scientific reports

Check for updates

OPEN

A study of extractive summarization of long documents incorporating local topic and hierarchical information

Ting Wang[1], Chuan Yang[1], Maoyang Zou[2]✉, Jiaying Liang[1], Dong Xiang[1], Wenjie Yang[1], Hongyang Wang[1] & Jia Li[1]

In recent years, the transformer-based language models have achieved remarkable success in the field of extractive text summarization. However, there are still some limitations in this kind of research. First, the transformer language model usually regards the text as a linear sequence, ignoring the inherent hierarchical structure information of the text. Second, for long text data, traditional extractive models often focus on global topic information, which poses challenges in how they capturing and integrating local contextual information within topic segments. To address these issues, we propose a long text extractive summarization model that employs a local topic information extraction module and a text hierarchical extraction module to capture the local topic information and document's hierarchical structure information of the original text. Our approach enhances the ability to determine whether a sentence belongs to the summary. In this experiment, ROUGE score is used as the experimental evaluation index, and evaluates the model on three large public datasets. Through experimental validation, the model demonstrates superior performance in terms of ROUGE-1, ROUGE-2, and ROUGE-L scores compared to current mainstream summarization models, affirming the effectiveness of incorporating local topic information and document hierarchical structure into the model.

Text summarization is an arduous task in the field of natural language processing (NLP)[1], wherein the goal is to generate a concise and logically connected summary of a given document. This process involves extracting crucial information and reduce the length of the document while preserving the essential meaning[2,3]. Text summarization can effectively reduce the information burden of users, enable users to quickly obtain information from redundant information, greatly reduce manpower and material resources. It plays an important role in various domains, including information retrieval, title generation and other related fields.

Based on the methodology employed, text summarization tasks can be categorized into two types: extractive summarization[4] and abstractive summarization[5]. The abstractive summarization method utilizes neural network-based approaches, such as the Sequence-2-Sequence (Seq2Seq) architecture[6], also known as encoder-decoder architecture. The principle of an encoder-decoder is similar to the way human think or write summaries. The encoder first encodes the full text, and then the decoder generates new sentences word by word to form a document summary. This method generates less redundant summary information, but might face challenges in maintaining fluency and grammatical correctness. In addition, the generation of new words or phrases may produce summaries that are inconsistent with the original statement[7]. These issues can be mitigated by directly selecting sentences from the source text and assembling them into summaries, i.e. the extractive summarization. The extractive method treats summarization as a classification problem, where important sentences are directly selected from the source text to construct a summary. Summaries generated through this approach often exhibit a good performance in fluency and grammar. For the extractive summarization task, the core challenge lies in learning comprehensive sentence context information and modeling inter-sentence relationships through the encoder, thereby enabling sentence classifiers to extract more valuable sentences. Traditional extractive methods usually employ graph-based methods or clustering-based methods for unsupervised summarization[8,9]. These approaches construct the correlation between sentences using cosine similarity, and then use sorting methods to

[1]School of Computer Science, Chengdu University of Information Technology, Chengdu 610225, Sichuan Province, China. [2]College of Blockchain Industry, Chengdu University of Information Technology, Chengdu 610225, Sichuan Province, China. ✉email: zoumy@cuit.edu.cn

calculate the importance of sentences. With the rapid development of deep learning, many extractive summarization methods use Recurrent Neural Network (RNN) to capture the relationship between sentences[10,11]. However, RNN-based methods are difficult to deal with long-distance dependencies, especially for long document summaries. In recent years, transformer[12] language model, which has been pre-trained by large-scale corpus, has achieved excellent results when fine-tuned for downstream tasks, and have found widespread application in the field of text summarization. Liu et al. [13] proposed the BERTSUM model by improving the BERT embedding layer. They applied the BERT model for the first time in the text summarization and achieved state-of-the-art (SOTA) performance on CNN/DailyMail dataset. Zhang et al. [14] designed a hierarchical transformer to capture long-range inter-sentence relationships. However, this method did not yield significant performance gains for summarization tasks and faced challenges such as slow training speed and potential overfitting. At the same time, some researchers introduced the neural topic model (NTM)[15] and graph neural network (GNN)[16] into the task of text summarization to capture global semantic information and further guide the generation of abstracts. Cui et al. [17] use NTM to capture the theme features of documents and GNN to represent documents as graph structures, thus obtaining the relationship between sentences.

