Leveraging Pre-Trained Language Model for Summary Generation on Short Text

SHUAI ZHAO, FUCHENG YOU, AND ZENG YUAN LIU

School of Information Engineering, Beijing Institute of Graphic Communication, Beijing 100000, China

Corresponding author: Fucheng You (youfucheng@bigc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61773229, and in part by the Beijing Municipal Natural Science Foundation under Grant KZ201710015010.

ABSTRACT  Bidirectional Encoder Representations from Transformers represents the latest incarnation of pre-trained language models which have been obtained a satisfactory effect in text summarization tasks. However, it has not achieved good results for the generation of Chinese short text summaries. In this work, we propose a novel short text summary generation model based on keyword templates, which uses templates found in training data to extract keywords to guide summary generation. The experimental results of the LCSTS data set show that our model performs better than the baseline model. The analysis shows that the methods used in our model can generate high-quality summaries.

INDEX TERMS  Summary generation, BERT, pre-trained language model, transformers.

I. INTRODUCTION

In deep learning research, when the target task training data is less, usually pre-training and fine-tuning methods can achieve outstanding results [1]. In recent years, we have witnessed the impressive results of pre-trained language models in several sub-tasks in the field of natural language processing (NLP) [2], such as dialogue systems, machine translation, and named entity recognition. It mainly includes ELMo [3], GPT [4], BERT [5], ALBERT [6] and other models. Among the pre-training methods mentioned above, BERT [5] has the most outstanding performance. The model uses masked language modelling and next sentence prediction methods for pre-training on a large number of corpora. It can be fine-tuned according to the specific requirements of downstream tasks and can achieve satisfactory results.

In this article, we explore how to apply the pre-trained language model to short text summary generation better. The purpose of the summary generation is to automatically generate a coherent summary from a given document by rewriting or extracting to shorten the document or paragraph, to alleviate the reading pressure caused by excessive information on users. Text summary generation methods are usually divided into two types: extractive methods and abstractive methods [7]. The extractive method directly selects significant and small redundant sentences or phrases from the text to form a summary. In contrast, the abstractive method is relatively complex, but can generate novel vocabulary and does not depend on the source document. This method is more in line with the standard for humans to write summaries. Compared with previous work, the summary generation model using the pre-trained language model has made significant progress. However, these works are based on long documents [8]. Experiments show that the BERT pre-training model does not perform well to generate Chinese short text summaries.

To solve this problem, we propose a short text summary generation model based on keyword templates. Template-based summaries are an effective method in the traditional summary generation, in which domain experts manually create a number of hard templates, and then use templates to guide summary generation [9]. However, it is unrealistic to create all templates manually, requiring a lot of experts and labour-intensive. Different from previous work, the templates used in our model are all from the training set without the need for experts to recreate it.

In this article, we introduce a short text summary generation model based on keyword templates, which makes the BERT model better applied to Chinese short text summarization generation tasks. Different from BERT’s sentence division method, we have added a keyword-based sentence division method based on the original sentence division method. In the training phase, we extract keywords in the
reference summary and divide the input text; in the testing phase, we use similarity calculations to find the most similar text in the training set and extract the keywords in the reference summary of the training text for sentence division. Experiments show that our proposed method has achieved good results in abstractive model and has generated higher-quality summaries.

Contributions made by this article:

1. We introduce a short text summary generation model based on keyword templates and improve the data preprocessing method of Chinese short text in summary generation tasks.
2. We showed how to apply the pre-trained language model to generate short text summaries efficiently, and verify it through the abstractive method.
3. Our model can be used as a stepping stone to improve the quality of the summary and make the pre-trained language model better used in the generation of short text summaries.

II. RELATED WORK
In this section, we will introduce relevant work research on pretrained language models, extractive models, and abstractive models.

