2023 IEEE 35th International Conference on Tools with Artificial Intelligence (ICTAI) | 979-8-3503-4273-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICTAI59109.2023.00053

DBERT-ELVA: Discourse-Aware Extractive Text Summarization with Autoencoder

Mahsa Abazari Kia
Computer and Data Science Department
Northeastern University London
London, United Kingdom
mahsa.abazari@nulondon.ac.uk

***Abstract*—In recent advancements in text summarization, BERT has gained popularity for encoding documents. However, sentence-based extractive models often lead to redundant or uninformative phrases in the generated summaries. Additionally, BERT, which is pretrained on sentence pairs rather than full documents, struggles to capture long-range dependencies present within a document. To overcome these challenges, we introduce DBERT-ELVA, a discourse-aware neural summarization model. DBERT-ELVA extracts sub-sentential discourse units, offering a more refined granularity for extractive selection, in contrast to traditional sentence-based approaches. To learn a compressed representation of these discourse units while capturing the inter-dependencies among them, an Autoencoder is designed, utilizing Extreme Learning for improved generalization performance. Experimental evaluations on popular summarization benchmarks demonstrate that the proposed model significantly outperforms state-of-the-art methods, including other BERT-based models, by a substantial margin.**

***Index Terms*—Text summarization, Autoencoder, BERT encoder, Extreme learning machine**

I. Introduction

The abundance and magnitude of digital documents accessible on the internet have significantly increased as a result of the expansion of social media and user-generated content. Consequently, there is a demand for numerous applications in Natural Language Processing (NLP) to analyze this vast amount of data. One such application is Automatic Text Summarization (ATS), which is a progressively expanding and complex task within the field of NLP. The objective of ATS is to generate a concise rendition of a lengthy text document while retaining the key concepts conveyed in the original source [1].

The proliferation of text-based content from various sources such as social networks, forums, sensors, and news websites has led to a significant increase in information volume. As a result, the importance of text summarization systems has grown considerably. These systems offer users the ability to grasp the essence of a text without the need to scroll through extensive pages, which can save hours of searching and enable users to focus on their intended goals. By quickly identifying the most relevant information, text summarization systems eliminate the necessity of examining the entire document to determine its relevance [2]. This becomes especially crucial when users have limited time to make critical decisions. Consequently, there is a gradual recognition of the need for automatic summarization software, driven by the potential cost savings resulting from automation [3]. Moreover, text summarization finds a crucial application in information retrieval systems, where search engines display a concise snippet of text summarizing the ranked page. This assists users in selecting the content that aligns best with their information requirements.

Text summarization can be broadly classified into two distinct categories: extractive and abstractive summarization. Extractive summarization, also known as sentence ranking, involves the process of ranking and extracting sentences based on their relevance and importance within the text. On the other hand, abstractive summarization entails the creation of new sentences that capture the essential idea of the original text through a process of rewriting and rephrasing [4].

Abstractive summarization requires extensive linguistic resources and human-created ontologies. However, the scarcity of natural language resources makes abstractive approaches challenging, resulting in the widespread adoption of extractive methods instead [5]. Document representation holds significant importance in machine learning algorithms employed in the field of NLP. This critical phase involves converting text into numerical values, which are then utilized as input vectors for these algorithms. By transforming text into this numerical representation, NLP algorithms can effectively process and analyze the data.

In recent times, the utilization of Bidirectional Encoder Representations from Transformers (BERT) [6] has become prevalent for document encoding in cutting-edge text summarization models. BERT incorporates attention mechanisms, allowing it to focus on important words throughout the entire document during the encoding process. By taking into account the broader context and utilizing contextualized representations, BERT proficiently captures the underlying meaning and subtleties of the text, resulting in a comprehensive document encoding.

In this study, our focus is on exploring a text similarity measure that relies on discourse-aware representations and features derived from an unsupervised model. The primary objective is to predict the importance of concepts and subsequently select the most significant pieces of text to be included in the summary. To accomplish this, we propose a deep learning

979-8-3503-4273-4/23/$31.00 ©2023 IEEE
DOI 10.1109/ICTAI59109.2023.00053

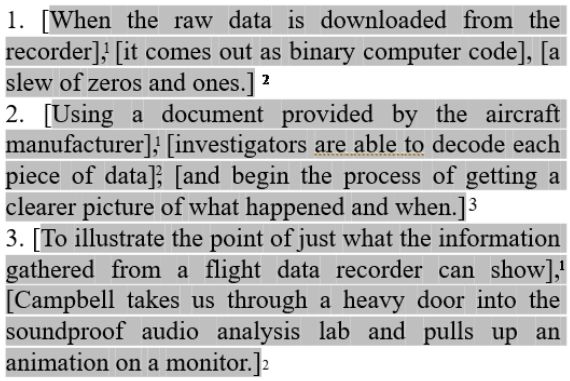

Fig. 1. An example of discourse segmentation, where sentences are broken down into Elementary Discourse Units (EDUs)

model that computes the semantic similarity between the text discourse units. Our approach utilizes BERT representation and offers an automatic text summarization model. Given the limited availability of labeled data for training supervised models, we find unsupervised deep learning techniques more suitable, especially when there is an abundance of unlabeled data. Therefore, we propose an unsupervised deep learning model that can extract meaningful features from unlabeled data. As a result, the issue of insufficient labeled data becomes obsolete, and we can overcome this limitation effectively.

