Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **8**, Number **3**, August **2023**

Long Document Extractive Summarization Method Based on Pre-training Model and Transformer

Xinxin Zhou*

School of Computer Science
Northeast Electric Power University
Jilin, Jilin, China
zxx51@qq.com

Yuechen Guo

School of Computer Science
Northeast Electric Power University
Jilin, Jilin, China
gyc1998625@gmail.com

Yuning Huang

School of Computer Science
Northeast Electric Power University
Jilin, Jilin, China
m18757944766@163.com

Yuming Yan

School of Computer Science
Northeast Electric Power University
Jilin, Jilin, China
yumingyandoffodil@163.com

Maoyuan Li

School of Computer Science
Northeast Electric Power University
Jilin, Jilin, China
734405854@qq.com

*Corresponding author: Xinxin Zhou

Received October 7, 2022, revised December 5, 2022, accepted February 23, 2023.

Abstract. *Due to the acceleration of the era of big data, more and more text resources are uploaded to the Internet. In many scenarios, long texts will be processed. These long text data have many problems of information redundancy and information overload, resulting in the text summary model. The robustness decreases when it is processed. Aiming at the problem that the model cannot be trained due to the lack of labeled data and sentence vectors procured is complicated, a summarization extraction model for long documents is proposed in this paper. First, aiming at the problem that the supervised model cannot be trained due to the lack of labeling of the original data source, a novel weighted similarity index is proposed to label the original data source and construct the training data. Second, in the coding stage, the pre-trained BERT model is introduced to code processing of the text data. The identifier is adopted in embedding layer to identify the sentence. And a sentence vector construction method based on the combination of average pooling and maximum pooling is proposed. The word vector is processed by suggested programmed to obtain the sentence vector. Finally, the Transformer model is used to learn sentence vector features, and extracted summarization is acquired after training and classification. To confirm the effectiveness of suggested means, a comparative experiment is carried out on the CAIL2020 dataset. The experimental results reveal that the method has good performance on the three Rouge indicators, and the accuracy of extracting summaries are improved, Laying the groundwork for follow-up research.*
Keywords: Extractive Summarization, Transformer, BERT, Weighted Similarity

1. **Introduction.** Text summarization technology refers to condensing, summarizing, and refining the content of large-scale digitized texts to generate a concise summary that can reflect the theme of the text, thereby helping readers quickly obtain useful information contained in the original lengthy documents. Text summarization technology is an important means to solve information redundancy, and is widely used in many domains such as news [1], medicine [2], and justice [3]. Long text summarization is a branch of the text summarization domain that has the characteristics of long text length and large information content. Compared with English texts, synonyms and synonyms in long Chinese texts are confused with each other, that fetches certain difficulties for semantic analysis of the text. On the basis of a disparate generation ways for text summarization, there are mainly two methods: extractive abstract and generative abstract. Extractive abstract [4, 5, 6, 7] is to directly select phrases or sentences that can represent the main content of the primeval text in the original text, and obtain a sententious summary after splicing. Generative abstracts [8, 9, 10] refer to the generation of words, phrases, and sentences that do not exist in the original text through the understanding and simplification of the primeval text and form a sententious summary that can represent the original content.

Many researchers have explored the generative approach to extractive summaries. Some scholars have proposed the use of statistical methods for extraction tasks, which are mainly based on features such as word frequency and sentence position information. Aone et al. [11] used TF-IDF to capture important words in sentences and score them according to their importance to obtain summaries. The essence of the abstract extraction method based on statistical methods [12, 13, 14, 15] is to judge the importance of words and sentences through features and extract the most important sentences to form summaries. With the development of deep learning in the natural language processing domain, neural networks perform summarization tasks by obtaining semantic information by learning data and using vectors to represent corresponding words. A convolutional neural network (CNN) [16] is an efficient feature extractor in the natural language processing domain. Another classic network is the recursive neural network (RNN) [17, 18], which can capture sentence-level features regardless of sentence length and learn the current word at each time step by memorizing the previous text, thereby obtaining long-distance dependencies.

The features extracted by the above methods cannot express the semantics well, and they cannot solve problems such as polysemy. For the sake of figuring out the above problems, Liu [19] proposed a BERT-based method for the extractive summarization assignment. BERT (Bidirectional Encoder Representations from Transformers) [20] is a new language model proposed by Google researcher Devlin et al. in 2018 and has achieved excellent results in 11 natural language processing (NLP) tasks. The main structure of BERT is stacked with encoders of Transformer [21], which can effectively improve the effect of downstream tasks and solve the problem of polysemy to a certain extent. By modifying the input sentence, coding it into sentence units to obtain document-level output, introducing contextual information into it, and introducing BERT into the extractive summary generation task, Liu achieves excellent results.

Due to the acceleration of the era of big data, more and more text resources are uploaded to the Internet. In many scenarios, long texts will be processed. These long text data have a large number of problems of information redundancy and information overload, resulting in the text summary model. The robustness decreases when it is processed. Compared with the problems of small amount and slight data redundancy in conventional text summarization, the problem of text summarization of long texts is mainly that there is a large amount and heavy redundant information. Due to the length limit of the model processing text data, too long text will cause the model to fail to extract accurate feature information from the text, which is manifested as a reduction in training speed and a decrease in model performance. Therefore, long text data in text summaries is more difficult to process than general data, and super-long text data with longer text length will not only make the problems such as data redundancy to be solved more serious, but also increase the processing burden of the model and problems such as unregistered words, and ultimately lead to more duplicate data and useless data generated by the model. Therefore, this paper mainly studies the BERT extractive abstract generation method in long text. At present, the main challenges in the field of long text abstract extraction are as follows:

1) Long text summarization data usually lacks labeled data, and the training of neural networks requires labeled data for training, so it is necessary to construct a reasonable data labeling method.

2) Traditional vectorization coding methods such as Word2vec do not consider polysemy, static vectors are difficult to obtain context features, and cannot be better applied to long texts, which affects the generation of subsequent summaries of the model.

3) Compared with short text, the model is more difficult to extract features when processing long text, and will make the model more complicated and require more training time.

The paper puts forward the abstract generative approach based on pre-training model and Transformer. This article mainly has certain innovation in the following aspects:

1) Aiming at the problem that long text summarization data lacks extractive labeling, a weighted similarity data labeling method is adopted. The original data source is processed by constructing a weighted similarity index, and the original unlabeled data is converted into labeled data to make data preparation for subsequent model training.

2) Introduce the BERT pre-trained model to code text data, and introduce identifiers in its embeddings layer to identify sentences. A sentence vector construction method based on the combination of average pooling and maximum pooling is proposed in the coding output stage. The proposed method processes the word vector to obtain the sentence vector. Finally, the use of the Transformer as the extraction body model is introduced to improve the ability to process long texts.

3) To demonstrate the effectiveness for a suggested approach, relevant experiments were put into effect. The experimental consequences display that the means achieves better results than other extractive summarization methods.

The rest of this paper is arranged as follows: Section 2 provides a detailed introduction and explanation of the proposed method. Section 3, conducts experiments and analyzes the experimental results. Section 4 summarizes the full text and points out future research directions.

