Building Method of a BERT-Based Model for Key Information Extraction from Chemical Engineering Literature

Zhenhua Liu and Shoulong Ma(✉)

School of Artificial Intelligence and Big Data, Hefei University, Hefei 230000, Anhui, China
398007321@qq.com

Abstract. For researchers, it is essential to access various literature in order to stay up-to-date with the latest advancements and trends in scientific research. Chemical engineering literature, characterized by its diverse range, lengthy articles, complex experimental conditions, and numerous references to chemical compounds, poses a challenge in terms of manually extracting research content from a massive volume of literature. Relying solely on human effort to extract information from chemical engineering literature would be extremely time-consuming and resource-intensive. To enhance the speed at which chemical engineering researchers acquire knowledge and reduce the time spent on reading literature, this paper proposes a text summarization model based on BERT (Bidirectional Encoder Representations from Transformers). The model aims to generate concise summaries corresponding to the key information found in chemical engineering literature. Based on the textual content generated by the model, it has achieved satisfactory results in extracting key information from chemical engineering literature.

Keywords: Literature reading · information extraction · BERT model · text summarization

1 Introduction

With the development of education and research in recent years, a large number of literature sources have emerged, containing the latest research findings and methodologies. More and more researchers rely on reading literature to learn and grasp the latest and cutting-edge knowledge. Similarly, researchers and professors in the field of chemical engineering also need to read a massive amount of literature to acquire the latest knowledge [1]. By leveraging deep learning methods and building a BERT-based model for information extraction from chemical engineering literature, we can extract key information from these documents.

To expedite research work, researchers aim to read the minimum number of words while obtaining the crucial content from the literature. Through text summarization techniques, we can summarize the key information from the literature, saving reading time and improving information utilization efficiency. Therefore, the proposed model

F. Huang et al. (Eds.): ICAIE 2023, AHCS 15, pp. 410–417, 2024.
https://doi.org/10.2991/978-94-6463-242-2_51

for key information extraction from a collection of chemical engineering literature holds significant importance [2].

Text summarization has been evolving for over 60 years since its inception in the late 1950s. Despite being one of the major research directions in NLP with numerous application scenarios, progress in this field has been relatively slow. The emergence of deep learning has propelled the development of text summarization, and the introduction of BERT pre-training models has taken text summarization to new heights, making it a significant area of research in recent years.

2 Related Research

The core technology used for key information extraction from a collection of chemical engineering literature is text summarization, which can be broadly categorized into two main approaches: extractive summarization and abstractive summarization [3]. In the field of semantic mining of texts, various classical classification and clustering algorithms have been proposed. Initially, summary techniques relied on statistical-based methods using word frequency and sentence position [4]. However, over the past decade, with the rapid development of machine learning (ML) and natural language processing (NLP) [5], many accurate and efficient text summarization algorithms have been introduced. Literature reading is a necessity for researchers, but it often consumes a significant amount of time. To address this time-consuming and inefficient reading process, text summarization plays a critical role. By condensing large volumes of text, it allows users to quickly obtain key information, saving time and enhancing the speed of accessing literature content.

Abstractive summarization is a method that utilizes natural language processing algorithms to generate summaries by rephrasing and replacing sentences from the original text, without using any existing sentences or phrases [6]. With the rapid development of deep learning in recent years, an increasing number of deep learning methods have been employed in text summarization. Cho et al. and Sutskever et al. first proposed the seq2seq model composed of an encoder and decoder. Tan et al. introduced a graph-based attention mechanism neural model, which achieved good results in text summarization tasks. Siddiqui et al. [7]. Improved upon the sequence-to-sequence model proposed by the Google Brain team by using local attention mechanisms instead of global attention mechanisms, effectively addressing the issue of generating duplicate information. Celikyilmaz et al. proposed a deep communication agent algorithm based on the encoder-decoder architecture to generate summaries for long documents. Khan et al. [8]. Presented a framework based on semantic role labeling, using deep learning methods to accomplish multi-document summarization from a semantic role understanding perspective.

The BERT (Bidirectional Encoder Representations from Transformers) model was introduced by Google in 2018. It utilizes a bidirectional Transformer network with stronger semantic representation capabilities to train language models. BERT is a universal "language understanding" model that is pre-trained on a large corpus and represents the first unsupervised, deep bidirectional system used in pre-training NLP tasks. With just an additional output layer and fine-tuning of the pre-trained BERT, it can be adapted to various tasks without modifying the model specifically for each task. BERT

has achieved breakthrough progress in tasks such as sentence relationship judgment, extractive tasks (e.g., SQuAD), sequence labeling tasks (e.g., named entity recognition), and classification tasks (e.g., SWAG).

