T-BERTSum: Topic-Aware Text Summarization Based on BERT

Tinghuai Ma, *Member, IEEE*, Qian Pan, Huan Rong, Yurong Qian, Yuan Tian, and Najla Al-Nabhan

***Abstract*— In the era of social networks, the rapid growth of data mining in information retrieval and natural language processing makes automatic text summarization necessary. Currently, pretrained word embedding and sequence to sequence models can be effectively adapted in social network summarization to extract significant information with strong encoding capability. However, how to tackle the long text dependence and utilize the latent topic mapping has become an increasingly crucial challenge for these models. In this article, we propose a topic-aware extractive and abstractive summarization model named T-BERTSum, based on Bidirectional Encoder Representations from Transformers (BERTs). This is an improvement over previous models, in which the proposed approach can simultaneously infer topics and generate summarization from social texts. First, the encoded latent topic representation, through the neural topic model (NTM), is matched with the embedded representation of BERT, to guide the generation with the topic. Second, the long-term dependencies are learned through the transformer network to jointly explore topic inference and text summarization in an end-to-end manner. Third, the long short-term memory (LSTM) network layers are stacked on the extractive model to capture sequence timing information, and the effective information is further filtered on the abstractive model through a gated network. In addition, a two-stage extractive–abstractive model is constructed to share the information. Compared with the previous work, the proposed model T-BERTSum focuses on pretrained external knowledge and topic mining to capture more accurate contextual representations. Experimental results on the CNN/Daily mail and XSum datasets demonstrate that our proposed model achieves new state-of-the-art results while generating consistent topics compared with the most advanced method.**

***Index Terms*— Bidirectional Encoder Representations from Transformers (BERTs), neural topic model (NTM), social network, text summarization.**

I. Introduction

AUTOMATIC text summarization is the process of compressing and extracting information effectively from input documents while still retaining its key content. This technique plays an important role in information retrieval and natural language processing (NLP) [1], which is a research hotspot in a multitude of fields, such as computer science, multimedia, and statistics [2]. Currently, typical summarization methods include extractive and abstractive [3]. The extractive method selects salient sentences or reorganizes sentences from the original text, and the abstractive method generates novel words or phrases with comprehension. Generally, most of the existing methods are designed to encode paragraphs and then decode with different mechanisms. Nevertheless, there is a large amount of information loss in the encoding and decoding stages. Therefore, existing works on summarization mainly focus on word embedding or contextual contents.

However, the advantages of word embedding are limited by specific small datasets, which require richer contextual information. The future direction of summarization is to predict words from full context by language modeling and representation learning [3]. Consequently, in this article, the challenging research problem is studied: how to use the pretrained language model for text representation and generation.

Unfortunately, it is an open challenge to generate sentences related to a topic with overall coherence and discourse-relatedness [4]. In order to well summarize the original text, we propose a topic-aware extractive and abstractive summarization model named T-BERTSum. The proposed model T-BERTSum has several challenges. First, the model is expected to obtain accurate and updatable topics corresponding to the specific article. Second, the model is necessary to match topic information with word embedding in an end-to-end manner to guide the generation of topic-aware summarization. Third, the overall framework of the extractive model and the abstractive model needs to summarize as much information as possible with low redundancy.

To overcome these challenges, the text is first represented as word embedding by the most advanced pretrained language model, Bidirectional Encoder Representations from Transformers (BERTs) [5], greatly guarantees the semantics of the context. Inspired by the work [6], the topic distribution is trained by NTM [7] in the neural variational inference framework. Technically, global information is represented through the integration of powerful token embedding and potential topic embedding, which contains quite relevant quantity of information. Then, the topic-aware sequence is put into the decoder with the transformer to learn soft alignments between summaries and source documents. T-BERTSum extends the successful transformer for text encoding and decoding [5] based on the attention network. Finally, the long short-term

Manuscript received August 11, 2020; revised February 28, 2021; accepted April 8, 2021. Date of publication June 24, 2021; date of current version May 30, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFE0104400 and in part by the Deanship of Scientific Research at King Saud University under Grant RGP-1441-33. *(Corresponding author: Tinghuai Ma.)*

Tinghuai Ma and Qian Pan are with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: thma@nuist.edu.cn).

Huan Rong is with the School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: hrong@nuist.edu.cn).

Yurong Qian is with the School of Software, Xinjiang University, Urumqi 830008, China.

Yuan Tian is with the School of Computer, Nanjing Institute of Technology, Nanjing 211167, China.

Najla Al-Nabhan is with the Department of Computer Science, King Saud University, Riyadh 11362, Saudi Arabia.

Digital Object Identifier 10.1109/TCSS.2021.3088506

2329-924X © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Sathyabama Institute of Science and Technology. Downloaded on December 20,2024 at 20:57:20 UTC from IEEE Xplore. Restrictions apply.

memory (LSTM) layers are stacked on top of the output layer to classify whether the sentence belongs to summaries for the extractive model. Besides, a gated network is added for the abstractive model to remove the useless information. As a consequence, the dependencies of the sentence with a relatively long span can be captured. The T-BERTSum adopts joint learning of topic modeling and text summarization and separates the optimizer of the encoder and the decoder to accommodate the fact that the former is pretrained and the latter must be trained from scratch. In addition, a two-stage summarization framework is constructed, based on T-BERTSum, to generate summaries on the extracted sentences, sharing information while greatly reducing sentence redundancy. Generally, the main contributions of this work are given as follows.

1) The new proposed T-BERTSum that applies BERT in text summarization, which introduces rich semantic features, based on the modified transformer architecture that achieves efficient and paralleled computation.
2) The background information is considered to be integrated into the encoding as an additional knowledge, which is encoded to be an adjustable topic representation, aiming to guide the generation of summaries in an end-to-end manner.
3) The ability to generate smooth summaries with low redundancy can be improved by sharing information between different tasks in a two-stage extractive–abstractive model.

The rest of this article is organized as follows. In Section II, we review some related works that can be adapted for the text summarization task in this article. In Section III, we elaborate on our proposed framework T-BERTSum in detail followed by the experimental analysis in Section IV. According to the experimental results, we make conclusions in Section V.

II. Related Works

Current works on text summarization mainly focus on word embedding, which represents each element in someway [8], [9]. However, word embedding cannot completely solve the problem of polysemy. In order to alleviate this problem, Embeddings from Language Models (ELMOs) [10] adopted bidirectional LSTM to train the language model, where hierarchical LSTM can grasp different granularity information. The LSTM captures the word features, syntactic features, and semantic features, respectively, from the shallow to the deep, but the parallelism is poor. By contrast, the transformer [5] accelerates the deep learning training process based on the attention mechanism, greatly improves the feature extraction capability, and contributes to parallel processing. Moreover, generative pretraining (GPT) [11] obtained a better context representation by using a feature extractor with a unidirectional transformer. Furthermore, BERT [5] trains a corpus of 33 million words via masked language modeling and next sentence prediction, which has generated better word embedding. In recent years, BERT has been successfully applied to various NLP tasks, such as text implication, name entity recognition, and machine reading comprehension. RoBERTa [12] has made several adjustments on the basis of BERT, namely, more training data, larger batch size, longer training time, and removal of next predict loss. In our study, we focused on BERT to extract context information effectively for sequence encoding. We believe that the BERT-based model can achieve better performance.

