
Return clustered document C
End

3.5. Feature extraction

After the clustering, the following features: phrase frequency; word; sentence length; position in the document having certain phrases; saliency features; Noun Anaphora; verb/adverb anaphora; zero anaphora and one anaphora were extracted from the clusters using the An-POS feature extractor. The features are extracted to frame the keywords, which are significant for TS. Therefore, extracted feature set (F) is exemplified as follows:

$$F = \{f_1, f_2, ...f_{\wp}\} \, or \, f_z, \; z = 1, 2, ..., \wp. \tag{16}$$

Here, $f_{\wp}$ implies the $\wp^{th}$ extracted feature.

3.6. Sentence ranking

After the features are extracted, the sentences are ranked using ELU-GAN to determine whether the sentence should be present in the summary or not; this helps to avoid the rankings of repetitive sentences. The semi-supervised generative adversarial network (GAN) works based on two units: a generator, and a discriminator. The generator creates a random input, which is then converted into a data instance. By comparing the original features with the generated features, the discriminator performs a similarity evaluation. For similar sentences, the ranking will not be provided [24]. Nevertheless the learning of long sentences reduces the coverage in the existing GAN. Therefore, the activation of the GAN was modified to an exponential linear unit (ELU) in the proposed ELU-GAN in order to overcome this problem.

Primarily, the features are given as inputs in ELU-GAN; with the features, the sentences are ranked according to the learned features. The value function of ELU-GAN is modeled as $T(\zeta, \mathfrak{F})$. Here, $\zeta, \mathfrak{F}$ symbolize the generator and discriminator of ELU-GAN. $\zeta$is designed to minimize loss function $[log\,(1 - \mathfrak{F}\,(\zeta\,(N)))\,]$. Here, N exemplifies the generated feature value. The generator generates fake data, and the discriminator discriminates the data to determine the real features, which is a min-max game on $T(\zeta, \mathfrak{F})$; this is formulated as follows:

$$\min_{G} \max_{D} T(\zeta, \mathfrak{F}) = E_{e \sim p(f_z)} \left[log \mathfrak{F} \;\; (N) \right] + E_{e \sim p(N)} \left[log\,(1 - \mathfrak{F}\,(\zeta\,(N)))\,\right], \tag{17}$$

where $E_{e \sim p(f_z)}$ illustrates the expectation of incoming data to the discriminator,(e) is the correspondence to the probability distribution of real features $(p\,(f_z))$, and

$E_{e\sim p(N)}$ specifies e correspondence to the probability distribution of generated features ($p\,(N)$). The discriminator in the ELU-GAN is a classifier that uses the ELU to perform Equation (17). The ELU is represented as follows:

$$U = \begin{cases} e & \text{if } e \geq 0 \\ \zeta\,(exp\,(e) - 1) & \text{if } e < 0 \end{cases}, \tag{18}$$

where U elucidates the ELU activation function, and ς implies a random distribution constant. With ELU's activation, the discriminator determines the following:

$$\mathfrak{F}^*\,(e) = \frac{p\,(f_z)}{p\,(f_z) + p\,(N)}. \tag{19}$$

$\mathfrak{F}^*\,(e)$ *or* a_v exemplifies the discriminator-ranked output, and $p\,()$ implies the probability distribution. Therefore, the final sentence-ranked document (A) is expressed as follows:

$$A = \{a_1, a_2, ..., a_\ell\} \text{ or } a_v, v = 1, 2, ..., \ell, \tag{20}$$

where a_ℓ symbolizes the ℓ^{th} ranked sentence (feature). The pseudo code of ELU-GAN is given as follows:

Input: Extracted features $\{f_1, f_2, ...f_\wp\}$ *or* f_z
Output: Ranked features (sentences)
Begin
Initialize features, ζ, $\mathfrak{F}$, discriminator input $\boldsymbol{e}$
Generate samples in the generator by minimizing $[log\,(1 - \mathfrak{F}\,(\zeta\,(N)))\,]$
For each input feature of the discriminator $\boldsymbol{e}$ **do**
Evaluate $min_G\, max_D\, T\,(\zeta, \mathfrak{F})$
Activate the hidden neurons using ELU
If $(\boldsymbol{e < 0})\{$
Compute $\zeta\,(exp\,(e) - 1)$
$\}$ **Else** $\{$
Compute output with e
$\}$
End If
End For
Return $\mathfrak{F}^*\,(e)$ *or* a_v
End

3.7. Saliency score assignment

After all of the sentences are ranked for each sentence a_v, a saliency score is assigned that is grounded on the number of keywords that are found in sentence a_v. The saliency score assignment is given as follows:

$$ss\,(a_v) = \begin{cases} 0 & \text{if } k = 0 \\ V & \text{if } k > 0 \end{cases}, \tag{21}$$

where $ss\,(a_v)$ exemplifies the saliency score of a_v, V exhibits the saliency score based on keywords, and κ epitomizes the keywords. By performing this, each sentence carries a unique score value. Thereafter, the sentences with high saliency scores are only considered for summary, and the sentences with $ss\,(a_v) = 0$ are neglected.