2. **Related works.** Correct coding representation of text is the basis for text summarization, text classification, and other related tasks in the natural language processing (NLP) domain [22, 23]. Existing research employment has exhibited that using the benign language model to text representation could effectively enhance the performance of downstream assignments. To better represent text and reduce costs, Google Labs proposed the Word2Vec model to represent text information [24]. This method can generate low-density, low-latitude representations and introduce semantic information. Glove [25] improved Word2Vec, replacing the objective function and weight function so that it could be trained on the large-scale corpora, which has the characteristics of fast training speed. However, the word vector generated by this method is static, and the problem of polysemy is not considered, so it is difficult to obtain contextual features for long texts.

Therefore, the performance of different NLP tasks can be significantly improved by pre-training word embedding methods on large-scale corpus [26, 27]. Kenton et al. [20] proposed the BERT pre-trained model based on Transformer. BERT is a bidirectional language model that can better consider contextual information and capture long-distance dependencies more efficiently. Thanks to the particularity of the structure of the BERT model, it has made major breakthroughs in the fields of medicine and cross-modality [28, 29, 30]. BERT can dynamically represent words according to the contextual content of the current words, which could effectively settle a matter of polysemy of word, then is more suitable people's verbal comprehension. BERT is to code vectors in units of words, which can be calculated with the text related assignments [31], but the automatic generation of text summarization is more suitable for coding in units of sentences. Therefore, inspired by this, this paper improves the BERT coding method and obtains sentence-level features by performing sentence-level embeddings in units of sentences. See Chapter 2 for details.

In the wake of rapid advancement for the machine learning, deep learning methods have gradually become the mainstream methods in the NLP domain. Rush et al. [32] was the first to use the deep learning method in the text summarization domain. This method introduces local attention into the model, generates a summary set according to the sentence, and then uses Rouge to score to obtain the final summary. However, this method cannot handle long text data well. Ling [33] introduced a hierarchical attention mechanism. The first layer selects keywords from the text, and then performs post-processing through the Seq2Seq model's afferent second layer. The attention model is trained by reinforcement learning. This method can process long texts, but the performance is not high. Cohan et al. [34] proposed a summary model, which introduces a hierarchical encoder and a perceptual decoder to improve the information acquisition ability of the model. Xiao et al. [35] proposed a multi-level encoder that combines section-level and sentence-level information. Cui et al. [36] proposed obtaining longer contextual information by sliding coding on chapters, which improves the receptive field of the model. Huang et al. [37] designed an efficient Transformer model with sparse attention to extraction, saving training time and greatly improving the efficiency of the model. Cao et al. [38] proposed incorporating text structure into the calculation of attention score and injecting hierarchical deviation. However, such methods based on the BERT model are

usually applied to the abstract generation of short texts, and the complexity of processing long texts increases quadratic with the input length, which means they cannot be well applied to the task of long text summarization of thousands of words, let alone very good attention to the semantic relationship between sentences [39, 40, 41, 42, 43].

3. Extractive Summarization Method Based on Pre-training Model and Transformer.

3.1. Data Construction Based on Weighted Similarity.

When conducting supervised deep learning model training, supervised data is the premise of model training, and data's quality will affect an effect of training. Existing data construction methods often only focus on one indicator, such as Rouge-1 and Rouge-2, but a single indicator cannot fully reflect the similarity of the data. Multiple indicators can better reflect the similarity of the data. Therefore, in order to improve data quality and enhance an impact of model training, the paper comes up with the data preprocessing method based on weighted similarity. First, divide the original data source according to punctuation to construct fine grain clauses. Then, a weighted similarity method is constructed according to the three evaluation indicators, each sentence of the reference abstract and the original text is weighted to solve, and the top-K sentences before similarity are selected and put into the candidate set. Figure 1 shows the data preprocessing process.

Among them, there are "M" sentences in the original text, and there are "N" sentences in the reference abstract, $X = \{x_1, x_2, \ldots, x_M\}$, "X" is the original text. $Y = \{y_1, y_2, \ldots, y_N\}, Y$ delegate a reference abstract. Experimental analysis and comparison according to 3.3.1, we construct the weighted similarity index according to the ratio of Rouge-1, Rouge-2, and Rouge-L according to 2:2:6, as shown in Equation (1).

$$Fs\,(x_j y_j) = 0.2^{*} Rouge - 1 + 0.2^{*} Rouge - 2 + 0.6^{*} Rouge - L \tag{1}$$

Where $Fs\,(x_i y_j)$ is the similarity score between x_i and y_j. According to the above weighting method, Solve the similarity index between the reference abstract $y_j (j \in N)$ and the original sentence $x_i (i \in M)$, and the score $S_{ij} = Fs\,(x_i y_j)$ of each sentence is obtained. According to score matching, the first top- K sentence corresponds to each reference abstract X_K :

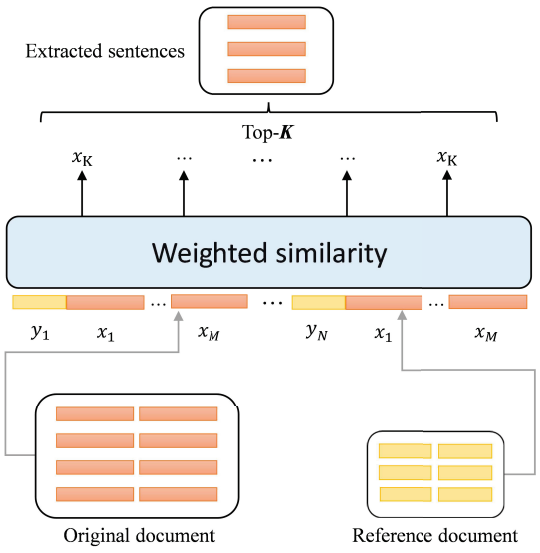

FIGURE 1. Data transformation method based on weighted similarity

$$X_K = top - K\left(FS\left(x_i, y_j\right)\right), x_i \in X, y_j \in Y \tag{2}$$

Then, select among the remaining sentences, and judge whether the connection between the sentence and the selected sentence can improve the similarity index score until the iteration of all remaining sentences is completed. The sentences corresponding to the final candidate set are marked as 1, and the remaining sentences are marked as 0, in the experiment we set K to 1.

3.2. **Vectorization Method Based on Improved BERT.** Since BERT vectorization is a word embedding-based method, it can be more suitable for paragraph or short sentence tasks, but not suitable for document-level tasks such as long text extraction summarization. Therefore, in order to accomplish the long-text summarization extraction assignment, the BERT embeddings representation method needs to be improved. An idea for ameliorate could be to add the "$< CLS >$" and "$< SEP >$" logo before each sentence to obtain the document-level sentence logo. And abandoning truncation, a sentence vector representation method is proposed. By pooling the word vector with sentence identification after BERT coding, and combined with maximum pooling, the sentence vector is obtained. For long texts much larger than 512 characters, abandoning text truncation greatly reduces information loss and ensures that the model can receive all the information. The text input TE is:

$$TE =< CLS >, X_1, < SEP >, \ldots, < CLS >, X_M < SEP > \tag{3}$$

$< CLS >$ and $< SEP >$ are symbols used by the BERT model to classify or summarize sentences, where $< CLS >$ intervals sentences. The embeddings layer converts the text content into a vector $\{TEv_i\}_{i=1}^{M} = \{TEv_1, TEv_2, \ldots, TEv_M\}$, where TEv_i is calculated in Equation (4):

$$TE_{V_i} = TE_p^{tok} + TE_i^{seg} + TE_i^{pos} \tag{4}$$

In Equation (4), TE_i^{tok} represents the mark of the sentence, TE_i^{seg} represents paragraph information, TE_i^{pos} represents position information, and i is the i -th sentence in the text.