Most importantly, one of the limitations of automatic summarization is how to reflect the implicit information conveyed between different texts and the background influence [13]. Akhtar *et al.* [14] used the latent Dirichlet allocation (LDA) to label the documents with topics and used formal concept analysis (FCA) to automatically organize in a lattice structure. In this way, topics' identification and documents' organization help better with text mining. Roul *et al.* [15] proposed a heuristic method that used the LDA technique to identify the optimum number of independent topics present in the corpus, ensured that all the important contents from the corpus of documents are captured in the extracted summary. In addition, the two-tiered topic model based on the pachinko allocation model (PAM) is combined with the TextRank method for summarization [16]. Word topic distribution of LDA combines the sequence-to-sequence model to improve abstractive sentence summarization [17]. Yang *et al.* [18] introduced a novel neighborhood preserving semantic (NPS) measure to capture the sparse candidate topics under that low-rank matrix factorization model. These techniques used the topic model, as an additional mechanism, to improve text generation. Nonetheless, these models have some sparseness problems and difficult to train. Our approach utilizes the neural topic model (NTM) to induce implicit topics in neural networks, which is easy to explain and expand. We have also demonstrated the influence of topics on text summarization.

From another aspect, text summarization is basically divided into extractive and abstractive models. Sadiq *et al.* [19] consider that the target user lacks background knowledge or reading ability and propose a linear combination of feature scores for social networks. NeuSum [20] integrated the selection strategy into the scoring model, which solves the problem of segmentation of sentence scoring and sentence selection previously and has able to end-to-end training without human intervention. ExDoS [21] is the first approach to combine both supervised and unsupervised algorithms in a single framework for document summarization; it iteratively minimizes the error rate of the classifier in each cluster with the help of dynamic local feature weighting. Wang *et al.* [22] used the seq2seq model of convolution and the strategy gradient algorithm to summarize and optimize the text. With the emergence of self-attention [23], increasing methods [24], [25] adapts self-attention instead of the RNN sequence model and uses multihead attention to capture different semantic information based on the fact that each element contributes differently to the sequence. Su *et al.* [26] propose a two-stage method for variable-length abstractive summarization; it consists of a text segmentation module and a two-stage transformer-based summarization module and has achieved good results in capturing the relationship between sentences.

In addition, some works [27], [28] have focused on summarizing via using pretraining language models recently.

Authorized licensed use limited to: Sathyabama Institute of Science and Technology. Downloaded on December 20,2024 at 20:57:20 UTC from IEEE Xplore. Restrictions apply.

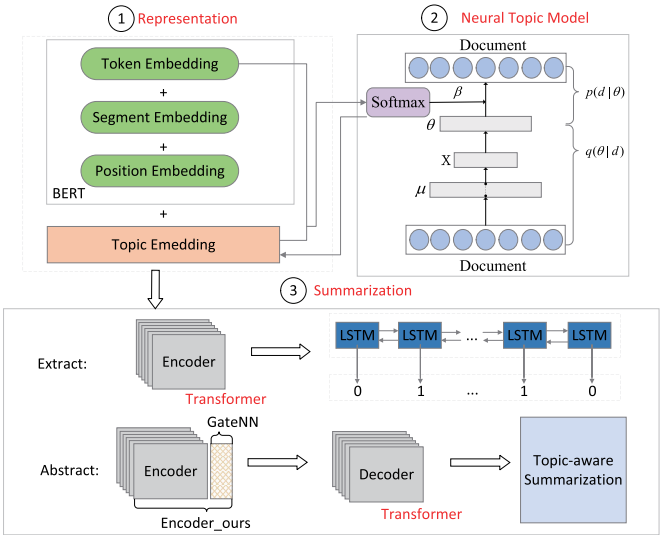


Fig. 1. Overall framework of our summarization model T-BERTSum.

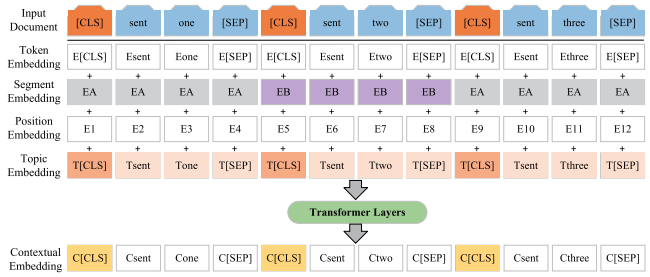


Fig. 2. Overview architecture of the T-BERTSum embedding. Differentiate sentences and tags with different color markers.

These models [29], [30] first applied BERT to fine-tuning on text summarization. Wang *et al.* [31] used BERT word embedding as input and integrated the extractive network and the generation network into a unified model by reinforcement learning. Srikanth *et al.* [32] use the existing BERT model to produce extractive summarization by clustering the embeddings of sentences by K-means clustering and introduce a dynamic method to decide the suitable number of sentences to pick from clusters. Different from the previous work that only uses BERT on text generation, our goal is to match the word embedding to the topic information and comprehensively and accurately express the context information of each sequence. To the best of our knowledge, this is still a relatively unexplored area, and the proposed method can also be applied to other NLP tasks. In conclusion, inspired by the above works, in this article, we intend to adopt the embedding from BERT, combine with the topic representation, and guide the summaries' generation in powerful transformer architecture.

III. Proposed Model T-BERTSum

In this section, we describe the general structure of extractive and abstractive, which generates multisentences' summaries from a given source document. The overall structure of our model is shown in Fig. 1. There are three major components (from left to right).

1) *Representation:* It adds the matching of token embedding, segment embedding, position embedding, and topic embedding of the text with BERT to fully express the semantic features of the sequence (as described in Section III-A).
2) An NTM to induce latent topics (as described in Section III-B).
3) *Summarization* It consists of an extractive mode and an abstractive mode.

The former utilizes LSTM for classification to select crucial sentences, while the latter generates sentences by stacking gated networks and transformer layers (as described in Section III-C). These three components can be updated simultaneously by joint learning. For the problem of training mismatch, we fine-tune the optimizer to train the three parts. This process will be introduced in Section III-D.

A. Representation

Formally, the input text is first preprocessed by inserting two special tokens. The token [CLS] is inserted into the beginning of each sentence, and the output calculated by this token is used to integrate the information in each sequence, while the token [SEP] is inserted into the ending of each sentence, as an indicator of sentence boundaries. The preprocessed text is then represented as a sequence of token $X = \{w_1, w_2, \ldots, w_n\}$. Each instance x is processed into representation: bag-of-words (BoW) term vector $x_{\text{BOW}} \in R^V$, where V is the vocabulary size.

