Journal of Computers Vol. 31 No. 2, 2020, pp. 241-249
doi:10.3966/199115992020043102020

A BERT Based Single Document Extractive Summarization Model

Wei Liu[1*], Pei-Ran Song[2], Rui-Li Jiao[2]

[1] School of Information and Communication Engineering, Beijing Information Science and Technology University, Beijing 100101, China
liuwei941213@163.com

[2] Department of Electronic Information Engineering, Beijing Information Science and Technology University, Beijing 100101, China
{peiransong, jiaoruili}@bistu.edu.cn

Received 20 January 2020; Revised 20 February 2020; Accepted 30 March 2020

Abstract. BERT, a pre-trained Transformer model, has already become one of the most common model in multiple natural language processing (NLP) tasks. It has been customized for extractive summarization via the fine-tuned BERTSUM model. Different from the other NLP tasks, extractive summarization relies heavily on the sentence position information at the document level. However, this crucial feature has not been fully studied in the existing models, either BERT or BERTSUM. In this paper, we propose a novel single document extractive summarization model, which incorporate the sentence positions through an extra documental position embedding module. The proposed model has been tested on the well-known CNN/DaliyMail dataset. Results show that the performance of our model is competitively against the state-of-the-art models on this task. Ablation experiments prove that the quality of the extracted summary can be improved by adding the documental sentence position embedding module.

Keywords: BERT, extractive summarization, sentence position embedding, single document

1 Introduction

Nowadays, people have to confront an enormous amount of textual materials on a daily basis. These documents could be news articles, web pages, blogs, status updates, etc. There is a great need to reduce the text data to shorter, focused summaries which capture the salient information. In order to assist us to have a more effective navigation as well as a quick check to filter out the non-relevant materials.

Single document summarization aims to automatically generating a shorter version of a given document while retaining its salient information. Methods for this task can be generally categorized as abstractive and extractive, based on their output type. Abstractive summarization generates entirely new phrases and sentences to capture the meaning of the source document. Classical methods operate by selecting and compressing content from the source document [1-5]. Although it is more closer to the approach ultimately used by humans, this approach is more challenging due to the requirement of the complex natural language understanding. Therefore, abstractive summarization methods are not yet state-of-the-art compared to extractive methods. Extractive summarization involves the selection of phrases and sentences from the source document to make up the new summary. Techniques involve ranking the relevance of phrases and sentences, in order to choose only those most relevant to the meaning of the source.

Traditional techniques for extractive summarization are based on statistical approaches [6-8], where sentences are ranked based on stemmed word frequencies, term frequency-inverted document frequency

[*] Corresponding Author

weights, etc. Recently, with the prosperity of machine learning, many methods start to tackle this task from the classification perspective. For example, decision tree models [9], graph based approaches [10], and integer linear programming methods [11]. In these approaches, extraction decisions are made based on primitive feature selections. As the features become more and more composite, deep learning methods start to enter the stage. Cheng and Lapata [12] proposed the first attention mechanism encoder-decoder model; Nallapati et al. [13] treated extractive summarization as a binary classification task and proposed a sequence model based on Recurrent Neural Networks (RNN); Zhou et al. [14] presented an end-to-end neural network framework for extractive document summarization by jointly learning to score and select sentences. Given the complexity of the task, these neural models have reached a bottleneck on the improvement of automatic metrics like ROUGE [15]. Thanks to the development of the language representation model Transformer [16] and its pre-training model BERT (**B**idirectional **E**ncoder **R**epresentations from **T**ransformers) [17], the performances of many NLP tasks have been greatly boosted. BERT model has been customized for extractive summarization through its variant BERTSUM (**BERT** architecture for **SUM**marizaiton) [18-19], which achieves the state-of-the-art outcomes on this task. In this model, inter-sentence Transformer layers have been structured on top of the BERT model to better express the semantics of the document and modify the sentence representations. However, Transformers do not encode the sequential nature of their inputs, and the Position Embeddings layer in BERT is not able to provide sentence position information at the document level. Therefore, the BERTSUM model does not take advantage of the sentence position information, which is the important auxiliary information for summary extraction.

In this paper, we propose a novel single document extractive summarization model, which incorporate the sentence positions via an extra documental learned positional embedding module. For better utilization of the sentence position information, in order to obtain more effective sentence representations in documents, hence render more efficient extractive summarizations.