A. PRETRAINED LANGUAGE MODELS
The research of pre-training language models is mainly aimed at language understanding tasks, which can usually be classified into feature-based models and fine-tuning-based models according to their characteristics [1]. Feature-based methods mainly use pre-training models to provide language representations and features for the downstream tasks [10]. EMLo [3] used a bidirectional LSTM [11] language model to obtain a context-sensitive pre-trained representation. In the supervised task, they are spliced into the word vector input or the top level representation of the model as features. GPT [4] used Transformer [12] network instead of LSTM [11] as a language model to better capture long-distance language structures. When applied to downstream tasks, GPT [4] does not need to rebuild a new model structure, and can effectively improve the generalization ability of supervised models and accelerate convergence. The fine-tuning methods are mainly to pre-train the model on the language modeling target, and then fine-tune the model on the downstream tasks with supervised data. The BERT model uses “masked language modelling” and “next sentence prediction” methods to train on a large-scale corpus. BERT can be widely used because it can be applied to multiple downstream tasks of natural language processing by fine-tuning and achieving outstanding results. Unlike the BERT pre-training model, ALBERT [6] is faster to train and uses less memory. ALBERT used factors such as factorization and cross-layer parameter sharing to reduce model parameters and improve training speed effectively.

In past research, pre-trained language models are usually applied to natural language understanding tasks to improve their performance. Recently, many scholars have applied pre-trained language models to generation tasks. For example, BERT can fine-tune its parameters with specific generation task parameters. In this article, we try to use the BERT model for the text summary generation task.

B. EX extractive models
The extractive summary first scores all sentences in the document according to their importance, then sorts the sentences according to the score, and finally select multiple sentences with the highest scores to form a summary. The early summary generation model mainly used human feature engineering, and its common methods include context matching [13], graph model [14], but the model effect is not very satisfactory. With the advancement of artificial intelligence, deep neural networks are widely used in summary generation tasks. Yin et al. [15] applied neural networks to extractive summarization tasks, mapping sentences into vectors, and selecting vital sentences to form summaries. Yasunaga et al. [16] combined a recurrent neural network with a graph convolutional network to calculate the importance of each sentence, and then select the sentence to form a summary. Shashi et al. [17] performed global optimization of ROUGE [18] metrics through reinforcement learning, conceptualized extractable single document summaries as sentence ordering tasks and proposed a novel training algorithm. This model improves the use of cross-entropy as the loss function in the model training process, which may lead to summary length and excessive redundant information. The SUMO [19] model proposed an end-to-end extractive text summary generation method, which regards single-document extractive summaries as a tree induction problem. The model subtree is the sentence in the original document related to the summary or explains the summary, thereby improving the correlation between the model generated summary and the text. Recently, pretrained language models has been shown to be useful for improving text summarization tasks. Yang Liu [20] applied BERT [5] to extractive summaries for the first time, the experimental results show that the pre-training model is also suitable for text summarization generation. Hongwang et al. [21] tried three different pre-training strategies, Mask, Replace and Switch, and used self-supervised methods to capture the global features of the document to more accurately grasp the main content of the article. The HIBERT [22] model treated the task as a sequence labelling task and marks whether a sentence appears in summary in the original text. Zhong et al. [23] proposed a completely new method to transform the extractive summary task into a semantic matching problem. The advantage is that the candidate summary (several sentences) can be directly extracted instead of sentence-level (sentence by sentence).

C. ABSTRACTIVE MODELS
Compared with the extractive method, the abstractive method is closer to the standard of human summary writing. In recent years, there has been more and more research on abstractive methods. The abstractive method regards the task of summary generation as a sequence-to-sequence
problem and used neural networks to solve it. Nallapati [24] and Chopra et al. [25] used RNN to replace traditional the encoder and decoder and achieved good results. Lin et al. [26] proposed a global coding framework based on context information to improve the model’s ability to control global details. The model uses a gated convolution unit to ensure that the core information is retained and filters redundant information. Chen et al. [11] proposed an accurate and fast summary model, first select some crucial sentences, and then generate operations on these sentences. The model uses a novel sentence-level strategy gradient method to connect the sentence extraction network and the summary generation network. Not only the best experimental results are obtained, but also a significant speed increase in decoding and training. Zhang et al. [27] applied the BERT model to abstractive summaries for the first time. The model uses BERT as the encoder to extract the input document’s features and then uses the Transformer to decode to generate the initial results, masked the output, and then made predictions through another BERT. Logan Lebanoff [28] proposed to map single sentences and pairs of sentences to a unified space for sorting. According to this sorting, single sentences and paired sentences that are important for the summary are selected. Finally, by compressing the single sentences, the paired sentences are merged to generate the summary.