The modification of the input embedding forms the preferable sequence. As illustrated in Fig. 2, for each input sentence, the output is the summation of four types of embedding: token embedding, segment embedding, position embedding, and topic embedding. Among them, the topic feature embedding is newly introduced to capture the underlying topic information, and the other three embeddings are the designs of the original BERT. A bidirectional transformer with multiple layers dealing with these characteristics from the output to generate the contextual embedding for each token is given as follows:

$$\tilde{h}^l = \text{LN}(h^{l-1} + \text{MHAtt}(h^{l-1})) \quad (1)$$

$$h^l = \text{LN}(\tilde{h}^l + \text{FFN}(\tilde{h}^l)) \quad (2)$$

where $h^0 = x$ are the output vectors, LN is the layer normalization operation [33], MHAtt is the multihead attention operation [23], and FFN is a two-layer feedforward network operation. Superscript l indicates the depth of the stacked layer.

In T-BERTSum, the symbol [CLS] is used to aggregate the features of a sequence, while the sequence is ended with a special ending element [SEP]. Token embedding represents the embedding vector of each word. Segment embedding is used to distinguish sentences, and we follow Liu and Lapata [30] to embed sentence to represent odd or even sentences by embedding E_A or E_B and achieve the purpose of learning adjacent sentence features or paragraph sentence features at different layers. Position embedding learns embedding at each position to represent the sequence order information.

Authorized licensed use limited to: Sathyabama Institute of Science and Technology. Downloaded on December 20,2024 at 20:57:20 UTC from IEEE Xplore. Restrictions apply.

Topic embedding is appended to each input sequence, which is the output representation of topic information in the implicit sequence trained by NTM described in Section III-B. The main contribution of topic embedding is to represent the topic information hidden in each word or sequence that can mine the gist of this article and solve the problem of polysemy. As an example, the word "*novel*" can be understood as fiction or new in different contexts. Therefore, it is necessary to dig out and represent the background information of each word.

B. Neural Topic Model

Our topic model is inspired by NTM that induces latent topics in neural networks. We assume topic vector $t \in R^{K\times H}$, where K is the number of topics and H is the dimension of the embedding. The token embedding e indicates the meaning of each token. The topic distribution over words for a given topic assignment z_n is $p(w_n|\beta, z_n) = \text{Multi}(\beta_{z_n})$, where Multi is the multinomial topic distribution, which is generated by computing token embedding and topic embedding as follows:

$$\beta_K = \text{soft}\max\left(e \cdot t_K^T\right). \tag{3}$$

We assume that β is achieved by calculating the semantic similarity between topics and words. Here, the prior parameters of μ and σ in Fig. 1 are defined as $G(\theta|\mu_0, \sigma_0^2)$ in which the Gaussian sample $x \sim N(x|\mu_0, \sigma_0^2)$, $N \sim (\mu|\sigma_0)$, represents the Gaussian distributions, where μ_0 and σ_0^2 are hyperparameters that set for a zero mean and unit variance Gaussian. We use Gaussian softmax to generate $\theta = g(x)$ and variational inference [34] to approximate a posterior distribution $q(\theta|d)$ over x. The loss function of topic model is defined as

$$L_{\text{NTM}} = E_{q(\theta|d)}\left[\sum_{n=1}^{N}\log\sum_{z_n}\left[p(w_n|\beta_{z_n})p(z_n|\theta)\right]\right] - D_{\text{KL}}\left[q(\theta|d)||p\left(\theta|\mu_0, \sigma_0^2\right)\right] \tag{4}$$

where $q(\theta|d)$ is the variational distribution approximating the true posterior $p(\theta|d)$. The KL term in (4) can be easily integrated as a Gaussian KL-divergence. We generate variational parameters $\mu(d)$ and $\sigma(d)$ through the inference network for document d so that we can estimate the variational lower bound by sampling θ from $q(\theta|d) = G(\theta|\mu(d), \sigma^2(d))$. We leave out the derivation details and refer the readers to [7].

C. Summarization

The transformer is used to be an encoder based on the self-attention mechanism. Compared with RNN that needs to process the input sequence word by word, it calculates the context vector of each word through self-attention, which has excellent parallelism and low computational complexity. We stacked a six-layered transformer that each layer has multihead attention and forward feedback layers. The final output of the encoder is contextual embedding, as described in Section III-A. As shown in Fig. 1, two modes are built up for summarization: the extractive mode and the abstractive mode.

1) Extractive: Extractive summarization can be defined as the task of assigning a label $Y_t \in \{0, 1\}$ to each sentence, indicating whether the sentence should be included in the summary or not. Moreover, LSTM combined with the transformer still has its unique advantages [35], which adds a forget gate to the simple RNN model to control historical state information. Therefore, the LSTM is combined for classifying summarization. After the encoder, the sentence embedding $S = \{s_1, s_2, \ldots, s_n\}$ is obtained to be the input of extractive model and filter key sentences by document-level features. At t-time step, the input is the vector s_t, and the output calculation process is given as follows:

$$f_t = \sigma(w_f s_t + b_f) \tag{5}$$

$$i_t = \sigma(w_i s_t + b_i) \tag{6}$$

$$o_t = \sigma(w_o s_t + b_0) \tag{7}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(w_c s_t + b_c) \tag{8}$$

$$h_t = o_t \odot \tanh(c_t) \tag{9}$$

where σ is a sigmoid function; s_t is the current input; f_t, i_t, and o_t are forget gates, input gates, and output gates; c_t and h_t are the context vector and the output vector; and w_f, b_f, w_i, b_i, w_o, b_o, w_c, and b_c are the weight and the bias of the forget gate, the input gate, the output gate, and the context vector, respectively. The final output layer uses a sigmoid function to calculate the final prediction score $\hat{Y}_t$, as shown in (10), to determine whether the sentence should be included in the summarization. The loss of the whole model is the binary classification entropy of $\hat{Y}_t$ against gold label Y_t

$$\hat{Y}_t = \text{sigmoid}(w_o h_t + b_o). \tag{10}$$

2) Abstractive: There are many words in the sequence, and only a few of them capture the key information of the entire sequence, which is exactly what we need. In order to filter the key information of the input sequence, the gated network is added to improve the transformer before the decoder, as shown in Fig. 3. Generally, the gated network is used to control the information flow from the input sequence to the output sequence, which makes the decoder focus on generating summaries from key information and removing unnecessary information. Here, the input of the gated network is the sentence representation s. The corresponding hidden layer representation of [CLS] is used as the representation of the input sequence vector, that is, $s = h_0$. At t-time step, the output is a new vector $\hat{h}_t$ obtained by filtering h_t; the gated network generates a threshold as follows:

$$g_t = \text{sigmoid}(W_g[h_t, s] + b) \tag{11}$$

where W_g is a linear transformation, b is the bias, g indicates the importance of the word, and $\hat{h}_t$ controls the filtered information, which is computed as

$$\tilde{h}_t = g_t h_t. \tag{12}$$

The filtered sequence is put into the N-layer transformer decoder, which is shown on in Fig. 3 (right). In order to efficiently decode the sequence and better capture the information passed by the encoder, the transformers' multihead attention

Authorized licensed use limited to: Sathyabama Institute of Science and Technology. Downloaded on December 20,2024 at 20:57:20 UTC from IEEE Xplore. Restrictions apply.