BERT is composed of multiple Transformer encoders in series. The vector obtained by Equation (4) is input into L consecutive Transformer, each Transformer consists of a multihead attention mechanism and a feedforward network. The vector is encoded as a deeper representation $T^L = \left\{T_i^L\right\}_{i=1}^{M} = \left\{T_1^L, T_2^L, \ldots, T_M^L\right\}$, where $T_i^L = T_{i1}^L, T_{i2}^L, \ldots, T_{in}^L$, where n is the total of n word vectors in the sentence.

$$T^L = \begin{cases} Transformer\left(TEv_1, \ldots, TEv_M\right), L = 1 \\ Transformer\left(T_1^{L-1}, \ldots, T_M^{L-1}\right), M \geq L > 1 \end{cases} \tag{5}$$

In Equation (5), T^L is the TEv_1 vector processed by the L layer Transformer. T^L is composed of corresponding multiple word vectors, but the scattered word vector cannot be used as a sentence vector, so average pooling is performed on all word vectors in a sentence, and combined with maximum pooling, and the pooling result is used as a sentence vector, as shown in Equation (6):

$$\bar{T}^L = \frac{1}{2}\left(\frac{1}{n}\sum_{j\in n} T_{ij}^L + max\left(\sum_{j\in n} T_{ij}^L\right)\right) \tag{6}$$

Where T_{ij}^L is the word vector in the T_i^L sentence, n is the number of word vectors in the sentence, and $\bar{T}^L$ refers to the sentence vector after combining average pooling and maximum pooling.

3.3. **Model Flow.**

3.3.1. *Data Preprocessing.* Firstly, denoising the text data, eliminating excess punctuation and spaces in the original data source and summary data, which will affect the subsequent segmentation effect. Then divide the sentences to construct fine grain clauses, and then the weighted similarity method is constructed according to the Equation (1) proposed in Section 3.1, and the similarity score is solved according to the Equation (2) according to this method, and the training data is constructed.

3.3.2. *Embeddings Representation.* Improve BERT embeddings. represent text according to Equation (3), increase sentence identification, and obtain embedding representation of text according to Equation (4) to prepare for vectorization.

3.3.3. *Sentence Vector Acquisition.* Using the BERT model as the encoder, the text is vectorized according to the Equation (5) through the 12-layer Transformer Model, and then Average pooling of the word vectors in the acquired sentences by formula Equation (6) and combine with the maximum pooling, and the average pooling result is used as the sentence vector.

3.3.4. *Extractor.* After obtaining the sentence vector, fine-tuning is performed by building an extraction layer, multiple Transformer layers are applied in the sentence representation; and deeper document-level features are obtained through the output of the coding layer, as shown in Equation (7) and Equation (8):

$$\tilde{h}^L = LN\left(h^{L-1} + MHAtt\left(h^{L-1}\right)\right) \tag{7}$$

$$h^L = LN\left(\tilde{h}^L + FFN\left(\tilde{h}^L\right)\right) \tag{8}$$

Where L represents the number of stacked Transformer layers, $MHAtt$ represents the multi-head attention mechanism, LN is the layer normalized, and FFN is the feedforward neural network.

3.3.5. *Training Loss.* After Transformer extraction, the final representation of the text is afferent to the output classification layer, and the prediction result is mapped to the interval of (0,1) through the sigmoid-based classifier, as shown in Equation (9):

$$\hat{Y}_i = sigmid\left(W^o h^L + b^o\right) \tag{9}$$

b^o and W^o are the parameter deviation and weight required to train the model, and $\hat{y}_i$ represents the predicted fraction ($\hat{y}_i \in [0,1]$) for each sentence of the text. h^L refers to the vector Transformer the Lth layer (top layer).

Using cross-entropy as the model loss function during the training phase, the calculation Equation (10) is:

$$F = \sum_i -y_i log\hat{y}_i - (1-y_i)\, log\,(1-\hat{y}_i) \tag{10}$$

Where y_i represents the real label of the text sentence with a value of 0 or 1. The model is trained to minimize the cross-entropy loss between the real label and the predicted score. The overall method architecture is shown in Figure 2.

4. **Experiment and Analysis.**

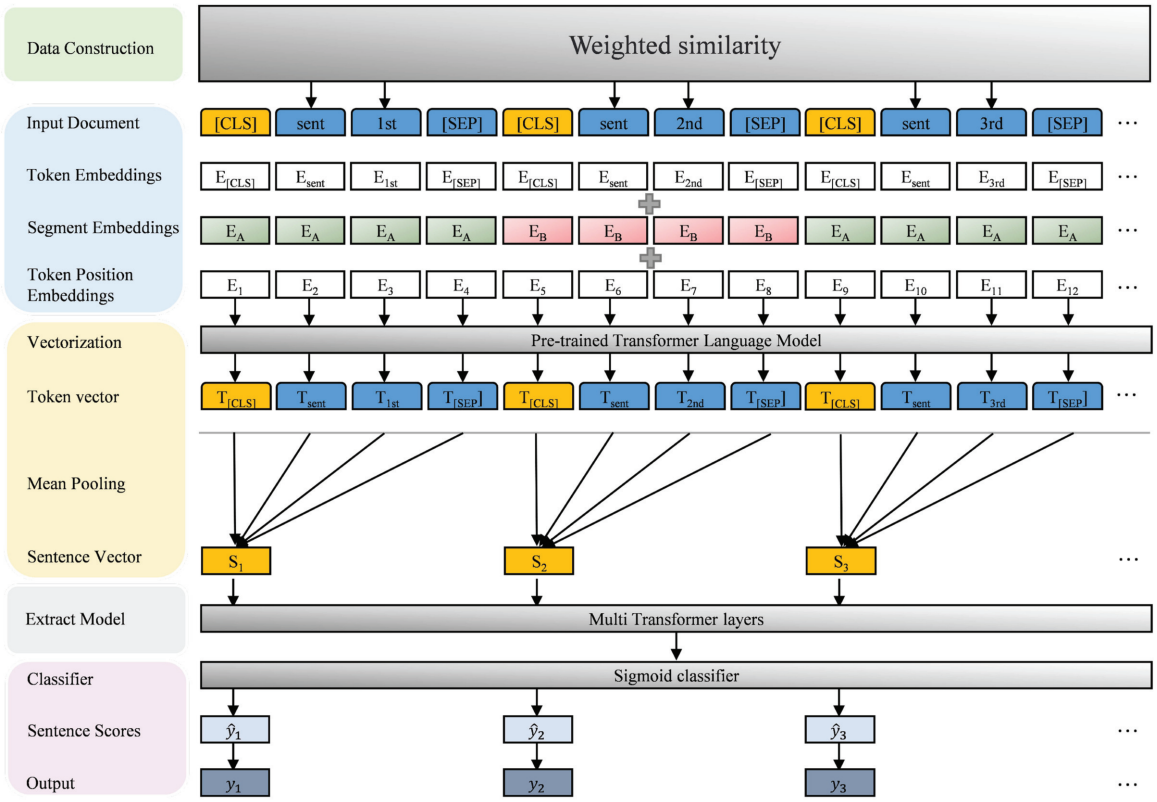

FIGURE 2. Architecture of the model

4.1. **Experimental Dataset.** This paper uses two-part dataset provided by CAIL2020, namely CAIL-small including 4047 pieces of data, and CAIL-big including 5437 pieces of data, The length distribution of the data set is shown in Figures 3, 4. The original data source is distributed into a training set and a test set in an 8:2 ratio.