3 Construction of BERT Model

3.1 Description of Bert Model

Language model is an important concept in the field of natural language processing. After describing the objective facts with language model, we can get the language representation that can be processed by computer. In recent years, researchers have adopted pre trained neural networks as language models, and have achieved good results by fine-tuning and processing vertical tasks based on this language model. A typical language model calculates the probability of the next word from left to right, as shown in formula (1):

$$\mathrm{p}(S) = p(w_1 w_2, ..., w_m) = \prod_{i=1}^{m} p(w_i | w_1 w_2, ..., w_m) \tag{1}$$

The word vector obtained through the traditional neural network language model is single and fixed, and there are problems such as the inability to represent the ambiguity of words. The pre-trained language model solves this problem well, and can represent words in combination with their contextual content. This BERT model uses a bidirectional Transformer as an encoder for feature extraction and is combined with the Seq2Seq model, which can obtain more context information and greatly improve the ability of the language model to extract features. The Transformer encoding unit consists of two parts: the self-attention mechanism and the feedforward neural network. The input part of the self-attention mechanism is composed of three different vectors from the same word, respectively Query vector (Q), Key vector (K) and Value vector (V). The similarity between the input word vectors is expressed by multiplying the Query vector and the Key vector, which is recorded as QKT, and scaled by dk to ensure that the obtained result is moderate in size. Finally, the normalization operation is performed through softmax to obtain the probability distribution, and then the weight sum representation of all word vectors in the sentence is obtained. The word vector obtained in this way combines the context information, and the representation is more accurate. The calculation formula is as follows:

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

Encode the input text into BERT's hidden representations and feed these representations into a Seq2Seq model. In the Seq2Seq model, the hidden representation of BERT is used as the input to the encoder. An encoder encodes an input text sequence into a fixed-length vector representation. Then, use the decoder to gradually generate the target digest sequence based on the output of the encoder and the generated digest sequence (Fig. 1).

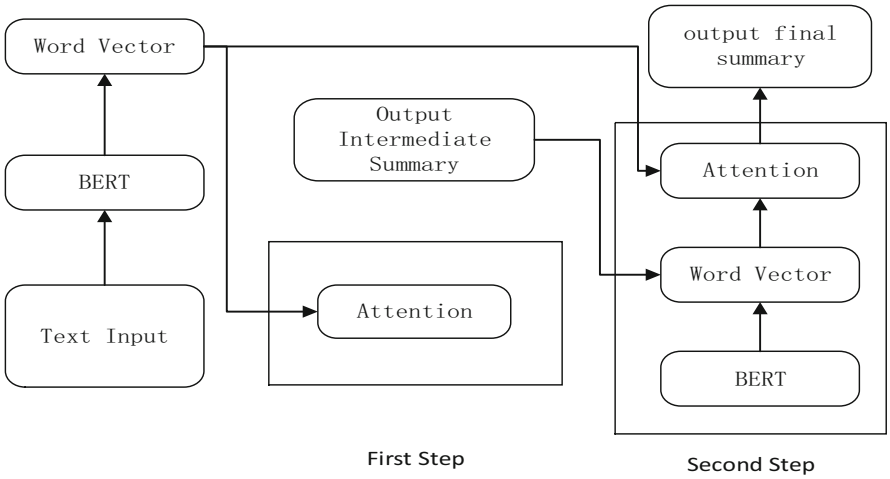

Fig. 1. Schematic diagram of model structure

3.2 Implementation Scheme Based on BERT Model

1) To obtain the corresponding text for the key content in the literature as input data for the model, processing the literature is an important task. First, it is necessary to confirm whether the literature contains content related to the keywords. If there is such content, Python programming tools can be used to extract the text content for further input to the model.
2) Selection of a specific dictionary. For chemical engineering literature, where there are numerous English references to compounds, these words carry relatively high weights. Therefore, it is important to include these unique compound references in the vocabulary of the model.
3) Import the pretrained BERT model and add the aforementioned specific dictionary to the vocab.txt file. Additionally, include attention layers in the model configuration (model_config) to handle the added vocabulary.
4) Model output, making full use of the language understanding ability of BERT and the sequence generation ability of the Seq2Seq model.
5) According to the output results, the model is fine-tuned to achieve the best performance of the model (Table 1).