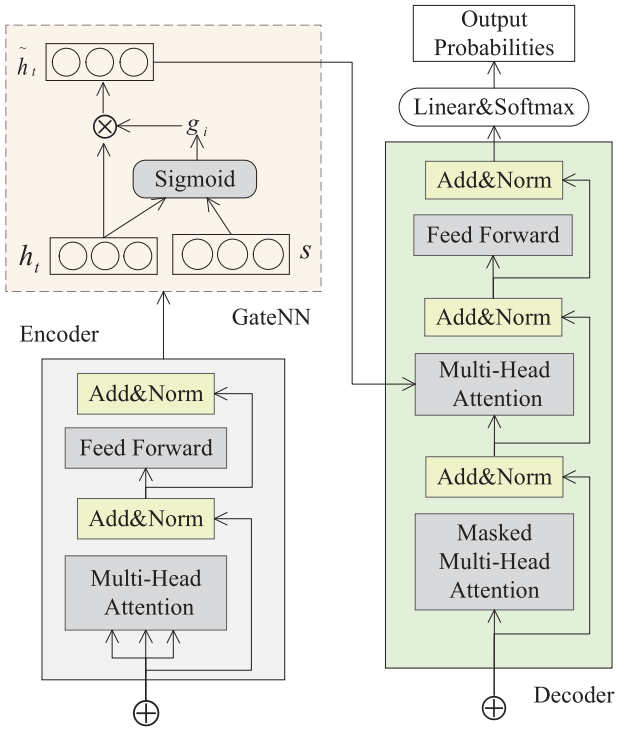


Fig. 3. *encoder_ours* architecture with the transformer, which consists of the gated network and the encoder–decoder framework based on multiattention.

is chose to help the decoder learn the soft alignment between the summary and the source document. The decoders learning objective is to minimize negative likelihood of conditional probability

$$L_{\text{dec}} = \sum_{t=1}^{|a|} -\log P(a_t = \hat{y}_t | a < t, H). \tag{13}$$

In addition, we propose a two-stage text summarization model based on the above work. In text summarization, the extracted sentences first processed by the encoder in the previous stage to obtain the encoder output sequence representation $X = \{x_1, x_2, \ldots, x_n\}$. The representation is used to be the input of the decoder; then, decoder predicts the final summary representation $Y = \{y_1, y_2, \ldots y_n\}$ by the transformer with gated network. We found that the model can take advantage of the information shared between these two tasks without fundamentally changing its architecture to provide a more complete sequence.

D. Joint Learning

The entire model integrates the NTM and the text summarization, which can be updated simultaneously in one framework. In this framework, we jointly deal with topic modeling and summaries' generation and define the loss function of the overall framework as follows:

$$L_{\text{final}} = \begin{cases} \lambda L_{\text{NTM}} + L_{\text{Ext}}, & \text{mod } e = \text{ext} \\ \lambda L_{\text{NTM}} + L_{\text{Abs}}, & \text{mod } e = \text{abs} \\ \lambda L_{\text{NTM}} + L_{\text{Ext}} + L_{\text{Abs}}, & \text{mod } e = \text{two-stage} \end{cases} \tag{14}$$

TABLE I

BASIC STATISTICS OF THE DATASET: SIZE OF TRAINING, VALIDATION, TEST SETS, AND AVERAGE DOCUMENT AND SUMMARY LENGTH (IN TERMS OF WORDS AND SENTENCES)

Datasets	# docs (train/val/test)	avg.doc length		avg.summary length		% novel bi-grams
		words	sentences	words	sentences	in gold summary
CNN	90,266/1,220/1,093	760.50	33.98	45.70	3.59	52.90
DailyMai	196,961/12,148/10,397	653.33	29.33	54.65	3.86	52.16
XSum	204,045/11,332/11,334	54.70	19.77	23.26	1.00	83.31

where L_{NTM} represents the loss of NTM and L_{Ext} and L_{Abs}, respectively, represent the losses of extractive summarization and abstractive summarization. L_{Ext} equals (9), and L_{Abs} equals (12). λ is the tradeoff parameter, controlling the balance between topic model and text summarization. To accommodate the fact that the encoder is pretrained and the decoder must be trained from scratch, the encoder's optimizer is separated from the decoder. We follow Liu and Lapata [30] to use different warm-up steps, learning rates, and two Adam optimizers for the encoder and the decoder, respectively. This will make the fine-tuning more stable.

IV. EXPERIMENT

A. Data Preparation

We conduct experiments on two benchmark datasets, namely, CNN/Daily Mail [36] and XSum [6], which are both common and well-known corpora for text summarization [1]. In recent years, the former has been widely used in automatic text summarization tasks due to its large data volume and long text content. The latter spurs further research toward the summarization models for its single news outlet and uniform summarization style. These datasets have different scales and methods of generation, some prefer to abstract, and some prefer to extract with a prominent sentence or first sentence. Table I demonstrates the statistics of these datasets as follows, including data segmentation, average text length, average summary length, and the percent of novel bigrams in gold summary. We used the standard splits of [36] for training, validation, and testing (90 266/1220/1093 CNN documents and 196 961/12 148/10 397 DailyMail documents). We used the splits of Narayan *et al.* (2018a) [6] for training, validation, and testing (204 045/11 332/11 334 XSum documents).

1) CNN/Daily Mail: There are 287 227 data for training, 13 368 data for validation, and 11 490 data for testing. CNN/Daily Mail consists of news articles with a summary corresponding to annotate several highlighted sentences manually. There are 52.90% novel bigrams in the CNN reference summaries and 52.16% in DailyMail. It is widely used in automatic text summarization tasks due to its large corpus and long text, and suitable for extractive and abstractive models. The original dataset can be applied here.[1]

2) XSum: It contains 226 711 BBC articles and is accompanied by a sentence summary, which answers the question regarding what this article is about. There are 83.31% novel bigrams in the XSum reference summaries. The articles and summaries in the XSum dataset are shorter, but the vocabulary

[1] https://github.com/abisee/cnn-dailymail

Authorized licensed use limited to: Sathyabama Institute of Science and Technology. Downloaded on December 20,2024 at 20:57:20 UTC from IEEE Xplore. Restrictions apply.

is large enough to be compared to CNN. The original dataset can be applied here.[2]

B. Experimental Setup

The experimental setup is described from the aspect of model settings, comparison models, and evaluation metrics. Among them, the evaluation metrics include automatic evaluation and manual evaluation.

1) Model Settings: All models were implemented on the PyTorch[3] [37] version of OpenNMT [38]. To reduce GPU memory, we choose the "BERTbase"[4] for fine-tuning, which has 110M total parameters. The size of the vocabulary is 30 522, and the dimensions of word embedding are 768. For the number of topics, we set $K = 1$. When $K = 1$, the guidance of summary generation is the best. When $K > 1$, the model will be a little bit disturbed; we observed that the ability of words to represent multiple topics is deficient. However, on the whole, multiple topics will not deviate from the topic too far, which is closer to the reference summary than the effect of setting K to 0. We obtained a probability distribution for each word over topics, and the topic distribution can be inferred for any new document. We follow the grid search of Miao *et al.* [7] by tuning the hyperparameters in NTM on the development set for achieving the held-out perplexity. We check sparsity (between le-3 and 0.75) to estimate perplexity. In order to achieve the optimal parameter setting, $\gamma = 0.8$ and $\lambda = 1.0$ for controlling the effects of NTM and summarization. μ_0 and σ_0^2 are hyperparameters that set for a zero mean and unit variance Gaussian. For extractive summarization, we get the score of each sentence according to the output layer, then arrange the sentences from high to low, and select the first three sentences as the key sentences. For abstractive summarization, we use beam search whose size is set to 4. During beam search, we set the probability of duplicate words to 0 and delete sentences with less than three words from the result set until an end-of-sequence token is emitted. We use the decoder of a six-layer transformer with 512 hidden units, six-head attention blocks, and 2048 hidden feedforward layers. The batch size is 140 with gradient accumulation every five steps.

