Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 3591894, 12 pages
https://doi.org/10.1155/2021/3591894

Research Article

Automatic Generation of the Draft Procuratorial Suggestions Based on an Extractive Summarization Method: BERTSLCA

Yufeng Sun,[1] **Fengbao Yang**,[1] **Xiaoxia Wang**,[1] **and Hongsong Dong**[1,2]

[1]*School of Information and Communication Engineering, North University of China, Taiyuan 030051, China*
[2]*Department of Computer Science, Lüliang University, Lüliang 033000, China*

Correspondence should be addressed to Fengbao Yang; yfengb@163.com

Received 10 April 2021; Revised 21 May 2021; Accepted 8 June 2021; Published 16 June 2021

Academic Editor: Ali Ahmadian

The automatic generation of the draft procuratorial suggestions is to extract the description of illegal facts, administrative omission, description of laws and regulations, and other information from the case documents. Previously, the existing deep learning methods mainly focus on context-free word embeddings when addressing legal domain-specific extractive summarization tasks, which cannot get a better semantic understanding of the text and in turn leads to an adverse summarization performance. To this end, we propose a novel deep contextualized embeddings-based method BERTSLCA to conduct the extractive summarization task. The model is mainly based on the variant of BERT called BERTSUM. Firstly, the input document is fed into BERTSUM to get sentence-level embeddings. Then, we design an extracting architecture to catch the long dependency between sentences utilizing the Bi-Long Short-Term Memory (Bi-LSTM) unit, and at the end of the architecture, three cascaded convolution kernels with different sizes are designed to extract the relationships between adjacent sentences. Last, we introduce an attention mechanism to strengthen the ability to distinguish the importance of different sentences. To the best of our knowledge, this is the first work to use the pretrained language model for extractive summarization tasks in the field of Chinese judicial litigation. Experimental results on public interest litigation data and CAIL 2020 dataset all demonstrate that the proposed method achieves competitive performance.

1. Introduction

Procuratorial suggestions serve as a means for the people's procuratorates to supervise administrative organs, which protect the public interests from harm. With the development of the economy and society, more and more public interest litigation cases flood into the court, which brings a great burden to the case-handling personnel. Case investigators need to summarize some summary characteristics of the text from the complex case to form a procuratorial suggestion document. In such a background, the technique of automatic text summarization has attracted more and more attention from researchers [1–3].

Recently, automatic text summarization has been used in many fields, such as biomedical domain [4, 5], medical domain [6–8], and meteorological domain [9]. Automatic procuratorial suggestion document summarization belongs to another application of automatic document summarization, which is to generate short and simplified summaries as procuratorial suggestions from public interest litigation document.

According to the summary result, document summarization is mainly divided into abstractive summarization [10, 11] and extractive summarization [12–16]. The extractive summarization task is to extract sentences that represent the important information from the original document as the summary, while abstractive summarization aims to generate words, phrases, and sentences that do not appear in the original document to compose the summary. Extractive summarization techniques are more attractive and wildly used in terms of difficulty, especially abstractive summarization method also faces the problem of lack of relevant data in the Chinese domain. In this paper, we focus on the extractive summarization method for the Chinese public interest litigation cases document.

Researchers have explored many techniques for extractive summarization methods. Previously, some researchers proposed statistical approaches for extraction tasks. The statistical approaches are mainly based on term frequency, word importance, and position information features of sentences. Alone et al. [17] developed a DimSum system that employed TF-IDF features to capture important words in a sentence. Matsuo et al. [18] made use of word cooccurrence to catch keywords from the text and achieved higher performance than the TF-IDF feature-based method. The essence of the statistical methods is based on the concept that the word importance determines the significance of the sentences.

With the development of machine learning techniques in NLP, some publications appeared that combined statistical features and machine learning to obtain document extracts. Methods such as naïve Bayes classifiers [19], decision trees [20], and hidden Markov models [21] are combined with hand-designed feature to explore document extraction task. The above methods need human-annotated features, which is time-consuming. The emergence of deep learning methods has attracted attention from manually designed features to data-driven approach, which helps the network to classify whether the sentence belongs to summary or not automatically. In the concept of deep learning methods, with sufficient data, the network learns to represent words with vectors, thus capturing syntactic and semantic information, which serves as features to conduct a summarization task. Convolutional neural network (CNN) [22] has proved to be a powerful feature extractor in the natural language processing (NLP) field. Another classic network is the recurrent neural network [23] that captures the sentence-level feature regardless of the sentence length. At each time step, the network tries to understand the present word with the memory of previous words, which captures the long-distance dependency. Researchers have explored CNN and RNN in extractive document summarization tasks and achieved great improvement [24].

However, the above method is based on the word2vec approach, which cannot solve the problem of polysemy, and the features extracted on this basis cannot accurately express the meaning of the sentence. To solve the problem, Liu [25] employed a BERT-based method called BERTSUM to conduct an extractive document summarization task. Bidirectional Encoder Representations from Transformers (BERT) is a new language model created by Google researchers, Devlin et. al [26], which obtains the state-of-the-art results on 11 NLP tasks. BERT is mainly based on the encoder of Transformer [27]; the word BERT represents taking the polysemy into account which helps to improve the downstream task. Since BERT has no decoding component, it is not suitable for generating tasks. Liu made several modifications to make BERT possible for the extraction task. They modify the input sentence and encode it in sentence units to produce document-level output, and also contextual information is taken into account. At the summarization layer, they employ RNN and Transformer layers to fine-tuning the summarization layers and achieve an outstanding result on CNN/Dailymail dataset.

In our work, we investigate the usefulness of BERTSUM in the Chinese domain, especially in Chinese public interest litigation cases document. However, some challenges exist if BERTSUM is directly fine-tuned to the extraction task. (1) BERTSUM achieves good performance based on combining with Transformer, which makes the model more complex, requires a complex network to fit the data, and ultimately takes more time to train. (2) Public interest litigation cases are often multisentence documents. The interrelationship between sentences is ignored, which affects the expression of document information and may eventually reduce the index value of the extraction task.

To solve the above limitations, we propose an improved BERTSUM called BERTSLCA in this paper, which obtains better quality of multisentences document summarization. BERTSLCA first employs BERTSUM for sentence-level encoding and obtains sentence representations for documents, which allows modeling the relationship between sentences and summaries. Secondly, we introduce BiLSTM as the first component of the extraction layer to catch the long dependency relations between different sentences and then three layers of convolutional neural networks with different kernel size cascading designs as the second component of the extraction layer to get the relationship between adjacent sentences. Last, the attention mechanism enables the model to assign different attention scores to different sentences, and the important sentences tend to have higher scores, which also conforms to the process of human understanding of documents. In summary, the contribution of the paper is as follows:

(1) An improved BERTSUM model BERTSLCA is designed, which is capable of conducting extractive summarization to generate procuratorial suggestions from public interest litigation documents with many sentences and long ones.

(2) We investigate the usefulness of the advanced BERTSUM model for document representation. Sentence embeddings are initialized using interval segment embeddings. We introduce a new extraction mechanism to capture relations between sentences within multisentence documents. Not only the long dependency between sentences is considered, but also the relationship between adjacent sentences is extracted.