The average length of the original input data is 2568 words, and the maximum number of words is 13064, of which 99% of the input data length is greater than 1000, and most of the input data length is concentrated between 2000-3000 words; the average length of the abstract data is 283 words, and the maximum number of words is 474, of which 95% of the abstract data length is greater than 327, and the training data statistics are shown in Table 1. By analyzing a dataset, it can be seen that this is a text summary task with 3000 words input and 300 words output, so this data set belongs to the long text summary dataset.

TABLE 1. Training Data Statistics

Total	Source	Mean	Standard Deviation	Max	Minimum
9484	Input	2568	1122	13064	866
	Output	283	36	474	66

4.2. **Experimental Evaluation Criteria.** The paper employs the three widely used indicators Rouge-1, Rouge-2, and Rouge-L on the text summarization domain and calculates the accuracy rate (P), recall (R), and $F1$ value of each indicator for experimental comparison. The calculation methods of P, R, and $F1$ of Rouge-1 and Rouge-2 are as

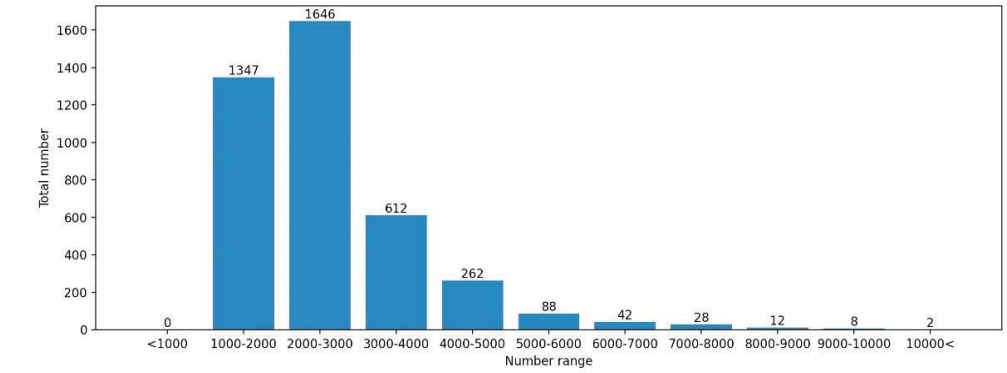

FIGURE 3. CAIL-Small word count distribution diagram

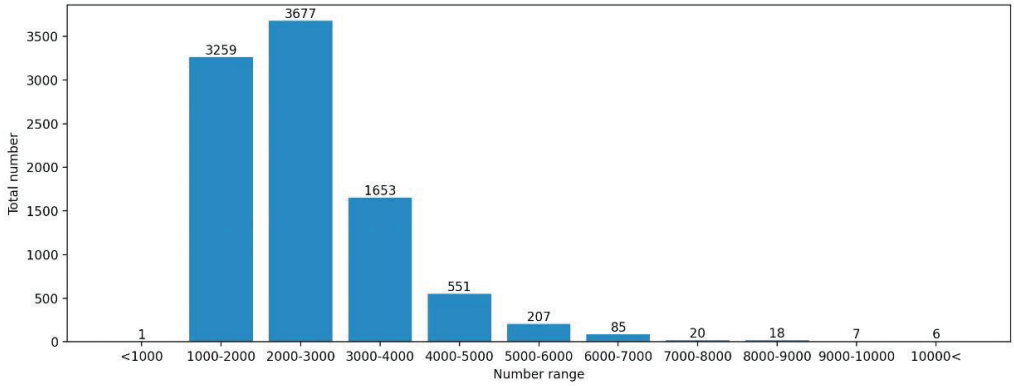

FIGURE 4. CAIL-Big word count distribution diagram

follows:

$$p = \frac{extractedsummary \cap referencesummary}{extractedsummary} \tag{11}$$

$$R = \frac{extractedsummary \cap referencesummary}{referencesummary} \tag{12}$$

$$F1 = \frac{2PR}{P+R} \tag{13}$$

Rouge-1 refers to the unary overlay between the extracted abstract and the key abstract, and Rouge-2 refers to the binary overlap between the extracted abstract and the reference abstract. The P, R, and $F1$ calculation methods of Rouge-L are as follows:

$$P_{lcs} = \frac{LCS(X,Y)}{|X|} \tag{14}$$

$$R_{lcs} = \frac{LCS(X,Y)}{|Y|} \tag{15}$$

$$F_{lcs} = \frac{(1+\beta^2)P_{lcs}R_{lcs}}{\beta^2 P_{lcs} + R_{lcs}} \tag{16}$$

$LCS(X,Y)$ refers to the length of the longest common sub-job family (LCS) between the extracted digest X and the reference digest Y. $|X|$ refers to the word's number with

an extracted digest, the $|Y|$ refers to the word's number in the reference digest. β is the acquiescent parameter, set one.

4.3. **Experimental Baselines Model and Model Parameter Setting.** The experimental environment is Ubuntu, Pytorch, and 8 GPUs (NVIDIA A100 GPUs with 40GB of memory). 1. The coding layer is initialized with the 'RoBERTa-base' model weight for text coding, the layers = 12, the hidden layer = 768, and the attention headers = 12; 2. For the abstract extraction layer, some layers are stacked Transformer Model and Hidden Layer = 768. Use the Adam optimizer for model optimization, $\beta_1 = 0.9, \beta_2 = 0.999$, and the learning rate is set to 1e-6.

The comparison models used mainly include the following models:

LEAD [44]: This method was the baseline method for summarizing tasks.

TextRank [45]: This method introduced graphs into summary tasks.

Word2Vec+fully connected Classifier: The method used the Word2Vec model for vectorization and fully connected classifiers for binary classification.

BERT+ fully connected Classifier: This method was a method proposed by Liu [19], which used the BERT model to replace the traditional word vector method as the encoder.

BERT+ LSTM: This method was proposed by Liu [19], and abstracts were extracted through the Bi-LSTM neural network.

BERT+ CNN: This method was proposed by Sun et al. [3] to introduce the CNN model into the summarization task.

BERT+ Transformer: This method was proposed by Liu [19] and used the Transformer as the extraction model.

4.4. **Experimental Results and Analysis.**

4.4.1. *Results and analysis based on different weighted similarities.* This part experiments and analyzes different weighted similarity ratios. We set up two controlled experiments, 9 and 12 controlled experiment groups, and introduced the BERT-Base model for text vectorization, using the most used CLS vector as the sentence vector. In the first part of the 9 experiments, we first set up 3 groups of data construction methods that only consider one Rouge indicator, which is also the main method used in the existing work. In addition, we set up 6 groups of data construction methods that consider three Rouge indicators at the same time, and set the weights of the three indicators more evenly, as shown in Table 2; through analysis of the experimental results in a first part, we found that Rouge-1 and Rouge-L have a relatively high impact on model indicators. Therefore, in the 12 groups of experiments in the second part, we increased the weights of Rouge-1-F1 and Rouge-L-F1 respectively, and analyzed the experimental results, as shown in Table 3.