We use the Adam optimizer [39] and follow Vaswani *et al.* [23] with a learning rate of $2e^{-3}$ and 0.05 for training the encoder and the decoder. In addition, we set two Adam optimizers with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for the encoder and the decoder, respectively. Model checkpoints were saved and evaluated on the validation set every 2000 steps. The maximum length of the sentence of the summary is set to 512. For regularization, we use dropout [40] and set the dropout rate to 0.1.

In addition, we did three comparative experiments: T-BERTSum(Ext) extracts vital sentences based on pretrained encoder and stacked LSTM, T-BERTSum(Abs) combines six-layer transformer encoder–decoder and gated network to generate summaries, and T-BERTSum(ExtAbs) integrates extractive and abstractive to generate sentence-level sequences.

2) Comparison Models: To further illustrate the superiority of our model over the two datasets, we compare the performance with many recent methods. Two groups are divided based on whether they are extractive models or abstractive models. All comparison models are described in detail as follows.

1) *Leading Sentences (Lead-3):* Lead3 is a baseline, which directly extracts the first three sentences of the article as a summary. This model is the extractive baseline.
2) *SummaRuNNer:* It is proposed by Nallapati *et al.* [41] to complete the sequence classification problem and selects the final subset of articles by RNN, as an extractive baseline.
3) *Refresh:* It is proposed by Narayan *et al.* [42], which optimizes the rouge evaluation by combining the maximum likelihood cross entropy and the reinforcement learning objective to make sentence ordering more accurate, as an extractive baseline.
4) *HSSAS:* It is proposed by Al-Sabahi *et al.* [43] to create sentence and document embeddings by a hierarchical self-attention mechanism, as an extractive baseline. We followed Al-Sabahi *et al.* [43] to set the maximum sentence length to 50 words and the maximum number of sentences per document to 100. At training time, the batch size was set to 64.
5) *BERT + Transformer:* It is a simple variant of BERT, which combines with the transformer to integrate sentences for extracting abstracts proposed by Liu [44]. We followed the original text to select the top-three checkpoints based on the evaluation losses and use the trigram blocking to reduce redundancy.
6) *Pointer-Generator + Coverage:* See *et al.* [45] copy the words directly from the original text through pointer and retain the ability to generate new words through generators, as an abstractive baseline.
7) *Bottom-Up:* It is proposed by Gehrmann *et al.* [46] to identify phrases in the source document that should be part of the summary by using a data-efficient content selector as a bottom-up attention step and an abstractive baseline.
8) *DCA:* Çelikyilmaz *et al.* [47] have multiple agents to represent documents and a hierarchical attention mechanism that decodes the agents. It serves as the best abstractive model in 2018, as a baseline for abstracting.
9) *BERTSum:* It is proposed by Liu and Lapata [30] to use pretrained language models to effectively summary in generation tasks, which can be used as a baseline for new methods.[5]
10) *BEAR:* It is proposed by Wang *et al.* [31] to use BERT word embedding as input and integrated the extractive network and the generation network into a unified model by reinforcement learning as an abstractive baseline. We followed the original text to set the learning rate of machine learning to 3×10^{-4}, the maximum length

[2]https://github.com/EdinburghNLP/XSum/tree/master/XSum-Dataset
[3]https://pytorch.org/
[4]https://github.com/huggingface/pytorch-pretrained-BERT
[5]https://github.com/nlpyang/BertSum

Authorized licensed use limited to: Sathyabama Institute of Science and Technology. Downloaded on December 20,2024 at 20:57:20 UTC from IEEE Xplore. Restrictions apply.

of the input text sequence sentences is set to 100, and the maximum number of sentences is set to 60.

3) Metrics:

a) Automatic evaluation: We apply ROUGE-1 that measures unigram recall between the summary and document, ROUGE-2 that measures bigram recall similarly, and ROUGE-L that measures the longest common subsequence between the summary and document as automatic evaluation. Rouge [48] compares the automatic generated summaries with the manual standard summaries by counting the overlapping lexical units between the two. This method has become the metric for evaluating the generated summary model and computed as follows:

$$\text{ROUGE}-N = \frac{\sum_{S\in\{\text{Ref}\}}\sum_{n-\text{grams}\in S}\text{Count}_{\text{match}}(n-\text{gram})}{\sum_{S\in\{\text{Ref}\}}\sum_{n-\text{grams}\in S}\text{Count}(n-\text{gram})} \tag{15}$$

where n is the length of the $n-$grams. $\text{Count}_{\text{match}}(n-\text{gram})$ is the number of $n-$grams appearing in a candidate summaries and a reference summaries simultaneously. $\text{Count}(n-\text{gram})$ is the number of $n-$grams in the reference summaries. The scores are computed by python pyrouge[6] package.

b) Manual evaluation: Since the automatic evaluation has great limitations in semantic and syntax information, the manual evaluation is further used to assist with automatic evaluation to verify our models. We implemented a small-scale manual evaluation and selected 100 articles from the test set to conduct anonymous experiments on the extractive model [Lead-3, BERTSum, and T-BERTSum(Ext)] and the abstractive model [PTGEN + COV, BEAR, and T-BERTSum(Abs)]. Due to the limited human resources, we randomly selected three highly educated volunteers to anonymously grade the summaries with six models, ranging from 1 to 5. A higher score is indicative of higher model capabilities. The evaluation includes the following.

1) *Salient:* The ability to reproduce the original information or viewpoint.
2) *Coherence:* Whether the summary has a consistent topic, meaning that whether the generated content is chaotic.
3) *Redundancy:* Whether the summary contains less redundant words.

C. Experimental Result and Analysis

We not only analyze the quality of the model from the standard automatic evaluation but also discuss the ability to generate text from the novelty percentage. Then, the ablation studies are done to verify the role of each component of the model. Finally, the differences between the original text and the generated summaries are shown in the case study, and the importance of topic-aware is verified.

1) Result: Tables II and III show the results of automatic evaluation of our model and the comparative model on the CNN/Daily Mail and XSum datasets, respectively. As shown in Table II, the first part of the table is the comparative baseline of the extractive model. The second part of the table is the comparative baseline of the abstractive model. The last part is our model, T-BERTSum(Ext), T-BERTSum(Abs), and T-BERTSum(ExtAbs). The topic-aware model based on pretraining performs better than traditional models in different evaluation standards, indicating that the transformer's good architecture based on attention can capture key information and summarize valid text instead of using traditional recurrent neural networks. Whether our model is compared with the extractive baseline or the abstractive baseline, the score reflects the superiority of the model, which implies the necessity of introducing the theme to guide the generation.