3.8. Sentence sorting

After the sentences with high saliency scores are extracted, there is a possibility that the sentences that comprise the same saliency score are the same sentences that might repeat during the summarization. To resolve this problem, the sentences are re-ranked and sorted in reverse order by utilizing the rear-entry xtreme tree (REXT) sorting algorithm. In the REXT algorithm, each sentence is considered as the node of a tree, which is mathematically expressed as follows:

$$G = \{y_1, y_2, ..., y_{nn}\} \ or \ y_\tau, \tau = 1, 2, ..., nn. \tag{22}$$

Here, G is the tree (text file), and y_{nn} indicates the nn^{th} node (sentence) of G. The score value is calculated to rank the sentences, which can be represented as follows:

$$\sigma\,(y_\tau) = \frac{1-\varkappa}{nn} + \varkappa \sum_{\tau=\tau-1}^{nn} \varpi_{(\tau-1,\tau)} \sigma\,(y_{\tau-1}), \tag{23}$$

where nn describes the total number of nodes, $\aleph$ signifies the constant damping factor, $\sigma\,()$ specifies a score value, and $\varpi_{\tau,\tau-1}$ implies the weight of the edge directing from node $y_{\tau-1}$ to node y_τ. This ranking process is continued until all of the nodes are ranked. If the re-ranked values are not very similar to other nodes, the sentences are added to the summary. Then, the nodes are sorted in reverse order with $\sigma\,(y_\tau)$. If similar ranks are found, one of the similar-ranked sentences is eliminated; this solves the redundancy problem. Hence, the sorted document (sd) is represented as follows:

$$sd = \{\mathfrak{R}_1, \mathfrak{R}_2, ..., \mathfrak{R}_W\} \ or \ \mathfrak{R}_\Omega, \tag{24}$$

where $\mathfrak{R}_W$ implies the W^{th} sorted sentence.

3.9. Cosine-similarity score

The sorted sentences are then assigned with a cosine-similarity score, which aids in determining the content relevance of the sentence. Cosine-similarity score (μ) with cosine-similarity score function $\nabla\,()$ are calculated as follows:

$$\mu = \nabla\,(\mathfrak{R}_\Omega, \mathfrak{R}_{\Omega+1}) = \frac{\mathfrak{R}_\Omega . \mathfrak{R}_{\Omega+1}}{\|\mathfrak{R}_\Omega\| \,.\, \|\mathfrak{R}_{\Omega+1}\|}. \tag{25}$$

3.10. Summary generation

After obtaining the cosine-similarity score, the index with a non-zero similarity score is given to Ad-DL-BERT to obtain an extractive summarized output. Owing to reliable

sentence embedding, BERT's [18] performance gives better-summarized results than the other algorithms. In the BERT model, however, processing with more layers is a time-consuming problem; hence, a dropout layer is added (which removes the unwanted layers in BERT). This removal and adding of layers is named Ad-DL-BERT, and the working steps of Ad-DL-BERT are given in Figure 2.

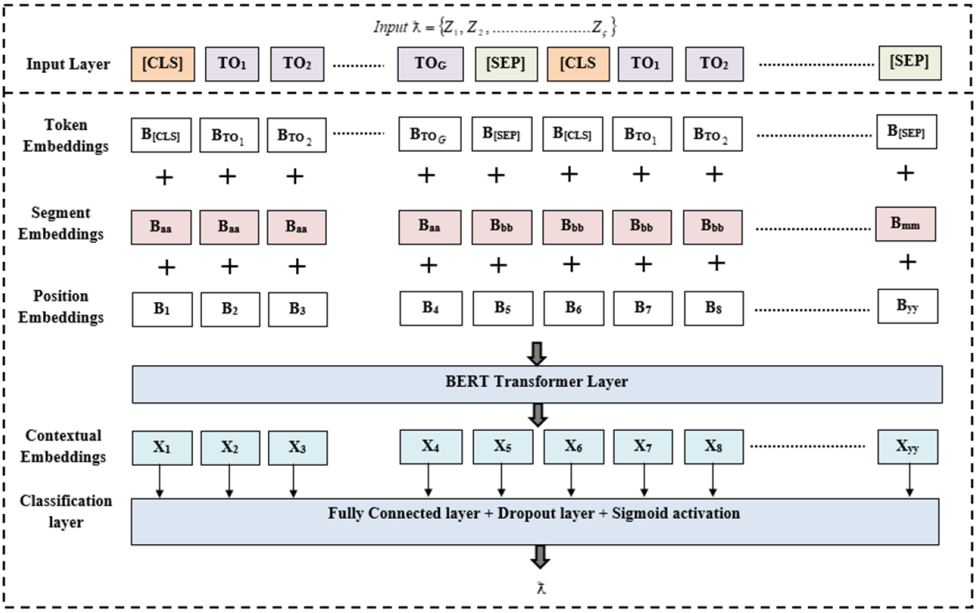

Figure 2. Ad-DL-BERT model

To summarize the scored sentences, Ad-DL-BERT predicts the next sentence in the sequence automatically from left to right or right to left. Ad-DL-BERT is comprised of the input layer, embedding layer, BERT transformer, contextual embedding layer, and output layer.