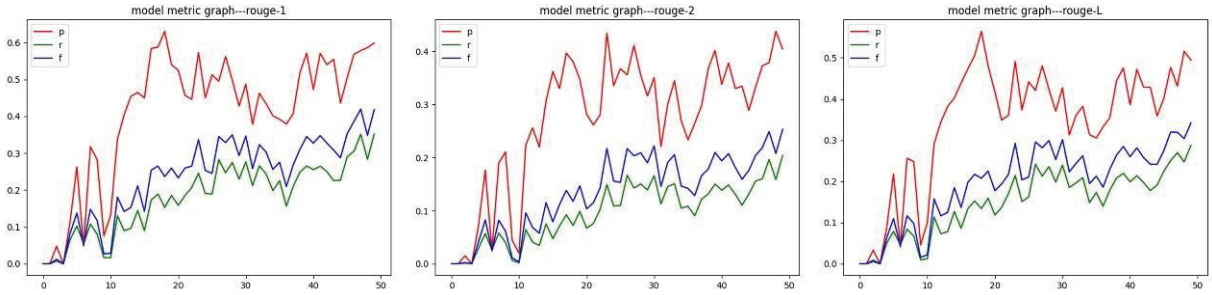

FIGURE 5. The weight ratio is 0:0:10

Table 2. The first set of control experiments

Proportion	Rouge-1-F1	Rouge-2-F1	Rouge-L-F1
10:0:0	40.89	24.80	33.43
0:10:0	39.27	23.47	31.20
0:0:10	**41.81**	**25.31**	**34.16**
4:3:3	35.34	21.02	27.57
3:4:3	33.67	20.38	27.79
3:3:4	36.08	22.47	29.27
2:4:4	32.75	19.02	25.09
4:2:4	31.29	19.84	25.16
4:4:2	31.83	19.75	25.17

Table 3. The second set of control experiments

Proportion	Rouge-1-F1	Rouge-2-F1	Rouge-L-F1
6:2:2	35.25	22.07	28.91
2:6:2	39.62	23.63	32.06
2:2:6	**45.10**	**26.19**	**35.92**
5:2:3	34.59	21.88	28.48
3:2:5	35.79	22.49	29.39
4:1:5	35.74	22.62	29.42
5:1:4	31.82	19.60	26.03
6:1:3	33.99	21.03	28.13
3:1:6	37.58	23.56	31.14
4:0:6	36.93	22.62	29.58
6:0:4	39.20	22.95	31.08
5:0:5	38.93	22.63	30.99

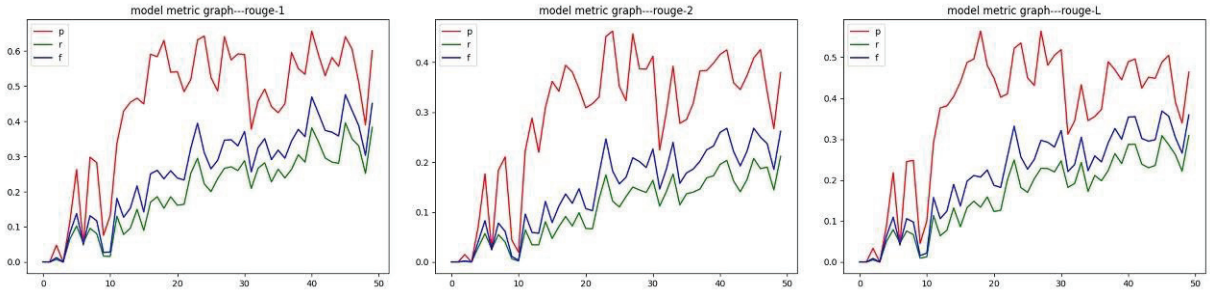

Figure 6. The weight ratio is 2:2:6

4.4.2. *Results and analysis based on different models.* This part mainly explores the model performance based on several extraction models. We choose LSTM, CNN, Transformer Model as extraction models to verify the superiority of this method.

(1) Research on Extraction Method Based on LSTM

This part uses LSTM as the extraction model, and introduces three BERT models for text vectorization, namely BERT-base, BERT-Ext, and RoBERTa, and uses four vectorization construction methods under each model, namely the most common CLS vector to represent sentence vector, and the average pooling and maximum pooling vectorization

methods are introduced. Finally, the PoolMax vectorization construction method combining average pooling and maximum pooling proposed in this paper is used. We have carried out a large number of comparative experiments to verify the superiority of the method proposed in the paper. As shown in Table 4, we can see that under each BERT model, the PoolMax vectorization construction method we proposed can achieve the best property, and best model performance is achieved under RoBERTa. Rouge-1-F1, Rouge-2-F1, Rouge-L-F1 reached 49.84, 29.09, 38.50 and other index scores, respectively. Finally, the evaluation indicators are visualized to prove the correctness of the index scores, as shown in Figure 7.

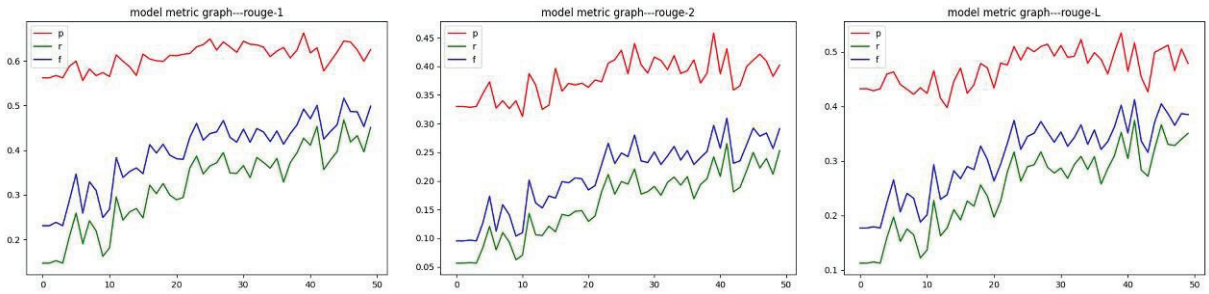

FIGURE 7. Evaluation index distribution based on LSTM

TABLE 4. F1 score based on LSTM

Model	Method	Rouge-1-F1	Rouge-2-F1	Rouge-L-F1
BERT-base	CLS	45.10	26.19	35.93
	Average	44.42	25.98	34.70
	Max	48.54	27.49	37.62
	PoolMax	**49.97**	**28.51**	**37.71**
BERT-Ext	CLS	43.22	25.23	33.95
	Average	48.86	28.22	38.13
	Max	49.03	27.81	36.41
	PoolMax	**49.42**	**28.55**	**38.07**
RoBERTa	CLS	40.89	24.28	32.92
	Average	48.98	28.13	38.14
	Max	49.71	28.21	37.51
	PoolMax	**49.84**	**29.09**	**38.50**

In this part, for the sake of study the impress of extracting the number of model layers for model property, we fixed all parameters, and used the above RoBERTa model and the proposed PoolMax vectorization method to explore the model performance at layers 1-6, as shown in Table 5. Through analysis, it is observed that when number of LSTM layers is 2, the model performance is relatively optimal, and the highest index scores are achieved in Rouge-1-F1 and Rouge-2-F1, and the score on Rouge-L-F1 is only lower than the score of 5-layer LSTM.