(3) Experiment results show that the proposed method achieves competitive results compared to other baselines. Compared to the previous best-performing method BERTSUM with transformer, our method takes less time while ensuring the quality of summary generation.

The reminder of the paper is listed as follows. Section 2 describes related works. Section 3 presents the details of the proposed method. Section 4 introduces the experiment results and analysis. Finally, Section 5 presents the conclusion.

2. Related Works

This section is to show the techniques that are relevant to extractive summarization.

2.1. Word Representations. Text summarization is one of the difficult subtasks of natural language processing, as with any other subtask such as text classification and reading comprehension, the text needs to be represented correctly. So, our work is closely related to word representations [28]. Previous work has shown that establishing a good language model for the correct representation of text can improve downstream tasks. One-hot vector is a binary encoding method, which is obtained by establishing a vocabulary, then setting the index dimension of each word to 1 according to the vocabulary, and setting the other elements to 0. Although one-hot vector is the simplest and most efficient way to code, it suffers from the data-sparse problem and does not consider the similarity between words. To solve the problem, scholars put forward the word2vec method (e.g., CBOW and skip-gram model) to express words [29]. Vectors generated by word2Vec are a low-latitude, dense representation that takes the semantic information of words into account, but one of the shortcomings is that the word vector it generates is static and does not consider the polysemy problem.

Researchers suggest that pretrained word embeddings on large corpora can elevate performance on different NLP tasks [30, 31]. BERT is one of the most successful pretrained methods and mainly based on the transformer model, which is a bidirectional representation language model and considers both the left and right context information. It is this language model structure that makes BERT achieve overwhelming results in many fields, such as disease diagnosis [32], irony detection [33], and image processing [34]. The word embeddings obtained by BERT is dynamic and has multiple representations of a word. The word can be represented dynamically according to the context of the word, thus solving the polysemy problem, which is more in line with human's understanding of language. BERT encodes word vectors in word units, which is very suitable for text classification tasks [35], and sentence units are more suitable for text summarization tasks at the sentence level. The latest research described a BERTSUM model, a variant of BERT, in which sentence-level features are obtained by coding with the sentence as a unit. Inspired by this, our method is mainly based on BERTSUM for sentence-level embeddings.

2.2. Text Summarization. Previously, researchers mainly focused on machine learning methods on summarization tasks on the legal document. For instance, Hachey et al. [36] conduct rhetorical role classification experiments for legal summarization. They chose location, thematic words, sentence length, quotation, and so on as feature sets for the SVM classifier. To judge the rhetorical category of the current sentence by its features and choose the sentence that is the most summary-worthy sentence. Saravanan et al. [37] propose an approach to identify rhetorical roles from legal documents. They ask for law experts to annotate rhetorical roles and generate training data. Then, conditional random field model is taken as the classifier to identify the rhetorical role with rich features. The application of the machine learning model brings out an effective summary in the final stage. However, the approaches in this direction have relied on hand-engineered features and large labeled data, which makes it restricted to a narrow subdomain.

Recently, researchers have explored deep learning methods on the task of legal text summarization and have achieved great performance. Anand et al. [38] proposed a model that utilizes CNN and RNN for the Indian legal judgment document summarization task. Huang et al. [39] presented a sequence-to-sequence model for the summarization of legal public opinion news by incorporating topic information. Their model is mainly based on an RNN-based encoder-decoder architecture. To improve the understanding ability of the model, they selected topic words as domain information to assist the model and achieved good performance under the rouge metrics. However, the abovementioned deep learning methods are mainly dependent on the word2Vec representation method and there exists a problem of insufficient understanding of the text, which affects the results of the follow-up summary task.

In the general domain, researchers use the powerful feature extraction ability of BERT to complete the text summarization task in related fields. Moradi et al. [5] proposed to utilize BERT to get contextualized embeddings without the need for incorporating domain knowledge and annotating in biomedical text summarization tasks. Shi et al. [9] presented a summarization model to automatically summarize Weibo posts in the meteorological domain, which took BERT as an encoder. The BERT model is mainly utilized to get the prior meteorological domain knowledge and initialize the word embedding. However, these models are mainly based on word-level BERT, and the relationship between the sentences is ignored. Also, they rarely focus on the task of a specific domain.

Oriented to the needs of the judicial field and inspired by the general domain, our method employs context- and sentence-aware embeddings BERTSUM. However, in order to enhance the ability of text feature extraction and further obtain the relationship between sentences, we have constructed a high-performance extracting layer to aggregate sentence information and capture document-level features for extracting summaries. The details are shown in Section 3. In this paper, we pay close attention to Chinese legal documents. For all we know, our work represents the first attempt to use the contextualized language model for extractive summarization tasks in the field of the Chinese legal domain.

3. Proposed Method: BERTSLCA

In this section, the proposed method is introduced. The architecture of the method contains five parts: model pretraining layer, an embedding layer, an encoding layer, an extracting layer, and the output layer.

(I) Model pretraining layer is to transfer the knowledge learned from larger datasets to the downstream task.

(II) Embedding layer which turns each word in the sentence to a vector by the BERTsum embedding manner rather than original BERT.

(III) Encoding layer first established by the BERTsum architecture, which also consists of several transformer encoder blocks as BERT.

(IV) Extracting layer contains three components: BiLSTM, convolutional gated unit, and attention mechanism. BiLSTM aims to capture the interaction information among different input sentences, and the convolutional gated unit is to catch the local information within sentences. Attention is to strengthen the ability to retain important sentences.

(V) Output layer which predicts whether the present sentence belongs to the summary.

The overall framework is shown in Figure 1, and the details are introduced in the following.

3.1. Pretraining Layer. The essence of deep learning is to optimize the value of weight parameters so that it can reach an optimal solution state. Among them, the layers that need to update the weight include the encoding layer, batch normalization layer, and fully connect layer. In the process of optimization, the initialization of weight is an important step to get the optimal solution. If the weight initialization is not appropriate, the model may fall into the local optimal solution, which leads to a bad prediction effect and even the oscillate loss function; as a result, the model does not converge.

The model pretraining aims to accelerate the training process, make the model converge faster, and improve the performance of the downstream task. The pretraining model provides weight parameters that are trained from a large amount of data, which is usually used to initialize the model parameters in the downstream task, to make full use of the language knowledge learned in the pretraining process and transfer it to the learning of the downstream task.

The pretraining model is mainly based on the autoencoding language model, which is shown in

$$\max_{\theta}\left(\log\ p_{\theta}(\overline{x}|\widetilde{x})\right) \approx \sum_{t=1}^{T} m_t \log\ p_{\theta}(\overline{x}|\widetilde{x}) = \sum_{t=1}^{T} m_t \log \frac{\exp\left(H_{\theta}(\widetilde{x}_t)^T e(x_t)\right)}{\sum_{x'} \exp\left(H_{\theta}(\widetilde{x}_t)^T e(x')\right)}, \tag{1}$$

where $\hat{\mathrm{x}}$ denotes the context of the current word, T is the number of the words in the sentence, θ is the parameter in the process, $H_{\theta}(\widetilde{\mathrm{x}}_t)$ denotes the probability density, and $m_t$represents whether the current word is masked. The essence of the autoencoder language model is to mask some words randomly, and the training goal is to maximize the prediction of the masked words according to the context.