Input: Ad-DL-BERT's input layer is the cosine-similarity score's predicted sentences; these are represented as follows:

$$\lambda = \{Z_1, Z_2, ..., Z_\zeta\} \quad or \quad Z_{cc}, \tag{26}$$

where λ implies the input document, and the final sentence in the document is signified as Z_ς. In the input layer, the tokens of each word of a sentence Z_{cc} are determined and represented within the different classes; these are symbolized as $[CLS]$, and each $[CLS]$ is separated by separation function $[SEP]$. The token of Z_{cc} is signified as follows:

$$Z_{cc} = [To_1, To_2, ..., To_G]\,. \tag{27}$$

Embedding layers: The tokens from the input layer are given to the embedding layer, which performs the token embedding, segment embedding, and position embedding. Embeddings are the representations of words in vector form.
(i) Token embedding: Here, the words are transformed into a fixed dimensional vector with $[CLS]$, and $[SEP]$ is added to the beginnings and ends of the sentences.
(ii) Segment embedding: The segment embedding for a word is useful in classifying the different inputs with binary coding.
(iii) Position embedding: Position embedding is utilized for differentiating the contextual meaning of a word in sentences because the position of the word differs from the meanings of the sentences.
BERT transformer encoder layer: The token, segment, and position embeddings are summed up and given to the BERT transformer encoder layer, which is comprised of 12 transformers with 122 attention mechanisms and millions of parameters. The transformers are a combination of a set of encoders and decoders. Different attention layers are encompassed in the encoders and decoders. The encoder encodes the words, and the decoder determines the significant keywords and gives contextual embeddings $\{X_1, X_2, ..., X_{yy}\}$.
Summarization (classification) layers: Here, any strong links between the sentences are determined, which aids in the summarization. The summarization layer is comprised of a simple classifier model, where the dropout layer is added with the classifier's fully connected layer (which deletes the insignificant hidden layer neurons), and the output is predicted at the sigmoid output layer. The dropout regularization and sigmoid output layer are signified as follows:

$$DL = \frac{1}{2}\left(tar - \sum_{in=1}^{\varsigma} \psi_{in} \upsilon_{in} \vartheta_{in}\right), \tag{28}$$

where DL indicates the dropout regularization, tar illustrates the target output score, ψ_{in} epitomizes the dropout rate of the neuron (sentence) in, υ_{in} elucidates the weight value of the neuron, and ϑ_{in} exemplifies the incoming neuron. With DL, the neurons of the hidden layers are regularized, and the output sentences of the dropout layer are represented as ξ_{wd}. Afterwards, the scores of the sentences in the document are estimated in the sigmoid output layer, and the high-scored sentences are given as summarized output; this is expressed as follows:

$$\hat{\lambda} = \phi\left(\upsilon_{out} \xi_{wd}\right), \tag{29}$$

where $\widehat{\lambda}$ implies the summarized output score value, and $\phi()$ specifies the sigmoid function.

4. Results and discussions

This section discusses the outcomes that were obtained during the experimental evaluation. The experiments were performed on the working platform of Python. The link

for the data set that was used is as follows: https://ieee-dataport.org/documents/sentence-embeddings-document-sets-duc-2002-summarization-task.

4.1. Data set description

The proposed system's experiments were conducted on the DUC2002 data set. DUC2002 was comprised of 59 clusters that contained 567 English documents of news reports. Figure 4 elucidates the extractive summarization that was obtained on the DUC2002 data set (from Figure 3) with Ad-DL-BERT.