Different from the previous work, we propose a short text summary generation model based on keyword templates. Our model uses different data processing methods, which can better apply the pre-trained language model to generate short text summaries in Chinese and generate high-quality summaries.

III. MODEL

In this section, we will introduce the overall structure of the model from two parts: data preprocessing, model architecture.

A. DATA PREPROCESSING

Unlike the BERT data preprocessing method, we added a keyword-based sentence division form based on the original sentence division model. The experimental results show that simple sentence division does not apply BERT to the generation of short text summaries. Therefore, based on the original sentence division model, we extract the reference summary keywords in the training set and divide the text twice.

In the model training stage, we extract keywords from the reference summary and divide the input text twice. In the model testing stage, we use the similarity calculation tool (macropodus) [29] to find the most similar data in the training set to the test document. The keywords in the reference summary of the data are extracted to divide the test document twice.

As shown in the left half of Figure 1, we divide the text according to the sentence structure. In the right half of the figure, we first extract “the summary generation” as a keyword, and divide the text again based on the original sentence division.

We insert [CLS] before each sentence (or phrase) obtained after preprocessing and insert [SEP] at the end of each sentence (or phrase) to represent each sentence (or phrase). And use the inserted [CLS] symbols to collect sentence features. The document is represented as a sequence of tokens $X = [w_1, w_2, \ldots, w_n]$. Each token $w_i$ consists of token embedding, segment embedding, and position embedding, as shown in Figure 1. Among them, token embedding converts each word into a fixed-dimensional vector; position embedding represents the position information of each token in the text; segmentation embeddings divide the sentence into $E_A$ or $E_B$ mainly according to whether $i$ is even or odd [20]. For example, suppose the text contains a total of six sentences, we divide $[\text{sent}_0, \text{sent}_1, \text{sent}_2, \text{sent}_3, \text{sent}_4, \text{sent}_5]$ into $[E_A, E_B, E_A, E_B, E_A, E_B]$. 

B. MODEL ARCHITECTURE

To verify our proposed method’s effectiveness in generating short text summaries, we use the traditional Encoder-Decoder structure and fine-tune BERT. The encoder uses BERT for initialization, and the decoder uses eight Transformers stacked with random initialization [8]. We name our
model BSA. If the model uses keyword templates to divide sentences, it named BSA*.

We represent the preprocessed data as \([sent_1, sent_2, sent_3, \ldots, sent_n]\) and input it into the BERT layer. The vector \(i_t\) is the feature vector collected by the \(i\) th [CLS] symbol output by the BERT layer, which can be used to represent the sentence \(i\). After the Encoder, we use the Decoder stacked by Transformers to decode the BERT output:

\[
\begin{align*}
\tilde{h}_i &= LN(h_{i-1} + MHAtt(h_{i-1})) \\
 h_i &= LN(\tilde{h}_i + FFN(\tilde{h}_i))
\end{align*}
\]

where \(LN\) refers to the normalization layer; \(MHAtt\) is the multi-headed attention composed of multiple self-attention mechanisms connected [20]; the symbol \(l\) represents the depth of the model stack. \(FFN\) is feedforward neural network between the encoder and decoder of each layer.

\[
FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2
\]

In formula (1), \(h_0\) is calculated by \(PosEmb(T)\); vector \(T\) means the sentence vectors output by the encoder layer. \(PosEmb\) refers to the function of adding positional embeddings (refers to the position information of each sentence in the text) to vector \(T\).

\[
\begin{align*}
PE(pos, 2i) &= \sin(pos/10000^{2i/dmodel}) \\
 PE(pos, 2i + 1) &= \cos(pos/10000^{2i/dmodel})
\end{align*}
\]

where \(pos\) indicates the position and \(i\) refers to the dimension.