The rest of the paper is organized as follows. Section 2 introduces the extractive summarization problem along with its mathematical problem formulation. Section 3 presents our learned positional embedding module and the structure of the proposed extractive summarization model. Ablation tests and comparison experiments with the state-of-the-art models are conducted in Section 4. Finally, Section 5 draws some conclusions.

2 Problem Description

Let D denote a document containing n sentences $\{s_1,s_2,\cdots,s_n\}$, where s_i represents the textual sequence of the i-th sentence in the document. Assume that the summary sentences represent the salient content of the document. Then the single document extractive summarization task can be defined as a label assignment problem, where each sentence s_i has a corresponding label $y_i \in \{0,1\}$, indicating whether the sentence should be included in the summary. Therefore, the problem can be deduced into two folds: first, build proper representations for sentences, $x \triangleq [x_1,x_2,\cdots,x_n]$, where x_i is a vector representation of the i-th sentence; second, apply a binary classifier over the representations to predict whether the label y_i should equals 1 (i.e. the i-th sentence should be included in the summary) or 0 (i.e. the i-th sentence should not be included).

Given gold labels $\{y_1,y_2,\cdots,y_n\}$ and predicted scores $\{r_1,r_2,\cdots,r_n\}$, with $r_i \triangleq f(x_i,W)$ being a function of sentence representation x_i and system parameter W, the mathematical problem formulation can be described as follow:

$$\underset{w}{\text{minimize}} \quad -\frac{1}{n}\sum_{i=1}^{n}\left(y_i \ln\left(r_i\right)+\left(1-y_i\right)\ln\left(1-r_i\right)\right), \tag{1}$$

where, the objective function of problem (1) is the average cross-entropy loss.

3 Extractive Summarization Models

The original architecture of BERT is shown in Fig. 1. Input text sequence is preprocessed by inserting special tokens [CLS] and [SEP]. Token [CLS] is attached to the beginning of the sequence, the output representation of this token aggregates information of the whole sequence. [SEP] is inserted after each sentence to indicate sentence boundaries. The modified text is then represented as a sequence of tokens, where each token is a superposition of three kinds of embeddings, namely, token embeddings, segment embeddings, and position embeddings. These three embeddings correspond to encode the token meaning, sentence-pair discrimination, and the token position within the sequence, respectively. The summed embedding vector is fed to a multi-layer bidirectional Transformer, to obtain an output vector for each token with contextual information.

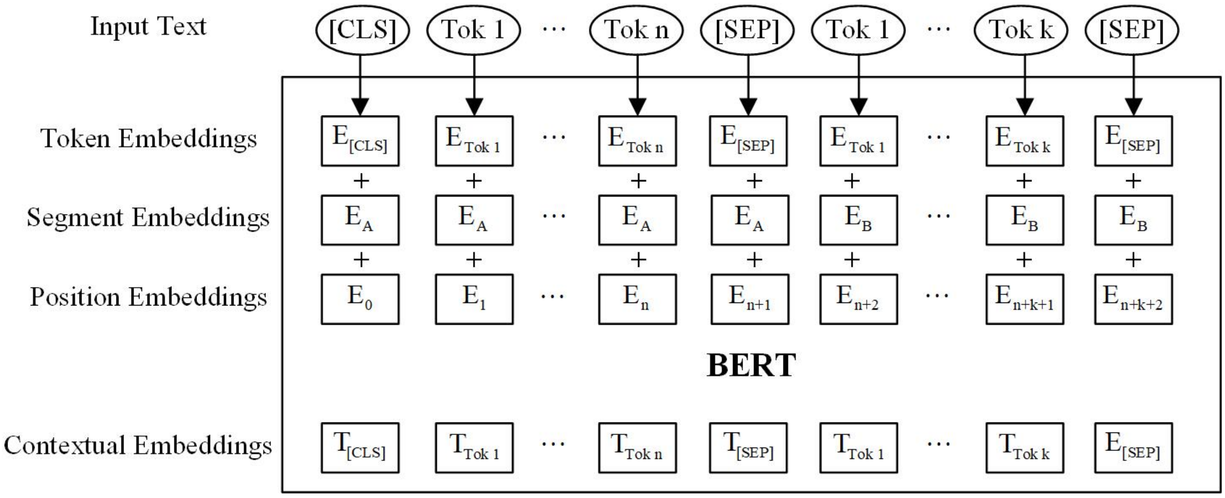

Fig. 1. Architecture of the original BERT model. Token [CLS] is appended to the beginning of the sequence, and token [SEP] is inserted after each sentence as an indicator of sentence boundaries. Every token, in the sequence, is a superposition of token, segment, and position embeddings. BERT will generate an output vector for each token with contextual information