the cardigan welsh corgi is one of two separate dog breeds known as welsh corgis that originated in wales , the other being the pembroke welsh corgi . it is one of the oldest herding breeds .
cardigan welsh corgis can be extremely loyal family dogs . they are able to live in a variety of settings , from apartments to farms . for their size , however , they need a surprising amount of daily physical and mental stimulation . cardigans are a very versatile breed and a wonderful family companion .
pembrokes and cardigans first appeared together in dog shows in 1925 when they were shown under the rules of the kennel club in britain . the corgi club was founded in december , 1925 in carmarthen in south wales . it is reported that the local members favored the pembroke breed , so a club for cardigan enthusiasts was founded a year later -lrb- 1926 - rrb- . both groups have worked hard to ensure the appearance and type of breed are standardized through careful selective breeding . pembrokes and cardigans were officially recognized by the kennel club in 1928 and were lumped together under the heading welsh corgis . in 1934 , the two breeds were recognized individually and shown separately .
cardigans are said to originate from the teckel family of dogs , which also produced dachshunds .
they are among the oldest of all herding breeds , believed to have been in existence in wales for over 3,000 years .
there is an old folktale that says that queen victoria was traveling down a country road one day until her carriage came up on a fallen tree branch . while wondering how she would get across , a fairy came out of nowhere and , in order to assist the queen , produced two corgis out of thin air . one was the pembroke welsh corgi and the other the cardigan welsh corgi . the two corgis moved the tree for the queen , and they say that is why the breed is currently prized by the british queen , elizabeth ii .
another old folktale features a cardigan welsh corgi battling an ancient dragon .
cardigans have never had the same popularity as pembrokes , probably due to the influence of the royal family . however , they have found their own place in many parts of the world .
cardigan welsh corgis can compete in dog sports also known as dog agility trials , obedience , showmanship , flyball , tracking , and herding events .
the phrase `` cor gi '' is sometimes translated as `` dwarf dog '' in welsh . the breed was often called `` yard-long dogs '' in older times . today 's name comes from their area of origin : ceredigion in wales .
modern breed .

Figure 3. Input document

the cardigan welsh corgi is one of two separate dog breeds known as welsh corgis that originated in wales , the other being the pembroke welsh corgi . it is reported that the local members favored the pembroke breed , so a club for cardigan enthusiasts was founded a year later -lrb- 1926 -rrb- . pembrokes and cardigans were officially recognized by the kennel club in 1928 and were lumped together under the heading welsh corgis . cardigans are said to originate from the teckel family of dogs , which also produced dachshunds . while wondering how she would get across , a fairy came out of nowhere and , in order to assist the queen , produced two corgis out of thin air . the two corgis moved the tree for the queen , and they say that is why the breed is currently prized by the british queen , elizabeth ii . cardigan welsh corgis can compete in dog sports also known as dog agility trials , obedience , showmanship , flyball , tracking , and herding events .

Figure 4. Summarized output

4.2. Performance analysis

For proving the proposed technique's superiority over the prevailing approaches, the outcomes of the models were evaluated in three segments: clustering, sentence ranking, and summary generation.

4.2.1. Performance analysis of summary generation

Here, proposed summary-generation model Ad-DL-BERT's performance was comparatively analyzed with conventional summarization models like BERT, BERTSUM, LSTM, and recurrent neural network (RNN) concerning the execution times and the bi-lingual evaluation understudy (BLEU) scores. Moreover, the ROUGE scores of recall and precision were evaluated. The similarity score between the predicted summarized results and the human-generated ground truth was calculated by the ROUGE score.

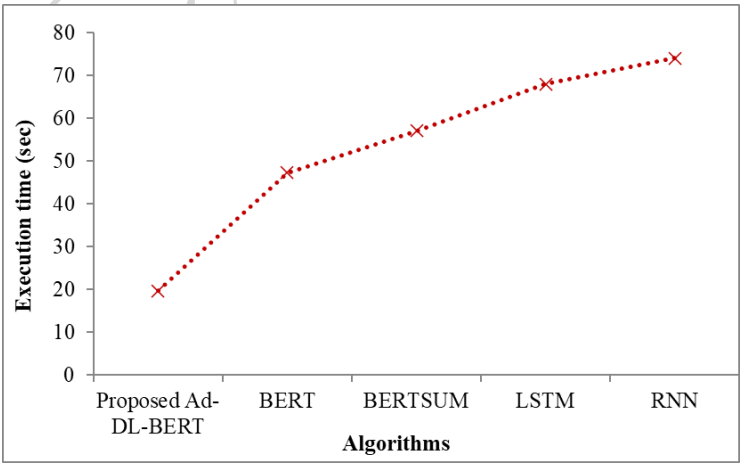

Figure 5. Graphical analysis of execution time

The time that was taken for executing the summarization process was called the execution time. Figure 5 unveils the proposed Ad-DL-BERT's execution-time analysis as compared to BERT, BERTSUM, LSTM, and RNN. Here, the time that was taken for executing the summarization task by the proposed technique was 19.75277s, which was lower than conventional BERT (47.27682s), BERTSUM (57.06456s), and RNN (74.06782s). This displayed that, as the dropout layer was added to the BERT model, the input document was summarized in less time than in the other conventional approaches.

Table 1
ROUGH-1 recall score

Algorithms	**Recall (%)**
Proposed Ad-DL-BERT	95.96736
BERT	94.83782
BERTSUM	92.38565
LSTM	88.92923
RNN	87.82121

The average recall values that were calculated for the ROUGE-1 scores are exemplified in Table 1. Here, the recall value of the existing BERT model was higher (94.83782%) among the existing algorithms. Still, the added dropout layer in the existing BERT increased the recall value by 1.191% more than conventional BERT. With Ad-DL-BERT, the summarization could therefore be done efficiently after the anaphora problem was solved in the proposed technique.

