Automatic Extractive Summarization using GAN Boosted by DistilBERT Word Embedding and Transductive Learning

Dongliang Li*, Youyou Li, Zhigang ZHANG
College of Artificial Intelligence, Jiaozuo University, Jiaozuo, 454000, China

***Abstract*—Text summarization is crucial in diverse fields such as engineering and healthcare, greatly enhancing time and cost efficiency. This study introduces an innovative extractive text summarization approach utilizing a Generative Adversarial Network (GAN), Transductive Long Short-Term Memory (TLSTM), and DistilBERT word embedding. DistilBERT, a streamlined BERT variant, offers significant size reduction (approximately 40%), while maintaining 97% of language comprehension capabilities and achieving a 60% speed increase. These benefits are realized through knowledge distillation during pre-training. Our methodology uses GANs, consisting of the generator and discriminator networks, built primarily using TLSTM - an expert at decoding temporal nuances in timeseries prediction. For more effective model fitting, transductive learning is employed, assigning higher weights to samples nearer to the test point. The generator evaluates the probability of each sentence for inclusion in the summary, and the discriminator critically examines the generated summary. This reciprocal relationship fosters a dynamic iterative process, generating top-tier summaries. To train the discriminator efficiently, a unique loss function is proposed, incorporating multiple factors such as the generator's output, actual document summaries, and artificially created summaries. This strategy motivates the generator to experiment with diverse sentence combinations, generating summaries that meet high-quality and coherence standards. Our model's effectiveness was tested on the widely accepted CNN/Daily Mail dataset, a benchmark for summarization tasks. According to the ROUGE metric, our experiments demonstrate that our model outperforms existing models in terms of summarization quality and efficiency.**

***Keywords*—Extractive text summarization; generative adversarial network; transductive learning; long short-term memory; DistilBERT**

I. Introduction

In the digital era, there is an overwhelming amount of online information. Manually extracting insights from this vast data is challenging. Automatic Text Summarization (ATS) is a solution that extracts essential details efficiently. Summarization involves creating a concise version of text from one or multiple sources, capturing the main information for specific users or purposes [1].

There is a plethora of approaches for extractive summarization. Some lean on machine learning techniques such as support vector machines [2] and clustering [3], optimization algorithms [4-7], while others adopt graph-based strategies [8], where sentences are portrayed as graphs, the nodes represent words, and edges signify the relationship between those words. As deep learning evolves, it is becoming more dominant in natural language processing, overshadowing conventional machine learning techniques. Thanks to its complex architecture, deep learning autonomously discerns word, sentence, or document attributes. However, even with numerous deep learning-driven summarization techniques, many grapple with extractive summarization intricacies. Crucial aspects of summarization, like sentence evaluation and choice, often present hurdles. Many existing methods tend to be overly selective, meaning after picking a valuable sentence, they might overlook its relevance in subsequent selections, reducing the overall efficacy of the summary.

Generative Adversarial Networks (GANs) are advanced machine learning tools that consist of a generator and a discriminator. The generator strives to create lifelike outcomes, such as images or text. In contrast, the discriminator tries to discern between genuine and fabricated content [9]. GANs hold potential for optimizing sentence selection in extractive text summarization. When generating a summary, they evaluate the entirety of the document, giving the generator a holistic grasp. Such understanding helps the generator pick subsequent sentences more judiciously, recalling previously identified important sentences and ensuring better summary quality. Adversarial Training significantly boosts the GANs' capability to address the issue of biased sentence selection. The discriminator offers critical insights into the generator about the cohesiveness and quality of the formed summary, allowing the generator to refine its methods. This results in selecting impactful sentences that also harmonize with earlier pivotal sentences. Thanks to adversarial training, GANs adeptly address the risk of omitting key sentences, delivering more unified summaries.

LSTM has been fundamental in numerous sequence-oriented tasks, such as video categorization, machine interpretation, and text summarization [10, 11]. However, standard LSTMs, rooted in inductive learning, shape a universal model from all training datasets. This can sometimes neglect nuances within the data, potentially hampering the adaptability of the model. TLSTM introduces a solution by incorporating a transductive learning method. This approach emphasizes performance improvement around novel data points, accentuating localized nuances. In the structure of TLSTM, this is accomplished through a tailored weighting system, adjusting weights concerning their closeness to the assessment data. Closest data points to the test get a higher weightage, ensuring optimal performance in those areas. By combining the strengths of LSTM with the adaptability of transductive learning, TLSTM offers a more tailored time-series prediction method. It captures long-term dependencies while addressing overlooked data nuances, making it suitable for intricate time-series challenges [12].

The BERT model in [13] is a notable NLP tool with many parameters. Larger models, while effective, increase computational and environmental costs. DistilBERT [14], a streamlined transformer model, is a distilled version of BERT. It functions 60% swifter and requires 40% less parameters compared to BERT, as evidenced in the GLUE benchmark. In comparison with predecessors like BERT and RoBERTa [15], DistilBERT is a more streamlined variant.

The paper presents an extractive summarizer, founded on DistilBERT word embedding, GAN, and attention mechanism-based TLSTM. GANs are made up of two generator and discriminator components that compete in a process. In this context, the generator's goal is to rate each sentence of the document, while the goal of the discriminator is to distinguish the real from the fake summary, which enhances the performance of the generator. In a non-greedy way, the generator determines the possibility of the presence of sentences in summary at once. The contributions of this article are as follows:

- Using GAN for summarization, the generator improves based on feedback from the discriminator, incorporating both real and fake summaries.
- We use TLSTM to design the generator and discriminator, enhancing accuracy in text summarization.
- By introducing varied noise levels during training and testing, we produce diverse summaries, with a voting system determining the final summary.
- Our proposed model utilizes DistilBERT word embedding to automatically learn and extract complex and meaningful text representations from the input data.

The remainder of the paper is structured as following. Section II covers some related works, while Section III introduces our proposed text summarization method. Section IV presents experimental results, and Section V concludes the paper.

II. Related Works

Abstractive summarization methods, a notable strategy in NLP, aim to produce summaries that don't merely pick and reorganize existing sentences or phrases [16]. These techniques endeavor to grasp the essence of the text and formulate new, succinct, and cohesive statements that reflect the main ideas of the original content [17]. Abstractive summarization seeks to produce summaries with a human-like touch, capturing the heart of the source material without restricting itself to direct extractions. By discerning the core semantics, connections, and subtleties of the document, abstractive methods can potentially craft summaries that are richer, more concise, and linguistically smooth. To realize this, such techniques frequently utilize advanced tools like neural networks and natural language generation models [18, 19]. These models employ methods such as sequence-to-sequence frameworks, attention systems, and reinforcement learning to craft summaries that hold semantic significance and flow smoothly. By grasping the underlying context and essence of the text, abstractive summarization models can reword and restructure the original material, introducing fresh phrases, reshaping statements, and even creating unique expressions to highlight the primary details. This capability to transcend basic extraction allows abstractive summarization to deliver shorter summaries that still encapsulate the primary intent of the original text. Yet, this approach comes with its set of challenges. The crafted summaries must walk the fine line of being brief yet informative, ensuring logical flow and upholding the truthfulness of the source. Moreover, abstractive techniques often demand vast training data and intricate models to effectively decipher the subtleties and variances in natural language [20].

Numerous extractive summarization methods, spanning graph-based to deep learning techniques, have been explored [21, 22]. LeClair et al. [23] delved into code summarization advancements via Graph Neural Network (GNN) application, enhancing summary insightfulness. Zhong et al. [24] derived word features from documents and then determined sentence scores based on word scores. Yousefi et al. [25] scored sentences using the cosine similarity between them and their topics. Cao et al. [26] employed recurrent neural networks for sentence ranking, treating sentences as trees with words as leaves, and deriving sentence scores from a non-linear procedure. Rosca et al. [27] utilized reinforcement learning for summary creation, where sentence coherence was the reward. The policy was crafted as a multilayer perceptron assigning scores to sentences. Abdi et al. [28] showcased a deep learning methodology for creating opinion-focused multi-document summaries. This involved components like sentiment analysis embedding space (SAS), text summarization embedding spaces (TSS), and an opinion summarizer module (OSM) [29]. The SAS uses an RNN with LSTM to capture sequential data, and the TSS applies linguistic knowledge for improved word embeddings. Fitrianah and Jauhari [30] employed LSTM and GRU models in their approach for ETS summarization, leveraging feature engineering techniques. Hin et al. [31] presented LineVD, a deep learning model that identifies vulnerabilities using a Graph Neural Network (GNN) and notably omits vulnerability status in its analysis. Nallapati et al. [32] introduced Summarunner, an RNN-based model, where two RNN layers embed words and sentences, followed by logistic regression for sentence classification. Kobayashi et al. [33] proposed a method centered on embeddings and document-level similarities, representing words through embeddings, treating sentences as word collections, and documents as sentence collections. Chen et al. [34] introduced a deep reinforcement learning technique and an encoder-extractor framework for single-document summarization, extracting sentences post key feature selection [35]. Mikael et al. [36] leveraged continuous vector representations in RNN and achieved top results on the Opinosis dataset. Yin et al. [37] devised a unique sentence selection strategy ensuring a balance between sentence significance and diversity after developing an unsupervised CNN for phrase representation learning.