3.1 BERTSUM Model

Inspired by the sequence preprocessing technique in BERT, BERTSUM [18-19] further modified the input sequence and embeddings by inserting a [CLS] token before each sentence. Hence enable the model to encode multiple sentences and try to get features of sentences by using the [CLS] symbols. The structure of the BERTSUM model is shown in Fig. 2. After obtaining the sentence vectors from BERT, several Transformer layers have been stacked on top of the BERT outputs to improve the sentence representations. For each sentence s_i, $i=1,\cdots,n$, the score r_i is predicted for calculating the binary classification entropy against the gold label y_i.

3.2 Sentence Position Embeddings

Different from the RNN based sequence models, the self-attention layer of a Transformer causes identical words at different positions to have the same output representation. Therefore, positional embeddings have to be introduced for recovering position information. In this subsection, two versions of positional embeddings will be discussed along with their applications in extractive summarization models. **Sinusoidal positional embeddings**. Sinusoidal positional embeddings generate relative position information using sine and cosine functions, for example [16]:

$$\begin{cases} PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{\text{model}}}\right) \\ PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{\text{model}}}\right) \end{cases}, \tag{2}$$

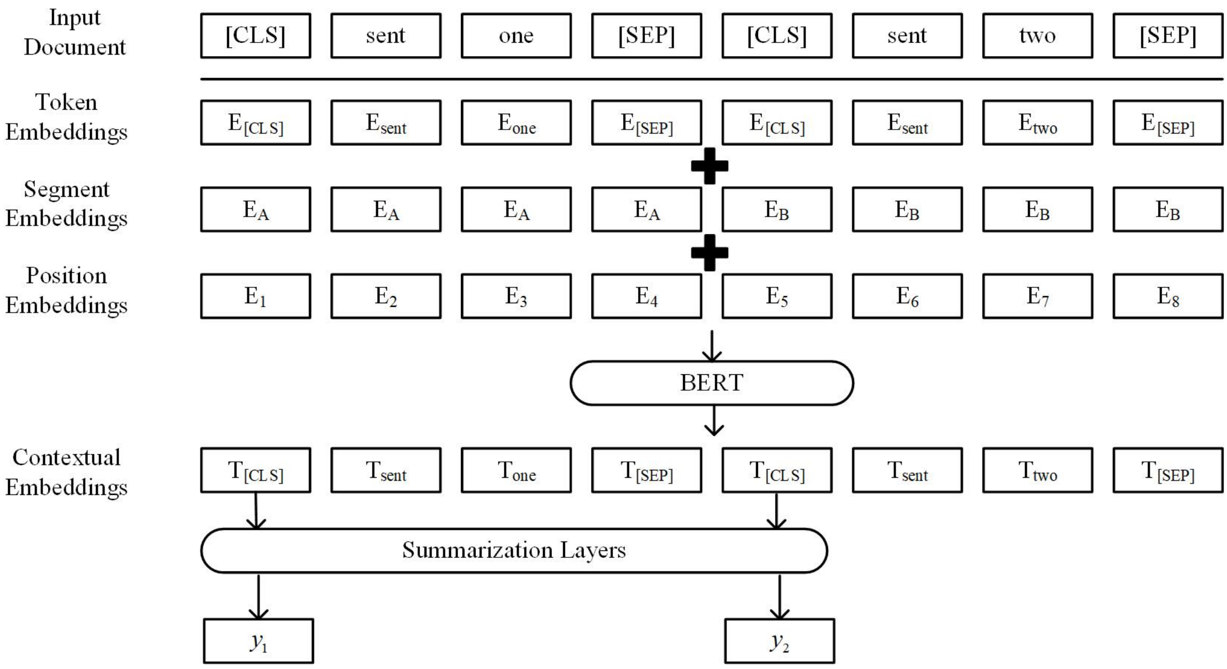

Fig. 2. Structure of BERTSUM model. Token [CLS] is appended to the beginning of every sentence. Features of sentences can be obtained by using the [CLS] representations. Sentences scores are predicted for calculating the binary classification entropy against the gold label

where d_{model} is the model output dimension, pos denotes the position, i is the number of sinusoids, i.e. each sinusoid corresponds to a positional encoding dimension. By using eq. (2), it would allow the model to learn the relative positions of the tokens in a given sequence.