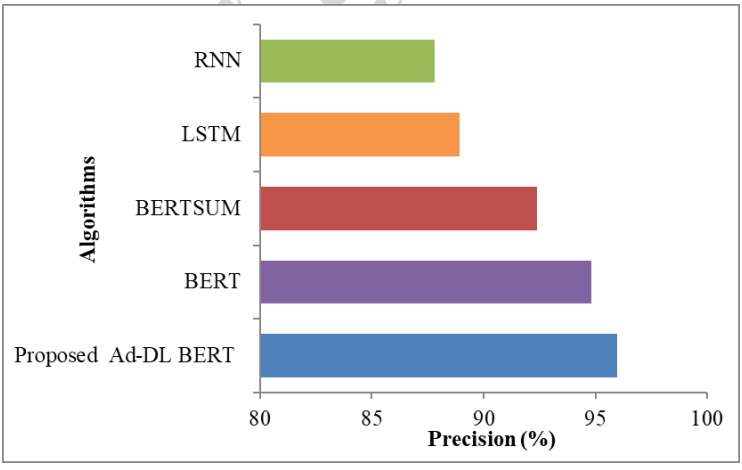

Figure 6. Precision level for ROUGE-1 score

Figure 6 displays the average precision values of the proposed summarization and baseline summarization techniques for the ROUGE-1 score on the DUC-2002 data set.

The pictorial representation shows that the precision value of RNN (87.821%) made it less reliable for the summarization model. However, the highest precision value of the proposed Ad-DL-BERT (95.967%) made it more reliable for the proposed ATS model.

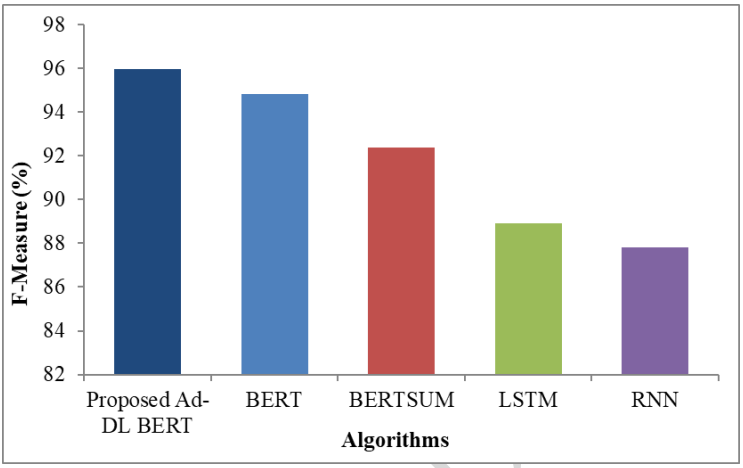

Figure 7. F-measure analysis

Grounded on the recall and precision values, the F-measure was calculated; the calculated F-measure result is pictorially epitomized in Figure 7. Here, the poor F-measure for the ROUGE-1 score was obtained by RNN among the existing techniques, followed by the LSTM and BERTSUM models. Yet, the proposed model achieved 9.35, 7.914, and 3.876% higher F-measure levels, respectively; this exhibited the quality of the summary that was generated by the proposed Ad-DL-BERT.

The similarity between the automatically predicted summary and the reference manually generated summary was measured by the BLEU score. The BLEU score values of the proposed Ad-DL-BERT and the conventional BERT, BERTSUM, LSTM, and RNN models are depicted in Figure 8. Here, the BLEU score that was obtained by the proposed Ad-DL-BERT was 0.904, which was higher than that of traditional BERT (0.812), BERTSUM (0.738), LSTM (0.667), and RNN (0.452). This increased value of the proposed Ad-DL-BERT showed the proposed technique's superiority.

4.2.2. Performance analysis of clustering

Regarding the clustering time and clustering accuracy, proposed clustering model ATDS's performance was assessed in comparison with the conventional auto-encoder clustering (AEC), k-means algorithm (KMA), fuzzy c-means (FCM), and Clustering LARge Applications (CLARA) techniques.