This paper introduces DBERT-ELVA, a neural extractive summarization model that incorporates both BERT and Autoencoder while considering discourse information. Unlike traditional extractive summarization models that operate at the sentence level, we utilize Elementary Discourse Units (EDUs) as the minimal selection unit, derived from the concept of Rhetorical Structure Theory (RST) [7], [8]. This enables us to simultaneously compress and extract information, reducing redundancy across sentences.

Fig. 1 illustrates an example of discourse segmentation, where sentences are broken down into EDUs, indicated by brackets. By working at the EDU level, our model can eliminate redundant details within sub-sentences, allowing for a more concise and informative summary. This approach provides additional capacity to include essential concepts or events, resulting in more meaningful summaries. The model learns a latent representation of the data by using a Autoencoder which is trained by an Extreme Learning (EL) technique [9] .

The main objective of this paper is to evaluate the usefulness of fine-tuning BERT with an unsupervised feature learning model while considering the discourse information and computing the semantic similarity for text unit selection in documents summarization task.

To demonstrate the effectiveness of our proposed approach, we conduct experiments using two distinct publicly available datasets specifically designed for evaluating the quality of text summarization systems. This allows us to showcase the complementarity of our approach across different evaluation scenarios and datasets. The main contributions are listed below:

- We propose an extractive summarization model, DBERT-ELVA, which functions at a granular level called subsentential discourse units. This approach enables the generation of summaries that are both concise and informative, while minimizing redundancy.
- The Autoencoder has been utilized with fine-tuned discourse-aware BERT and Extreme Learning to reduce the training and generate comprehensive extractive summaries.
- We propose an Extreme Learning technique to enhance the generalization and performance by utilizing regularization.
- The content of the extractive summary is selected based on the similarity of the text EDUs' abstract representation obtained by the designed deep neural network.

II. Related Work

A. Neural Extractive Summarization

The objective of ETS (Extractive Text Summarization) is to categorize sentences within a document based on whether they should be included in the summary. Liu and Lapata [10] refined BERT by incorporating stacked Transformer layers and a sigmoid classifier, known as BERTSUMEXT. Instead of directly utilizing the existing Transformer encoder for document encoding, Zhang et al. [11] employ a hierarchical Transformer encoder, comprising a sentence encoder and a document encoder (HIBERT), which is pre-trained and fine-tuned for ETS. For lengthy documents, Xiao and Carenini [12] propose ExtSum-LG, an ETS model that utilizes a recurrent neural network (RNN) to consider both the global and local context. To tackle redundancy in extractive summaries, the authors further enhance their research by introducing redundancy reduction techniques [13].

B. Discourse and Summarization

Louis et al. [14] investigated the advantages of utilizing discourse relations' graph structure for summarization purposes. Hirao et al. [15] and Yoshida et al. [16] approached the summarization problem by considering the pruning of the document discourse tree. Durrett et al. [17] introduced a system that combined sentence extraction and compression, employing ILP (Integer Linear Programming) methods with discourse structure. Li et al. [18] demonstrated that employing EDUs for content selection resulted in improved summarization performance. In comparison to these previous works, our proposed approach represents a neural end-to-end summarization model that uses EDUs as the basis for selection. Xu et al. [19] presented an extractive approach with structural discourse graphs constructed based on RST trees and coreference mentions, encoded with Graph Convolutional Networks.

C. Extreme Learning Machine

ELM (Extreme Learning Machine), introduced by Huang et al. [9], was developed as a means to efficiently and quickly learn Single-Layer Feedforward Networks (SLFNs). Initially, ELM was applied to supervised regression and classification tasks [20]. Subsequently, it was adapted for semi-supervised tasks through the incorporation of manifold regularization [21]. In contrast to classical feedforward neural networks trained using the BackPropagation (BP) learning algorithm, which often suffer from slow learning speeds and local minimum issues, ELM achieves a shorter learning time while maintaining superior generalization performance. This efficacy has been demonstrated in various computer vision applications such as image segmentation and classification [22], human action recognition [23], and face classification [24]. In a similar study [25] proposed a novel clustering method that utilizes ELM as an unsupervised feature learning technique.

Our model uses a similar configuration to encode the document with BERT as DISCOBERT did, but we use BERT discourse document encoder with Autoencoders for learning features and labeling the EDUs based on them while preventing overfitting.

III. Methodology

An outline of the proposed model is presented in Fig. 2, which comprises a Document Encoder, an Autoencoder, and a voting component based on similarity. The Document Encoder utilizes a pre-trained BERT model to encode the entire document at the token level. Following that, a self-attentive span extractor is employed to derive EDU representations from the corresponding text spans. The Autoencoder (AE) module takes the Document Encoder's output, mapping it to a latent space and subsequently refining the EDU representations. These refined representations are then used for label prediction, incorporating a voting component that measures the semantic similarity.