In the research, the training process is associated with two tasks: masked language model (MLM) and next sentence prediction (NSP). MLM pretraining objective is introduced to get a context-dependent bidirectional feature representation. In the input sequence, 15% of the words are randomly blocked, and the task is to predict those words. NSP is to model the relationship between two sentences, which helps to understand the nature of the language modeling. To be specific, NSP is to predict whether the latter one is the next sentence of the former one.

3.2. Embedding Layer. The input of our task is at the document level, which may consist of different numbers of sentences for each document, so for the embedding layer, a document-level embedding method is needed. Vanilla BERT is a word embedding-based method, which is suit for sentence and paragraph level tasks and not for document-level tasks such as long text classification or extractive summarization. Inspired by Liu, we apply the improved BERT method, that is, BERTSUM, to obtain the sentence vectors, which aims to capture document-level features. Here, BERT and BERTSUM share different input representations. Assume a document consists of n sentences $S = \{S_i\}_{i=1}^{n} = \{S_1, S_2, ..., S_n\}$; BERTSUM gets the following input document ID:

$$\mathrm{ID} = \langle CLS\rangle, S_1, \langle SEP\rangle, \langle CLS\rangle, S_2, \langle SEP\rangle, ..., \langle CLS\rangle, S_n, \langle SEP\rangle, \tag{2}$$

Here, $\langle CLS\rangle$is a symbol of each sentence for classification or summarization, and $\langle CLS\rangle$is a token to separate different sentences.

The embedding layer is to transform the document to the vectors, and the input document vectors $\mathrm{IDV} = \{\mathrm{IDV}_i\}_{i=1}^{n} = \{IDV_1, IDV_2, ..., IDV_n\}$, for IDV_i; its vector is constructed as follows:

$$\mathrm{IDV}_i = \mathrm{ID}_i^{\mathrm{tok}} + \mathrm{ID}_i^{\mathrm{seg}} + \mathrm{ID}_i^{\mathrm{pos}}, \tag{3}$$

where $\mathrm{ID}_i^{\mathrm{tok}}$, $\mathrm{ID}_i^{\mathrm{seg}}$, and $\mathrm{ID}_i^{\mathrm{pos}}$depict token, segment, and position embeddings for sentence ID_i, as shown in Figure 1.

3.3. Encoding Layer. This section uses BERT to encode sentences and obtains the final document features. BERT or BERTSUM shares the same architecture that consists of multiple transformers. The document vector $\mathrm{IDV}_i$is fed into L successive transformer blocks. Each block has two main components: a multihead attention mechanism and a fully connected feed-forward network. The details about the transformer block can be found in [27]. After L successive transformer blocks, the input document vector $\mathrm{IDV} = \{\mathrm{IDV}_i\}_{i=1}^{n} = \{IDV_1, IDV_2, ..., IDV_n\}$ is encoded to deeper feature representations $E^L = \{E_i^L\}_{i=1}^{n} = \{E_1^L, E_2^L, \ldots, E_n^L\}$:

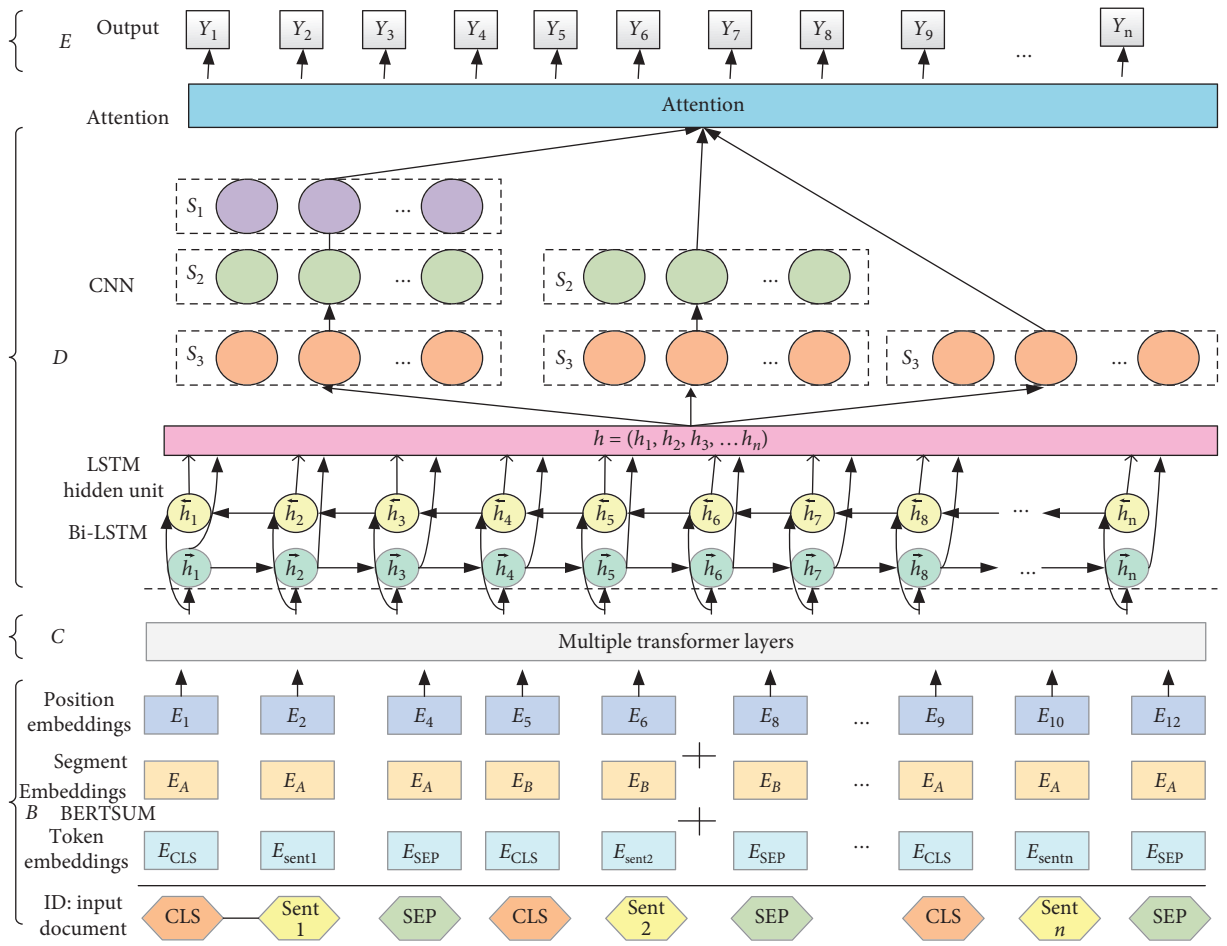

FIGURE 1: The proposed method consists of five parts: (a) the pretraining layer, (b) the embedding layer, (c) the encoding layer, (d) the extracting layer, and (e) the output layer; for simplicity, the pretraining layer is omitted.

$$\begin{cases} E^L = \text{Transformer}^L\left(E^{L-1}\right) = \text{Transformer}^L\left\{E_1^{L-1}, E_2^{L-1}, \ldots, E_n^{L-1}\right\}, & 1 < L \le n, \\ E^L = \text{Transformer}^L\left(E^0\right) = \text{Transformer}^L\{I\ DV_1, I\ DV_2, \ldots I\ DV_n\}, & L = 1, \end{cases} \tag{4}$$

where E_i^L denotes sentence vector $\text{ID}V_i$ being processed by L transformer blocks, and the process is shown in Figure 2.