Using Transformers as a feature extractor can effectively extract document-level features and generate higher quality and more coherent summaries. The decoder focus on extract document-level features and generate higher quality summaries. The decoder focus on improving the experimental results under different topic data by both the BERT layer. The vector \(i_t\) is the feature vector collected by the \(i\) th [CLS] symbol output by the BERT layer, which can be used to represent the \(i\) th sentence. After the Encoder, we use the Decoder stacked by Transformers to decode the BERT output:

\[
\begin{align*}
\tilde{h}_i &= LN(h_{i-1} + MHAtt(h_{i-1})) \\
 h_i &= LN(\tilde{h}_i + FFN(\tilde{h}_i))
\end{align*}
\]

where \(LN\) refers to the normalization layer; \(MHAtt\) is the multi-headed attention composed of multiple self-attention mechanisms connected [20]; the symbol \(l\) represents the depth of the model stack. \(FFN\) is feedforward neural network between the encoder and decoder of each layer.

\[
FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2
\]

In formula (1), \(h_0\) is calculated by \(PosEmb(T)\); vector \(T\) means the sentence vectors output by the encoder layer. \(PosEmb\) refers to the function of adding positional embeddings (refers to the position information of each sentence in the text) to vector \(T\).

\[
\begin{align*}
PE(pos, 2i) &= \sin(pos/10000^{2i/dmodel}) \\
 PE(pos, 2i + 1) &= \cos(pos/10000^{2i/dmodel})
\end{align*}
\]

where \(pos\) indicates the position and \(i\) refers to the dimension.

Using Transformers as a feature extractor can effectively extract document-level features and generate higher quality and more coherent summaries. The decoder focus on extract document-level features and generate higher quality summaries. The decoder focus on improving the experimental results under different topic data.

IV. EXPERIMENTAL DETAILS

In this section, we introduce the data set used in the experiment, model parameter settings, and the baseline models compared with the experiment.
by classifying the training set data according to the topic. CGU [34] is a seq2seq model base on the convolutional gated unit and the attention mechanism. RTCS [35] is the conventional seq2seq model with convolutional neural network.

Besides, we have proved the effectiveness of our proposed method through ablation experiments.

V. EXPERIMENT ANALYSIS

In this section, we verify the experimental results of the BSA and BSA* models in the short text summary and conduct a comparative analysis. Besides, we also showed two summary examples to illustrate that our model can generate high-quality summaries.

A. RESULTS ON LCSTS DATASET

Table 1 shows the experimental results of our model and baseline model on the LCSTS data set. Our model achieves a better performance with 44.2 F-score of ROUGE-1, 28.9 ROUGE-2 and 39.2 ROUGE-L. Compared with the RTCS model, our model is improved by 4.3, 7.4, and 1.3 on ROUGE-1, -2, and -L. Besides, compared with the unimproved model(BSA), the BSA* model is improved by 3.8, 3.0, and 2.5 on ROUGE-1, -2, and -L.