In recent times, there has been a transformative shift in the realm of natural language processing, largely attributed to BERT. BERT, a model based on the transformer architecture, has brought about a significant evolution in the domain of NLP by introducing contextual comprehension of words and sentences. In contrast to prior models that processed sentences in a linear manner, BERT takes into account both antecedent and subsequent words, thereby capturing a deeper insight into contextual interdependencies present within the text. This bidirectional approach empowers BERT to construct word representations that carry more profound significance, accurately reflecting their contextual applications. The integration of BERT has yielded notable enhancements across diverse NLP tasks, encompassing aspects such as text classification, identification of named entities, evaluation of sentiment, and particularly, condensing texts. By incorporating BERT into frameworks for text summarization, researchers have achieved summaries that are not only more precise but also informed by the context. BERT's proficiency in grasping linguistic intricacies and generating comprehensive portrayals has led to a paradigm shift in the way we handle and comprehend natural languages. This has, in turn, paved the way for the development of more intricate and efficient NLP applications. As influence of the BERT model continues to stimulate progress in language modeling and comprehension, it carries immense potential for further reshaping the landscape of natural language processing. Koto et al. [38] introduced techniques for probing discourse at the document level, which were utilized to detect connections between documents and appraise the performance of pre-trained language models. They employed BERT, BART, and RoBERTa as model choices to assess the outcomes derived from their assessment. In a separate study, Abdel-Salam and Rafea [39] conducted an evaluation of diverse variations of BERT-based models intended for text summarization. They introduced an unsupervised strategy for creating summaries from multiple documents, leveraging the transfer learning capabilities of the BERT sentence embedding model. The researchers adjusted the BERT model through supervised intermediate tasks extracted from GLUE benchmark datasets. This adjustment included the use of both single-task and multi-task fine-tuning methodologies to enhance the learning of sentence representations. In a different context, Srikanth et al. [40] harnessed the potential of the BERT model to produce extractive summaries through the clustering of sentence embeddings using K-means clustering. Alongside this, they introduced a dynamic approach to ascertain the suitable quantity of sentences to be chosen from the clusters.

A considerable portion of prior deep learning methods face a constraint during sentence selection. They often exhibit an inclination to excessively prioritize the selection of the sentence with the highest score, neglecting its pertinence within the context of subsequent sentence selection. As a result, this methodology contributes to a reduction in the overall excellence of the produced summary.

III. The Proposed Method

To tackle our research challenge, we have combined the strengths of DistilBERT for word embeddings and TLSTM for analyzing temporal data.

DistilBERT, a streamlined variant of BERT, excels in converting words into pertinent vectors. It mirrors BERT's bidirectional transformer architecture but is more efficient due to fewer layers, resulting in quicker computations. Notably, its training employs dynamic rather than static masking.

TLSTM excels at handling sequential data by grasping both short and long-term patterns. Its unique gating mechanism modulates data retention and recall over time.

We have also harnessed GANs for text summarization to boost extractive techniques and combat issues like greediness. Our GAN setup includes a generator, which creates synthetic data, and a discriminator that distinguishes between authentic and fabricated content. We've further enhanced our model by conditioning it on sentence features, allowing more accurate and relevant outputs. Integrating noise into the document representation lets our generator craft varied but consistently high-quality summaries.

In the following, the details of each component are explained.

A. DistilBERT-based word Embedding

The objective of word embedding is to transform words into vectors for utilization in deep learning algorithms. Word embedding has demonstrated its credibility in generating reliable vectors for words, drawing from the surrounding context. Diverse methodologies for word embedding have been introduced, each designed to produce substantial representations suitable for deep models. These approaches encompass Skip-gram [41] and matrix factorization techniques such as GloVe [42].

BERT stands as a profound bidirectional language model, capable of furnishing contextual portrayals. It is frequently subjected to fine-tuning through a hefty neural network layer tailored to various classification undertakings. Its training data encompasses extensive datasets such as Wikipedia. The initial, broadly applicable significance acquired during pre-training can be effectively harnessed to capture context- or issue-specific nuances through the process of fine-tuning, also readying it for classification purposes. BERT adopts a bidirectional transformer architecture, wherein representations are concurrently influenced by both preceding and subsequent context, spanning all tiers. This distinctive characteristic distinguishes BERT from models like GloVe and Word2Vec, which offer embeddings in a singular direction that disregards the contextual intricacies.

DistilBERT integrates the concept of information distillation, wherein a condensed system, the student, learns from a more expansive system's patterns, labeled as the teacher. The student system's learning curve is shaped by a specific loss criterion, reflecting the teacher's probability benchmarks:

$$L_{ce} = \sum_i t_i * \log\ (s_i) \qquad (1)$$

Here, t_i and s_i represent the probabilities obtained from the teacher and student techniques, respectively. DistilBERT employs a structure akin to that of BERT, yet with fewer layers. DistilBERT exhibits a 40% reduction in width, operates with 60% enhanced speed, while still retaining 97% of the performance capabilities inherent to BERT. The core aim of the distillation process lies in approximating the comprehensive output distributions of BERT by means of a more condensed model, exemplified by DistilBERT. Consequently, the quantity of layers within the BERT architecture has been curtailed from 12 to 6. The pre-trained model encompasses a total of 66 million parameters, a comparison to the 110 million presents in the BERT model. Notably, DistilBERT's training duration amounts to 3.5 GPU days (using 8 × V100), in contrast to BERT's 12 GPU days (also with 8 × V100). DistilBERT, much like BERT, undergoes training using a dataset of 16 GB sourced from English Wikipedia, specifically the Toronto books corpus. During the training process of DistilBERT, a substantial batch size is employed in conjunction with gradient accumulation. This methodology entails the local amalgamation of gradients from multiple mini-batches prior to the modification of trainable parameters in each phase. Additionally, the training regimen of DistilBERT does not incorporate objectives such as next-sentence prediction and segment embedding learning, which are observed in BERT training. Moreover, the dynamic masking technique employed during inference replaces the static masking mechanism used in the BERT model.

B. TLSTM

LSTM has risen in prominence as a widely embraced and potent method applied to sequence data across diverse domains. Its inherent aptitude to apprehend extended temporal relationships and manage sequential information renders it exceptionally fitting for tasks involving time series analysis and prognosis [43]. An eminent virtue of LSTM lies in its capacity to unravel intricate temporal structures and seize the fluid dynamics of systems that undergo time-driven fluctuations. By dissecting the inherent motifs and inclinations embedded within the data, LSTM architectures can unveil nuanced interconnections that may not be readily discerned using established statistical or machine learning methodologies.

LSTM networks possess the capability to grapple with input sequences of varying lengths, thereby endowing them with adaptability for scenarios involving sequence data wherein the historical data's length may fluctuate across instances [44]. This adaptative trait proves invaluable when confronted with diverse systems or equipment beset with differing operational states or maintenance schedules. Furthermore, LSTM's prowess in dealing with both brief and extensive dependencies within sequence data sets it apart. Traditional methods like autoregressive models or moving average approaches might falter in capturing prolonged trends or subtle intricacies concealed within the data fabric [45]. In contrast, LSTM's memory cells empower it to retain information over extended intervals, thereby empowering the model to apprehend dependencies spanning multiple time increments [46, 47].

The groundbreaking concept of LSTM was originally developed by Hochreiter and Schmidhuber [48]. From its genesis, an array of methods aimed at enhancing its performance have been introduced [49]. In this research, we put into practice the well-accepted architecture advanced by Gers et al. [50], a blueprint that has been leveraged in a multitude of academic endeavors, including [51, 52]. The LSTM mechanism revolves around a gating system which regulates how information is retained over time, skillfully overseeing how long it is stored and determining the appropriate moment for its access through the memory cell. This paper places particular emphasis on the scrutiny of the LSTM cell, as expounded in Graves' work [51]. LSTM employs three gates to optimize information processing. Let i_t, f_t, o_t, c_t, and h_t symbolize the input gate, forget gate, output gate, memory cell, and hidden state at the sequence time t, respectively. When x_t represents the system's input at the same time, the architecture of the LSTM cell can be described as follows [51, 53]:

$$i_t = \sigma(W_{xi}x_t + \ W_{hi}h_{t-1} + W_{ci}c_{t-1} + \ b_i) \tag{2}$$

$$f_t = \sigma\left(W_{xf}x_t + \ W_{hf}h_{t-1} + W_{cf}c_{t-1} + \ b_f\right) \tag{3}$$

$$c_t = f_t c_{t-1} + i_t tanh(W_{xc}x_t + \ W_{hc}h_{t-1} + \ b_c) \tag{4}$$

$$o_t = \sigma(W_{xo}x_t + \ W_{ho}h_{t-1} + W_{co}c_t + \ b_o) \tag{5}$$

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

The sigmoid function, σ(.), acts as an activation function. The logistic sigmoid and hyperbolic tangent are applied element-wise. Weight matrices, W_{xk} and W_{ck} , are linked to the input, forget, output gates, and memory cell. The number of neurons in these gates is preset, with Eq. (2) to Eq. (6) affecting each neuron separately. If n represents the neuron count, then $\{i_t, f_t, c_t, o_t, h_t\}$ are in $R^{n\times 1}$. In discussing the LSTM model, we use w_{lstm} and b_{lstm} for weights and biases. The LSTM equations are presented as follows:

$$\begin{cases} c_t = f(c_{t-1}, h_{t-1}, x_t; w_{lstm}, b_{lstm}) \\ h_t = g(h_{t-1}, c_{t-1}, x_t; w_{lstm}, b_{lstm} \end{cases} \tag{7}$$

Considering Eq. (2) through (6), we derive $g(\cdot)$ and $f(\cdot)$. Assuming $z(\eta)$ represents an unseen series, the state space depiction of the T-LSTM is expressed as:

$$\begin{cases} c_{t,\eta} = f(c_{t-1,\eta}, h_{t-1,\eta}, x_t; w_{lstm,\eta}, b_{lstm,\eta}) \\ h_{t,\eta} = g(h_{t-1,\eta}, c_{t-1,\eta}, x_t; w_{lstm,\eta}, b_{lstm,\eta} \end{cases} \tag{8}$$

In Eq. (8), the model structure markedly deviates from what is outlined in Eq. (7). While in Eq. (7) the model parameters remain stable irrespective of the evaluation point, in Eq. (8) they are molded by the feature vector of that specific evaluation point. The subscript η is introduced to underscore the variation in model parameters resulting from the inclusion of data point z(η). It is imperative to underscore that the evaluation label is presumed to be undisclosed. Throughout the training phase, the primary purpose of the evaluation point is to ascertain the relevance of training data points by examining the affinity between their feature vectors and the vector of the evaluation point.