(2) Research on Extraction Method Based on CNN

In this part, CNN is used as the extraction model, and three BERT models are introduced for text vectorization, namely BERT-base, BERT-Ext, and RoBERTa, and four vectorization construction methods are used under each model. The most common CLS vector represents the sentence vector, and the average pooling and maximum pooling

Table 5. F1 scores of LSTM with different layers

6layer	Rouge-1-F1	Rouge-2-F1	Rouge-L-F1
1	48.88	27.71	35.59
2	**49.84**	**29.09**	38.50
3	47.46	26.48	35.85
4	48.13	26.79	38.62
5	47.66	27.51	**38.66**
6	48.14	26.15	37.60

vectorization methods are introduced. Finally, the PoolMax vectorization construction method combining average pooling and maximum pooling proposed in this paper is used. As shown in Table 6, we can see that under each BERT, our suggested PoolMax vectorization construction method can obtain better performance. And the best model performance was achieved under RoBERTa. Rouge-1, Rouge-2, Rouge-L reached 51.61, 36.77, 46.00 and other index scores, respectively, and only Rouge-1-F1 score was lower than Pool's index score. Finally, the evaluation index was visualized to prove the correctness of the index score, as shown in Figure 8.

Table 6. F1 score based on CNN

Model	Method	Rouge-1-F1	Rouge-2-F1	Rouge-L-F1
BERT-base	CLS	40.94	31.34	38.96
	Average	47.14	32.60	40.61
	Max	48.68	34.81	**44.43**
	PoolMax	**52.77**	**36.00**	42.70
BERT-Ext	CLS	50.42	33.52	42.56
	Average	**51.79**	34.36	43.43
	Max	50.33	34.37	44.15
	PoolMax	50.18	**35.55**	**44.76**
RoBERTa	CLS	51.72	33.71	43.18
	Average	**52.50**	34.10	43.43
	Max	48.42	32.92	42.15
	PoolMax	51.61	**36.77**	**46.00**

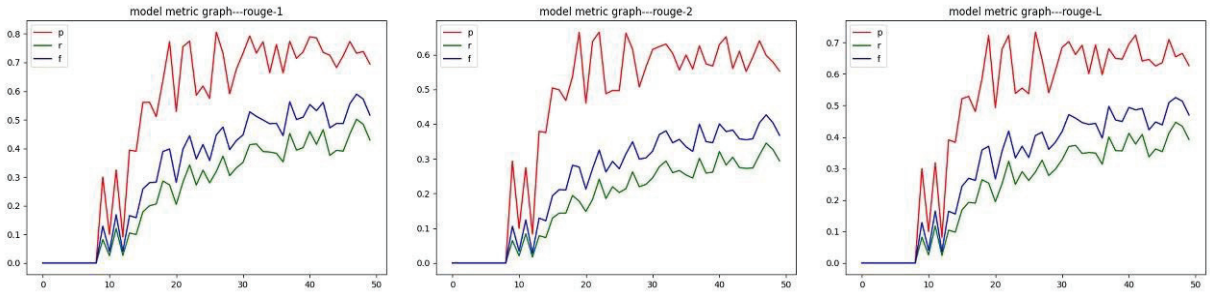

Figure 8. Evaluation index distribution based on CNN

In this part, for the sake of study the impress of CNN layers for model property, we fixed all parameters, and used the above RoBERTa model and the proposed PoolMax vectorization method to explore the model performance at layers 3-6, as shown in Table 7.

Through analysis, it is observed that when CNN has 6 layers, the performance is optimal, and the highest index scores are achieved in Rouge-1-F1, Rouge-2-F1, and Rouge-L-F1, reaching 51.61, 36.77, and 46.00 respectively.

TABLE 7. F1 scores of CNN with different layers

layer	Rouge-1-F1	Rouge-2-F1	Rouge-L-F1
3	44.60	32.19	40.39
4	47.60	33.35	42.89
5	46.45	32.81	42.53
6	**51.61**	**36.77**	**46.00**

(3) Research on Extraction Method Based on Transformer

This part takes Transformer as the extraction model, and introduces three BERT models for text vectorization, namely BERT-base, BERT-Ext, RoBERTa, and uses four vectorization construction methods under each model, respectively, the most common CLS vector to represent the sentence vector, and the average pooling and maximum pooling vectorization methods are introduced, and finally the PoolMax vectorization construction method combining average pooling and maximum pooling proposed in this paper is used. As shown in Table 8, we find that under each BERT, the suggested PoolMax vectorization construction method can achieve the best performance, and the best model performance is achieved under RoBERTa. Rouge-1, Rouge-2, Rouge-L reached 56.37, 33.39, 46.41 and other index scores, respectively. Finally, the evaluation indicators are visualized to prove the correctness of the index scores, as shown in Figure 9.

TABLE 8. F1 score based on Transformer

Model	Method	Rouge-1-F1	Rouge-2-F1	Rouge-L-F1
BERT-base	CLS	49.64	28.60	37.94
	Average	50.46	32.06	**39.61**
	Max	49.37	27.88	37.74
	PoolMax	**51.50**	**33.62**	39.17
BERT-Ext	CLS	50.40	29.70	38.33
	Average	51.45	30.33	40.20
	Max	52.91	31.02	41.57
	PoolMax	**53.29**	**31.88**	**44.38**
RoBERTa	CLS	47.60	26.98	36.35
	Average	53.24	33.34	40.57
	Max	54.29	32.17	41.94
	PoolMax	**56.37**	**33.39**	**46.41**

In this part, for the sake of study the impress of the figure of Transformer layers for model performance, we fixed all parameters, and used the above RoBERTa model and the proposed PoolMax vectorization method to explore the model performance in layers 1-6, as shown in Table 9. Through analysis, it can be seen that when the number of Transformer layers is 2, the model property is correspondingly optimal, and the highest index scores are achieved in Rouge-1 and Rouge-2, reaching 56.37 and 33.39 scores, and the score on Rouge-L is only lower than the score of the third layer Transformer.

4.4.3. *Results and analysis based on different methods.* This section analyzes the results between different methods. Table 10, 11, and 12 show the results of different methods in P, R, and $F1$ of Rouge-1, Rouge-2, and Rouge-L, then analyze the consequences.

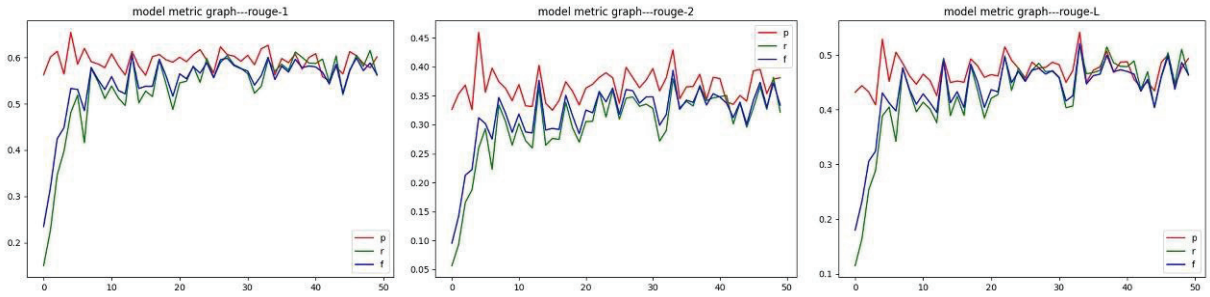

FIGURE 9. Evaluation index distribution based on Transformer

TABLE 9. F1 scores of Transformer with different layers

layer	Rouge-1-F1	Rouge-2-F1	Rouge-L-F1
1	55.35	33.16	44.37
2	**56.37**	**33.39**	46.41
3	55.72	33.25	**46.69**
4	51.58	30.09	42.35
5	49.61	27.59	38.90
6	46.51	26.56	35.80