However, for long document summarization tasks, the above methods have two shortcomings. The first one is that they fail to recognize the explicit hierarchical structures and section headings inherent within the long document. When manually summarizing text, we tend to focus on the main sections. For example, in the context of scientific papers, more attention may be given to sections like "Methodology", "Experimental" and "Conclusion", but "Background" or "Related Work" may not receive as much emphasis. In addition, sentences within a section have stronger relationships compared to those outside the section. Understanding the logical relationship between sentences and the hierarchical structure within the document helps the model better identify the important sentences. However, the traditional transformer-based text summarization methods often regard the text as a sequential structure, and struggle with longer documents. The second shortcoming is that the longer the document, the more topics it may discuss, because each section presents different topic information. In summary, the aforementioned methods focus on the overall topic information of the entire document, that is, the global information, neglecting the local topic information of individual sections. In order to address these issues, this paper proposes a long-document extractive summarization model that integrates local topic information and document hierarchy information into current topic segment.

The main contributions of this paper can be summarized as follows:

(1) Introduction of an innovative long-document extractive summarization model. This model consists of a text encoder, a module for extracting local topic information, and a module for embedding hierarchical structure information of the document. The information is integrated into the sentence representation of the document, enhancing the quality of the generated summaries.
(2) This paper utilizes LSTM-Minus[18] to obtain distributed representations of local information and combines it with text summarization tasks. Instead of employing a fixed three-segment approach for text paragraphing, the paper adopts a dynamic method based on the number of sentences to determine paragraph length, thereby calculating the starting and ending positions of each paragraph in the text. Paragraph segments are divided based on these positions, and their topic information is computed.
(3) Experimental results conducted on the PubMed dataset reveal excellent performance of the proposed method when compared to several baseline models.

Related Work

Extractive summarization method

With the rapid development of neural networks, significant achievements have been made in extractive summarization tasks. At present, the extractive methods are mainly regarded as sentence sorting task or binary sequence labeling tasks. In the sentence sorting paradigm, models are required to assign scores to each sentence in the text and place higher-scored sentences at the front of the summary list while lower-scored sentences are placed towards the back. This process yields an ordered list of sentences, and the top few sentences are selected as the summary. Narayan et al. [19] proposed a topic-aware convolutional neural network model. This model first extracts features from the documents using convolutional neural networks and then weights the features according to the topic. Finally, a selection-based sorting method is employed to choose the most relevant sentences as the summary. Experiments results on multiple datasets show that this approach can generate concise summaries that still preserve valuable information. Li et al.[20] proposed a method for evaluating sentence importance in multi-document summarization using variational autoencoder. Different from the traditional method based on feature engineering, this method directly learns the abstract semantic representation directly from the original data. KL divergence is introduced to constrain the generated sentence representations to be close to the prior distribution, thereby improving the generalization ability of the model.

Regarding the second paradigm, which considers extractive text summarization as a sequence labeling task, this approach involves extracting and encoding features for each sentence or paragraph. The encoded features are then input into a decoder for labeling prediction to determine which sentences should be selected for the summary. The sequence labeling method has been widely applied in extractive text summarization and has achieved good results. Nalapati et al.[4] proposed the SummaRuNNer model for text summarization, which is a sequence model based on RNN. This model generates document summarization by learning the importance of each sentence within the document. It has demonstrated good summarization performance on multiple text datasets. Zhang et al.[21] introduced a latent variable extractive model, which treats sentences as latent variables and infers summaries using sentences with activated variables.