C. *Model*

In this research, we use adversarial generating networks for extractive summarization. We employ this network to improve the problems of previous methods, including greed. We will first have a description of this network, and then the proposed model is presented.

Generative adversarial networks (GANs) were first proposed by Goodfellow et al. [54]. These networks consist of two separate networks that are similarly trained: the generator and discriminator networks. The purpose of the generator is to produce data such as images, text, etc., which are structurally similar to real data but are fake. On the other hand, the task of the discriminator network is to strengthen the generator.

These two networks play a two-player min-max game with a value function $V(D, G)$ as follows:

$$min_G\, max_D\, V\,(D.G) = E_{x\sim p_{data}(x)}[\log(D(x))] + E_{z\sim p_z(z)}[log(1 - D(G(z)))] \qquad (9)$$

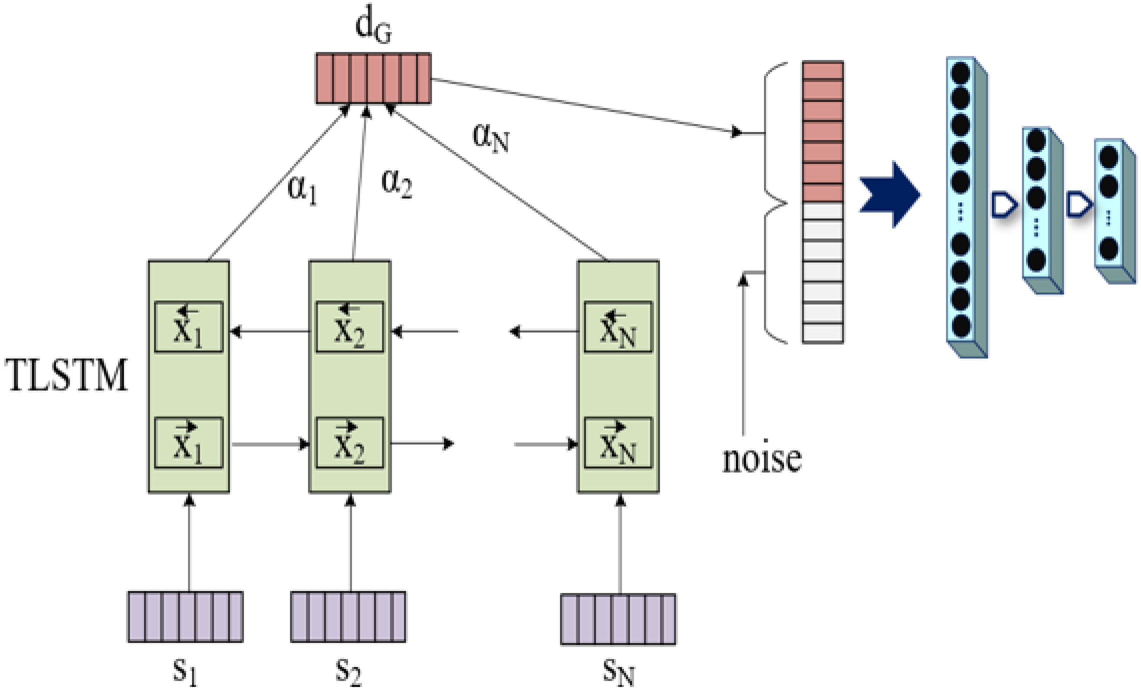


Fig. 1. Generator architecture.

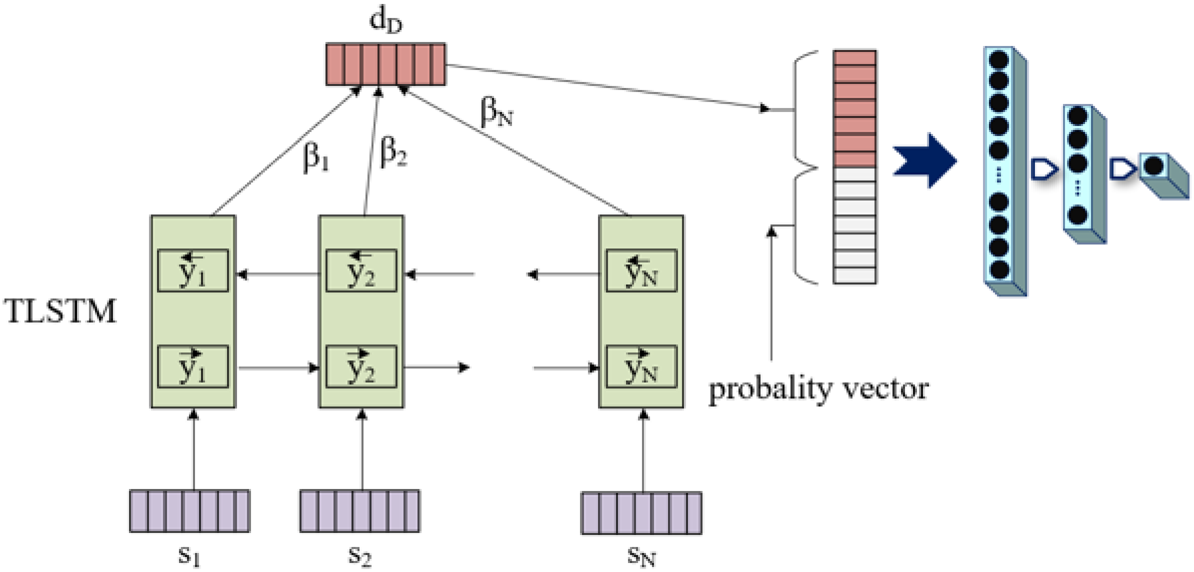


Fig. 2. Discriminator architecture.

Where x and z are input data and noise, respectively. G and D mean the generator and discriminator, respectively. $p_{data}(x)$ and $p_z(z)$ represent the input data distribution and the noise distribution, respectively. E is mathematical expectation.

Generative adversarial networks can be extended to a conditional model If condition y is added to the generator and discriminator input. The value function, in this case, changes as follows:

$$min_G\, max_D\, V\,(D.G) = E_{x\sim p_{data}(x)}[\log(D(x|\boldsymbol{y}))]+$$

$$E_{z\sim p_z(z)}[log(1 - D(G(z|\boldsymbol{y})))] \qquad (10)$$

The proposed generator and discriminator model are shown in Fig. 1 and Fig. 2, respectively. We use sentence features as a condition in the generator and discriminator. Let $D = \{s_1.s_2 \ldots. s_N\}$ represents the document, where the $s_i \in \mathbb{R}^d$ is the extracted features of the i-th sentence. N is the length of document D, which is equal to the number of restricted sentences in each document. The attention mechanism calculates the representation vector of the document in the generator and the discriminator according to the following equations:

$$d_G = \sum_{i=1}^{N} \alpha_i\, [\overleftarrow{x}_i.\vec{x}_i] \qquad (11)$$

$$d_D = \sum_{i=1}^{N} \beta_i\, [\overleftarrow{y}_i.\vec{y}_i] \qquad (12)$$

where, $\overleftarrow{x}_i \in \mathbb{R}^{d_1}$, $\overleftarrow{x}_i \in \mathbb{R}^{d_1}$, $\overleftarrow{y}_i \in \mathbb{R}^{d_2}$, $\vec{y}_i \in \mathbb{R}^{d_2}$ are the output of step i in BLSTM. α_i and β_i are the coefficients of attention for the i-th sentence in the generator and the discriminator, respectively, which are formulated as follows:

$$\alpha_i = \frac{e^{u_i}}{\sum_{i=1}^{N} e^{u_i}} \qquad (13)$$

$$\beta_i = \frac{e^{v_i}}{\sum_{i=1}^{N} e^{v_i}} \qquad (14)$$

$$u_i = tanh(W_u[\overleftarrow{x}_i.\vec{x}_i] + b_u) \qquad (15)$$

$$v_i = tanh(W_v[\overleftarrow{y}_i.\vec{y}_i] + b_v) \qquad (16)$$

where, $W_u \in \mathbb{R}^{2.d_1}. b_v \in \mathbb{R}$, $W_v \in \mathbb{R}^{2.d_2}$, and $b_v \in \mathbb{R}$ are the parameters of the attention mechanism for documents.