Assume that a document D is divided into a total of n EDUs, represented as $D = \{d_1, d_2, \cdots, d_n\}$, where d_i represents the $i-th$ EDU. Following the approach of Liu and Lapata [10], we approach extractive summarization as a task of sequential labeling. Each EDU d_i is assigned a score by neural networks, and decisions are made based on the scores of all EDUs. A sequence of binary labels is generated, where 1 indicates selection and 0 indicates non-selection. These labels are denoted as $Y = \{y_1^*, y_2^*, \cdots, y_n^*\}$. During training, our objective is to predict the sequence of labels Y given the document D. During training, we must also consider discourse dependency to ensure the coherence and grammatical correctness of the resulting summary.

A. Document Encoder

In this section, we first introduce the Discourse Analysis and then explain how we construct a discourse aware document encoder.

1) Discourse Analysis: Discourse analysis focuses on the relationships between sentences in a document or conversation. In the framework of RST, the organization of discourse in text can be depicted as a tree structure. The entire document can be divided into consecutive and non-overlapping segments of text known as EDUs. Each EDU is categorized as either Nucleus or Satellite, indicating its level of importance or salience. Nucleus nodes typically hold more central positions, while Satellite nodes are more peripheral and carry less significance in terms of content and grammatical reliance. The EDUs exhibit dependencies that represent their rhetorical connections. In this study, we consider the EDU as the smallest unit for selecting content in text summarization.

Fig. 3 illustrates an instance of discourse segmentation and the parse tree of a sentence. Within these EDUs, the rhetorical relations signify the roles performed by different discourse units. As noted in the study by Louis et al. [14], the RST tree structure already provides a strong indication for content selection. However, the agreement among rhetorical relations tends to be lower and more ambiguous. Consequently, our model does not explicitly encode rhetorical relations. When it comes to content selection in text summarization, our objective is for the model to choose the most concise and essential concept in the document, minimizing redundancy. Traditional extractive summarization methods require the model to select entire sentences, even if certain parts are unnecessary. In contrast, our proposed approach enables the selection of fine-grained EDUs, thereby reducing redundancy in the generated summaries. This forms the basis of our DBERT-ELVA model.

2) Discourse-based BERT: BERT is a pre-trained deep bidirectional Transformer encoder [6], [26]. Following the approach of Liu and Lapata [10], we employ BERT to encode the entire document and fine-tune the model specifically for summarization purposes. Initially, BERT was trained to encode either a single sentence or a pair of sentences. However, news articles typically consist of more than 500 words, necessitating certain adjustments to utilize BERT for document encoding. To address this, we insert special tokens, namely ⟨CLS⟩ at the beginning and ⟨SEP⟩ at the end of each sentence. Additionally, for encoding longer documents like news articles, we extend the maximum sequence length accepted by BERT from 512 to 768 in all our experiments.

The input document after tokenization is denoted $D = \{d_1, \cdots, d_n\}$, and $d_i = \{w_{i1}, \cdots, w_{i\ell_i}\}$ where ℓ_i is the number of BPE tokens in the $i-th$ EDU. If d_i corresponds to the first EDU in a sentence, an additional ⟨CLS⟩ token is added at the beginning of d_i. Similarly, if d_j corresponds to the last EDU in a sentence, a ⟨SEP⟩ token is appended to d_j (see Fig. 3 for visualization). This approach of inserting ⟨CLS⟩ and ⟨SEP⟩ tokens follows the method employed by Liu and Lapata [10]. For the sake of simplicity, these two tokens are not explicitly shown in the equations. Subsequently, the BERT model is utilized to encode the document:

$$\left\{\mathbf{h}_{11}^B, \cdots, \mathbf{h}_{n\ell_n}^B\right\} = \mathrm{BERT}\left(\{w_{11}, \cdots, w_{n\ell_n}\}\right), \quad (1)$$