3.4. Extracting Layer. After the sentence vectors are encoded with document-level features, the document feature representations E_L are fed into an extracting layer that consists of three parts: BiLSTM unit, convolutional gated unit, and attention mechanism. BiLSTM is used to encode the document representation further and capture the long-range information between different sentences within the document. To obtain the local features of the sentence, a convolutional gated unit is applied on the top of the BiLSTM unit. Attention is used to strengthen the ability to distinguish sentences.

The LSTM unit is used to capture the long dependency between sentences, and we adopt the following implementation:

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f \cdot x_t + \mathbf{W}_{f'} \cdot \mathbf{h}_{t-1} + \mathbf{b}_f\right), \tag{5}$$

$$\mathbf{i}_t = \sigma\left(\mathbf{W}_i \cdot x_t + \mathbf{W}_{i'} \cdot \mathbf{h}_{t-1} + \mathbf{b}_i\right), \tag{6}$$

$$\mathbf{o}_t = \sigma\left(\mathbf{W}_o \cdot x_t + \mathbf{W}_{o'} \cdot \mathbf{h}_{t-1} + \mathbf{b}_o\right), \tag{7}$$

$$\widetilde{c}_t = \tanh\left(\mathbf{W}_c \cdot x_t + \mathbf{W}_{c'} \cdot \mathbf{h}_{t-1} + \mathbf{b}_c\right), \tag{8}$$

$$c_t = \mathbf{f}_t \otimes c_{t-1} + \mathbf{i}_t \otimes \widetilde{c}_t, \tag{9}$$

$$\mathbf{h}_t = \mathbf{o}_t \otimes \tanh\left(c_t\right), \tag{10}$$

where W and b are the weight and bias parameters need to train, x_t is the output of the encoding layer, h_{t-1} is the hidden state at a time $t-1$, c_t is the memory cell state, f_t, i_t, and o_t are the forgetting gate, input gate, and output gate, $\widetilde{c}_t$ is the output of the memory gate at the time t, and h_t is the hidden state at the time$t$.

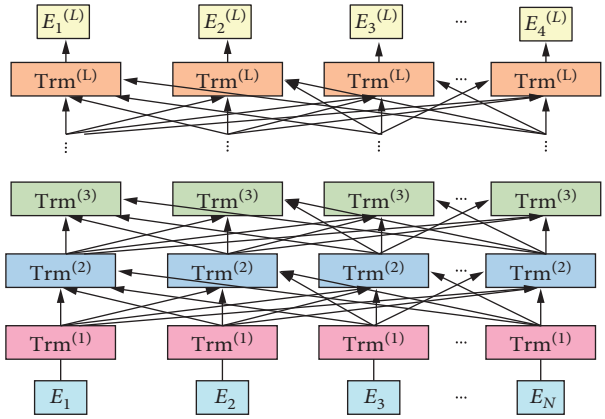

Figure 2: Multiple Transformer layers.

The convolution operation is to capture the correlations between adjacent sentences; even those sentences are spaced 2,3,4 apart but can be achieved by the given convolution kernel different filter sizes. The input of this section is $h = \{h_i\}_{i=1}^{n} \in R^{n\times \dim}$, the convolution kernel $W \in R^{s\times \dim}$, and s is the filter size (receptive field). The convolution between h and W generates a feature map $c = \{c_i\}_{i=1}^{n-s+1} \in R^{n-s+1}$; each feature c_i is calculated as follows:

$$c_i = f\left(W_s * h + b_s\right)_i = f\left(W_s \otimes h_{i:\, i+s-1} + b_s\right), \tag{11}$$

where W_s and b_s are weight and bias parameters, $h_{i:\, i+s-1}$ is the concatenation of vector h_i to h_{i+s-1}, which is also the same dimension with W_s, $\otimes$ is dot product operation, and f is the nonlinear activation function.

In our work, we design the convolution of three layers of connection, and the size of each layer is different. The specific connection mode is as follows: the first layer has m convolution kernels and the size of the kernel is S_1. For the second layer, there are $2 * m$ convolution kernels (the size of the first m convolution kernels is S_1, and the size of the last m convolution kernels is S_2). The third layer has $3 * m$ convolution kernels (the sizes of each m convolution kernel are S_1, S_2, and S_3 respectively). The output of BILSTM goes through the first-layer convolution kernel and gets the output O_1; the output of the second-layer convolution kernel is O_2, and for the third layer, the output is O_3. The final output of the convolutional neural network is the joint of the three, which can be denoted by c in (11).

After several validation tests, we use query-free type attention, which is originally from [40]. This mechanism learns contributions different sentences are made without query vector, which reduces the computational complexity. Take the CNN output as input of the attention component, which generates a new rating mechanism:

$$m(c) = \tanh(c), \tag{12}$$

Then, $m(c)$ is processed with softmax function to get the distribution of attention α:

$$\alpha = \text{soft max}\left(W^T m(c)\right), \tag{13}$$

where W^T is the trainable parameters of the softmax layer. Lastly, the final representation of the document is a weighted sum of the feature map vectors:

$$h' = \text{attention}(c) = c\alpha^T, \tag{14}$$

3.5. Output Layer. After the attention layer, the final representation of the document is fed into the final output layer, and the final output layer is a sigmoid classifier as (9):

$$\widehat{Y}_i = \text{sigmoid}\left(W^o h' + b^o\right), \tag{15}$$

where the sigmoid function maps the predicted score to an interval from 0 to 1. W^o and b^o are the weights and bias parameters the model needed to train. $\widehat{y}_i$ depicts the predicted score $\widehat{y}_1, \widehat{y}_2, \ldots, \widehat{y}_n$ ($\widehat{y}_i \in [0, 1]$) for each sentence in the document; those parameters are learned at the fine-tuning stage of the BERTSUM.

In the training stage of the model, we employ cross-entropy as the loss of the model, which is shown in (10):

$$L = \sum_i -y_i \log \widehat{y}_i - (1 - y_i)\log\left(1 - \widehat{y}_i\right), \tag{16}$$

where y_i (either 0 or 1) is the ground-truth label of the sentence in the document. The training objective of the model is to minimize the cross-entropy loss function of the predicted score and real tags.

4. Experiments

In this part, we introduce the dataset and data preprocessing. Then, we give the related parameter settings and evaluation measures. Finally, we conduct comprehensive experiments, which consist of comparisons between our method and baseline models, training loss convergence comparison, several ablation studies, and investigations of running efficiency. To verify the effectiveness of the method in this paper, we experiment on the open data set CAIL2020 in the legal domain. Finally, a statistical test is carried out.