3.3 Experimental Model Setup

The publicly available BERT model is pretrained on Chinese Wikipedia and general text data. To apply it to the task of text summarization, it is necessary to fine-tune the model specifically for this task [11]. It is found that removing the next sentence prediction task and masking consecutive segments yields better results. Additionally, the length of a typical summary sentence is not expected to exceed one-third of the original text length [12]. Therefore, in this study, the original text is concatenated with the summary sentences.

Table 1. Dictionary special identifier

serial number	identifier	effect
0	<unk>	unknown word
1	<s>	Sentence-initial identifier
2	</s>	Sentence-ending identifier
3	[PAD]	Padding identifier
serial number	identifier	effect
4	[CLS]	Sentence relationship identifier
5	[SEP]	Sentence separator identifier
6	[MASK]	Masking identifier

[CLS] Original Text [SEP] Summary Sentence

Continuing with the process, the summary sentence is masked as a whole to create the following data format for training and fine-tuning the BERT model for masked language prediction task.

[CLS]Original Text[SEP][MASK][MASK]…[MASK]

In the second stage, the fine-tuned BERT model is used to extract text features from the original text, which are then fed into the Transformer decoder for text summarization generation.

The encoder side adopts the BERT pre-trained language model architecture with the following specifications:

12 hidden layers, 768 hidden units in each layer, 12 attention heads, Dropout probability of 0.1 for both attention and hidden layers, Maximum positional encoding of 512.

The activation function for the hidden layers is the Gaussian Error Linear Units (GELU) [13], represented as:

$$GELU(X) = Xp(X \leq \mathrm{x}) = x\Phi(x) \tag{3}$$

In the equation, $\Phi(\mathrm{x})$ [14] represents the cumulative distribution function of the Gaussian normal distribution.

In the decoder, a 6-layer Transformer is used with a hidden size of 768, 12 attention heads, and a maximum target text length of 32.

The loss function is implemented using Label Smoothing, which is represented as:

$$\mathrm{L} = -(1-\varepsilon)\log(y\_t) - \frac{\varepsilon}{V} sum(\log(y\_i)) \tag{4}$$

Among them, ε is the smoothing parameter, which controls the degree of smoothing. y_t is the predicted probability of the model for the t-th word in the target sequence. Sum(log(y_i)) means taking the logarithm of the predicted probabilities of words in the entire vocabulary and summing them. Adjusting the loss function by introducing a smoothing parameter ε and a uniform distribution over the entire vocabulary (ε/V) makes the model more robust during training.

4 Experiment

4.1 Experimental Environment Description

Experiment with the python3.10 version of the Windows platform.

4.2 Text Acquisition

Since the model requires text as input while dealing with literature, the first step is to process the literature and extract the desired text content.

Python provides various libraries for parsing PDF files, such as PyPDF2, PDFMiner, PyMuPDF, ReportLab, and more. In this case, we primarily use the PyMuPDF library to extract the text corresponding to the key information in the literature. The advantage of PyMuPDF is that it preserves the original document structure and keeps the entire paragraph with line breaks intact as it appears in the PDF document.

4.3 Experimental Procedure

The obtained text mentioned above serves as the input to the model, which undergoes the encoding and decoding process. Although the Encoder-Decoder model is a classic approach, it has limitations in terms of requiring a fixed semantic vector length. In the Encoder-Decoder model, the encoder needs to compress the entire input sequence, which can introduce drawbacks such as the compressed semantic vector not fully representing the original information. Therefore, the BERT model incorporates the Attention mechanism. When generating the output, the model creates an attention scope to highlight the important parts of the input sequence and generates the next output based on these focused parts. Figure 2 provides a general illustration of the model.

After the text is input, after passing through the BERT model with attention (Attention), the summary content of the text will be output. The summary content is a simplification of the important content in the chemical literature. Table 2 is an example of the model output.

Compared to the original input text, the generated output consists of approximately one-fifth of the original text, yet it contains the essential information from the original text. This indicates that the model achieves a high level of accuracy. Therefore, it can be concluded that the BERT model developed in this study effectively extracts key information from chemical engineering literature.