In Fig. 1, the document's representation vector is linked with the noise vector, entering a feed-forward neural network. The final layer of this network computes the likelihood of each sentence's presence. The introduction of noise prompts the generator to generate diverse outputs. Each document undergoes multiple iterations of summarization by the generator, with varied noises, leading to distinct outputs. The generator aims to create varied yet similarly high-quality summaries for each document. This process empowers the generator to identify diverse sentence combinations suitable for crafting the summary. Consequently, sentences that might lack individual significance for the summary can contribute to a quality summary when positioned alongside other sentences.

Within the discriminator network, the probability vector of sentences interfaces with the document's representation vector (see Fig. 2). In this context, the probability vector of sentences represents the count of sentences within a document, and each element assumes a value of either zero or one.

1) Real summary: In a typical GAN framework, the output of the generator functions as synthetic data for training the discriminator. Moreover, an authentic target is garnered for each individual sample. In this study, to acquaint the discriminator with quality summaries, more than one summary is drawn from each document, displaying similar levels of quality. Simultaneously, several summaries of inferior quality are generated for each document. Given that actual summaries are text and unsuitable as target data, a method is required to represent the presence or absence of each sentence in a summary as a numerical value. To serve this purpose, a vector with N elements is designated for every document, where N signifies the sentence count. Each element of this vector holds a value of either zero or one, with a value of one indicating the inclusion of the sentence in the summary. This vector is constructed employing a greedy approach as delineated in Fig. 3. Initially, a vector of length N with M ones is generated, where M corresponds to the sentences within the summary. The ones are distributed randomly throughout the vector. Sentences associated with a value of one are then concatenated within this vector to compose a summary, and the quality is assessed using the ROUGE metric.

Subsequently, a one is selected at random and transformed into zero, while a randomly selected zero is converted to one. The ROUGE score is recalculated, and if it surpasses the prior value, the alteration is retained. This sequence is reiterated Itr times, culminating in the selection of the most favorable vector throughout the process, which is then designated as the outcome. It is important to note that for the generation of any genuine target, the algorithm must be restarted from the beginning. The procedure for devising a synthetic target closely mirrors that of a real target, with the exception that if the ROUGE score is lower, the vector supersedes the previous one. The length of all documents is confined to N sentences, with longer documents being truncated to N sentences and shorter ones being padded with zeros.

2) Loss function: The Loss function is calculated based on the discriminator output for the generator as follows:

$$Loss_G = E_{i\sim Dataset}\left[E_{z\sim p_{z(z)}}\left[log\left(1 - D\big(G(z|y_i)\big)\right)\right]\right] \qquad (17)$$

where, *Dataset* is a set of documents, y_i is features of sentences in document i, and E is the mathematical expectation. The Loss function for the discriminator is computed based on the generator output, real and fake summaries as follows:

$$Loss_D = E_{i\sim Dataset}[\,E_{z\sim p_{z(z)}}[log(1 - D(G(z|y_i)|\,y_i))] +$$

$$E_{k\sim p_{Fake_i}}\big[log\big(1 - D(k|y_i)\big)\big] + E_{k\sim p_{Real_i}}[log(D(l|y_i))]] \qquad (18)$$

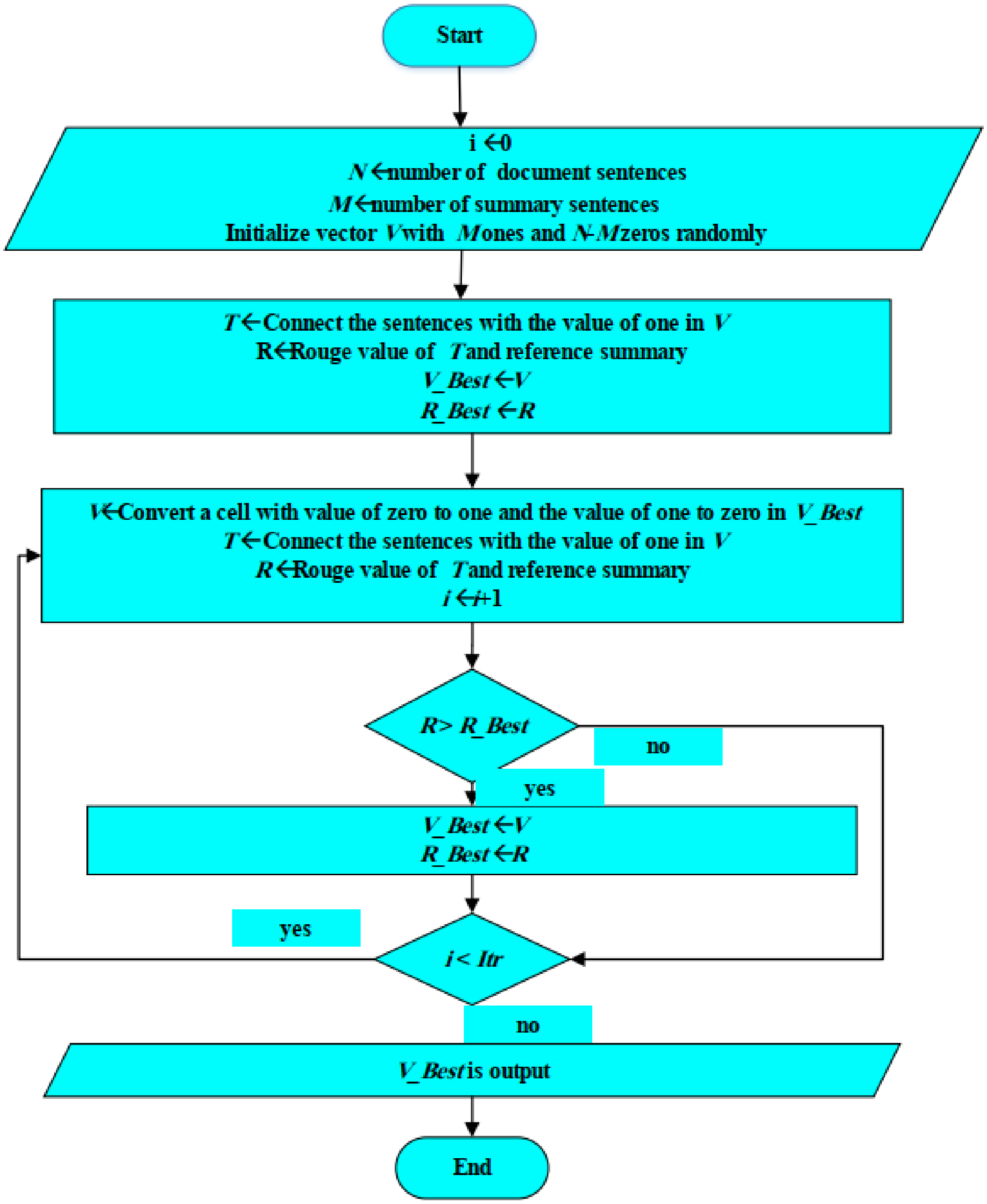

Start
i ←0
N ←number of document sentences
M ←number of summary sentences
Initialize vector V with M ones and N-M zeros randomly
T ←Connect the sentences with the value of one in V
R←Rouge value of T and reference summary
V_Best ←V
R_Best ←R
V←Convert a cell with value of zero to one and the value of one to zero in V_Best
T ←Connect the sentences with the value of one in V
R←Rouge value of T and reference summary
i ←i+1
R> R_Best
no
yes
V_Best ←V
R_Best ←R
yes
i< Itr
no
V_Best is output
End

Fig. 3. Generate a real summary for the document.

where, p_{Real_i} and p_{Fake_i} show the distribution of real and fake summaries for the document *i*. Eq. (18) forces the discriminator to learn a set of high-quality and low-quality summaries. On the other hand, be sensitive to the summaries produced by the generator and force the generator to produce a high-quality summary.

IV. Empirical Evaluation

A. Dataset

For our assessments, we employ a familiar dataset known as CNN/Daily Mail. This dataset amalgamates two distinct datasets devised for comprehension, extractive, and abstractive tasks, and it has garnered notable attention from researchers in the domain of automated summarization in recent years. The CNN/Daily Mail dataset is comprised of 287,226 documents earmarked for training, 13,368 for validation, and 11,490 for testing. Within the training data, the average document encompasses approximately 28 sentences. On average, each document's reference summary spans 3 to 4 sentences, and the mean word count per document in the training dataset is approximately 802 words [34]. You can delve into additional particulars outlined in Table I. This dataset exists in two versions: the first version features the replacement of all entities with specific words, while the second version retains the original data. For our model, we choose to adopt the second version of the dataset.

TABLE I. STATISTICS OF THE CNN / DAILY MAIL DATASET

	Train	Validation	Test
Pairs of data	287,113	13,368	11,490
Article length	749	769	778
Summary Length	55	61	58

TABLE II. THE PARAMETERS OF THE MODEL

Parameter	value
batch size	128
embedding dim	60
max sentence length	100
real summary per document	40
fake summary per document	40
activation fun (tlstm & dense)	relu
dense hidden layer	8

B. Detail of Model

For the execution of this study, the Python programming language and the PyTorch library have been harnessed for implementation purposes. The Jupyter environment has been employed as the platform to execute project codes. Additionally, the NLTK library, an instrumental component, has been utilized. This particular library furnishes an assortment of classes and methods dedicated to processing natural language within the Python context. Its capabilities span a broad spectrum of natural language processing tasks.