As can be seen from Tables 10, 11, and 12, the property for our means in the paper is the best upon the three evaluation indicators. In this method, P, R and $F1$ reached 60.14%, 56.18% and 56.37% respectively on Rouge-1, 38.10%, 32.20% and 33.39% on Rouge-2, and 49.34%, 46.31% and 46.41% on Rouge-L, which reveals that proposed means in the paper is effective, and our model improves the score index of $F1$, which is due to the improved ability to extract document-level features and the ability to understand the context. First, experiments reveal that a current supervised summarization means based on a neural network is significantly better than unsupervised methods such as LEAD and TextRank. second, experiments reveal that text vectorization means based on BERT is better than Word2Vec, Glove, and other vectorization methods under the condition of ensuring that the extraction parts are the same; third, use BERT for text vectorization and compare the results of different neural network models. This method achieves better results under the three evaluation indicators. Compared with BERT + Transformer, this method is effective in Rouge-1-F1, Rouge-2-F1, and Rouge-L-F1 increased by 6.73%, 4.79%, 8.74% (from 49.64% to 56.37%, from 28.60% to 33.39%, and from 37.94% to 46.41%). This shows that the proposed improved strategy can obtain better property indicators, and the property of the method is better. Figure 10 shows the score comparison of Rouge-1-F1, Rouge-2-F1, and Rouge-L-F1 between different methods. It can be seen in Figure 10 that this method achieves the best property on Record-Oriented Understudy for Gisting Evaluation-F1 score.

5. **Conclusion.** In the paper, a long text summarization extraction means ground on the basis of pre-training model and Transformer is proposed. In this proposed model architecture, firstly, a novel weighted similarity index is proposed to construct the training data of the model. Secondly, introduce a variety of BERT models to process text to obtain sentence-level vectors to better understand text content. Finally, The Transformer model is introduced to abstract the text, which is more suitable for the extraction of

Table 10. Comparison of Rouge-1 in different methods

Method	Model	P	R	$F1$
LEAD [44]	None	10.10	7.46	8.58
TextRank [45]		21.17	15.22	17.71
Word2Vec + FC [24]		24.80	16.85	20.07
BERT + FC [19]		30.38	18.72	23.21
BERT-Base [19]	LSTM	63.38	44.12	49.97
BERT-Ext		62.50	44.30	49.42
RoBERTa		62.51	45.10	49.84
BERT-Base [3]	CNN	**72.38**	42.69	52.77
BERT-Ext		62.18	42.11	50.18
RoBERTa		69.44	42.92	51.61
BERT-Base [19]	Transformer	62.30	44.51	49.64
BERT-Ext		60.22	51.07	53.29
Proposed method	Transformer	60.14	**56.18**	**56.37**

Table 11. Comparison of Rouge-2 in different methods

Method	Model	P	R	$F1$
LEAD [44]	None	8.78	7.12	7.86
TextRank [45]		15.43	12.65	13.90
Word2Vec + FC [24]		11.73	10.25	10.94
BERT + FC [19]		14.65	11.07	12.61
BERT-Base [19]	LSTM	40.35	23.96	28.51
BERT-Ext		39.99	24.47	28.55
RoBERTa		40.20	25.25	29.09
BERT-Base [3]	CNN	**56.18**	27.47	36.00
BERT-Ext		49.29	28.38	35.55
RoBERTa		55.27	29.42	**36.77**
BERT-Base [19]	Transformer	39.89	24.53	28.60
BERT-Ext		35.96	30.61	31.88
Proposed method	Transformer	38.10	**32.20**	33.39

Table 12. Comparison of Rouge-L in different methods

Method	Model	P	R	$F1$
LEAD [44]	None	9.25	7.08	8.02
TextRank [45]		14.10	11.52	12.68
Word2Vec + FC [24]		19.81	15.79	17.57
BERT + FC [19]		23.55	17.82	20.29
BERT-Base [19]	LSTM	47.58	33.34	37.71
BERT-Ext		47.86	34.27	38.07
RoBERTa		47.89	35.07	38.50
BERT-Base [3]	CNN	58.66	33.63	42.70
BERT-Ext		55.68	37.88	44.76
RoBERTa		**62.69**	39.23	46.01
BERT-Base [19]	Transformer	47.23	34.13	37.94
BERT-Ext		47.69	44.11	44.38
Proposed method	Transformer	49.34	**46.31**	**46.41**

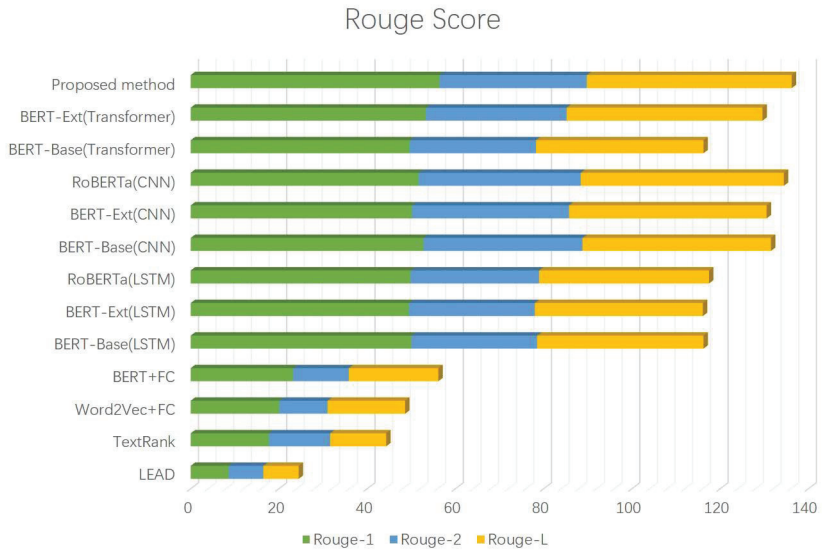

FIGURE 10. Comparison of F1 between different methods

long text data. The Transformer-based method is compared with the LSTM and CNN-based methods, and the classification layer is connected after the model to obtain the final summary. We have conducted many experiments on the CAIL2020 dataset, and the results show that the method proposed in this paper achieves the best F1 score under all three Rouge indicators.

Our work is mainly to extract long texts based on the extractive abstract method. However, the extraction method only simply extracts the text content, and does not understand and refine the article, so the abstract results obtained will be slightly blunt and not coherent enough. In the future, we will explore the generative approach based on the generative abstract or combine the extractive ways with the generative means to better understand this text and generate a summary that is more in line with language logic.

REFERENCES

[1] R. Dangovski, M. Shen, D. Byrd, L. Jing, D. Tsvetkova, and N. P, "We Can Explain Your Research in Layman's Terms: Towards Automating Science Journalism at Scale," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 14, pp. 12728-12737, 2021.

[2] C. Gulden, M. Kirchner, C. Schüttler, M. Hinderer, M. Kampf, and H. Prokosch, "Extractive summarization of clinical trial descriptions," *International Journal of Medical Informatics*, vol. 129, pp. 114-121, 2019.

[3] Y. Sun, F. Yang, X. Wang, and H. Dong, "Automatic Generation of the Draft Procuratorial Suggestions Based on an Extractive Summarization Method: BERTSLCA," *Mathematical Problems in Engineering*, vol. 2021, pp. 3591894-3591906, 2021.