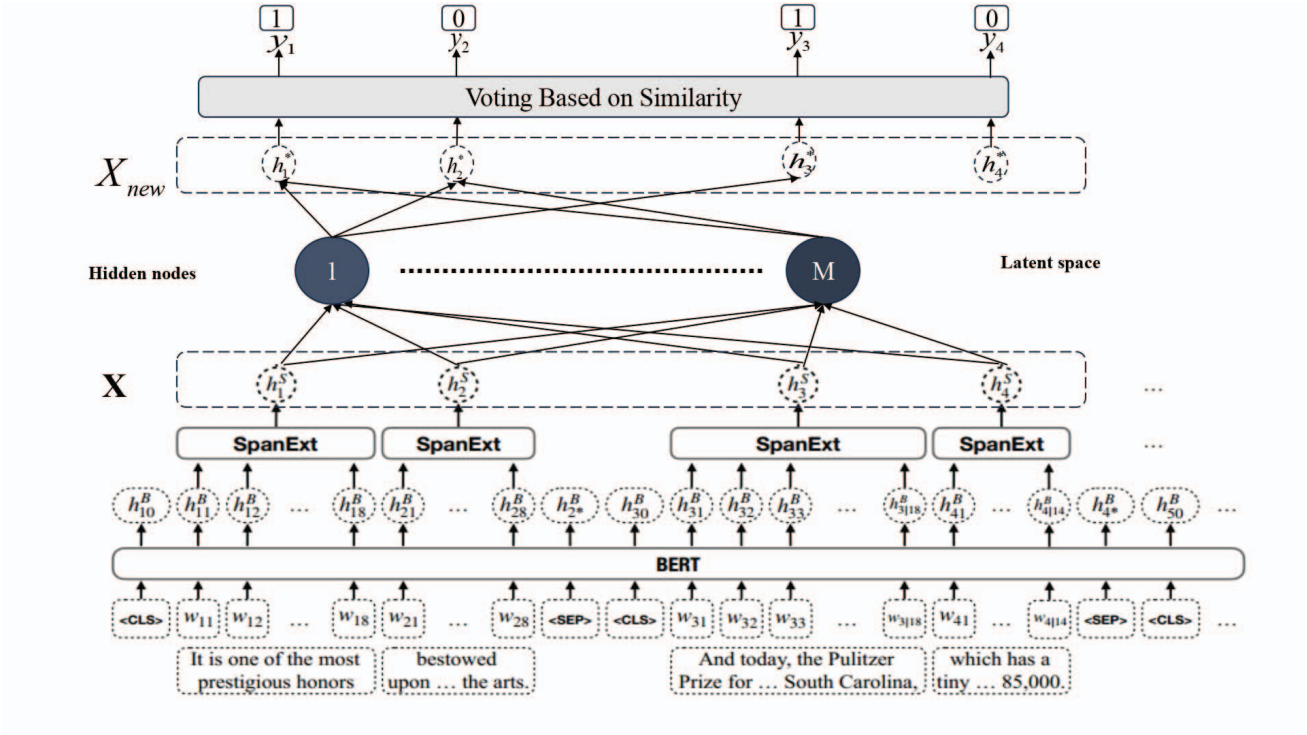

Fig. 2. Model architecture of DBERT-ELVA.

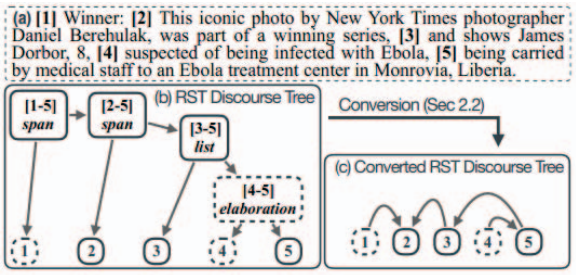

Fig. 3. Example of discourse segmentation and RST tree conversion. The original sentence is segmented into 5 EDUs in box (a), and then parsed into an RST discourse tree in box (b). The converted dependency based RST discourse tree is shown in box (c) [19].

The BERT output of the entire document, denoted as $\left\{\mathbf{h}_{11}^B, \cdots, \mathbf{h}_{n\ell_n}^B\right\}$, has the same length as the input. In the BERT encoder, the representation of the ⟨CLS⟩ token can typically serve as the sentence representation. However, this approach is not suitable for our specific context as we require the extraction of representations for EDUs. To address this, we employ a Self-Attentive Span Extractor (SpanExt), as proposed by Lee et al. [27], to learn the representations of the EDUs.

For the $i-th$ EDU with ℓ_i words, with the output from the BERT encoder $\left\{\mathbf{h}_{i1}^B, \mathbf{h}_{i2}^B, \cdots, \mathbf{h}_{i\ell_i}^B\right\}$, we obtain EDU representation as follows:

$$\alpha_{ij} = \mathbf{W}_2 \cdot \mathrm{ReLU}\left(\mathbf{W}_1 \mathbf{h}_{ij}^B + \mathbf{b}_1\right) + \mathbf{b}_2 \tag{2}$$

$$\mathbf{a}_{ij} = \frac{\exp\left(\alpha_{ij}\right)}{\sum_{k=1}^{\ell_i} \exp\left(\alpha_{ik}\right)}, \quad \mathbf{h}_i^S = \sum_{j=1}^{\ell_i} \mathbf{a}_{ij} \cdot \mathbf{h}_{ij}^B, \tag{3}$$

In the context of the provided text, the symbol α_{ij} represents the score assigned to the $j-th$ word within an EDU, while a_{ij} denotes the normalized attention of the $j-th$ word with respect to all the words within the span. The representation $\mathbf{h}_i^S$ of the EDU is obtained as a weighted sum of the hidden states from the BERT output. It's important to note that throughout the paper, the matrices W and vectors b are considered as parameters that are learned during the training process.