However, most of the methods mentioned above rely on RNN for extractive summarization. RNN-based methods face challenges in handling long-distance dependencies at the sentence level and may omit on language or structural information due to the input format of the original document. In order to address these issues, researchers have started utilizing transformer-based pre-training language model as encoders and representing documents through more intuitive graph structures. They have also incorporated NTM to extract topic features from the documents, further guiding the models to produce high-quality summaries. Jia et al.[22] proposed a method called deep differential amplifier for extractive summarization, which enhances the features of summary sentences by contrast to non-summary sentences using differential amplifiers. Shi et al.[23] proposed a star architecture-based extractive summarization method, where sentences in documents are modeled as satellite nodes, and a virtual central node is introduced to learn the inter-sentence relationships using the star structure. This approach achieved promising results on three public datasets. Ma et al.[24] embedded the topic features extracted by NTM into BERT to generate a vector representation with topic features, thus improving the quality of summaries.

Although the aforementioned methods have succeeded in modeling inter-sentence relationship and extracting global semantics, there is still a problem with extractive text summarization methods based on transformer pre-training language models. The length of the input document in text summarization is longer compared to general natural language processing task, and using just a transformer-based encoder is insufficient for effectively handling long texts and often leads to high computational costs. To better understand the original document, researchers have proposed various improvements. Xie et al.[25] first preprocessed the documents by dividing them into blocks with the same size, encoded each block with block encoding. They merged the block encoding results with NTM to generate global topic features. Finally, they established a comparison graph between topic features and sentence features to filter summary sentences. This method has achieved good results in both long documents and short news documents, with particular advantages in handling the former. Beltagy et al.[26] introduced the Longformer model, specifically designed for processing long documents. By replacing the self-attention mechanism of the transformer with a sliding window self-attention mechanism, the time complexity is reduced to linear level, enabling the model to handle long documents easily. Although the Longformer performs well in handling long documents, it fails to model local semantic information and document hierarchy structure, which affects its performance. Therefore, this paper uses the Longformer as the encoder and incorporates local contextual information of the current topic segment and hierarchical structure information of the document. This allow our model to prioritize local topic information and overall structural information when dealing with long scientific papers.

LSTM-Minus

Wang et al.[27] proposed the LSTM-Minus method for the first time, and applied it to dependency parsing and achieved good results. The LSTM-Minus method is a novel approach for learning embedding of text segments, utilizing subtraction between LSTM hidden vectors to learn the distributed representation of sentence segments. Initially, a sentence is divided into three segments (prefix, infix and suffix), and the segment from the word w_i to the word w_j is represented by the hidden vector $h_j - h_i$. This allows the model to effectively learn segment embedding from both external and internal information, thus enhancing its ability to obtain sentence-level information. In the same year, Cross et al.[28] extended the unidirectional LSTM-Minus to the bidirectional, using it as sentence span representation and achieving impressive performance in component syntactic analysis tasks. Build upon this idea, we applied this method to the field of text summarization to extract the contextual information from local topic segments.

Method

To address the limitations of the existing extractive text summarization methods, this paper proposes a long document extractive summarization model that integrates local contextual information and document-level hierarchical information from the current topic segment. The model is inspired by the long document extractive model proposed by Ruan et al.[29], which incorporates hierarchical structure information. The final model of this paper is obtained by incorporating local topic information. Experiments results show that the inclusion of local topic information further deepens the model's understanding of long texts. The task of long text extractive summarization is defined as: follow: Given an original document $\mathrm{D} = \{sent_1, \ldots, sent_n\}$, D contains n sentences, where each sentence denoted as the $sent_i$, represents the i-th sentence of the original document. The purpose of the extractive text summarization model is to select m sentences capturing the central idea of the original text as summaries, where m is the desired number of summary sentences ($m \ll n$). This task is typically treated as a sentence classification problem. For each sentence $sent_i$, there is a corresponding label $y_i \in \{0, 1\}$, where a label of 1 means that the sentence belongs to the summary, while 0 indicates that it does not.