Learned positional embeddings. Another encoding method of position information, embed the absolute position index with learnable parameters. Given a randomly initialized vector of the tokens at each position, training is performed on the data to obtain the position information of each token. Learned positional embeddings allow the model to know which portion of the input sequence is currently being processed, but also imposes a restriction on the maximum input sequence length.

BERT model utilized the latter positional embedding method, it incorporated the sequential nature of the input sequences by learning a vector representation for each position. It was designed to process input sequences of up to length 512 tokens [17]. This means that the Position Embeddings layer is a lookup table of size (512, d_{model}), where the first row is the vector representation of any word in the first position, the second row is the vector representation of any word in the second position, etc. Which infers that the vanilla BERT model may not suit for tasks dealing with long text sequences, such as documents. Unless, a structural modification has been launched. Thus, BERT model by itself cannot provide documental sentence representations.

BERTSUM, a variant of the BERT model, enabling BERT on the extractive summarization task by inserting extra token to obtain each sentence representation of a document. It then stacks multiple Transformer layers to acquire a document level sentence representations. The authors chose sinusoidal position embeddings for the documental sentence position information. As discussed above, the sinusoidal embedding method only involves the relative positions of elements. However, at extractive summarization regime, the most effective position information is the sentence index. For instance, people usually using the first or the last sentence of a document as a summary. Consequently, there is still room for improvement of automatic metrics like ROUGE.

3.3 Proposed Model

In this subsection, we introduce our BERT based extractive summarization model, which incorporates the advantages of both the BERT and BERTSUM models. The architecture of the proposed model has been shown in Fig. 3. As suggested by BERTSUM, to represent sentences separately, a pair of extra token [CLS] and [SEP] have been appended at the two ends of each sentence. Following the notations in Section 2, after the preprocessing, document D becomes

$$\tilde{D} = \{[[\mathrm{CLS}], s_1, [\mathrm{SEP}]], [[\mathrm{CLS}], s_2, [\mathrm{SEP}]], \cdots, [[\mathrm{CLS}], s_n, [\mathrm{SEP}]]\} . \quad \textbf{(3)}$$