4.1. Dataset and Data Preprocessing. The datasets of the experiment are from the people's procuratorate and related websites (https://"http://www.itslaw.com/home", https://wenshu.court.gov.cn/). There are five types of original datasets, including environmental pollution, resource destruction, food and drug safety, state-owned property protection, and state-owned land use transfer. Considering that environmental protection cases have attracted more attention in China, it is of greater significance to study this type of case. We conducted experiments on environmental pollution data, including a total of 4,000 samples, and the training set and the test set are divided according to the ratio of 5 : 1. The types of environmental pollution mainly include air pollution, air pollution, water pollution, soil pollution, and solid waste pollution. Each sample is a description of pretrial review and the corresponding draft

recommendations for prosecution is considered as extractive summarization (reference summary). And the reference summaries are annotated by PhD in law from the University of Political Science and Law.

The extractive text summarization is transformed into a binary classification task; for the sentences in the document, we need to consider whether the sentence belongs to the summarization. The aim of data preprocessing is to mark those sentences that belong to the summarization as 1 and others to 0.

Based on the pretrial review document and draft recommendations for prosecution (serve as reference summary), we obtain the ground truth summary, which is achieved by the greedy algorithm, and the main principle is to calculate the ROGUE between the reference summary and the sentences in the document. To be specific, an example is given in Table 1.

4.2. Parameter Settings and Evaluation Measures. The experimental environment is based on windows10, PyTorch, an NVIDIA with a 12 GB GPU. ***1*** . The "BERT-base" version is employed to implement the model, with a number of layers $L = 12$, hidden size $H = 768$, and the number of attention heads $A = 12.2$. For the BiLSTM layers, the hidden size of the BiLSTM unit is 768, and for the CNN unit, the filter size is 3, 5, 7. For the optimization method, Adam optimizer is used with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

Since the summarization task is based on an extractive method, we conduct the experiments with three metrics ROGUE-1, ROGUE-2, and ROGUE-L that are wildly used in the area. The ROGUE-1 and ROGUE-2 are calculated as follows:

$$P = \frac{\text{extracted summary} \cap \text{provided summary}}{\text{extracted summary}}, \tag{17}$$

$$R = \frac{\text{extracted summary} \cap \text{provided summary}}{\text{provided summary}}, \tag{18}$$

$$F_1 = \frac{2PR}{P + R}. \tag{19}$$

Here, ROGUE-1 is used to measure the unigram overlap between extracted summary and provided a summary, and ROGUE-2 is to measure the bi-gram overlap.

ROGUE-L is calculated as in the following formula:

$$P_{lcs} = \frac{LCS(X, Y)}{|X|}, \tag{20}$$

$$R_{lcs} = \frac{LCS(X, Y)}{|Y|}, \tag{21}$$

$$F_{lcs} = \frac{\left(1 + \beta^2\right) R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}}. \tag{22}$$

Here, ROGUE-L is used to calculate the longest common subsequence between extracted summary X and provided a summary Y, $|X|$ denotes the number of words in the extracted summary, and $|Y|$ is the number of words in the provided summary. β is a default parameter which sets to 1.

4.3. Experimental Results

4.3.1. Baselines. To demonstrate the effectiveness of the proposed method, we compared our method with other methods:

LEAD : This model is proposed by [41], which is a baseline for the summarization task.

NEURALSUM: This is an extractive summarization method composed of a CNN sentence encoder and a recurrent document encoder [42].

TextRank : This is a model from [43], which is a graph-based method and uses TextRank as a basis for sorting.

BERTSUM + fully connected classifier: This is a BERT-based method proposed by Liu [24], which employs BERTSUM as an encoder.

BERTSUM + Transformer: Liu [25] proposes using a Transformer replace classifier which achieves outstanding results.

BERTSUM + LSTM : Liu [25] proposes using BiLSTM to replace fully connected classifier.

BERTSUM + CNN: This is an ablation study of our proposed method.

BERTSUM + BiLSTM + CNN: This also is an ablation study of our proposed method.

4.3.2. Comparing to Baselines. This subsection investigates the result and analysis between our method and other baseline methods. Tables 2–4 show P, R, and $F1$ of the ROGUE-1, ROGUE-2, and ROGUE-L indexes between different methods. There are several observations in this part.

First, our proposed method performs best among all the methods on almost all metrics. For instance, our method achieves 51.0%, 19.3%, and 28.1% on ROGUE-1, 15.2%, 17.5%, and 16.3% on ROGUE-2, and 22.6%, 31.1%, and 26.2% on ROGUE-L, which demonstrate the effectiveness of our method; especially, our model improves the F1 index. This is mainly because of the context-dependent representations and document-level deep feature extraction that provide sufficient syntactic and semantic information.

Second, we compare to BERT-based methods. Our method goes beyond the previous best-performing method BERTSUM + Transformer, which gains 0.6%, 1%, and 2.4% on ROGUE-1-F1, ROGUE-2-F1, and ROGUE-L (from 27.5% to 28.1%, from 15.3% to 16.3%, and from 23.8% to 26.2%). This indicates that when substituting BiLSTM + CNN + attention for transformer, the model performs better and obtains the highest index. This mainly owes to the powerful feature extraction ability the former shows. BiLSTM learns the dependencies between different sentences within the document, followed by CNN to catch the N-gram features (such as 2, 4, and 6), and attention strengthens the ability. The combination shows better

Table 1: An example of annotating the ground truth labels for the document.

Document	According to the investigation conducted by the institute, it is found that **the xx road sub-district of XX district of XX shall be responsible for the comprehensive coordination of pollution prevention and control, carry out related greening work, and do a good job in the source management and coordination of construction waste treatment within its jurisdiction**. In XX lane XX road several points in the residential area exist the problem of open-air stacking of construction waste, which not only occupies the residential greening or public road resources but also easily causes dust pollution and damages the environment.
Reference summary	Your sub-district office shall be responsible for the comprehensive coordination of pollution prevention and control, carry out related greening work, and do a good job in the source management and coordination of construction waste treatment within its jurisdiction.
Ground summary	**The xx road sub-district of XX district of XX shall be responsible for the comprehensive coordination of pollution prevention and control, carry out related greening work, and do a good job in the source management and coordination of construction waste treatment within its jurisdiction**.

Table 2: ROGUE1 comparisons between different methods.

Method\value	*P*	*R*	*F*1
LEAD	19.3	14.4	16.5
NEURALSUM	20.5	30.1	24.4
TextRank	21.2	26.3	23.5
BERTSUM + FC	21.8	31.5	25.8
BERTSUM + Transformer	28.2	26.8	27.5
BERTSUM + LSTM	27.6	24.9	26.2
BERTSUM + BiLSTM + attention	23.1	33.6	27.2
BERTSUM + CNN	50.1	18.4	26.9
BERTSUM + BiLSTM + CNN	50.8	19.0	27.4
Proposed method	51.0	19.3	28.1

Note. The value of the proposed method is the average value after 10-fold cross-validation test.