We abstract the above Self-Attentive Span Extractor as:

$$\mathbf{h}_i^S = \mathrm{SpanExt}\left(\mathbf{h}_{i1}^B, \cdots, \mathbf{h}_{i\ell_i}^B\right) \tag{4}$$

After the span extraction step, the whole document is represented as a sequence of EDU representations: $\mathbf{h}^S = \left\{\mathbf{h}_1^S, \cdots, \mathbf{h}_n^S\right\} \in \mathbb{R}^{d_h \times n}$, which will be sent to the Autoencoder.

B. Autoencoder with Extreme Learning

Given an input data point x, a sequnece of EDU representations, the output of the Autoencoder with Extreme Learning (ELVA) model is given by a mapping function to M-dimensional random feature space:

$$f_M(x) = \sum_{i=1}^{M} \beta_i h_i(x) = h(x)\beta \tag{5}$$

where $\beta = [\beta_1, \ldots, \beta_M]^T$ is the output weight matrix between the hidden nodes and the output nodes, $h(x) = [h_1(x), \ldots, h_M(x)]$ are the hidden node outputs for input x, and $h_i(x)$ is the output of the ith hidden node. Given N training samples $\{(x_i, t_i)\}_{i=1}^{N}$, the following learning problem is addressed by ELM:

$$H\beta = T \tag{6}$$

where $\left[\ t_1, \ldots,\ t_N\ \right]^T$ are target labels, and $H = \left[\ h^T(x_1), \ldots,\ h^T(x_N)\ \right]^T$. The output weights matrix β is calculated using the following formulas:

$$\beta = H^{\dagger}T \tag{7}$$

where $H^{\dagger}$ is the Moore–Penrose generalized inverse (pseudoin-verse) of the output matrix H. While ELM (Extreme Learning Machine) offers notable advantages in terms of generalization and training speed, it often exhibits poor performance in terms of generalization. Deng et al. [28] tackle this issue by introducing a novel ELM variant known as Regularized Extreme Learning Machine (RELM). The RELM model aims to minimize the cost function associated with least squares estimation by incorporating a regularization coefficient C. The formulation incorporating this regularization coefficient is as follows:

$$\beta = \left(\frac{1}{C} + H^T H\right)^{-1} H^T T \tag{8}$$

Algorithm 1 provides an overview of the main steps involved in ELM. The original version of ELM is primarily designed to learn features from labeled data. However, with the increasing availability of unlabeled data in the era of digital transformation, there is a need for unsupervised techniques to learn, extract features, and reduce the dimensionality of such data. To address this challenge, Kasun et al. [29] introduced a new unsupervised variant of ELM called Extreme Learning Machine Autoencoder (ELAE). ELAE is a neural network that builds upon the ELM framework. It consists of a single hidden layer, where the input data is also the output. The initial weights and biases of the hidden nodes are randomly generated and should be orthogonal. The network architecture of ELM-AE is illustrated in Fig. 4.

The training process of an ELAE involves two main stages: the encoder stage and the decoder stage. In the encoder stage, the input features are transformed and mapped into an M dimensional feature space using one of three approaches, depending on the size of d and M: (1) Sparse architecture: When $d < M$, the encoder maps features from a lower-dimensional input data space to a higher-dimensional feature space. This allows for the representation of features in a more expressive and expanded feature space. (2) Compressed architecture: When $d > M$, the encoder compresses features from a higher-dimensional data space into a lower-dimensional feature space. This compression helps in reducing the dimensionality of the features while retaining their essential information. (3) Equal dimension: When $d = M$, the encoder represents features from an input data space dimension that is equal to the dimension of the feature space. This approach maintains the same dimensionality for both the input data and the feature space. These different approaches enable the ELAE to adapt to the specific requirements of the data and the desired dimensionality of the feature space.

In this particular study, the focus is on the compressed architecture of ELAE. In this architecture, the random orthogonal weights and biases of the hidden nodes are used to map the input data x_i to a lower-dimensional feature space M. This mapping is achieved using the following formula:

$$h(x_i) = g\left(a^T x_i + b\right) \tag{9}$$

$$a^T a = I, b^T b = 1 \tag{10}$$

where $a = [a_1, \ldots, a_M]$ are the orthogonal random weights, and $b = [b_1, \ldots,\ \ b_M]$ denotes the orthogonal random biases between the input and hidden nodes. $h(x_i) \in R^M$ corresponds to the output vector of the hidden layer concerning the input x_i ; $g(.)$ is an activation function which can be sigmoid, Gaussian function or so on; I is an identity matrix of order M. In this paper, the sigmoid function is used in the encoder stage of the ELAE:

$$\min_{\beta \in R^{M \times d}} L_{ELM-AE} = \min_{\beta \in R^{M \times d}} L_{ELM-AE} \frac{1}{2}\|\beta\|^2 + \frac{C}{2}\|X - H\beta\|^2 \tag{11}$$

where C is a penalty coefficient on the training errors. It balances experiential risk and structural risk. By setting the gradient of L_{ELM-AE} to zero, we have:

$$\beta + CH^T(X - H\beta) = 0 \tag{12}$$

According to the above equation, the output weights β of an ELAE can be computed in three different ways:

- When the number of training samples N is larger than the number of hidden layer nodes M, output weights are calculated by 13. This is a compressed ELAE representation.
- When the number of training samples N is smaller than the number of hidden layer nodes M, output weights are calculated by 14. This is a sparse ELAE representation.
- For equal dimension $(N = M)$, output weights can be expressed as 15. This is an equal ELAE representation.

$$\beta = \left(\frac{I_M}{C} + H^T H\right)^{-1} H^T X \tag{13}$$

$$\beta = H^T \left(\frac{I_N}{C} + HH^T\right)^{-1} X \tag{14}$$

$$\beta = H^{-1} X \tag{15}$$

where I_k is an identity matrix of dimension k. The primary objective of this paper is to utilize compressed data rather than the original input data for the automatic summarization task. Dimensionality reduction is achieved through the unsupervised ELAE, which involves projecting the input data X in the decoder stage. The new representation of the input data X

Algorithm 1 Extreme learning machine.

Input: Training set $S=\{(x_i, t_i)\}|\ x_i \in R^n, t_i \in R^m, i=1,\ 2,\ \ldots,\ N$, activation function $g(x)$, number of neurons in hidden layer M;
Output: Weight matrix β;
1. Initialize the input weight matrix W and hidden layer bias b with random values;
2. Using the activation function g, calculate the hidden layer output matrix H with:
$H=g(Wx+b)$
3. Calculate the network output weight matrix β using Eq. (10)

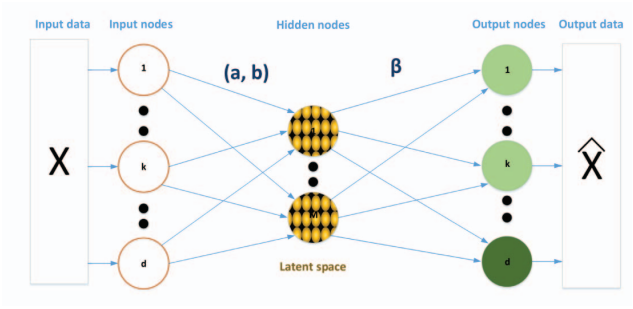

Fig. 4. ELAE model. The input X is the same as the output $\hat{X}$, (a, b) are the randomly generated hidden node parameters which are made orthogonal.

in the feature space of dimension n_h is determined by the following formula:

$$X_{new} = X\beta^T \tag{16}$$

Thereafter, the original data (X) is replaced by the new generated data (X_{new}) in the summarization task.

C. Voting based on Similarity and Summary generation

Each sentence EDU is projected into a concept space by a mapping function given by a specific model. An abstract representation h_i^* is produced and used in order to compute the similarity between two EDUs using the cosine similarity metric.

$$\text{sim}\left(h_i^*, h_j^*\right) = \frac{h_i^* h_j^*}{\|h_i^*\| \left\|h_j^*\right\|} \tag{17}$$

After measuring semantic similarity between the refined representations of EDUs, we sort the similarities in descending order, and select EDUs accordingly. Note that the dependencies between EDUs are also enforced in prediction to ensure grammacality of generated summaries.

IV. Experiments

In this section, we present the experimental results obtained from two widely used news summarization datasets. Our proposed model is compared against state-of-the-art baselines, and we perform a comprehensive analysis to validate the effectiveness of BERT-ELVA.

A. Datasets

We evaluate our models using two datasets: New York Times (NYT) and CNN and Dailymail (CNNDM). To extract summaries from the raw data, we utilize the script provided by See et al. [30]. For sentence boundary detection, tokenization, and parsing, we employ Stanford CoreNLP [31]. It's worth noting that BERT has a limitation in encoding up to 768 BERT BPEs. Regarding CNNDM, the dataset consists of 287,226 samples for training, 13,368 samples for validation, and 11,490 samples for testing. We utilize the un-anonymized version of CNNDM and for NYT, which is licensed by LDC6, we use 137,778 samples for training, 17,222 samples for validation, and 17,223 samples for testing, following the approach of Zhang et al. [11] and Xu et al. [32].

B. State-of-the-art Baselines

We compare our model with the following state-ofthe-art neural text summarization models.