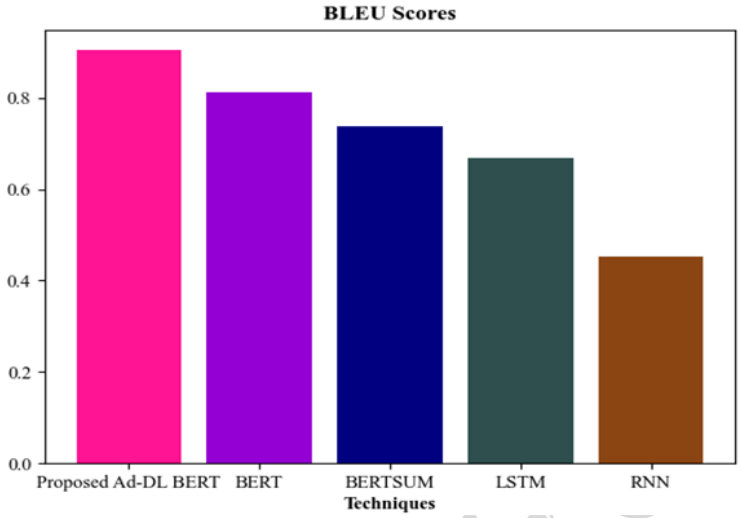

Figure 8. BLEU score analysis

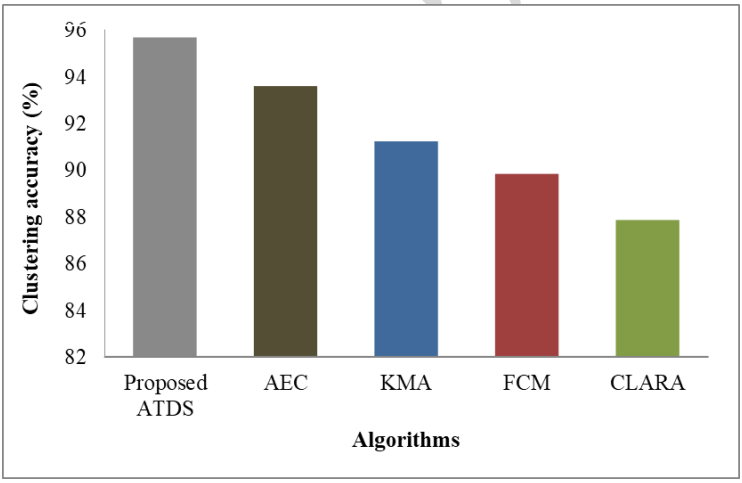

Figure 9. Analysis of clustering accuracy

The clustering accuracy results of the proposed ATDS approach and the traditional clustering approaches like AEC, KMA, FCM, and CLARA are exemplified in Figure 9. In the topic model, the results were given to the AEC model, which increased the clustering accuracy by 2.22, 4.87, 6.50, and 8.90% more than the AEC, KMA, FCM, and CLARA techniques, correspondingly. With the proposed ATDS approach, the sentences under the corresponding topics were, therefore, clustered more accurately in the proposed model than in the existing approaches.

Table 2
Clustering-time evaluation

Algorithms	Clustering time (seconds)
Proposed ATDS	125.1461
AEC	164.4896
KMA	173.1130
FCM	186.1473
CLARA	197.8727

An analysis of the clustering time is demonstrated in Table 2. The time that was taken for clustering those sentences that were relevant to the corresponding topic is called the clustering time. Here, the clustering time of the proposed ATDS was 39.3436, 61.0012, and 72.7266 s lower than the prevailing AEC, FCM, and CLARA techniques respectively. This lowest amount of time that was taken by the proposed ATDS scheme proved the time efficiency during the clustering.

4.2.3. Performance analysis of sentence ranking

This segment analyzes the sentence ranking performance of the proposed ELU-GAN and the conventional GAN, RankNet, LambdaRank, and LambdaMart techniques grounded on the sentence-ranking time.

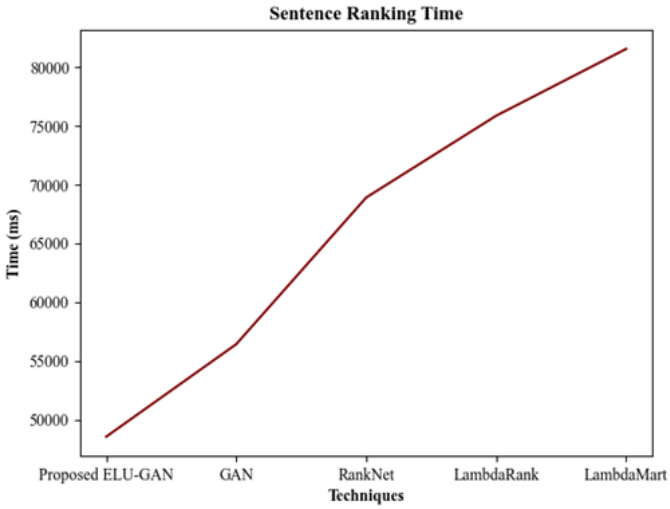

Figure 10. Time analysis for sentence ranking

The time that was taken by a system to rank the sentences based on the number of keywords is called the sentence-ranking time. Figure 10 elucidates a graphical

representation of the time that was taken to rank the sentences by the proposed and existing approaches. As per the graph, the time that was taken by the proposed ELU-GAN was 48,567 ms, which was 12,864, 25,337, 32,295, and 37,980 ms less than the existing ranking techniques (GAN, RankNet, LambdaRank, and LambdaMart), respectively. This lowest value that was obtained by the proposed ELU-GAN (owing to the embedding of ELU in the GAN model) proved the time efficiency of the proposed ranking model.