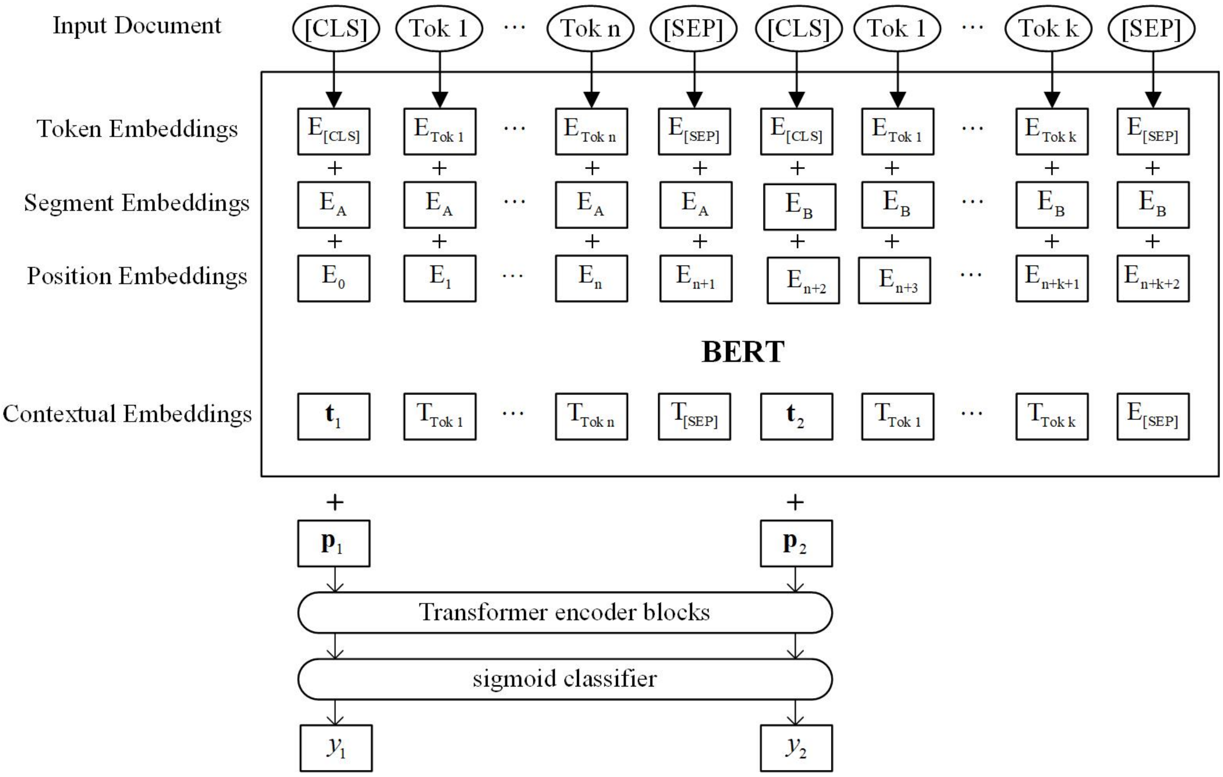

Fig. 3. Architecture of the proposed model. Token [CLS] and [SEP] are appended at the two ends of each sentence. t_i can be considered as the sentence representation of s_i, p_i is the corresponding learned position embedding at the document level

The preprocessed text sequences in $\tilde{D}$ are concatenated and fed into the BERT model as the input document. Due to the sequences concatenation, the input text is deliberately lengthened, which challenges the distinguishability of the position embeddings in BERT model. To overcome the position embedding limitation, we add more randomly initialized position embeddings and let them fine-tuned with other system parameters.

Let $t \triangleq [t_1, t_2, \cdots, t_n]$ denotes the output representations of BERT, where t_i is the corresponding vector of the i-th [CLS] token. It can be viewed as the representation of the i-th sentence s_i. Along with the documental sentence position embeddings, $p \triangleq [p_1, p_2, \cdots, p_n]$, the input vectors of the L-layered inter-sentence Transformer can be derived as $x^0 \triangleq t + p$, where p_i embeds the absolute position of s_i. Therefore,

$$x^l = \mathrm{Transformer}(x^{l-1}), \ \ l = 1, \cdots, L , \quad \textbf{(4)}$$

where $\mathrm{Transformer}(\bullet)$ represents the model structure introduced by Vaswani et al. [16]. Thus, according to the problem description in Section 2, the predicted sentence scores are calculated as follow:

$$r_i = f(x_i^L, W), \ i = 1, \cdots, n . \quad \textbf{(5)}$$

If $f(\bullet)$ happened to be the sigmoid function, then problem (1) reduces to the logistic regression problem.

4 Experiments

In this section, we present the implementation dataset and the evaluation protocols for testing and analyzing our model. Ablation tests and comparison experiments with the start-of-the-art models are conducted successively.

4.1 Summarization Dataset

The proposed model has been evaluated on the CNN/DailyMail news highlights dataset, which is the most famous benchmark dataset for summarization tasks. We follow the standard split of Hermann et al. [20] for training, validating, and testing (90,266/1,220/1,093 CNN documents and 196,961/12,148/10,397 DailyMail documents). There is no anonymous entities or lowercase tokens. Trail the previous research hypothesis in [19] that "story highlights" associated with each document are referred as gold-standard reference summaries. In order to conduct a fair comparison, we utilize the same sentence splitting and data preprocessing method as in [18]. Specifically, the sentences are split by CoreNLP and the dataset is preprocessed by following the method in [21].

4.2 Evaluation and Average Methods

Manual and semi-automatic evaluations of large-scale summarization models is costly and cumbersome. The ROUGE package [15] offers a set of automatic metrics based on the lexical over-lap between candidate and reference summaries. Overlap can be computed between consecutive(n-grams) and non-consecutive (skip-grams) sub-sequences of tokens.

Evaluation criteria. In the following experiments, ROUGE scores is computed to evaluate the quality of the extracted summaries. Unigram and bigram overlap (i.e. ROUGE-1 and ROUGE-2 scores) are used as a means to evaluate informativeness of the summaries, and the longest common subsequence (ROUGE-L) is used as a method to assess the textual fluency. In order to jointly evaluate the performance of the models in terms of both precision and recall, the harmonic mean of precision and recall (i.e. F_1 score) for each ROUGE score has been considered as the evaluation criterion.