The model architecture incorporates a dual-layer bidirectional TLSTM structure. Within generative adversarial networks, the discriminator tends to converge at a quicker rate than the generator, often impeding the generator's convergence. In light of this, we have designed a training strategy wherein the discriminator is trained once for every 15 iterations of generator training. Moreover, due to the interconnection of vectors within the two networks, we implement batch normalization prior to data entry into the feed-forward neural network. The parameter values are detailed in Table II.

C. Metrics

We employ the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) package [55] as an evaluation metric in our experiments. This metric calculates the similarity between the generated summary and the reference summary by counting the number of common units. Rouge-n recall between an extracted summary and a reference summary is calculated as follows:

$$\text{Rouge-}n = \frac{\sum_{s\in\{ref\ sum\}} \sum_{gram_n \in s} Count_{match}\ (gram_n))}{\sum_{s\in\{ref\ sum\}} \sum_{gram_n \in s} Count\ (gram_n))} \quad (19)$$

where, n stands for the length of n-ngram, $Count_{match}\ (gram_n)$ is the maximum number of n- gram co-occurring in the extracted summary and the reference summary. Rouge-1 and Rouge-2 are special cases of Rouge-n in which n = 1 or n = 2. R-L calculates the length of the longest common subsequence between the reference summary and the extracted summary. Based on previous works, Rouge-1(R-1), Rouge-2(R-2) and, Rouge-L(R-L) are most widely used in summarization. For this reason, we use these three metrics in all our experiments.

D. Experimental Results

In the execution of our project, we utilized a Windows operating system that runs on 64-bits, accompanied by 64 GB of RAM and an integrated GPU. For the CNN/Daily Mail dataset, the optimal model emerged after running through 50 epochs. Remarkably, the entirety of our training duration spanned a mere four hours.

Our innovative approach was subjected to a comparative analysis against various methodologies. These included three methodologies rooted in graph algorithms: BGSumm [56], TextRank [57], and EdgeSumm [8]. Additionally, we compared against seven methodologies anchored in deep learning paradigms: SummaRunner [32], RENS with Coherence [58], SHANN [59], HSSAS [60], T5 [61], BART [62], and DeepSumm [63]. Lastly, a foundational model, TLSTM, was part of our comparison. It's noteworthy to mention that the TLSTM model is exclusively reliant on the generator component we devised. To visualize the assessment results for our system using the CNN/Daily Mail dataset, please refer to Table III.

TABLE III. NUMERICAL COMPARISON OF THE PROPOSED METHOD AND OTHER METHODS ON THE CNN / DAILY MAIL DATASET

Model	R-1	R-2	R-L
BGSumm	33.20	12.50	31.74
TextRank	32.16	11.10	29.21
EdgeSumm	34.26	13.10	32.90
DeepSumm	42.91	18.18	37.95
SummaRunner	39.65	16.26	35.39
RENS with Coherence	41.29	18.90	37.79
SHA-NN	35.46	14.74	33.26
HSSAS	42.32	17.81	37.65
T5	42.48	18.08	37.77
BART	36.51	15.14	31.26
TLSTM	26.46	8.48	6.25
Proposed	44.51	18.46	38.90

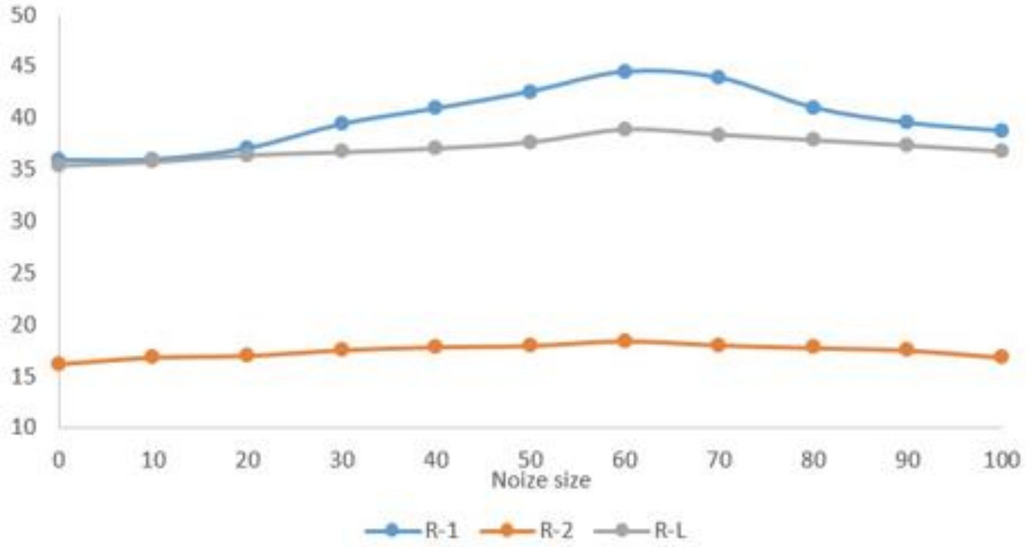


Fig. 4. Results of the proposed model for different noises on the CNN / Daily Mail dataset.

Drawing insights from the graph-centric models, the EdgeSumm model conspicuously outperformed its peers, inclusive of BGSumm, across all evaluated benchmarks. To quantify, EdgeSumm decreased errors by magnitudes of more than 33%, 32%, and 30% for the three primary metrics: R-1, R-2, and R-L, respectively. Intriguingly, even though BGSumm demonstrated its efficiency on a medical dataset, it couldn't replicate its performance for the CNN/Daily Mail dataset. Surpassing even the robust EdgeSumm model, our pioneering model showcased error enhancement rates of approximately 24.29%, 24.30%, and 25.41%. As many would anticipate, models anchored in deep learning exhibited greater efficacy than those rooted in graph algorithms. The RENS with Coherence approach, despite its integration of sentence coherence, didn't match the precision of our model. Among the pantheon of deep learning models, DeepSumm emerged as the frontrunner. However, even DeepSumm lagged behind our proposed model, registering weaker performance metrics of 25.28%, 25.39%, and 26.47%.

1) Explore noise: We undertook additional experimental trials to ascertain the impact of varying noise intensities on the generator's functionality. In these trials, we introduced noise of diverse magnitudes to the generator to observe its effect. The outcomes, specifically for metrics R-1, R-2, and R-L pertaining to the CNN/Daily Mail dataset, are graphically presented in Fig. 4. A noteworthy observation was that elevating the noise level to 60 improved the aforementioned metrics: R-1, R-2, and R-L. However, a declining trend in performance was evident when the noise level ranged between 60 and 100. From our analysis of this dataset, it appears that the optimal noise magnitude stands at 60. It is evident from the findings that the generator's efficiency is enhanced when noise is incorporated, with our proposed model displaying superior results in the presence of noise.

2) Word embedding: Word representations play a pivotal role in the realm of deep learning models. This is primarily because these models interpret input data as vectors; therefore, if there is any discrepancy or error in the embedding process, it could potentially misguide the model. In this research, we have employed the DistilBERT model for word embeddings, which is considered one of the latest advancements in this domain.

TABLE IV. RESULTS OF DIFFERENT WORD EMBEDDINGS ON THE MODEL

Model	R-1	R-2	R-L
One-Hot encoding	23.45	8.11	27.80
CBOW	35.86	12.01	30.80
Skip-gram	36.04	12.14	31.11
GloVe	39.30	15.26	35.39
FastText	40.14	16.98	36.89
BERT	43.40	17.33	37.44
DistilBERT	44.51	18.46	38.90

To rigorously assess the efficacy of various word embedding techniques in tandem with our model, we introduced five different embeddings for our evaluation: One-Hot encoding, CBOW, Skip-gram, GloVe, FastText, and the original BERT model [45]:

- One-Hot Encoding: This basic yet foundational technique is essential for translating categorical variables into a format that can be fed into deep learning models, thereby optimizing prediction and classification outcomes. The essence of this approach lies in representing each unique category with a distinct binary code, ensuring only one bit is "hot" or set to '1' for every class representation.

- CBOW and Skip-gram: These are sophisticated models that employ neural networks to associate words with their respective embedding vectors. Their operational methodologies might differ, but they share a common goal.

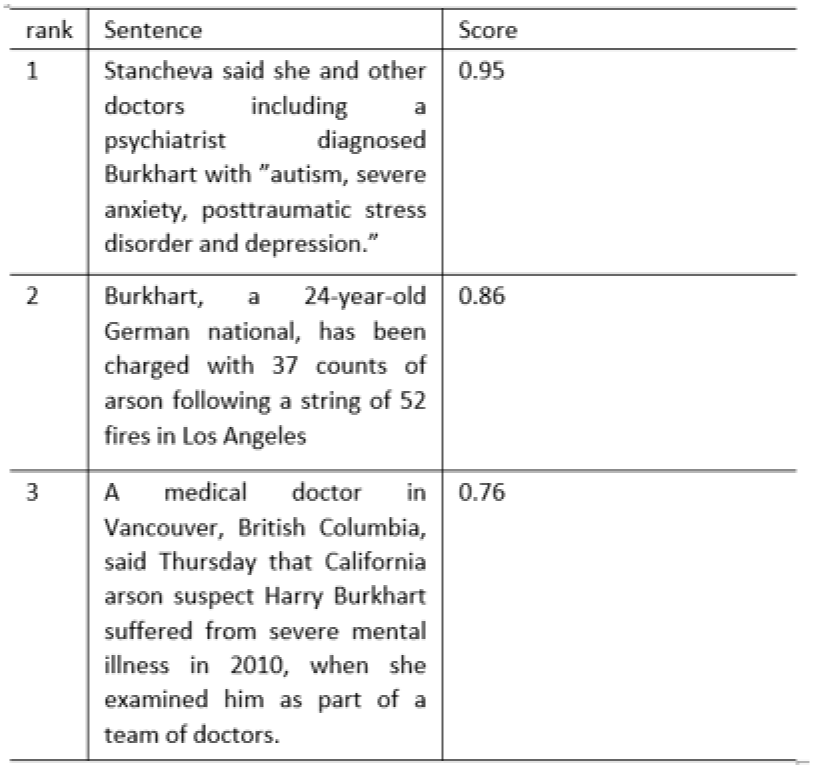
- GloVe: This unsupervised learning model taps into the aggregated co-occurrence data of global word pairs from a given corpus, providing a distinctive representation for words.