Table 3: ROGUE2 comparisons between different methods.

Method\value	*P*	*R*	*F*1
LEAD	14.1	12.4	13.2
NEURALSUM	14.3	13.3	13.8
TextRank	13.6	16.9	15.1
BERTSUM + FC	12.7	16.8	14.5
BERTSUM + Transformer	13.2	18.1	15.3
BERTSUM + LSTM	12.3	18.5	14.8
BERTSUM + BiLSTM + attention	13.1	17.5	15.0
BERTSUM + CNN	14.2	15.6	14.9
BERTSUM + BiLSTM + CNN	12.4	21.0	15.6
Proposed method	15.2	17.5	16.3

Notes. The value of the proposed method is the average value after 10-fold cross-validation test.

simulation ability of human understanding of the document process; as a result, the extraction indexes increase. When CNN or BiLSTM is removed from our method, the performance decreases; the same is true for the attention component. For instance, the ROGUE-1-F1 decreased by 1.05% for CNN and 0.5% for BiLSTM on average. This indicates that CNN and BiLSTM play an important part in the document summarization task, which may be due to the reason that when CNN is removed, the relation between adjacent sentences is ignored. And when BiLSTM is removed, the long-range information is ignored; the result is the decline in the representation of the document. Figure 3 is the average F1 of ROGUE-1-F1, ROGUE-2-F1, and

Table 4: ROGUE-L comparisons between different methods.

Method\value	*P*	*R*	*F*1
LEAD	18.3	13.3	15.4
NEURALSUM	19.1	14.6	16.6
TextRank	18.2	13.9	15.8
BERTSUM + FC	18.7	26.7	22.0
BERTSUM + Transformer	21.1	27.3	23.8
BERTSUM + LSTM	19.4	27.1	22.6
BERTSUM + BiLSTM + attention	19.5	28.0	23.0
BERTSUM + CNN	19.9	27.8	23.2
BERTSUM + BiLSTM + CNN	20.8	27.3	23.7
Proposed method	22.6	31.1	26.2

Note. The value of the proposed method is the average value after 10-fold cross-validation test layer (8.47), BERTSUM with CNN (0.7), and BERTSUM with Transformer (1.47). And at the final step, our method also achieves the lowest loss, 0.46. All this points to one fact, the BERTSUM-based methods are pretrained with large corpora providing weight that fits the downstream task. As a result, the models exhibit better document representation and the losses of training are small at the beginning. Our method leverages BiLSTM and CNN with attention mechanism exhibits higher feature extraction ability which benefits the document summarization task.

ROGUE-L-F1 between different methods. From Figure 3, we can see that our method achieves the best performance on average F1, which is 23.5%.

We can conclude that our method can better represent documents and shows strong feature extraction ability; as a result, there is an enhancement in the final document summarization task. Our model can better extract long dependency and relations between sentences.

4.3.3. Training Loss Convergence. To further investigate the convergence performance and fitting ability of the model, we make a loss convergence comparison between different methods. Figure 4 shows the convergence of the BERTSUM-based methods, from which we can see that our method has lower loss and faster convergence rates. In the beginning, the training loss of our method is 0.81, which is the lowest compared to BERTSUM that is fully connected.

4.3.4. Ablation Study. We conduct two ablation studies to reveal the contribution of different components of BERTSUM: interval segments embeddings and trigram blocking

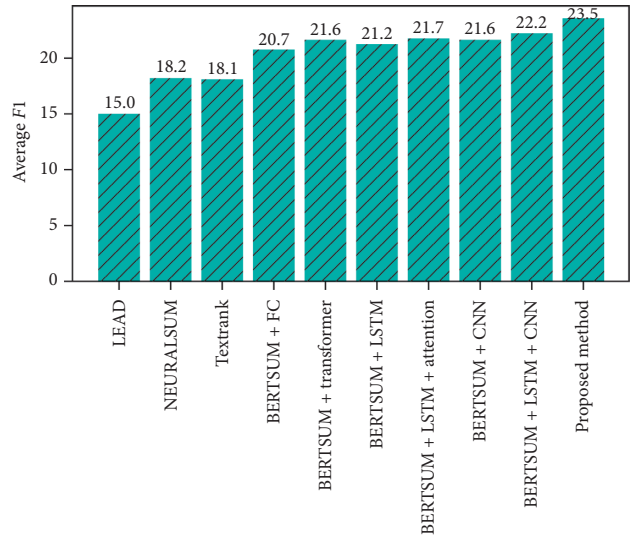

Figure 3: Comparison of average F1 between different methods.

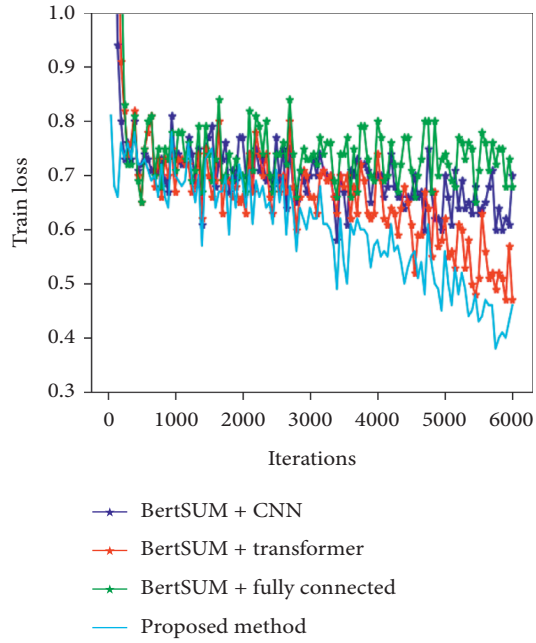

Figure 4: Training loss between different methods.

strategy. The results are shown in Table 5. With the addition of BERTSUM, BERT is improved compared to the original version. In terms of encoding, BERTSUM encodes at the document level and uses the interval segments embeddings to separate sentences in the document. To be specific, the interval segments mechanism assigns the sentence label according to their odevity. From Table 5, we can see that without the segment component, the model performance decreased by 0.8%, 1.4%, and 3.3% on ROGUE-1-F1, ROGUE-2-F1, and ROGUE-L-F1. This negative performance confirms the interval segments embeddings that have a better presentation than original BERT segment embeddings and thus improve the summarization results. Trigram blocking is a strategy that is used to reduce redundancy. The strategy plays an important role in the summarization task; the ablation study on trigram blocking is to explore this influence. Table 5 shows with trigram blocking, the summarization results are improved, which suggests that the strategy is helpful to obtain informative summaries.

4.3.5. Parameters and Running Speed. This section aims at revealing the efficiency of the methods, such as the number of the parameters involved, the training, and test time cost. Table 6 shows parameters and real-time capability comparisons between BERT-based methods, from which we can see that the BERT + Transformer method has 116M parameters, with 48.6 s to train at 50 steps and our method cost

Table 5: Ablation study on different strategies.