Automatic label method. As suggested in Section 2, our model eventually will have to solve a supervised label assignment problem. However the implementing dataset only contains the abstractive gold summaries, which are not readily suite for training the extractive summarization models. To deal with this problem, a greedy algorithm was used to generate the so-called oracle summaries for each document. Where the algorithm greedily select sentences that maximize the ROUGE scores to become one of the oracle sentences. The sentences that have been included in the oracle summaries are labeled as 1, and 0 otherwise. The acquired sentence labels play the role of gold labels in the model training procedure.

Average methods. Due to the statistical properties, neither one of the individual results or the checkpoints saved through the training process can represent the quality of the models and their performances. Hence, we need averaged results to show the overall effectiveness of the summarization models. There are mainly two kinds of averaging methods. One is first evaluate the model with ROUGE and then average over the resulting ROUGE scores. The other is first average the model outputs and then evaluate by ROUGE. Although, both methods can provide general ideas about the performance of the extractive summarization models, we are more fans of the latter method. Since averaging in this way, can provide us a way to select more effective models by combination.

4.3 Experimental Results

We use the "bert-base-uncased"[1] version of BERT to implement the model, the BERT and Transformer layers are jointly fine-tuned. The experimental results on CNN/Dailymail dataset are shown in Table 1, where we compare our model with several previously proposed systems. To ensure a fair comparison, we implement our proposed method based on exactly the same setups as in [19].

[1] https://github.com/huggingface/transformers

Table 1. Test set results on the CNN/DailyMail dataset using ROUGE F_1 scores. Results with * mark are taken from the corresponding papers. All implemented results are truncated to two decimal places to have a better comparison. The Models are sorted in descending order according to the ROUGE-1 results

Model	ROUGE-1	ROUGE-2	ROUGE-L
ORACLE	52.85	28.43	45.43
Proposed	**43.38**	**20.37**	**39.76**
BERTSUM*	43.25	20.24	39.63
NEUSUM*	41.59	19.01	37.98
REFRESH*	40.00	18.20	36.60
SummaRunner*	39.60	16.20	35.30
LEAD-3	37.33	16.30	31.55

ORACLE. We use the name "ORACLE" to indicate the enumeration and exhaustive search based methods [22]. They try to find the most possible best extractive summaries, at the cost of enormous computing time. The results from these methods can serve as upper bounds for the other extractive summarization approaches. In this experiment, we first sort the sentences of each document in descending orders according to their ROUGE scores, which are calculated with respect to the gold-standard reference summaries. Exhaustive search has been carried on the possible 3-sentence extracts among the top 10 high-scoring sentences of each document.

BERTSUM. As introduced in Section 3.1, BERTSUM model [18-19] is the state-of-the-art for extractive summarization. It is a fine-tuned BERT variant.

NEUSUM. It is a neural network framework for extracting document summaries by jointly learning scoring and selecting sentences [14]. The document sentence is first read using a hierarchical encoder to obtain a representation of the sentence. Then extract the sentences one by one to build the output summary.

REFRESH. REFRESH is proposed by Narayan et al. [23], it is an extractive summarization model trained by globally optimizing the ROUGE metric with reinforcement learning.

SummaRunner. SummaRuNNer [13] is a neural sequence model based on RNN for extracting document summarization. The model has the characteristics of strong interpretability, and is trained through a novel abstract mechanism to eliminate the need for extractive labels during training.

LEAD-3. It is an extractive baseline which uses the first-3 sentences of the document as a summary.

As shown in Table 1, the BERT-based models, our proposed model and BERTSUM, outperformed the others by a large margin. Results our model is comparable and slightly better than the state-of-the-art for all three ROUGE metrics, namely ROUGE-1, ROUGE-2, and ROUGE-L.

Ablation Tests. Ablation studies are conducted to show the contribution of sentence position embeddings in our model. The results are shown in Table 2, where two variants of the proposed model are involved in the comparison. "- Sentence Position Embeddings" represents the case where the inter-sentence position embeddings are removed. "w/ Sinusoidal Positional Embeddings" means the case where we replace the learned positional embeddings with the sinusoidal ones.