In the complete experimental group, the experimental results of our model are significantly improved compared with the earlier work and baseline model. As shown in Table II, we obtain the highest ROUGE score for the current abstractive summary compared with HSSAS, which has the ability to automatically learn distributed representations of sentences and documents and convert the summary task into a classification task by calculating the respective probabilities of the membership of the sentence summary. This model uses a hierarchical self-attention mechanism to create sentences and document embeddings, which is similar to our model. However, T-BERTSum(Ext) has increased by a few percentages,

[6]https://pypi.org/project/pyrouge/

TABLE II
PERFORMANCE COMPARISON OF MODELS WITH RESPECT TO THE BASELINES ON CNN/DAILY MAIL. R-AVG CALCULATES AVERAGE SCORE OF ROUGE-1, ROUGE-2, AND ROUGE-L

Model	ROUGE-1	ROUGE-2	ROUGE-L	R-AVG
Extractive				
Lead-3	40.34	17.70	36.57	31.54
SummaRuNNer[41]	39.60	16.20	35.30	30.37
Refresh[42]	40.00	18.20	36.60	31.60
HSSAS[43]	42.30	17.80	37.60	32.57
BERT+Transformer[44]	43.17	20.18	39.35	34.23
Abstractive				
Pointer-generator+coverage[45]	39.53	17.28	36.38	31.06
Bottom-Up[46]	41.22	18.68	38.34	32.75
DCA[47]	41.69	19.47	37.92	33.03
BERTSum[30]	41.72	19.39	38.76	33.29
BEAR[31]	41.90	20.16	39.39	33.82
Ours				
T-BERTSum(Ext)	**43.58**	20.43	**39.80**	**34.60**
T-BERTSum(Abs)	42.12	**20.45**	39.74	34.10
T-BERTSum(ExtAbs)	43.06	19.76	39.43	34.03

TABLE III
PERFORMANCE COMPARISON OF MODELS WITH RESPECT TO THE BASELINES ON XSUM. R-AVG CALCULATES AVERAGE SCORE OF ROUGE-1, ROUGE-2, AND ROUGE-L

Model	ROUGE-1	ROUGE-2	ROUGE-L	R-AVG
Lead-3	16.30	1.60	11.95	9.95
Abstactive				
Pointer-generator[45]	29.70	9.21	23.24	20.72
Pointer-generator+coverage[45]	28.10	8.02	21.72	19.28
BERTSUMEXT[30]	38.76	16.33	31.15	28.75
BERTSUMEXTABS[30]	38.81	16.50	31.27	28.86
Ours				
T-BERTSum(Abs)	**39.90**	**17.48**	**32.18**	**29.85**
T-BERTSum(ExtAbs)	39.62	17.26	31.67	29.52

Authorized licensed use limited to: Sathyabama Institute of Science and Technology. Downloaded on December 20,2024 at 20:57:20 UTC from IEEE Xplore. Restrictions apply.

indicating that, to greatly improve the quality of the generation, it requires not only the accurate embedding of the document but also the auxiliary extraction of additional information, so that the model can accurately locate the most relevant important sentences. Compared with BERTSum, which also uses BERT for the text embedding and transformer for the basic architecture, our abstractive model improves by 0.4%, 1.06%, and 0.98% on ROUGE-1, ROUGE-2, and ROUGE-L, respectively. The improvement of the score explains that it is indispensable to add topic awareness to the model by mapping text embeddings. Topic-aware word embeddings or sentence sequences can better help the model to classify and identify the text, so as to achieve the formation of a summary that does not deviate from the topic. Compared with the traditional model, we not only interpret and infer the text by encoding–decoding but also integrate the background information of the text into the text embedding from the perspective of text understanding, so as to make favorable use of the ability of BERT to generate the model of the pretraining language with large-scale corpus. We also try to combine the extractive model with the abstractive model so that the two promote each other and share useful information. The two-stage extractive–abstractive model also achieved the highest score on ROUGE-L compared with other extractive and abstractive models. The two-stage extractive–abstractive model T-BERTSum(ExtAbs) also scored the highest on ROUGE-L. Compared with the extractive model and the abstractive model, there is a small drop in scores, which means that the two-stage model can share information, causing less interference and reducing the redundancy of generation. It can be observed that our extractive model performs better than the abstractive model and the two-stage model, which is related to the CNN/Daily Mail dataset that is biased toward extracting prominent sentences as summaries.

For XSum, each article is accompanied by a sentence that tends to generate. As shown in Table III, our models significantly outperform the comparison models across all three variants of the ROUGE metric. Compared with the classic pointer network, pointer-generator + coverage, our method has been significantly improved. The two-stage model has improved by nearly 10%. This shows that the pointer network can well integrate the extractive and abstractive methods, but it lacks the ability of text embedding representation and topic awareness guidance, which are indispensable for the quality assurance of generation. Compared with the best model, which is also based on BERT as text embedding, the scores of ROUGE-1, ROUGE-2, and ROUGE-L are, respectively, improved 1.14%, 1.15%, and 1.03% for the abstractive model, which manifests that our model still has certain advantages in abstractive method and can focus on effective information. The NTM can effectively inference and enhance the model's understanding ability. The two-stage model has also improved by 0.81%, 0.76%, and 0.4% on ROUGE-1, ROUGE-2, and ROUGE-L. Compared with other baseline models, our model has obvious advantages in generative datasets, which shows that our model can capture context information and summarize it well.

We also evaluate the novelty of the abstractive model by calculating the proportion of newly appeared n-grams on

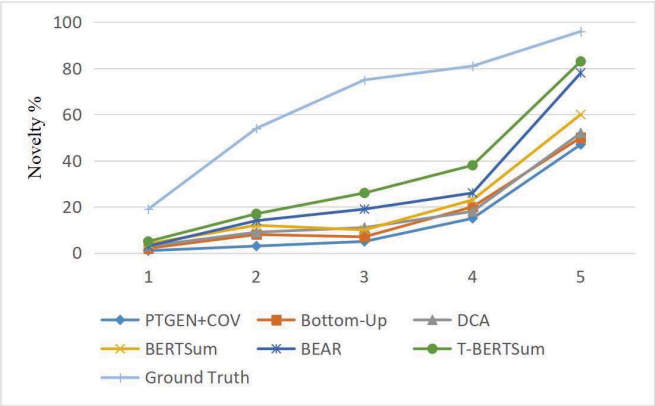


Fig. 4. Statistics of novel n-grams and sentences on the CNN/Daily mail dataset. Our model can generate far more novel n-grams and sentences than other baseline models.

CNN/Daily mail with the source text. Novelty measures the generative ability of a model. As shown in Fig. 4, our model copies next to 15% of sentences from the source text, and the copy rate is closest to the copy rate of the reference summaries. PTGEN + COV has low novelty, copying nearly 50% of the original sentence and generating nearly 1% of new words, because the pointer network tends to select words from the original text. Compared with the BERA model, our model has been greatly improved, with relatively high novelty, and nearly 5% of the token is newly generated because our model emphasizes the understanding of semantics and has the support of large-scale lexical reserves. It also proves that the introduction of topic representation can improve the encoding ability and memory ability of the encoder; the gated network can also remember valid information.