*F*1 index	ROGUE-1	ROGUE-2	ROGUE-L
Proposed method	28.1	16.3	26.2
Without interval segments	27.3	14.9	22.9
Without trigram blocking	26.8	14.3	22.3

Table 6: Parameters and efficiency of different methods.

Model (index)	Parameters (*M*)	Train speed per 50 step (s)
BERTSUM + FC	102	40.2
BERTSUM + Transformer	116	48.6
BERTSUM + BiLSTM	105	46.9
BERTSUM + BiLSTM + attention	105	45.2
BERTSUM + CNN	106	37.9
BERTSUM + BiLSTM + CNN	116	40.7
Proposed method	116	40.3

Table 7: Performance on CAIL 2020 dataset.

Method\value	ROGUE1-F1	ROGUE2-F1	ROGUEL-F1
LEAD	18.5	12.4	18.4
NEURALSUM	25.4	13.3	19.8
TextRank	22.5	16.9	17.6
BERTSUM + FC	25.7	16.8	22.0
BERTSUM + Transformer	26.5	18.1	22.7
BERTSUM + LSTM	26.1	18.5	23.6
BERTSUM + BiLSTM + attention	28.2	17.5	24.1
BERTSUM + CNN	28.9	15.6	23.2
BERTSUM + BiLSTM + CNN	29.4	21.0	25.8
Proposed method	29.8	20.5	29.1

Table 8: The result of the 10-fold cross-validation test.

Fold	Method	ROGUE-1-F1	ROGUE-2-F1	ROGUE-L-F1
1	BERTSLCA	28.3	16.8	25.8
2	BERTSLCA	27.9	16.5	25.7
3	BERTSLCA	28.0	16.0	26.8
4	BERTSLCA	28.4	17.2	27.0
5	BERTSLCA	27.8	16.4	25.9
6	BERTSLCA	27.6	15.8	26.0
7	BERTSLCA	27.5	15.4	26.1
8	BERTSLCA	28.2	16.4	26.5
9	BERTSLCA	28.0	16.5	26.3
10	BERTSLCA	29.3	16.0	25.9
Ove	BERTSLCA	28.1 ± 0.51	16.3 ± 0.52	26.2 ± 0.43

Note. Ove means the overall performance (mean ± std.).

40.3 s but achieves higher summarization results on *F*1 value. It indicates that our method achieves good results based on ensuring real-time ability. This is mainly because our method applies BiLSTM and CNN with an attention mechanism which presents an efficient calculation method that outperforms the Transformer method with less time to train.

4.3.6. Performance on CAIL 2020 DATA. For the research on Chinese text summarization, there are still few studies at present. Many studies are based on the data of microblogs, such as [44–46]; shi et al. crawled the data from microblogs related to the meteorological domain for the task of generation of meteorological briefing. For specialized fields, such as the legal domain, there is currently publicly available that called CAIL2020 judicial summary data (obtain judicial summarizations according to relevant legal documents). To further verify the validity of our model, we evaluate the method based on this data. From Table 7 we can see that our proposed method still achieves the highest performance on the CAIL dataset, which indicates that the proposed method has good generalization ability. It is not

only applicable to the legal documents of public interest litigation cases but also applicable to civil cases. This is mainly because the legal document of the case and the corresponding summarization are written in a standard manner. The trained model is well suited to capturing normative legal documents.

4.3.7. Statistical Test. A statistical test is further conducted to validate the performances. Table 8 is the experiment results of each fold of the proposed method. We can find that the proposed method BERTSLCA is consistently superior to the compared methods in the ROGUE-F1 index. We further conduct a t-test for statistical significance tests. Compared with other models, it is found that there is a significant difference ($P < 0.05$), and the average error rate of our method is smaller, and the performance is the best. The experimental results show that combining BERTSUM representations and the extracting components can steadily elevate the summarization performance.

5. Conclusion

In this paper, we have proposed an improved BERTSUM network called BERTSLCA to tackle the multisentences public interest litigation document for summarization of draft procuratorial suggestions. In BERTSLCA, a BERTSUM is first utilized to initial the word embeddings and get the sentence-level features, which is suitable for extracting summaries. We have constructed a high-performance extracting layer to aggregate sentence information and capture document-level features for extracting summaries. In the extracting layer, we find that each of the three components BiLSTM, CNN and attention mechanism plays a very important role in the contribution to the extractive summarization task. Furthermore, based on the previous document-level feature representation, each sentence is classified to determine whether it belongs to the summary, and the summary task is transformed into the classification task. Extensive experiments on real-world datasets and CAIL2020 have shown that the proposed BERTSLCA outperforms the state-of-the-art methods in almost all evaluation metrics, and the influence of the three parts of the extraction layer on the ROGUE index is also confirmed. An experiment on running time shows that the method in this paper costs the shortest time.

However, one limitation of our work is mainly based on the extractive method to complete the public interest litigation cases prosecutorial suggestions automatic generation; that is to say, we can only select the existing sentences from the prelawsuit review documents to complete the generation of prosecution recommendations. In the future, we will explore abstractive summary generation methods, which completes the real sense of understanding and achieves the process of creation out of nothing in order to be more realistic. On the other hand, due to the lack of a public corpus, we will explore the influence of domain knowledge on the pretraining model in order to improve the performance of the downstream summary generation task.

Data Availability

The dataset used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key R&D Program of China, under Grant no. 2018YFC0830800.

References

[1] N. Alami, M. Meknassi, N. En-Nahnahi, Y. El Adlouni, and O. Ammor, "Unsupervised neural networks for automatic Arabic text summarization using document clustering and topic modeling," *Expert Systems with Applications*, vol. 172, Article ID 114652, 2021.

[2] J.-G. Yao, X. Wan, and J. Xiao, "Recent advances in document summarization," *Knowledge and Information Systems*, vol. 53, no. 2, pp. 297–336, 2017.

[3] M. Mojrian and S. A. Mirroshandel, "A novel extractive multi-document text summarization system using quantum-inspired genetic algorithm: MTSQIGA," *Expert Systems with Applications*, vol. 171, Article ID 114555, 2021.

[4] Y. Du, Q. Li, L. Wang, and Y. He, "Biomedical-domain pre-trained language model for extractive summarization," *Knowledge-Based Systems*, vol. 199, Article ID 105964, 2020.

[5] M. Moradi, G. Dorffner, G. Dorffner, and M. Samwald, "Deep contextualized embeddings for quantifying the informative content in biomedical text summarization," *Computer Methods and Programs in Biomedicine*, vol. 184, Article ID 105117, 2020.

[6] D. A. B. Duy and D. F. Guilherme, "Extractive text summarization system to aid data extraction from full text in systematic review development," *Journal of Biomedical Informatics*, vol. 64, pp. 265–272, 2016.

[7] C. Gulden, M. Kirchner, and C. Schüttler, "Extractive summarization of clinical trial descriptions," *International Journal of Medical Informatics*, vol. 129, pp. 114–121, 2019.

[8] M. Hans, P. Laura-Maria, and H. Juho, "Comparison of automatic summarisation methods for clinical free teXt notes," *Artificial Intelligence in Medicine*, vol. 67, pp. 25–37, 2016.