TABLE 1. Results of our Model and Baseline Systems in LCSTS Dataset. Besides, the Table also Contains the Results of the Ablation Experiment

| Model  | ROUGE-1 | ROUGE-2 | ROUGE-L |
|--------|---------|---------|---------|
| RNN [11] | 21.5    | 8.9     | 18.6    |
| Bi-RNN [31] | 29.9    | 17.4    | 27.2    |
| SRB [32] | 33.3    | 20.0    | 30.1    |
| CopyNet [13] | 34.4    | 21.6    | 31.3    |
| TSNGH [33] | 38.4    | 26.6    | 36.1    |
| CGU [34] | 39.4    | 26.9    | 36.5    |
| RTCS [35] | 39.9    | 21.5    | 37.9    |
| BSA | 40.4 | 25.9 | 36.7 |
| BSA* | 44.2 | 28.9 | 39.2 |

Compared with the model using RNN as the feature extractor (RNN, Bi-MulRnn), our model uses Transformers as the feature extractor to capture the document’s in-depth features more efficiently. Besides, our model uses a pre-trained BERT model compared with other baseline models, which can effectively improve model performance and generate higher quality summaries.

Through the comparative analysis of ablation experiments, our model results show that the use of keywords to subdivide the text can effectively improve the quality of the generated summaries and better apply the pre-trained language model to Chinese short text summaries.

B. DISCUSSION

We show two examples of summaries generated by our model and compare them with RNN models and reference summaries. At the same time, we also show the summaries generated by our model under different data processing methods. As shown in the summary example in Table 2, compared with the RNN model, the summary generated by our model can highlight the main content of the text. In contrast, the BSA model can not accurately express the main idea of the text. For example, in the first example, there is no giant in Shanghai Internet, but the model’s summary thinks that Shanghai Internet is only a few giants. Compared with the BSA model, the BSA* model’s summary represents the text’s main content more accurately. Simultaneously, the second summary example can effectively illustrate that the pre-trained model can generate a higher quality summary.

VI. CONCLUSION

To improve the application of the pre-trained language model in Chinese short text summary generation, this article proposes a novel Chinese short text summary generation model based on the keyword templates. In our model, we divide the text twice by extracting keywords. Experimental results show that our model’s document sentence division method effectively improves the generated summary quality. We offer how to efficiently apply the pre-trained language model to the generation of short text summaries.

However, this method has only been verified on the abstractive model. In future work, we will verify whether the short text summary generation model based on keyword templates can improve the extractive model summary generation’s quality and verified on other pre-trained language models.

REFERENCES

[1] K. Song, “MASS: Masked sequence to sequence pre-training for language generation,” in Proc. ICLR Conf., 2019, pp. 1–11.
[2] Z. Lu, “VGCN-BERT: Augmenting BERT with graph embedding for text classification,” in Proc. ECIR Conf., 2020, pp. 369–382.
[3] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., 2018, pp. 1–15.

TABLE 2. Two Summary Examples Include Source Text, Reference Summary, RNN Model Generation Summary, BSA Model Summary, and BSA* Model Summary

| Example | Source | Reference | RNN Model | BSA Model | BSA* Model |
|---------|--------|-----------|-----------|-----------|-----------|
| First summary example | After careful calculation, there are many successful cases of Internet companies in Shanghai, but few of them eventually become giants. This can also explain why there are few Internet companies in the top 100 taxpayers list. There are some mergers and acquisitions, such as eBay, tudou.com, PPS, PPTV, Yihadian, etc. The other is that they are inclined to segment the market for several years. | Why can’t Shanghai be an Internet giant? | Shanghai’s Internet giant. | Why are Shanghai’s Internet giants less successful? | Why did Shanghai Internet’s giants fail? |
| Second summary example | The multi-person smoking crew of the Chengdu-Beijing flight clashed with non-smokers. Last night, on a China United Airlines flight from Chengdu to Beijing, it was found that many people were smoking. Due to the weather, the plane later landed at Taiyuan Airport, and several passengers were found smoking by the door. Therefore, some passengers requested a new security check, but the captain decided to continue the flight, causing conflict between the crew and non-smokers. At present, China United Airlines is contacting the crew for verification. | The multi-person smoking crew of the Chengdu-Beijing flight clashed with non-smokers. | Many non-smokers on the Chengdu-Beijing flight clashed with the crew. | Many passengers smoked on the flight from Chengdu to Beijing, and the crew clashed with non-smokers. |