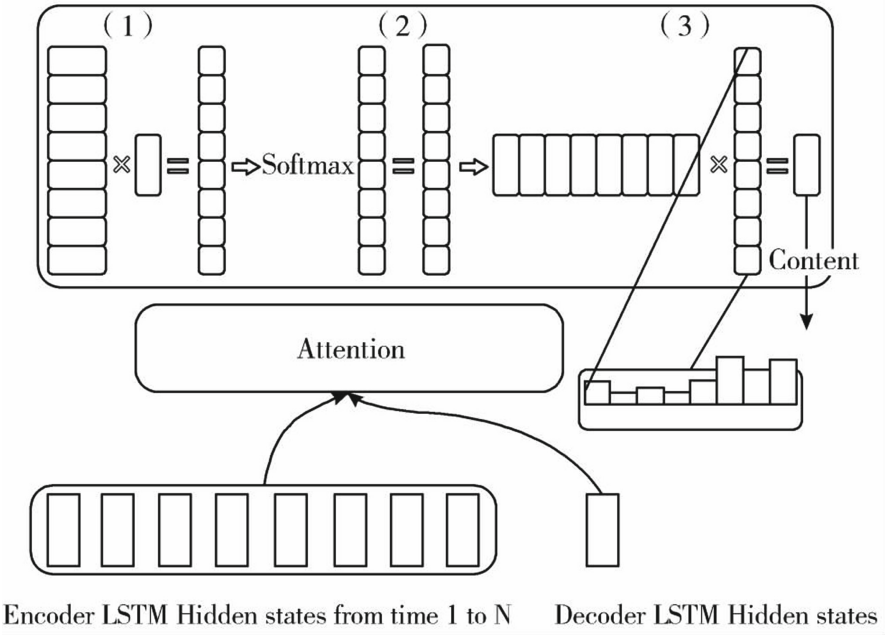

Fig. 2. Attention in BERT

Table 2. Example of the model output

PE and EVA were blended at the mass ratio of 1/1. The blend was selective solvent extraction of EVA. The weight loss after solvent extraction was used to calculate the continuity of EVA phase in the blend, revealing the continuity of EVA phase is about 85%

5 Conclusion and Prospect

This paper is based on the BERT pre-training model and applies natural language processing (NLP) to chemical engineering literature reading. It constructs a model for extracting key information from chemical engineering literature, which quickly generates a concise version of the essential information in the literature. This enables faster reading and learning of the literature, saving valuable time and significantly improving the efficiency of literature review. However, there are some limitations that need to be addressed. For example, the model currently does not incorporate image information from the literature. Important information in the literature can be contained in figures and tables. By combining text with visualized data in the form of graphs and tables, the understanding of the literature can be further enhanced. This is an area for improvement and optimization in future work. The paper will also include content related to graphs and tables to make the model more comprehensive, accurate, and user-friendly.

References

1. Du Xiuying. A Multi-Text Automatic Summarization Method Based on Clustering and Semantic Similarity Analysis [J]. Journal of Information, 2017, 36(06): 167-172.
2. Zhang Minghui. Multi-Document Automatic Summarization of Chinese and English Texts Based on Topic Models [D]. Suzhou University, 2011.
3. Wang Jiasong. Research on Multi-Document Automatic Summarization Based on Deep Learning [D]. Jilin University, 2017.
4. Cao Yang. Research on Single-Document Automatic Summarization Based on TextRank Algorithm [D]. Nanjing University, 2016.
5. Zhang C, Sah S, Nguyen T, et al Semantic Sentence Embeddings for Paraphrasing and Text Summarization [J]. 2018.
6. Zhao Meiling, Liu Shengquan, Liu Yan, GuoZhuwei, Fu Xianzhe. Research on Multi-Document Automatic Summarization Based on Improved K-means Clustering and Graph Model Combination [J]. Modern Computer (Professional Edition), 2017(17): 26-30.
7. Lan Z, Chen M, Goodman s, et al. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations[C]// International Conference on Learning Representations, 2019.
8. RUSH A M, CHOPRA S, WESTON J. A neural attention model for abstractive sentence summarization [J]. arXiv preprint arXiv:1509.00685, 2015.
9. NARAYAN S, COHEN S B, LAPATA M. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization [J]. arXiv preprint arXiv:1808.08745, 2018.
10. CAO Z, LI W, LI S, et al. Retrieve, rerank and rewrite: Soft template based neural summarization[CV/Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018: 152-161.
11. LINJ, SUN X, MA S, et al. Global encoding for abstractive summarization [J]. arXiv preprint arXiv:1805.03989, 2018.
12. MOROSHKOE, FEIGENBLAT G, ROITMAN H, et al. An editorial network for enhanced document summarization [J]. arXiv preprint arXiv:1902.10360, 2019.
13. KINGMA D P, BA J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.
14. SRIVASTAVA N, HINTON G, KRIZHEVSKY A, et al. Dropout: a simple way to prevent neural networks from overfitting [J]. The journal of machine learning research, 2014, 15(1): 1929-1958.