2) Ablation Study: In order to verify the validity of the representation of the pretrained language model, we conducted a simple comparison experiment. We modeled $\text{BERT}_{\text{large}}$ on the extractive model, another version of BERT, which consists of larger corpus training and architecture. We found that the model was slightly improved, as shown in Fig. 5. This also reflects BERT's strong representational ability. Compared with LEAD-3, the model has been significantly improved, which proves that the first step of the text task is to have a powerful feature representation to fully understand the original article as much as possible. Compared with the T-BERTSum(base), T-BERTSum(large) has a certain improvement, which is 0.63%, 0.7%, and 0.95%, respectively. We believe that BERT can further improve the performance of generating summaries through pretraining on large datasets and powerful architecture for learning complex features.

Ablation studies show the contribution of different components of T-BERTSum, and the results are shown in Fig. 6. We conducted two comparative experiments on the extractive model [see Fig. 6(a)]: one model only used a six-layer transformer to classify the sentences in the decoding stage instead of the LSTM network, and one model used the unpretraining transformer architecture that has fewer parameters. The former improves 0.3% on average of ROUGE, which is in line with our hypothesis that the combination of transformer and

Authorized licensed use limited to: Sathyabama Institute of Science and Technology. Downloaded on December 20,2024 at 20:57:20 UTC from IEEE Xplore. Restrictions apply.

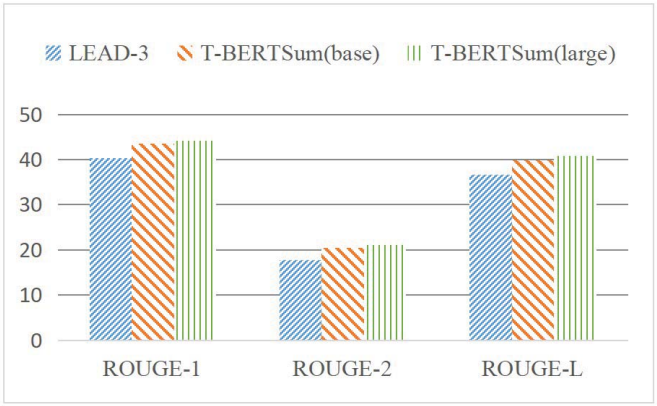


Fig. 5. Performance comparison between the $\text{BERT}_{\text{base}}$ model and the $\text{BERT}_{\text{large}}$ model.

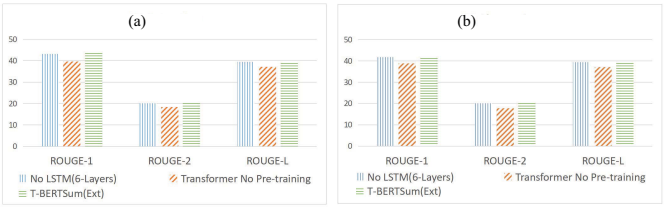


Fig. 6. Results of ablation study between (a) extractive model and (b) abstractive model.

TABLE IV

HUMAN EVALUATION OF SIX MODELS. WE COMPARE THE SCORE OF SALIENT, COHERENCE, AND REDUNDANCY

Model	Lead-3	BERTSum	T-BERTSum (Ext)	PTGEN +COV	BEAR	T-BERTSum (Abs)
salient	2.61	2.85	2.96	2.91	3.01	3.11
coherence	3.56	3.43	3.90	3.60	3.29	3.89
redundancy	-	-	-	3.31	3.11	3.13

LSTM has certain advantages. LSTM can make up for the shortcomings of the transformer in abandoning RNN. The latter improves the average of ROUGE by nearly 3% points. Pretraining improves the efficiency and accuracy of the model. Although there are a lot of parameters, we only need to train these parameters with different optimizers. We also performed two comparative experiments on the abstractive model [see Fig. 6(b)]: the model without the gated network and the model without pretraining. Although the former has only a slight improvement and a small contribution to the removal of useless information, it does not trigger a reverse effect on the model. The gated network memorizes the information to a certain extent, which is beneficial for capturing contextual information. It is obvious that pretraining can greatly improve the ability to generate and reduce the burden on the model.

3) Manual Evaluation and Demonstration: For simplicity, we only perform a manual evaluation on the CNN/Daily Mail dataset. The results of evaluation scoring are shown in Table IV. Consistent with the results of the automatic evaluation, our model obtained better scores in both extractive and abstractive, which further proves that our model can well model the original documents and accurately capture crucial information.

TABLE V

COMPARISON OF GROUND-TRUTH SUMMARY AND GENERATED SUMMARIES OF THE BASELINE BEAR MODEL AND OUR MODEL ON THE CNN/DAILY MAIL DATASET. FOR BREVITY, THIS ARTICLE HAS BEEN SHORTENED. FOR READABILITY, CAPITALIZATION WAS ADDED MANUALLY

Example	
Article snippet	Miami Heat superstar LeBron James gave a tou -ching tribute to one of his biggest fans last nig -ht - wearing her name on his sneakers just hou -rs after she lost a grueling six-year battle with cancer.Bella Rodriguez-Torres, ...she slipped aw -ay peacefully surrounded by family and friends. ...And at Tuesday night's Miami Heat Vs Pacers game, LeBron James and Dwyane Wade wore s -hoes with #LiveLikeBella written on them in honor of the brave Heat fan.... Shannah and Bell -a's father, Raymond, broke the sad news on thei -r daughter's Facebook page , which has more th -an 62,000 followers. ...But in 2007, at the tender age of just four, she became paralyzed from a tu -mor on her spine and was diagnosed with stage four Alveolar Rhabdomyosarcoma. But the dogg -ed illness returned in April 2009, when doctors found a tumor in her brain and she underwent fu -ll brain radiation plus more chemotherapy ...
Gold Summary	Bella Rodriguez-Torres was diagnosed with stage four Alveolar Rhabdomyosarcoma at age four.T -he brave 10-year-old fought the illness but it ca -me back four times. She passed away peacefully on Tuesday surrounded by family and friends. At Tuesday night's Miami Heat Vs Pacers game, Le -Bron James wrote #LiveLikeBella on his sneake -rs in honor of the young fan.
BEAR[31]	LeBron James gave a touching tribute to one of his biggest fans last night. she was diagnosed wi -th stage four Alveolar Rhabdomyosarcoma. Bell -a slipped away peacefully surrounded by family and friends. Bella's parents broke the sad news o -n their daughter's Facebook page with 62,000 fo -llowers.
T-BERTSum(Abs)	Bella Rodriguez-Torres was diagnosed with stage four Alveolar Rhabdomyosarcoma. but neoplasm recurrence, despite putting up another fierce fight and began failing.Basketball star LeBron James wore sneakers with #LiveLikeBella on them on Tuesday night in the little Heat fan's honor.