- FastText: Pioneering an evolution of the Skip-gram model, FastText takes a novel approach by encoding words as letter n-grams rather than representing them as unique vectors.

For a comprehensive understanding of our results, one should consult Table IV. It was anticipated, and the results confirmed, that One-Hot encoding lagged behind other embeddings, registering suboptimal performance. The improvement metrics for our proposed model using this method were approximately 62.70% (R-1), 9% (R-2), and 18% (R-L). Intriguingly, CBOW and Skip-gram, given their analogous architecture, exhibited similar performances, with both overshadowing the GloVe embedding. Among the lot, the BERT model emerged as the most competent word embedding technique. However, its effectiveness diminished slightly when juxtaposed with the DistilBERT model. In comparison to BERT, DistilBERT demonstrated a reduction in errors by 11% (R-1), 10% (R-2), and 19% (R-L).

The real summary is, “A Canadian doctor says she was part of a team examining Harry Burkhart in 2010, Diagnosis: autism, severe anxiety, post-traumatic stress disorder and depression, Burkhart is also suspected in a German arson probe, officials say, Prosecutors believe the German national set a string of fires in Los Angeles”.

3) Examples: Using a practical illustration of the generator's functionality, we've provided three sentences from a document within the CNN/Daily Mail dataset. These sentences, as extracted by the generator, can be viewed alongside their corresponding reference summary in Fig. 5. Upon inspection, it is evident that sentences sharing a greater number of words with the reference summary tend to receive elevated scores. This showcases the generator's ability to prioritize and assign higher scores to sentences that align more closely with the central themes or keywords present in the reference summary.



rank	Sentence	Score
1	Stancheva said she and other doctors including a psychiatrist diagnosed Burkhart with ”autism, severe anxiety, posttraumatic stress disorder and depression.”	0.95
2	Burkhart, a 24-year-old German national, has been charged with 37 counts of arson following a string of 52 fires in Los Angeles	0.86
3	A medical doctor in Vancouver, British Columbia, said Thursday that California arson suspect Harry Burkhart suffered from severe mental illness in 2010, when she examined him as part of a team of doctors.	0.76

Fig. 5. Three sentences extracted by the generator for the CNN / Daily mail dataset.

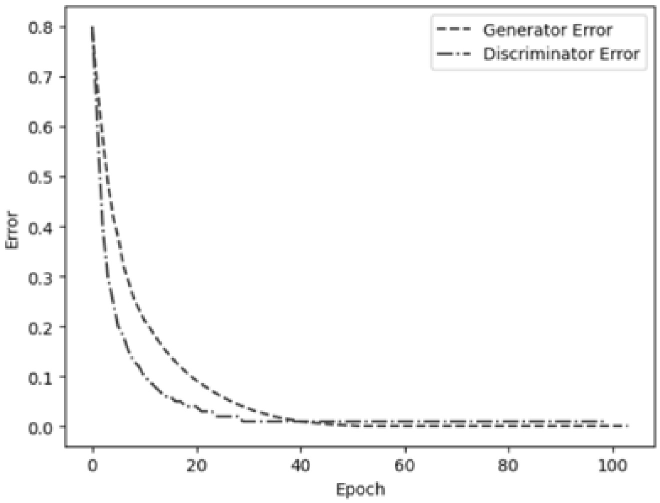


Fig. 6. Comparative diagram of error dynamics.

4) Discussion: This paper unveiled a groundbreaking approach to extractive text summarization, blending a multitude of advanced techniques. Our introduced methodology synergizes the power of a GAN-centric framework, the precision of DistilBERT word embeddings, and the adaptability of TLSTM. Central to efficacious summarization is the interplay between two components within the GAN architecture: the generator and the discriminator. The generator's primary task is to gauge the significance of each sentence in a prospective summary. In contrast, the discriminator's role is to critique and ascertain the caliber of the summaries generated. Such a dynamic within the GAN structure empowers the generator, encouraging it to sift through an array of sentence permutations. As a result, this culminates in the creation of summaries that are both concise and of superior quality. Adding another layer of sophistication is TLSTM, which harnesses the power of transductive learning. Transductive learning is distinctive in its approach, as it assigns augmented weights to data samples that are proximate to a specific test point. This ensures that the most relevant and closely aligned data samples exert greater influence, optimizing the summarization process.

Fig. 6 presents error diagrams for both the generator and discriminator in a GAN across various epochs. Initially, the generator's error is markedly higher at 0.789, revealing its challenges in generating samples mimicking the true data distribution. Yet, as training progresses, the error of the generator exhibits a clear decreasing trend. This suggests the generator is progressively getting better at simulating genuine data, capturing intricate patterns within the dataset. Simultaneously, the discriminator, starting with a slightly lower error of 0.8, undergoes its own evolution. Its role is to differentiate between real and synthetic data. Over the epochs, its error also reduces, though not as sharply as the generator. This indicates that even as the discriminator becomes more skillful, the generator is advancing at a slightly faster pace, producing ever more convincing samples. The interplay between these adversarial elements is crucial to the convergence of the GAN. A diminishing error for both entities across epochs implies a harmonious convergence in the GAN training. The generator refines its outputs, drawing them closer to real samples, while the discriminator sharpens its evaluative abilities. This consistent drop in error highlights the stability and continuous advancement of the GAN in training. The model aptly leverages the adversarial dynamic between its components to enhance performance over epochs. In essence, Fig. 5 underscores the iterative refinement in GAN training, each step bringing the system closer to generating more credible synthetic data distributions.

The paper touts the proposed model's preeminence, substantiating its claims through performance metrics derived from the ROUGE evaluation on the CNN/Daily Mail dataset. Nevertheless, this evaluation is tethered to just one dataset, raising questions about the model's adaptability across a spectrum of diverse datasets. To genuinely encapsulate the model's prowess, it would be prudent to undertake assessments across a myriad of datasets. Relying solely on the CNN/Daily Mail dataset might pave the way for dataset-specific biases. The rationale behind this exclusive dataset choice warrants elucidation, and an exploration into the model's versatility across varied datasets is imperative. Multiple datasets inherently encapsulate nuances in linguistic style, domain specificity, and content diversity, which undeniably bear implications on the outcomes of text summarization. A more holistic evaluation, spanning multiple datasets, would invariably render a more nuanced understanding of the model's capabilities.

The paper touts the proposed model's preeminence, substantiating its claims through performance metrics derived from the ROUGE evaluation on the CNN/Daily Mail dataset. Nevertheless, this evaluation is tethered to just one dataset, raising questions about the model's adaptability across a spectrum of diverse datasets [64]. To genuinely encapsulate the model's prowess, it would be prudent to undertake assessments across more datasets. For this, we can use datasets presented in [65]. Relying solely on the CNN/Daily Mail dataset might pave the way for dataset-specific biases. The rationale behind this exclusive dataset choice warrants elucidation and an exploration into the model's versatility across varied datasets is imperative. Multiple datasets inherently encapsulate nuances in linguistic style, domain specificity, and content diversity, which undeniably bear implications on the outcomes of text summarization. A more holistic evaluation, spanning multiple datasets, would invariably render a more nuanced understanding of the model's capabilities [66].

GANs are built upon a novel framework where two neural networks, the generator and the discriminator, work in tandem. The generator's primary goal is to produce outputs that are indistinguishable from real data, while the discriminator's objective is to differentiate between actual data and the data generated by the generator. This adversarial process, though powerful in theory, presents several practical challenges, particularly during the training phase. One primary concern is the issue of convergence. Given the dynamic nature of the adversarial relationship, ensuring that both networks converge to an optimal solution is not straightforward. If not carefully managed, the training can end up in a loop where each network constantly tries to outdo the other without reaching a stable equilibrium. This oscillatory behavior can make GANs particularly sensitive to hyperparameters, initialization, and the chosen architecture, often requiring extensive experimentation and fine-tuning. Additionally, the delicate balance between the generator and discriminator can easily be disrupted. If the discriminator becomes exceptionally adept early on in training, it can stifle the generator's ability to learn. The generator, facing constant rejection from the discriminator, may struggle to make any meaningful progress, leading to a stagnation in learning and potentially resulting in mode collapse, where the generator produces limited or repetitive outputs. On the other hand, if the generator dominates the learning process and continually manages to deceive the discriminator, the discriminator may fail to provide meaningful feedback. This can result in generated summaries that, while convincing at first glance, might stray from the original content's essence, compromising the quality and relevance of the output. Given these challenges, there's a growing consensus in the research community about the need for more refined training strategies for GANs [67]. Techniques such as gradient penalty [68], spectral normalization [69], and modified loss functions [70] have been proposed to stabilize GAN training.