The proposed model, as shown in Fig. 1, comprises three main modules: a pre-trained language model based encoder, a local topic information extraction module (referred to as the Topic Segment Representation module in the Fig. 1), and a text hierarchical information embedding module. Because this work deals with long text corpus, the encoder used is based on the Longformer, an improvement over the transformer pre-training language model, which allows for better encoding of long documents. Once the contextual representation of the document is obtained through the encoder, it is passed to the local topic information extraction module, which extracts the topic information of the sentence segment it belongs to. The specific structure of this module is shown in Fig. 2. Then, the local topic information representation is fused with the text contextual representation, resulting in a fusion of the local topic information and the textual context. The text hierarchical structure information embedding module embeds the hierarchical structure information of the text into the fused representation of the local topic information and textual context. By using a two-layer stacked transformer, this module learns

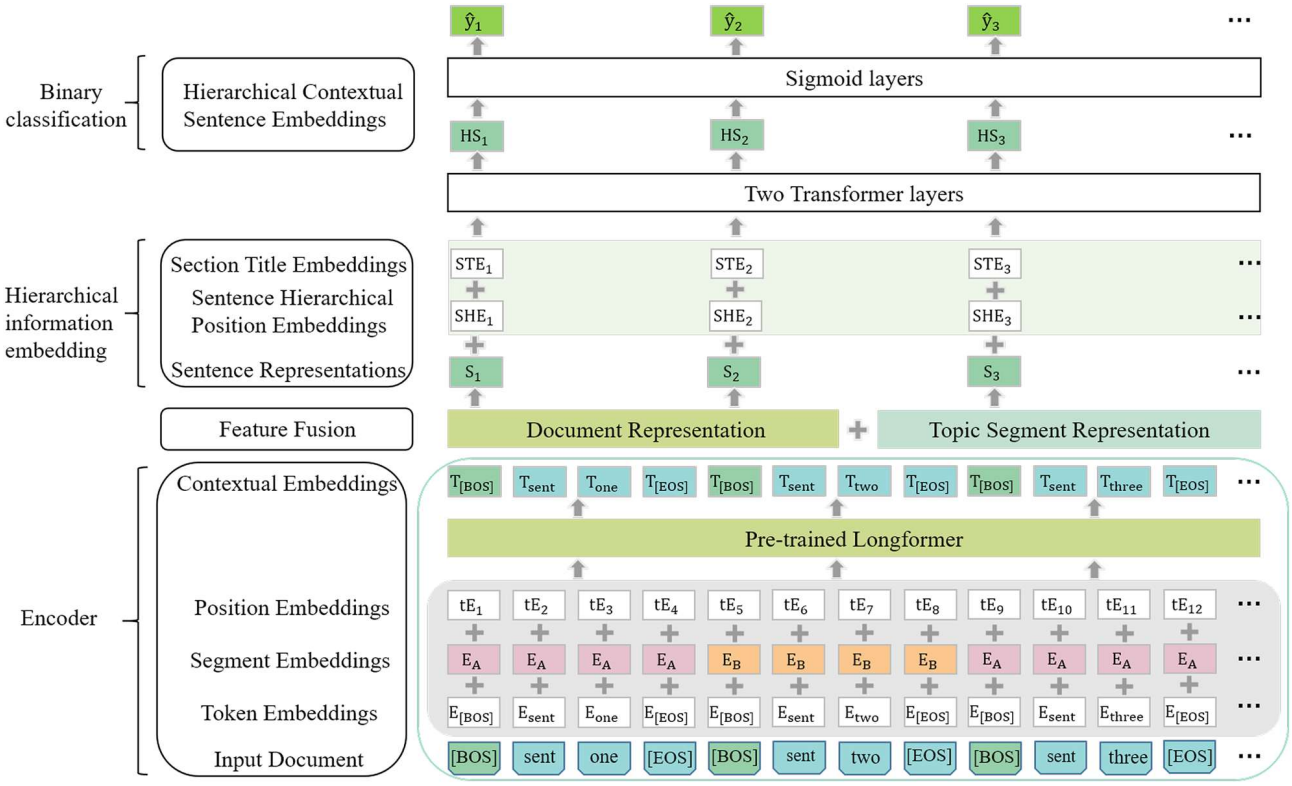

Figure 1. Overall structure diagram of the model.