4.3. Comparative analysis with related works

Here, the proposed Ad-DL-BERT's ROUGE-1 recall score was compared with the existing works of [23], [16], and [24].

Table 3
Comparative analysis of ROUGE-1 recall score on DUC-2002 data set

Works	**Recall (%)**
Proposed Ad-DL-BERT	95.96
[24]	67.33
[16]	48.39
[23]	48.80

The ROUGE-1 recall score of the proposed Ad-DL-BERT summarization model and the existing works on the DUC-2002 data set are exemplified in Table 3. The proposed summarization model was efficiently trained with anaphora-extracted features with ATDS clustering and ELU-GAN-ranked sentences, which improved the ROUGE-1 recall score to 95.96%. For the works of [24], [16], and [23], the ROUGE-1 recall scores were 67.33, 48.39, and 48.80%. This showed the reliability of the proposed Ad-DL-BERT model for ATS and proved the quality of the summarized text by Ad-DL-BERT.

5. Conclusion

This paper proposed a novel ETS approach with Ad-DL-BERT based on An-POS extracted features. The sentences were clustered by ATDS clustering; also, ELU-GAN was used for the sentence ranking. The proposed techniques' performances were estimated on the DUC-2002 data set. As per the experimental outcomes, grouping the sentences that were grounded on topic was minimized to 39.3436 s with the proposed ATDS approach, and the clustering accuracy was maximized to 95.68%. With the proposed Ad-DL-BERT model, the ROUGE-1 recall and precision scores were increased to 95.96%, and the execution time was reduced to 19.75277 seconds (which was lower than the existing BERT, BERTSUM, LSTM, and RNN techniques). This proved that Ad-DL-BERT gave a more accurate summary than any of the other algorithmsdid. Moreover, the reliability of the proposed technique for ATS was confirmed

by the comparative assessment of the ROUGH-1 recall score. Hence, the overall analysis proved that the proposed approaches were most suitable for extractive automatic single DS. The proposed system was evaluated with English language documents; thus, the proposed technique can also be extended for other languages in the future.

Acknowledgements

We thank the anonymous referees for their useful suggestions.

References

[1] Abdelwahab M.Y., Al Moaiad Y., Bakar Z.B.A.: Arabic Text Summarization using Pre-Processing Methodologies and Techniques, *Asia-Pacific Journal of Information Technology & Multimedia*, vol. 12(1), pp. 70–110, 2023. doi: 10.17576/apjitm-2023-1201-05.

[2] Adhikari S., *et al.*: NLP based machine learning approaches for text summarization. In: *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 535–538, IEEE, 2020. doi: 10.1109/iccmc48092.2020.iccmc-00099.

[3] Alami N., Meknassi M., En-nahnahi N., El Adlouni Y., Ammor O.: Unsupervised neural networks for automatic Arabic text summarization using document clustering and topic modeling, *Expert Systems with Applications*, vol. 172, p. 114652, 2021. doi: 10.1016/j.eswa.2021.114652.

[4] Alqaisi R., Ghanem W., Qaroush A.: Extractive multi-document Arabic text summarization using evolutionary multi-objective optimization with K-medoid clustering, *IEEE Access*, vol. 8, pp. 228206–228224, 2020. doi: 10.1109/access.2020.3046494.

[5] Dave H., Jaswal S.: Multiple text document summarization system using hybrid summarization technique. In: *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, pp. 804–808, IEEE, 2015. doi: 10.1109/ngct.2015.7375231.

[6] Debnath D., Das R., Pakray P.: Extractive single document summarization using an archive-based micro genetic-2. In: *2020 7th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, pp. 244–248, IEEE, 2020. doi: 10.1109/iscmi51676.2020.9311571.

[7] Debnath D., Das R., Pakray P.: Extractive single document summarization using multi-objective modified cat swarm optimization approach: ESDS-MCSO, *Neural Computing and Applications*, pp. 1–16, 2021. doi: 10.1007/s00521-021-06337-4.

[8] Debnath D., Das R., Pakray P.: Single document text summarization addressed with a cat swarm optimization approach, *Applied Intelligence*, vol. 53(10), pp. 12268–12287, 2023.

[9] Dilawari A., Khan M.U.G., Saleem S., Shaikh F.S., *et al.*: Neural attention model for abstractive text summarization using linguistic feature space, *IEEE Access*, vol. 11, pp. 23557–23564, 2023. doi: 10.1109/access.2023.3249783.

[10] Elbarougy R., Behery G., El Khatib A.: Extractive Arabic text summarization using modified PageRank algorithm, *Egyptian Informatics Journal*, vol. 21(2), pp. 73–81, 2020. doi: 10.1016/j.eij.2019.11.001.

[11] Ghodratnama S., Beheshti A., Zakershahrak M., Sobhanmanesh F.: Extractive document summarization based on dynamic feature space mapping, *IEEE Access*, vol. 8, pp. 139084–139095, 2020. doi: 10.1109/access.2020.3012539.