- **Extractive Models**: **BanditSum**, introduced by Dong et al. [33], approaches extractive summarization as a contextual bandit problem and employs policy gradient methods for training. On the other hand, **NeuSum**, proposed by Zhou et al. [34], is an extractive model that utilizes a sequence-to-sequence (seq2seq) architecture. In NeuSum, the attention mechanism is utilized to score the document and determine the index of the selected content for summarization.
- **Compressive Models**: **JECS** is a neural text compression-based summarization model using BLSTM as the encoder [32]. The first stage is selecting sentences, and the second stage is sentence compression by pruning the constituency parsing tree.
- **BERT-based Models**: BERT-based models have achieved significant improvement on CNNDM and NYT, when compared with LSTM counterparts. **BERTSUM** is the first BERT-based extractive summarization model [10]. It is built on top of BERT encoder by stacking several inter-sentence Transformer layers. **PNBERT**, introduced by Zhong et al. [35], is a BERT-based model that incorporates reinforcement learning and Pointer Networks. This model aims to enhance the performance of extractive summarization. **HiBERT**, proposed by Zhang et al. [11], is a hierarchical BERT-based model specifically designed for document encoding. It extends the pretraining process to include unlabeled data, enabling the model to learn more comprehensive representations. **DiscoBERT** [19] presented an extractive approach with structural discourse graphs constructed based on RST trees and coreference mentions, encoded with Graph Convolutional Networks.

C. Implementation Details

In our implementation, we utilize AllenNLP [36] as the code framework. The length of each document is truncated to 768 BPEs to fit the model's input requirements. We employ the pre-trained 'bert-base-uncased' model and fine-tune it for all our experiments. The models are trained for a maximum of 80,000 steps. For evaluation, we utilize ROUGE [37] as the evaluation metric, specifically focusing on the 'R-2' measure for validation purposes. The realization of discourse units and structure plays a crucial role in the preprocessing of EDUs. This process involves two main steps: discourse segmentation and RST parsing. In the segmentation phase, we employ a neural discourse segmenter based on the BiLSTM CRF (Bidirectional Long Short-Term Memory Conditional Random Fields) framework, as proposed by Wang et al. [38]. ELAE indicates the system based on the extreme learning machine Autoencoder model. In this paper, the ELAE is composed of one hidden layer with 50 hidden units.

D. Experimental Results

Results on CNNDM Table I shows results on CNNDM. Among all the baselines, DiscoBERT demonstrates the best performance, which can be attributed to its approach of proposing a BERT discourse graph. BERTSUM, on the other hand, ranks second among the baselines due to its utilization of BERT and inter-sentence Transformer layers. BanditSum, employing policy gradient methods, exhibits the worst performance among the baselines. In contrast, NueSum outperforms BanditSum by leveraging attention mechanisms and a seq2seq architecture. JECS surpasses both BanditSum and NueSum in R-1 measure by incorporating sentence compression through the pruning of the constituency parsing tree. PNBERT outperformed HiBERT by utilizing Reinforcement Learning. In comparison, our proposed model outperforms all the baselines and achieves improvements in the R-1, R-2, and R-L. This improvement is attributed to the combination of a discourse BERT encoder and an Autoencoder with Extreme Learning for updating EDU presentations while capturing their long-range dependencies.

Results on NYT Table II shows results on NYT. Similar to results on the CNNDM dataset, DiscoBERT demonstrates the best performance. HiBERT is in second place and outperformed JECS because of utilizing unlabeled data and proposing a pretraining process on BERT. The proposed model (DBERT-ELVA) surpasses the previous state-of-the-art baseline models due to integrating the BERT discourse document encoding and Extreme Learning Autoencoder.

TABLE I
RESULTS ON THE TEST SET OF THE CNNDM DATASET. ROUGE-1, -2 AND -L F1 ARE REPORTED.

Model	R-1	R-2	R-L
NeuSum [34]	41.59	19.01	37.98
HiBERT [11]	42.37	19.95	38.83
BanditSum [33]	41.50	18.70	37.60
JECS [32]	41.70	18.50	37.90
PNBERT [35]	42.39	19.51	38.69
PNBERT w. RL	42.69	19.60	38.85
BERTSUM [10]	43.25	20.24	39.63
DiscoBERT [19]	43.77	20.85	40.67
BEET-ELVA	**43.92**	**20.88**	**41.01**

TABLE II
RESULTS ON THE TEST SET OF THE NYT DATASET.

Model	R-1	R-2	R-L
JECS [32]	45.50	25.30	38.20
HiBERT [11]	48.38	29.04	40.53
DISCOBERT [19]	50.00	30.38	42.70
BERT-ELVA	**50.44**	**30.67**	**43.11**

V. CONCLUSION

In this paper we introduced DBERT-ELVA, a novel approach that employs discourse units as the minimal selection basis to mitigate redundancy in summarization. The model utilizes an Autoencoder with Extreme Learning to generate compressed representations and effectively capture long-range dependencies among the discourse units. To validate the effectiveness of our proposed approach, we conduct experiments on two well-known summarization datasets, and consistently observe improvements compared to baseline models. As future directions, we aim to explore the potential of large language models and graph encoding techniques. Additionally, we plan to extend the application of discourse encoding to other tasks that involve encoding lengthy documents.

REFERENCES

[1] N. Moratanch and S. Chitrakala, "A survey on abstractive text summarization," in *2016 International Conference on Circuit, power and computing technologies (ICCPCT)*. IEEE, 2016, pp. 1–7.

[2] A. Sinha, A. Yadav, and A. Gahlot, "Extractive text summarization using neural networks," *arXiv preprint arXiv:1802.10137*, 2018.