[4] B. R. C. Somnath, Z. Chao, and C. Snigdha, "Unsupervised Extractive Opinion Summarization Using Sparse Coding," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, Dublin, Ireland, vol. 1, pp. 1209-1225, 2022.

[5] J. Ruipeng, Z. Xingxing, C. Yanan, L. Zheng, W. Shi, and W. Furu, "Neural Label Search for Zero-Shot Multi-Lingual Extractive Summarization," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, Dublin, Ireland, vol. 1, pp. 561-570, 2022.

[6] T. Uckan, and A. Karcl, "Extractive multi-document text summarization based on graph independent sets," *Egyptian Informatics Journal*, vol. 21, no. 3, pp. 145-157, 2020.

[7] L. Cagliero, and Q. M. La, "Extracting highlights of scientific articles: A supervised summarization approach," *Expert Systems with Applications*, vol. 160, 113659, 2020.

[8] Z. Shengqiang, Z. Xingxing, B. Hangbo, and W. Furu, "Attention Temperature Matters in Abstractive Summarization Distillation," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, Dublin, Ireland, vol. 1, pp. 127-141, 2022.

[9] D. 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.

[10] P. Kouris, G. Alexandridis, and A. Stafylopatis, "Abstractive text summarization based on deep learning and semantic content generalization," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019, pp. 5082-5092.

[11] C. Aone and M. E. Okurowski, "A trainable summarizer with knowledge acquired from robust nlp techniques," *Advances in Automatic Text Summarization*, vol. 17, pp. 71-80, 1999.

[12] 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.

[13] G. Nianlong, A. Elliott, and H. Richard, "MemSum: Extractive Summarization of Long Documents Using Multi-Step Episodic Markov Decision Processes," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, Dublin, Ireland, vol. 1, pp. 6507-6522, 2022.

[14] F. Cosme, J. Milheiro, J. Pires, F. I. Guerra-Gomes, L. Filipe-Ribeiro, and F. M. Nunes, "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, 107979, 2021.

[15] S. Ruan, B. Chen, K. Song, H. Li, "Weighted nave Bayes text classification algorithm based on improved distance correlation coefficient," *Neural Computing and Applications*, vol. 34, no. 4, pp. 2729-2738, 2022

[16] L. Yang, S. Sheng, and L. Mirella, "Noisy Self-Knowledge Distillation for Text Summarization," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (ACL)*, Online, 2021, pp. 692-703.

[17] 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.

[18] H. Dong, F. Yang, and X. Wang, "Multi-label charge predictions leveraging label co-occurrence in imbalanced data scenario," *Soft Computing*, vol. 24, no. 23, pp. 17821-17846, 2020.

[19] Y. Liu, "Fine-tune BERT for extractive summarization," arXiv preprint arXiv:1903.10318. 2019.

[20] J. Devlin, M. W. Chang, K. Lee, K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

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

[22] 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.

[23] G. Rossiello, P. Basile, G. Semeraro, "Centroid-based text summarization through compositionality of word embeddings," in *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genresm*, 2017, pp. 12-21.

[24] 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.

[25] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532-1543.

[26] H. Zhang, J. Xu, and J. Wang, "Pretraining-based natural language generation for text summarization," arXiv preprint arXiv:1902.09243, 2019.

[27] D. Anh, and N. Trang, "Abstractive text summarization using pointer-generator networks with pretrained word embedding," in *Proceedings of the Tenth International Symposium on Information and Communication Technology*, 2019, pp. 473-478.

[28] R. Arpita, and P. Shimei, "Incorporating medical knowledge in BERT for clinical relation extraction," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online and Punta Cana, 2021, pp. 5357-5366.

[29] K. Fajri, H. L. Jey, and B. Timothy, "IndoBERTweet: A Pretrained Language Model for Indonesian Twitter with Effective Domain-Specific Vocabulary Initialization," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online and Punta Cana, 2021, pp. 10660-10668.

[30] C. Lei, and W. C. Hou, "Utilizing Cross-Modal Contrastive Learning to Improve Item Categorization BERT Model," in *Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP)*, Dublin, Ireland. Association for Computational Linguistics, 2022, pp. 217-223.

[31] S. Eyal, G. Ariel, H. Alon, D. Lena, C. Leshem, and A. Ranit et al., "Cluster & Tune: Boost Cold Start Performance in Text Classification," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, Dublin, Ireland, vol. 1, pp. 7639-7653, 2022.

[32] A. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," arXiv preprint arXiv:1509.00685, 2015.

[33] J. Ling, "Coarse-to-fine attention models for document summarization (Doctoral dissertation)," 2017.

[34] C. Arman, D. Franck, S. K. Doo, B. Trung, K. Seokhwan, and C. Walter et al., "A discourse-aware attention model for abstractive summarization of long documents," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (ACL)*, vol. 2, pp. 615-621, 2018.

[35] X. Wen, and C. Giuseppe, "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)*, Hong Kong, China. Association for Computational Linguistics, 2019, pp. 3011-3021.

[36] C. Peng, and H. Le, "Sliding selector network with dynamic memory for extractive summarization of long documents," in *Proceedings of the 2021 Conference of the North American Chapter of the As sociation for Computational Linguistics: Human Language Technologies (ACL)*, Online. As sociation for Computational Linguistics, 2021, pp. 5881-5891.

[37] H. Luyang, C. Shuyang, P. Nikolaus, J. Heng, and W. Lu, "Efficient Attentions for Long Document Summarization," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (ACL)*, Online. Association for Computational Linguistics, 2021, pp. 1419-1436.

[38] C. Shuyang, and W. Lu, "HIBRIDS: Attention with Hierarchical Biases for Structure-aware Long Document Summarization," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, Dublin, Ireland. Association for Computational Linguistics, vol. 1, pp. 786-807, 2022.

[39] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, L. Liu, "Application of quantum genetic optimization of LVQ neural network in smart city traffic network prediction," *IEEE Access*, vol. 8, pp. 104555-104564, 2020.

[40] S.-M. Zhang, X. Su, X.-H. Jiang, M.-L. Chen, T.-Y. Wu, "A traffic prediction method of bicycle-sharing based on long and short term memory network," *Journal of Network Intelligence*, vol. 4, no. 2, pp. 17-29, 2019.

[41] E. K. Wang, X. Zhang, F. Wang, T.-Y. Wu, C.-M. Chen, "Multilayer dense attention model for image caption," *IEEE Access*, vol. 7, pp. 66358-66368, 2019.

[42] M.-E. Wu, J.-H. Syu, C.-M. Chen, "Kelly-based options trading strategies on settlement date via supervised learning algorithms," *Computational Economics*, vol. 59, no. 4, pp. 1627-1644, 2022.

[43] S. Kumar, A. Damaraju, A. Kumar, S. Kumari, C.-M. Chen, "LSTM Network for Transportation Mode Detection," *Journal of Internet Technology*, vol. 22, no. 4, pp. 891-902, 2021.

[44] K., I. S. Ando, and A. Okumura, "Hybrid text summarization method based on the TF method and the lead method," in *Proceedings of the 2nd National Institute of Informatics Test Collection Information Retrieval (NTCIR)*, Workshop, no. 1, pp. 5-219, 2001.

[45] 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, pp. 404-411, 2004.