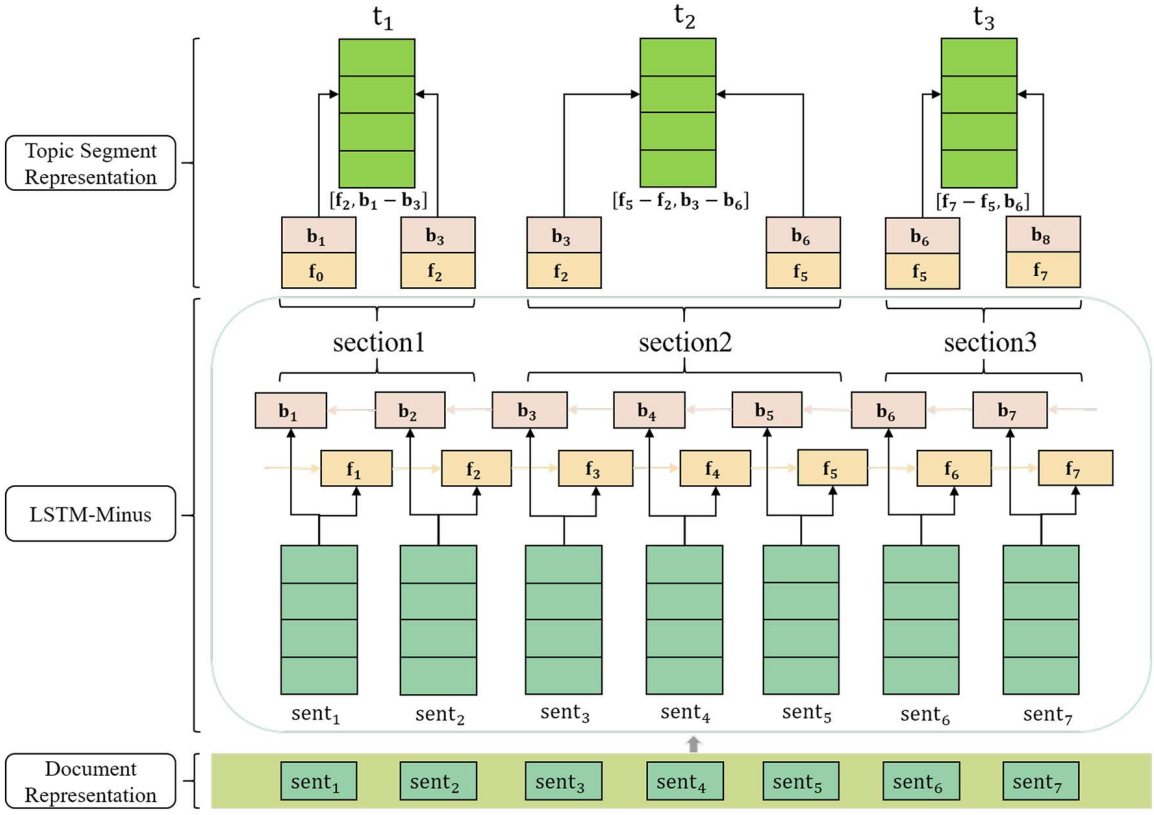

Figure 2. Local topic information extraction module.

the hierarchical structure information at both the sentence and document levels, enabling the model to gain a deeper understanding of the text context. Finally, the confidence score of each sentence is calculated through Sigmoid layer for each sentence to determine whether it should be included in the summary.