[3] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert systems with applications*, vol. 165, p. 113679, 2021.

[4] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, "Text summarization techniques: a brief survey," *arXiv preprint arXiv:1707.02268*, 2017.

[5] N. Moratanch and S. Chitrakala, "A survey on extractive text summarization," in *2017 international conference on computer, communication and signal processing (ICCCSP)*. IEEE, 2017, pp. 1–6.

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

[7] W. C. Mann and S. A. Thompson, "Rhetorical structure theory: Toward a functional theory of text organization," *Text-interdisciplinary Journal for the Study of Discourse*, vol. 8, no. 3, pp. 243–281, 1988.

[8] L. Carlson, D. Marcu, and M. E. Okurowski, "Building a discourse-tagged corpus in the framework of rhetorical structure theory," *Current and new directions in discourse and dialogue*, pp. 85–112, 2003.

[9] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[10] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," *arXiv preprint arXiv:1908.08345*, 2019.

[11] X. Zhang, F. Wei, and M. Zhou, "Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization," *arXiv preprint arXiv:1905.06566*, 2019.

[12] W. Xiao and G. Carenini, "Extractive summarization of long documents by combining global and local context," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3011–3021.

[13] ——, "Systematically exploring redundancy reduction in summarizing long documents," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 2020, pp. 516–528.

[14] A. Louis, A. Joshi, and A. Nenkova, "Discourse indicators for content selection in summarization," in *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2010, pp. 147–156.

[15] T. Hirao, Y. Yoshida, M. Nishino, N. Yasuda, and M. Nagata, "Single-document summarization as a tree knapsack problem," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1515–1520.

[16] Y. Yoshida, J. Suzuki, T. Hirao, and M. Nagata, "Dependency-based discourse parser for single-document summarization," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1834–1839.

[17] G. Durrett, T. Berg-Kirkpatrick, and D. Klein, "Learning-based single-document summarization with compression and anaphoricity constraints," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1998–2008.

[18] J. J. Li, K. Thadani, and A. Stent, "The role of discourse units in near-extractive summarization," in *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2016, pp. 137–147.

[19] J. Xu, Z. Gan, Y. Cheng, and J. Liu, "Discourse-aware neural extractive text summarization," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 5021–5031.

[20] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2011.

[21] G. Huang, S. Song, J. N. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE transactions on cybernetics*, vol. 44, no. 12, pp. 2405–2417, 2014.

[22] A. D. Andrushia and R. Thangarajan, "Visual attention-based leukocyte image segmentation using extreme learning machine," *International Journal of Advanced Intelligence Paradigms*, vol. 7, no. 2, pp. 172–186, 2015.

[23] A. Iosifidis, A. Tefas, and I. Pitas, "Human action recognition based on multi-view regularized extreme learning machine," *International Journal on Artificial Intelligence Tools*, vol. 24, no. 05, p. 1540020, 2015.

[24] A. A. Mohammed, R. Minhas, Q. J. Wu, and M. A. Sid-Ahmed, "Human face recognition based on multidimensional pca and extreme learning machine," *Pattern recognition*, vol. 44, no. 10-11, pp. 2588–2597, 2011.

[25] H. Huang, H. Ma, H. J. van Triest, Y. Wei, and W. Qian, "Automatic detection of neovascularization in retinal images using extreme learning machine," *Neurocomputing*, vol. 277, pp. 218–227, 2018.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[27] K. Lee, L. He, M. Lewis, and L. Zettlemoyer, "End-to-end neural coreference resolution," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017.

[28] W.-Y. Deng, Q.-H. Zheng, L. Chen, and X.-B. Xu, "Research on extreme learning of neural networks," *Chinese Journal of Computers*, vol. 33, no. 2, pp. 279–287, 2010.

[29] L. L. C. KASUN, H. ZHOU, G.-B. HUANG, and C. M. VONG, "Representational learning with elms for big data," *IEEE intelligent systems*, vol. 28, no. 6, pp. 31–34, 2013.

[30] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1073–1083.

[31] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 2014, pp. 55–60.

[32] J. Xu and G. Durrett, "Neural extractive text summarization with syntactic compression," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3292–3303.

[33] Y. Dong, Y. Shen, E. Crawford, H. van Hoof, and J. C. K. Cheung, "Banditsum: Extractive summarization as a contextual bandit," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3739–3748.

[34] Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, and T. Zhao, "Neural document summarization by jointly learning to score and select sentences," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 654–663.

[35] M. Zhong, P. Liu, D. Wang, X. Qiu, and X.-J. Huang, "Searching for effective neural extractive summarization: What works and what's next," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1049–1058.

[36] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. F. Liu, M. E. Peters, M. Schmitz, and L. Zettlemoyer, "Allennlp: A deep semantic natural language processing platform," in *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, 2018, pp. 1–6.

[37] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

[38] Y. Wang, S. Li, and J. Yang, "Toward fast and accurate neural discourse segmentation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 962–967.