[12] Gupta H., Patel M.: Method of text summarization using LSA and sentence based topic modelling with Bert. In: *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, pp. 511–517, IEEE, 2021. doi: 10.1109/icais50930.2021.9395976.

[13] Hassan A.Q., Al-onazi B.B., Maashi M., Darem A.A., Abunadi I., Mahmud A.: Enhancing extractive text summarization using natural language processing with an optimal deep learning model, *AIMS Mathematics*, vol. 9(5), pp. 12588–12609, 2024. doi: 10.3934/math.2024616.

[14] Hernández-Castañeda Á., García-Hernández R.A., Ledeneva Y.: Toward the automatic generation of an objective function for extractive text summarization, *IEEE Access*, vol. 11, pp. 51455–51464, 2023. doi: 10.1109/access.2023.3279101.

[15] Jugran S., Kumar A., Tyagi B.S., Anand V.: Extractive automatic text summarization using SpaCy in Python & NLP. In: *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pp. 582–585, IEEE, 2021. doi: 10.1109/icacite51222.2021.9404712.

[16] Khademi M.E., Fakhredanesh M.: Persian automatic text summarization based on named entity recognition, *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, pp. 1–12, 2020. doi: 10.1007/s40998-020-00352-2.

[17] Lamsiyah S., El Mahdaouy A., Espinasse B., Ouatik S.E.A.: An unsupervised method for extractive multi-document summarization based on centroid approach and sentence embeddings, *Expert Systems with Applications*, vol. 167, p. 114152, 2021. doi: 10.1016/j.eswa.2020.114152.

[18] Miller D.: Leveraging BERT for extractive text summarization on lectures, *arXiv preprint arXiv:190604165*, 2019.

[19] Mitra M., Singhal A., Buckley C.: Automatic text summarization by paragraph extraction. In: *Intelligent scalable text summarization*, 1997.

[20] Mojrian M., Mirroshandel S.A.: A novel extractive multi-document text summarization system using quantum-inspired genetic algorithm: MTSQIGA, *Expert Systems with Applications*, vol. 171, p. 114555, 2021. doi: 10.1016/j.eswa.2020.114555.

[21] Rani R., Lobiyal D.: An extractive text summarization approach using tagged-LDA based topic modeling, *Multimedia Tools and Applications*, vol. 80, pp. 3275–3305, 2021.

[22] Roul R.K.: Topic modeling combined with classification technique for extractive multi-document text summarization, *Soft Computing*, vol. 25, pp. 1113–1127, 2021.

[23] Sanchez-Gomez J.M., Vega-Rodríguez M.A., Pérez C.J.: The impact of term-weighting schemes and similarity measures on extractive multi-document text summarization, *Expert Systems with Applications*, vol. 169, p. 114510, 2021. doi: 10.1016/j.eswa.2020.114510.

[24] Sihombing J.J., Arnita A., Al Idrus S.I., Niska D.Y.: Implementation of text summarization on indonesian scientific articles using textrank algorithm with TF-IDF web-based, *Journal of Soft Computing Exploration*, vol. 5(3), pp. 310–319, 2024. doi: 10.52465/joscex.v5i3.475.

[25] Srikanth A., Umasankar A.S., Thanu S., Nirmala S.J.: Extractive text summarization using dynamic clustering and co-reference on BERT. In: *2020 5th International Conference on Computing, Communication and Security (ICCCS)*, pp. 1–5, IEEE, 2020. doi: 10.1109/icccs49678.2020.9277220.

[26] Tomer M., Kumar M.: Improving text summarization using ensembled approach based on fuzzy with LSTM, *Arabian Journal for Science and Engineering*, vol. 45, pp. 10743–10754, 2020. doi: 10.1007/s13369-020-04827-6.

[27] Tomer M., Kumar M.: Multi-document extractive text summarization based on firefly algorithm, *Journal of King Saud University-Computer and Information Sciences*, vol. 34(8), pp. 6057–6065, 2022. doi: 10.1016/j.jksuci.2021.04.004.

[28] Uçkan T., Karcı A.: Extractive multi-document text summarization based on graph independent sets, *Egyptian Informatics Journal*, vol. 21(3), pp. 145–157, 2020. doi: 10.1016/j.eij.2019.12.002.

[29] Widyassari A.P., Rustad S., Shidik G.F., Noersasongko E., Syukur A., Affandy A., Ignatius Moses Setiadi D.R.: Review of automatic text summarization techniques & methods, *Journal of King Saud University-Computer and Information Sciences*, vol. 34(4), pp. 1029–1046, 2022. doi: 10.1016/j.jksuci.2020.05.006.

Affiliations

Sunil Upadhyay
Amity University MP, Gwalior, sunilupadhyay27@gmail.com

Hemant Kumar Soni
Amity University MP, Gwalior, hksoni@gwa.amity.edu

Received: 11.06.2024
Revised: 22.09.2024
Accepted: 25.09.2025