Transductive learning is a unique learning paradigm that seeks to make predictions specifically for the given test set without generalizing to the broader population. By concentrating on samples near the test point, it can produce highly optimized results for a specific set of data. However, this precision comes with its own set of challenges, primarily related to model generalization [71]. The inherent nature of transductive learning to prioritize certain instances over others can inadvertently lead the model to capture noise or idiosyncrasies present in the training data. Such a model would be finely tuned to a particular dataset, but might falter when introduced to new, unseen data. This phenomenon, known as overfitting, means that while the model performs exceptionally well on its training data, its performance significantly drops on new, unfamiliar data. In practical scenarios, especially in dynamic environments like news summarization, social media analytics, or customer feedback systems, data distributions can shift rapidly. A model trained with a strong transductive bias might not adapt well to these changing scenarios, thus compromising its effectiveness and reliability. It would continuously require retraining or fine-tuning on new data points, which is resource-intensive and not always feasible. Addressing these challenges necessitates a more balanced approach to learning. One potential avenue is the incorporation of regularization techniques [72]. Regularization, in essence, adds a penalty to the loss function, discouraging the model from fitting too closely to every data point and, in turn, mitigating overfitting. Techniques such as L1 and L2 regularization or dropout [73] can be applied to ensure the model retains a level of generality. Furthermore, blending transductive learning with inductive learning offers another promising solution. While transductive learning focuses on specific test points, inductive learning aims to find a general pattern or hypothesis that can be applied to any input. By combining these two paradigms, one could harness the precision of transductive learning while maintaining the broader applicability provided by inductive learning. Such a hybrid approach would not only cater to specific data instances but also ensure that the model remains versatile and adaptable to a range of data distributions.

V. Conclusion

In the research presented, we introduce an innovative approach to extractive text summarization, leveraging a blend of GAN, the DistilBERT word embedding technique, and an attention-centric TLSTM methodology. Utilizing DistilBERT, we crafted a feature vector for each sentence, which was subsequently fed into the generator to deduce the likelihood of that sentence being part of the final summary. In tandem, a discriminator was employed to scrutinize the summaries churned out by the generator, thus honing its capabilities. We further innovated by designing a unique loss function tailored for the training of the discriminator. This function meticulously considers the output of the generator, as well as both authentic and contrived document summaries. An intriguing facet of our methodology is that each document is paired with distinct noise during both training and testing phases. Such a strategy empowers the generator, equipping it to explore a vast array of sentence amalgamations, laying the groundwork for the creation of superior quality summaries. Empirical evaluations conducted on the CNN/Daily Mail dataset lend weight to the efficacy of our model. The results not only underscore the effectiveness of our novel methodology but also highlight its superiority, outpacing other established text summarization techniques in performance metrics.

In forthcoming research endeavors, we intend to focus on enhancing the coherence among sentences within our model. Coherence plays an instrumental role in ensuring that the summarized text is not just a collection of sentences, but a fluid and cohesive narrative that is easy for readers to follow and understand. Addressing this aspect can significantly elevate the quality and readability of the generated summaries. One possible approach to achieve this would be to prioritize coherence during the construction of our target summaries. By doing so, the model would be trained to select sentences that not only contain critical information but also seamlessly connect with one another, ensuring a natural flow of ideas. Additionally, another promising avenue to explore is the incorporation of coherence as a loss function within the generator. By integrating coherence into the loss function, the generator would be incentivized to produce summaries where the sentences logically follow one another, leading to more cohesive and contextually relevant outputs. Introducing such modifications could provide dual benefits: improving the intrinsic quality of the summaries and enhancing the user experience, as coherent and logically structured summaries are more easily comprehensible. This, in turn, would further cement the model's applicability and usefulness in real-world scenarios, catering to a wider range of text summarization needs.

References

[1] R. Mitkov, Handbook for Language Engineers edited by Ali Farghaly, Computational Linguistics, 30 (2004) 397-399.

[2] T. Hirao, H. Isozaki, E. Maeda, Y. Matsumoto, Extracting important sentences with support vector machines, COLING 2002: the 19th international conference on computational linguistics, 2002.

[3] S. Akter, A.S. Asa, M.P. Uddin, M.D. Hossain, S.K. Roy, M.I. Afjal, An extractive text summarization technique for Bengali document (s) using K-means clustering algorithm, 2017 ieee international conference on imaging, vision & pattern recognition (icivpr), IEEE, 2017, pp. 1-6.

[4] S. Vakilian, S.V. Moravvej, A. Fanian, Using the artificial bee colony (ABC) algorithm in collaboration with the fog nodes in the Internet of Things three-layer architecture, 2021 29th Iranian Conference on Electrical Engineering (ICEE), IEEE, 2021, pp. 509-513.

[5] A. Kumar, A. Sharma, Systematic literature review of fuzzy logic based text summarization, Iranian journal of fuzzy systems, 16 (2019) 45-59.

[6] S. Vakilian, S.V. Moravvej, A. Fanian, Using the cuckoo algorithm to optimizing the response time and energy consumption cost of fog nodes by considering collaboration in the fog layer, 2021 5th International Conference on Internet of Things and Applications (IoT), IEEE, 2021, pp. 1-5.

[7] S.V. Moravvej, R. Alizadehsani, S. Khanam, Z. Sobhaninia, A. Shoeibi, F. Khozeimeh, Z.A. Sani, R.-S. Tan, A. Khosravi, S. Nahavandi, RLMD-PA: A reinforcement learning-based myocarditis diagnosis combined with a population-based algorithm for pretraining weights, Contrast Media & Molecular Imaging, 2022 (2022).

[8] W.S. El-Kassas, C.R. Salama, A.A. Rafea, H.K. Mohamed, EdgeSumm: Graph-based framework for automatic text summarization, Information Processing & Management, 57 (2020) 102264.

[9] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A.A. Bharath, Generative adversarial networks: An overview, IEEE signal processing magazine, 35 (2018) 53-65.

[10] S. Song, H. Huang, T. Ruan, Abstractive text summarization using LSTM-CNN based deep learning, Multimedia Tools and Applications, 78 (2019) 857-875.

[11] S.V. Moravvej, M. Joodaki, M.J.M. Kahaki, M.S. Sartakhti, A method based on an attention mechanism to measure the similarity of two sentences, 2021 7th International Conference on Web Research (ICWR), IEEE, 2021, pp. 238-242.

[12] S. Hochreiter, J. Schmidhuber, LSTM can solve hard long time lag problems, Advances in neural information processing systems, 9 (1996).

[13] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805, (2018).

[14] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108, (2019).

[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692, (2019).

[16] D.d.V. Feijo, V.P. Moreira, Improving abstractive summarization of legal rulings through textual entailment, Artificial intelligence and law, 31 (2023) 91-113.

[17] H. Oh, S. Nam, Y. Zhu, Structured abstract summarization of scientific articles: Summarization using full‐text section information, Journal of the Association for Information Science and Technology, 74 (2023) 234-248.

[18] P.K. Katwe, A. Khamparia, D. Gupta, A.K. Dutta, Methodical Systematic Review of Abstractive Summarization and Natural Language Processing Models for Biomedical Health Informatics: Approaches, Metrics and Challenges, ACM Transactions on Asian and Low-Resource Language Information Processing, (2023).

[19] M.T.R. Laskar, M. Rahman, I. Jahan, E. Hoque, J. Huang, CQSumDP: A ChatGPT-Annotated Resource for Query-Focused Abstractive Summarization Based on Debatepedia, arXiv preprint arXiv:2305.06147, (2023).

[20] Z. Chen, H. Lin, Improving named entity correctness of abstractive summarization by generative negative sampling, Computer Speech & Language, 81 (2023) 101504.

[21] T. Vo, An approach of syntactical text graph representation learning for extractive summarization, International Journal of Intelligent Robotics and Applications, 7 (2023) 190-204.

[22] S. Rai, R.C. Belwal, A. Sharma, Investigating the Application of Multi-lingual Transformer in Graph-Based Extractive Text Summarization for Hindi Text, International Conference on Data Management, Analytics & Innovation, Springer, 2023, pp. 393-403.

[23] L. Alex, H. Sakib, W. Lingfei, M. Collin, Improved code summarization via a graph neural network. In 2020 IEEE, ACM International Conference on Program Comprehension, 2020.

[24] S.-h. Zhong, Y. Liu, B. Li, J. Long, Query-oriented unsupervised multi-document summarization via deep learning model, Expert systems with applications, 42 (2015) 8146-8155.

[25] M. Yousefi-Azar, L. Hamey, Text summarization using unsupervised deep learning, Expert Systems with Applications, 68 (2017) 93-105.

[26] Z. Cao, F. Wei, L. Dong, S. Li, M. Zhou, Ranking with recursive neural networks and its application to multi-document summarization, Proceedings of the AAAI conference on artificial intelligence, 2015.

[27] M. Rosca, B. Lakshminarayanan, D. Warde-Farley, S. Mohamed, Variational approaches for auto-encoding generative adversarial networks, arXiv preprint arXiv:1706.04987, (2017).

[28] A. Abdi, S. Hasan, S.M. Shamsuddin, N. Idris, J. Piran, A hybrid deep learning architecture for opinion-oriented multi-document summarization based on multi-feature fusion, Knowledge-Based Systems, 213 (2021) 106658.