Text hierarchical information

Sentence hierarchical information

Due to scientific papers consisting of multiple sections, with each section containing several paragraphs that describe different topics, this paper uses paragraphs as the unit for hierarchical division of the article. The sentence-level hierarchical structure information includes the linear position of the paragraph to which the sentence belongs and the linear position representation of the sentence within the paragraph. The positions of paragraphs and sentences are represented by numerical serial numbers corresponding to them. For an original document $D = \{sent_1, \ldots, sent_n\}$, the hierarchical structure information of the i-th sentence $sent_i$ is expressed as a two-dimensional vector (s_s, s_g), which indicates the position of the sentence at this level, as shown in Formula (1).

$$vsent_i = (s_s, g_s) \tag{1}$$

where s_s represents the linear position of the paragraph containing the sentence relative to the entire article, and g_s represents the linear position of the sentence within its respective paragraph. All sentences within the same paragraph share the same value in the first dimension of the *vsent* vector, indicating a higher correlation among sentences within the same paragraph. And the g_s vector further indicates the linear relationship among sentences within the paragraph.

Section title information

Compared with short news articles, scientific papers often have section titles. The content within each section is usually highly relevant to the corresponding section title, as the section title serves as a concise summary of the content of the section. In this study, when encoding sentences, the section titles are incorporated as additional hierarchical information into the sentence encoding. However, for scientific papers, there are many similar section titles with the same meaning. For instance, "Method" and "Methodology" have similar meanings and can be grouped together under the "Method" category. Therefore, for the PubMed dataset used in this paper, eight section title categories are defined[29], including "introduction", "background", "case", "Method", "result", "discussion", "conclusion", and "additional information". If the section title of a section does not fall into any of the eight predefined categories, the original section title itself is directly used.

Encoder

Document encoding

The purpose of document encoding is to encode the sentences of the input document into a vector representation with a fixed length. Previous methods for extractive text summarization tasks often employed RNN and BERT[30] as encoders. BERT is a bidirectional transformer encoder that is pre-trained on large-scale corpus and has achieved excellent performance on various natural language processing tasks. However, for long text data, BERT cannot process the entire document, which will lead to information loss. Therefore, in this paper, we uses the Longformer pre-training language model as the text encoder. Longformer improves the self-attention mechanism of the traditional transformer into the sliding window self-attention, which makes it easy to handle documents with thousands of characters. In the traditional transformer self-attention mechanism, the calculation is performed by linearly transforming the input word embedding matrix to generate a Query matrix (Query, Q), a Key matrix (Key, K), and a Value matrix (Value, V) of dimension d. The specific calculation process is shown in Formula 2.

$$\text{Attention} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

where $(Q, K, V) \in R^{L\times d}$, and d represents the dimension of a word vector matrix, while d_k represents the dimension of the K matrix. Hence, the spatial complexity of the traditional transformer self-attention mechanism is $O(L^2)$, the spatial complexity of Longformer's sliding windows self-attention mechanism is $O(L)$, scaling linearly with the input sequence length L. As a result, Longformer has more advantages in encoding long texts.

As shown in Fig. 1, in order to obtain the representation of each sentence, we inserts [BOS] (beginning of sentence) and [EOS] (end of sentence) tags at the beginning and end of each sentence respectively. The model embedding layer includes Token Embeddings (TE), Segment Embeddings (SE) and Position Embeddings (PE). These features are sum to obtained the embedded representation of each word. Subsequently, the context of the input sequence is learned by using the pre-trained Longformer. The entire procedure is illustrated in Eqs. (3) and (4).

$$w_{i,j} = (TE + SE + PE) \tag{3}$$

$$\{h_{1,0}, h_{1,1}, \ldots, h_{N,0}, \ldots, h_{N,*}\} = \text{Longformer}(w_{1,0}, w_{1,1}, \ldots, w_{N,0}, \ldots, w_{N,*}) \tag{4}$$

where $w_{i,j}$ represents the j-th word of the i-th sentence, which is obtained by Formula 3. $w_{i,0}$ and $w_{i,*}$ represent the [BOS] and [EOS] tags of the i-th sentence respectively, and $h_{i,j}$ represents the hidden state of the corresponding word. After Longformer encoding, we use the [BOS] tag as the context representation of each sentence, that is,$H_s = (h_{1,0}, \ldots, h_{N,0})$.