Table 2. Results of ablation studies of sentence position information on CNN/Dailymail test set using ROUGE F_1

Model	ROUGE-1	ROUGE-2	ROUGE-L
Proposed Model	43.386	20.372	39.768
w/ Sinusoidal Positional Embeddings	43.371	20.315	39.754
- Sentence Position Embeddings	43.369	20.307	39.759

The results in Table 2 indicating that the inter-sentence position embedding module can help to improve the qualities of the extracted summaries in term of the ROUGE metrics, with only a negligible extra overhead. In extractive summarization tasks, the learned positional embeddings outperform the sinusoidal positional embeddings.

5 Conclusions

In this paper, we propose a single document extractive summarization model, which incorporate the sentence positions through an extra documental position embedding module. Experiments on CNN/DaliyMail dataset show that our model can compete against the state-of-the-art models. Ablation experiments prove that the quality of the extracted summary can be improved by adding the documental sentence position embedding module.

References

[1] A.M. Rush, S. Chopra, J. Weston, A neural attention model for abstractive sentence summarization, in: Proc. EMNLP Conference in Lisbon, 2015.

[2] J. Gu, Z. Lu, H. Li, V.O.K. Li, Incorporating copying mechanism in sequence-to-sequence learning, in: Proc. ACL Conference of the 54th Annual Meeting, 2016.

[3] A. See, P.J. Liu, C.D. Manning, Get to the point: summarization with pointer generator networks, in: Proc. ACL Conference of the 55th Annual Meeting, 2017.

[4] R. Paulus, C. Xiong, R. Socher, A deep reinforced model for abstractive summarization. <https://arxiv.org/abs/1705.04304>, 2017.

[5] S. Narayan, S.B. Cohen, M. Lapata, Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization, in: Proc. EMNLP Conference in Brussels, 2018.

[6] H.P. Luhn, The automatic creation of literature abstracts, IBM Journal of Research and Development (1958) 159-165.

[7] P.E. Baxendale, Machine-made index for technical literature--an experiment, IBM Journal of Research and Development (1958) 354-361.

[8] H.P. Edmundson, New methods in automatic extracting, Journal of the Association for Computing Machinery (1969) 264-285.

[9] Y. Liu, I. Titov, M. Lapata, Single document summarization as tree induction, in: Proc. ACL Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2019.

[10] G. Erkan, D.R. Radev, Lexpagerank: prestige in multi-document text summarization, in: Proc. EMNLP Conference, A Meeting of Sigdat, 2004.

[11] K. Woodsend, M. Lapata, Automatic generation of story highlights, in Proc. ACL Conference of the 48th Annual Meeting, 2010.

[12] J. Cheng, M. Lapata, Neural summarization by extracting sentences and words, in: Proc. ACL Conference of the 54th Annual Meeting, 2016.

[13] R. Nallapati, F. Zhai, B. Zhou, Summarunner: a recurrent neural network based sequence model for extractive summarization of documents, in: Proc. AAAI Conference of the 31st Annual Meeting, 2017.

[14] Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, T. Zhao, Neural document summarization by jointly learning to score and select sentences, in: Proc. ACL Conference of the 56th Annual Meeting, 2018.

[15] C.Y. Lin, ROUGE: a package for automatic evaluation of summaries, in: Proc. the Workshop on Text Summarization Branches Out, 2004.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need. <https://arxiv.org/abs/1706.03762>, 2017.

[17] J. Devlin, M. Chang, M. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proc. NAACL Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2019.

[18] Y. Liu, Fine-tune BERT for extractive summarization. <https://arxiv.org/abs/1903.10318>, 2019.

[19] Y. Liu, M. Lapata, Text summarization with pretrained encoders, in: Proc. EMNLP Conference & International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019.

[20] K.M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, P. Blunsom, Teaching machines to read and comprehend, in: Proc. Advances in Neural Information Processing Systems, 2015.

[21] A. See, P.J. Liu, C.D. Manning, Get to the point: summarization with pointer-generator networks, in: Proc. ACL Conference of the 55th Annual Meeting of the Association for Computational Linguistics, 2017.

[22] T. Hirao, M. Nishino, J. Suzuki, M. Nagata, Enumeration of extractive oracle summaries. in: Proc. 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 386-396.

[23] B. Narayan, S.B Cohen, M. Lapata, Ranking sentences for extractive summarization with reinforcement learning, in: Proc. 2018 NAACL, 2018.