[29] H. Zareiamand, A. Darroudi, I. Mohammadi, S.V. Moravvej, S. Danaei, R. Alizadehsani, Cardiac Magnetic Resonance Imaging (CMRI) Applications in Patients with Chest Pain in the Emergency Department: A Narrative Review, Diagnostics, 13 (2023) 2667.

[30] B.T. Hammad, A.M. Sagheer, I.T. Ahmed, N. Jamil, A comparative review on symmetric and asymmetric DNA-based cryptography, Bulletin of Electrical Engineering and Informatics, 9 (2020) 2484-2491.

[31] D. Hin, A. Kan, H. Chen, M.A. Babar, LineVD: statement-level vulnerability detection using graph neural networks, Proceedings of the 19th International Conference on Mining Software Repositories, 2022, pp. 596-607.

[32] R. Nallapati, F. Zhai, B. Zhou, Summarunner: A recurrent neural network based sequence model for extractive summarization of documents, Proceedings of the AAAI conference on artificial intelligence, 2017.

[33] H. Kobayashi, M. Noguchi, T. Yatsuka, Summarization based on embedding distributions, Proceedings of the 2015 conference on empirical methods in natural language processing, 2015, pp. 1984-1989.

[34] L. Chen, M. Le Nguyen, Sentence selective neural extractive summarization with reinforcement learning, 2019 11th International Conference on Knowledge and Systems Engineering (KSE), IEEE, 2019, pp. 1-5.

[35] S. Danaei, A. Bostani, S.V. Moravvej, F. Mohammadi, R. Alizadehsani, A. Shoeibi, H. Alinejad-Rokny, S. Nahavandi, Myocarditis Diagnosis: A Method using Mutual Learning-Based ABC and Reinforcement Learning, 2022 IEEE 22nd International Symposium on Computational Intelligence and Informatics and 8th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics (CINTI-MACRo), IEEE, 2022, pp. 000265-000270.

[36] M. Kågebäck, O. Mogren, N. Tahmasebi, D. Dubhashi, Extractive summarization using continuous vector space models, Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC), 2014, pp. 31-39.

[37] W. Yin, Y. Pei, Optimizing sentence modeling and selection for document summarization, Twenty-fourth international joint conference on artificial intelligence, 2015.

[38] F. Koto, J.H. Lau, T. Baldwin, Discourse probing of pretrained language models, arXiv preprint arXiv:2104.05882, (2021).

[39] S. Abdel-Salam, A. Rafea, Performance study on extractive text summarization using BERT models, Information, 13 (2022) 67.

[40] A. Srikanth, A.S. Umasankar, S. Thanu, S.J. Nirmala, Extractive text summarization using dynamic clustering and co-reference on BERT, 2020 5th International Conference on Computing, Communication and Security (ICCCS), IEEE, 2020, pp. 1-5.

[41] C. Ma, T. Wang, L. Zhang, Z. Cao, Y. Huang, X. Ding, Distributed Representation Learning with Skip-Gram Model for Trained Random Forests, Neurocomputing, (2023) 126434.

[42] S. Aburass, O. Dorgham, J.A. Shaqsi, A Hybrid Machine Learning Model for Classifying Gene Mutations in Cancer using LSTM, BiLSTM, CNN, GRU, and GloVe, arXiv preprint arXiv:2307.14361, (2023).

[43] S.V. Moravvej, S.J. Mousavirad, M.H. Moghadam, M. Saadatmand, An LSTM-based plagiarism detection via attention mechanism and a population-based approach for pre-training parameters with imbalanced classes, Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part III 28, Springer, 2021, pp. 690-701.

[44] S.V. Moravvej, S.J. Mousavirad, D. Oliva, G. Schaefer, Z. Sobhaninia, An improved de algorithm to optimise the learning process of a bert-based plagiarism detection model, 2022 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2022, pp. 1-7.

[45] S.V. Moravvej, S.J. Mousavirad, D. Oliva, F. Mohammadi, A Novel Plagiarism Detection Approach Combining BERT-based Word Embedding, Attention-based LSTMs and an Improved Differential Evolution Algorithm, arXiv preprint arXiv:2305.02374, (2023).

[46] M.S. Sartakhti, M.J.M. Kahaki, S.V. Moravvej, M. javadi Joortani, A. Bagheri, Persian language model based on BiLSTM model on COVID-19 corpus, 2021 5th International Conference on Pattern Recognition and Image Analysis (IPRIA), IEEE, 2021, pp. 1-5.

[47] L. Hong, M.H. Modirrousta, M. Hossein Nasirpour, M. Mirshekari Chargari, F. Mohammadi, S.V. Moravvej, L. Rezvanishad, M. Rezvanishad, I. Bakhshayeshi, R. Alizadehsani, GAN‐LSTM‐3D: An efficient method for lung tumour 3D reconstruction enhanced by attention‐based LSTM, CAAI Transactions on Intelligence Technology, (2023).

[48] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation, 9 (1997) 1735-1780.

[49] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078, (2014).

[50] F.A. Gers, N.N. Schraudolph, J. Schmidhuber, Learning precise timing with LSTM recurrent networks, Journal of machine learning research, 3 (2002) 115-143.

[51] A. Graves, Generating sequences with recurrent neural networks, arXiv preprint arXiv:1308.0850, (2013).

[52] W. Zaremba, I. Sutskever, O. Vinyals, Recurrent neural network regularization, arXiv preprint arXiv:1409.2329, (2014).

[53] S.V. Moravvej, M.J.M. Kahaki, M.S. Sartakhti, A. Mirzaei, A method based on attention mechanism using bidirectional long-short term memory (BLSTM) for question answering, 2021 29th Iranian Conference on Electrical Engineering (ICEE), IEEE, 2021, pp. 460-464.

[54] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in neural information processing systems, 27 (2014).

[55] A.R. Zehan, Web Interface for Rouge Automatic Summary Evaluator, Invited Lectures, 57.

[56] M. Moradi, Frequent itemsets as meaningful events in graphs for summarizing biomedical texts, 2018 8th International Conference on Computer and Knowledge Engineering (ICCKE), IEEE, 2018, pp. 135-140.

[57] R. Mihalcea, P. Tarau, Textrank: Bringing order into text, Proceedings of the 2004 conference on empirical methods in natural language processing, 2004, pp. 404-411.

[58] Y. Wu, B. Hu, Learning to extract coherent summary via deep reinforcement learning, Proceedings of the AAAI conference on artificial intelligence, 2018.

[59] J.-Á. González, E. Segarra, F. García-Granada, E. Sanchis, L.ı.-F. Hurtado, Siamese hierarchical attention networks for extractive summarization, Journal of Intelligent & Fuzzy Systems, 36 (2019) 4599-4607.

[60] K. Al-Sabahi, Z. Zuping, M. Nadher, A hierarchical structured self-attentive model for extractive document summarization (HSSAS), IEEE Access, 6 (2018) 24205-24212.

[61] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, The Journal of Machine Learning Research, 21 (2020) 5485-5551.

[62] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, arXiv preprint arXiv:1910.13461, (2019).

[63] A. Joshi, E. Fidalgo, E. Alegre, L. Fernández-Robles, DeepSumm: Exploiting topic models and sequence to sequence networks for extractive text summarization, Expert Systems with Applications, 211 (2023) 118442.

[64] S. Gong, Z. Zhu, J. Qi, W. Wu, C. Tong, SeburSum: a novel set-based summary ranking strategy for summary-level extractive summarization, The Journal of Supercomputing, (2023) 1-29.

[65] A.P. Widyassari, S. Rustad, G.F. Shidik, E. Noersasongko, A. Syukur, A. Affandy, Review of automatic text summarization techniques & methods, Journal of King Saud University-Computer and Information Sciences, 34 (2022) 1029-1046.

[66] S.V. Moravvej, A. Mirzaei, M. Safayani, Biomedical text summarization using conditional generative adversarial network (CGAN), arXiv preprint arXiv:2110.11870, (2021).

[67] S. Moravvej, M. Maleki Kahaki, M. Salimi Sartakhti, M. Joodaki, Efficient GAN-based method for extractive summarization, Journal of Electrical and Computer Engineering Innovations (JECEI), 10 (2022) 287-298.

[68] S. Bourou, A. El Saer, T.-H. Velivassaki, A. Voulkidis, T. Zahariadis, A review of tabular data synthesis using GANs on an IDS dataset, Information, 12 (2021) 375.

[69] Z. Li, M. Usman, R. Tao, P. Xia, C. Wang, H. Chen, B. Li, A systematic survey of regularization and normalization in GANs, ACM Computing Surveys, 55 (2023) 1-37.

[70] Z. Pan, W. Yu, B. Wang, H. Xie, V.S. Sheng, J. Lei, S. Kwong, Loss functions of generative adversarial networks (GANs): Opportunities and challenges, IEEE Transactions on Emerging Topics in Computational Intelligence, 4 (2020) 500-522.

[71] I. Triguero, S. García, F. Herrera, Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study, Knowledge and Information systems, 42 (2015) 245-284.

[72] A.A. Syed, F.L. Gaol, A. Boediman, T. Matsuo, W. Budiharto, A Survey of Abstractive Text Summarization Utilising Pretrained Language Models, Asian Conference on Intelligent Information and Database Systems, Springer, 2022, pp. 532-544.

[73] M.Q. Pham, B. Oudompheng, J.I. Mars, B. Nicolas, A Noise-Robust Method with Smoothed ℓ1/ℓ2 Regularization for Sparse Moving-Source Mapping, Signal Processing, 135 (2017) 96-106.