[9] S. Kaize, L. Hao, and Y. F. Zhu, "Automatic generation of meteorological briefing by event knowledge guided summarization model," *Knowledge-Based Systems*, vol. 192, Article ID 105379, 2019.

[10] D. S. Moirangthem and M. Lee, "Abstractive summarization of long texts by representing multiple compositionalities with temporal hierarchical pointer generator network," *Neural Networks*, vol. 124, pp. 1–11, 2020.

[11] A. M. Azmi and N. I. Altmami, "An abstractive Arabic text summarizer with user controlled granularity," *Information Processing & Management*, vol. 54, no. 6, pp. 903–921, 2018.

[12] X. Mao, H. Yang, S. Huang, Y. Liu, and R. Li, "Extractive summarization using supervised and unsupervised learning," *Expert Systems with Applications*, vol. 133, pp. 173–181, 2019.

[13] R. Elbarougy, G. Behery, and A. E. Khatib, "Extractive Arabic text summarization using modified PageRank algorithm," *Egyptian Informatics Journal*, vol. 21, pp. 73–81, 2019.

[14] L. Cagliero and M. L. Quatra, "Extracting highlights of scientific articles: a supervised summarization approach," *Expert Systems with Applications*, vol. 1, p. 160, Article ID 113659, 2020.

[15] T. Ukan and A. Karc, "Extractive multi-document text summarization based on graph independent sets," *Egyptian Informatics Journal*, vol. 21, pp. 145–157, 2020.

[16] H. K. Thakkar, P. K. Sahoo, and P. Mohanty, "DOFM: domain Feature Miner for robust extractive summarization," *Information Processing & Management*, vol. 58, no. 3, Article ID 102474, 2021.

[17] C. Aone and M. E. Okurowski, "A trainable summarizer with knowledge acquired from robust nlp techniques," *Advances in Automatic Text Summarization*, vol. 17, 1999.

[18] Y. Matsuo and M. Ishizuka, "Keyword extraction from a single document using word co-occurrence statistical information," *International Journal on Artificial Intelligence Tools*, vol. 13, no. 1, pp. 157–169, 2004.

[19] S. Ruan, "Weighted nave Bayes text classification algorithm based on improved distance correlation coefficient," *Neural Computing and Applications*, pp. 1–10, 2021.

[20] B. Fca, "Authentication of Douro DO monovarietal red wines based on anthocyanin profile: comparison of partial least squares – discriminant analysis, decision trees and artificial neural networks," *Food Control*, vol. 125, 2021.

[21] S. Ali and N. Bouguila, "Maximum a posteriori approximation of hidden markov models for proportional sequential data modeling with simultaneous feature selection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 99, 2021.

[22] Z. Y. Khan and Z. Niu, "CNN with depthwise separable convolutions and combined kernels for rating prediction," *Expert Systems with*, vol. Applications, p. 114528, 2020.

[23] H. Liu, Y. Hao, W. Zhang, H. Zhang, F. Gao, and J. Tong, "Online urban-waterlogging monitoring based on a recurrent neural network for classification of microblogging text," *Natural Hazards and Earth System Sciences*, vol. 21, no. 4, pp. 1179–1194, 2021.

[24] H. Dong, F. Yang, and X. Wang, "Multi-label charge predictions leveraging label co-occurrence in imbalanced data scenario," *Soft Computing*, vol. 24, p. 9, 2020.

[25] Y. Liu, "Fine-tune BERT for extractive summarization," 2019, http://arxiv.org/abs/1903.10318.

[26] J. Devlin, "BERT: pre-training of deep bidirectional transformers for languageunderstanding," in *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies*, Association for Computational Linguistics, Minneapolis,MN, USA, 2019.

[27] A. Vaswani, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ACM, Long Beach, CA, USA, 2019.

[28] R. A. Sinoara, J. Camacho-Collados, R. G. Rossi, R. Navigli, and S. O. Rezende, "Knowledge-enhanced document embeddings for text classification," *Knowledge-Based Systems*, vol. 163, pp. 955–971, 2019.

[29] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, Sydney, Australia, 2013.

[30] A. Elnagar, R. Al-Debsi, and O. Einea, "Arabic text classification using deep learning models," *Information Processing & Management*, vol. 57, no. 1, Article ID 102121, 2020.

[31] F. Gargiulo, S. Silvestri, M. Ciampi, G. De Pietro, and G. De Pietro, "Deep neural network for hierarchical extreme multi-label text classification," *Applied Soft Computing*, vol. 79, pp. 125–138, 2019.

[32] X. Zhang, Y. Zhang, Q. Zhang et al., "Extracting comprehensive clinical information for breast cancer using deep learning methods," *International Journal of Medical Informatics*, vol. 132, Article ID 103985, 2019.

[33] J. Á. González, L.-F. Hurtado, and F. Pla, "Transformer based contextualization of pre-trained word embeddings for irony detection in Twitter," *Information Processing & Management*, vol. 57, no. 4, Article ID 102262, 2020.

[34] J. He, L. Zhao, H. Yang, M. Zhang, and W. Li, "HSI-BERT: hyperspectral image classification using the bidirectional encoder representation from transformers," *Ieee Transactions on Geoscience and Remote Sensing*, vol. 58, no. 1, pp. 165–178, 2020.

[35] W. Fang, H. Luo, S. Xu, P. E. D. Love, Z. Lu, and C. Ye, "Automated text classification of near-misses from safety reports: an improved deep learning approach," *Advanced Engineering Informatics*, vol. 44, Article ID 101060, 2020.

[36] B. Hachey and C. Grover, "Sentence classification experiments for legal text summarisation," in *Proceedings of the Conference on Legal Knowledge & Information Systems*, Berlin, Germany, 2004.

[37] M. Saravanan, *Automatic identification of rhetorical roles using conditional random fields for legal document summarization, Artificial Intelligence and Law*, 2008.

[38] D. Anand and R. Wagh, "Effective deep learning approaches for summarization of legal texts," *Journal of King Saud University - Computer and Information Sciences*, 2019.

[39] Y. Huang, "Legal public opinion news abstractive summarization by incorporating topic information," *International Journal of Machine Learning and Cybernetics*, vol. 3, 2020.

[40] P. Zhou and W. Shi, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 2, Berlin, Germany, 2016.

[41] S. S. Narayan and S. B. Cohen, "Ranking sentences for extractive summarization with reinforcement learning," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, New Orleans, LA, USA, 2018.

[42] J. P. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, 2016.

[43] R. Mihalcea and P. Tarau, "Textrank: bringing order into text," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Hong Kong, China, 2004.

[44] S. Ma, X. Sun, W. Li et al., "Query and output: generating words by querying distributed word representations for paraphrase generation," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 196–206, New Orleans, LA, USA, 2018.

[45] S. Ma, X. Sun, J. Xu et al., "Improving semantic relevance for sequence-to-sequence learning of Chinese social media text summarization," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 635–640, Vancouver, Canada, 2017.

[46] S. Gao, X. Chen, P. Li et al., Abstractive text summarization by incorporating reader comments, 2018, https://arxiv.org/abs/1812.05407.