In terms of coherence, LEAD-3 is only second to our model in terms of coherence that the method selects the first three sentences of the original article as the summary, which is a high probability to describe the same point of standpoint. Nevertheless, our model still gets a high score, proving that the model makes positive use of the background information of the original text and the guidance to generate the summary with topics. It does not make much sense to evaluate the redundancy of the extractive model. In terms of redundancy, it does not make much sense to evaluate the redundancy on the extractive model because the extractive method will remove repeated

Authorized licensed use limited to: Sathyabama Institute of Science and Technology. Downloaded on December 20,2024 at 20:57:20 UTC from IEEE Xplore. Restrictions apply.

sentences. On the redundancy evaluation of the abstractive model, PTGEN + COV obtained the highest score because the method incorporated a replication mechanism based on the pointer network to avoid the generation of overlapping words. However, our model score is not too low since we take the attitude that the gated network plays a certain role in filtering information at each step. In terms of salient, we got the highest evaluation score by manual evaluation, which is the recognition of our model ability and the quality of the summaries. We found that our model has a better comprehensive ability, which proves that integrating the topic information can better summarize the original text based on the premise of strong representation ability.

We present the example in Table V for comparison with our model and the baseline model. We can see that our model does not lose crucial information because of the long distance for the long article while capturing the topic reliably. The words of the table marked with underlines are the important topics of the text. As an example, our model captures core ideas, such as "$fierce\ fight$" around the topic compared to the baseline model, which can well reflect the event itself described by the article. When both models capture the same topic, our model can also generate new recurrence vocabulary based on the topic, which is effective and accurate.

V. Conclusion and Future Work

In this work, we propose a general model of extractive and abstractive for text summarization, which is based on BERT's powerful architecture and additional topic embedding information to guide contextual information capture. For a good summary, an accurate representation is extremely important. This article introduces the representation of a powerful pretraining language model (BERT) to lay the foundation of the source text encoding and emphasizes the subjectivity of the generated content. The fusion of topic embedding is a direct and effective way to achieve high-quality generation through NTM inferring. The combination of token embedding, segment embedding, position embedding, and topic embedding can more abundantly embed the information that the original text should contain. Stacking the transformer layer in the encoding stage is able to enhance the BERT's ability to represent source texts, make full use of self-attention, and judge the importance of different components of the sentence through different focus scores. The two-stage extractive–abstractive model can share information and generate salient summaries, which reduces a certain degree of redundancy. The experimental results show that the model proposed in this article achieves the state-of-the-art results on the CNN/Daily Mail dataset and the XSum dataset. The analysis shows that the model can generate high-quality summaries with outstanding consistency for the original text.

Although the model has made some progress in text summarization, it also has some limitations. For long articles with multiple topics, our model has limited processing power. In future work, we will try to extend our work to multitopic with the transformer network to capture multiple topics hierarchically by imitating multihead self-attention and further prove the validity of this article. In addition, we need to further solve another big problem, that is, the generated summaries do not match the facts of the source text: on the one hand, how to introduce additional structured knowledge so that the encoder can not only consider the contextual representation but also consider additional knowledge information; on the other hand, how to extend the topic information so that we can obtain multiple topics and subtopics of the article to enhance sentence information and consolidate document-level knowledge. Finally, we can consider how to process topic information and additional structured knowledge in parallel on the basis of the method in this article, so as to make a qualitative leap in the task of text summarization while keeping the generated summary consistent with the original facts.

References

[1] M. Allahyari *et al.*, "Text summarization techniques: A brief survey," 2017, *arXiv:1707.02268*. [Online]. Available: http://arxiv.org/abs/1707.02268

[2] T. Ma, Q. Liu, J. Cao, Y. Tian, A. Al-Dhelaan, and M. Al-Rodhaan, "LGIEM: Global and local node influence based community detection," *Future Gener. Comput. Syst.*, vol. 105, pp. 533–546, Apr. 2020, doi: 10.1016/j.future.2019.12.022.

[3] A. Khan and N. Salim, "A review on abstractive summarization methods," *J. Theor. Appl. Inf. Technol.*, vol. 59, no. 1, pp. 64–72, 2014.

[4] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: A survey," *Artif. Intell. Rev.*, vol. 47, no. 1, pp. 1–66, Jan. 2017, doi: 10.1007/s10462-016-9475-9.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. Minneapolis, MN, USA: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186, doi: 10.18653/v1/n19-1423.

[6] S. Narayan, S. B. Cohen, and M. Lapata, "Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization," 2018, *arXiv:1808.08745*. [Online]. Available: http://arxiv.org/abs/1808.08745

[7] Y. Miao, E. Grefenstette, and P. Blunsom, "Discovering discrete latent topics with neural variational inference," 2017, *arXiv:1706.00359*. [Online]. Available: http://arxiv.org/abs/1706.00359

[8] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. 2014 Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1724–1734, doi: 10.3115/v1/d14-1179.

[9] H. Su *et al.*, "Improving multi-turn dialogue modelling with utterance ReWriter," 2019, *arXiv:1906.07004*. [Online]. Available: http://arxiv.org/abs/1906.07004

[10] M. E. Peters *et al.*, "Deep contextualized word representations," 2018, *arXiv:1802.05365*. [Online]. Available: http://arxiv.org/abs/1802.05365

[11] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI Company, San Francisco, CA, USA, Tech. Rep., 2018.

[12] Y. Liu *et al.*, "RoBERTa: A robustly optimized BERT pre-training approach," 2019, *arXiv:1907.11692*. [Online]. Available: http://arxiv.org/abs/1907.11692

[13] T. Ma, Y. Zhao, H. Zhou, Y. Tian, A. Al-Dhelaan, and M. Al-Rodhaan, "Natural disaster topic extraction in sina microblogging based on graph analysis," *Expert Syst. Appl.*, vol. 115, pp. 346–355, Jan. 2019, doi: 10.1016/j.eswa.2018.08.010.

[14] N. Akhtar, H. Javed, and T. Ahmad, "Hierarchical summarization of text documents using topic modeling and formal concept analysis," in *Data Management, Analytics and Innovation*. Singapore: Springer, 2019, pp. 21–33.

[15] R. K. Roul, S. Mehrotra, Y. Pungaliya, and J. K. Sahoo, "A new automatic multi-document text summarization using topic modeling," in *Proc. Int. Conf. Distrib. Comput. Internet Technol. (ICDCIT)*, vol. 11319. Cham, Switzerland: Springer, Jan. 2019, pp. 212–221, doi: 10.1007/978-3-030-05366-6_17.

[16] C. Lin and E. H. Hovy, "The automated acquisition of topic signatures for text summarization," in *Proc. 18th Int. Conf. Comput. Linguistics (COLING)*. San Mateo, CA, USA: Morgan Kaufmann, Jul./Aug. 2000, pp. 495–501. [Online]. Available: https://www.aclweb.org/anthology/C00-1072/

Authorized licensed use limited to: Sathyabama Institute of Science and Technology. Downloaded on December 20,2024 at 20:57:20 UTC from IEEE Xplore. Restrictions apply.

[17] H. Pan, H. Liu, and Y. Tang, "A sequence-to-sequence text summarization model with topic based attention mechanism," in *Proc. Int. Conf. Web Inf. Syst. Appl.*, vol. 11817. Cham, Switzerland: Springer, Sep. 2019, pp. 285–297, doi: 10.1007/978-3-030-30952-7_29.

[18] Z. Yang, Y. Yao, and S. Tu, "Exploiting sparse topics mining for temporal event summarization," in *Proc. IEEE 5th Int. Conf. Image, Vis. Comput. (ICIVC)*, Jul. 2020, pp. 322–331.

[19] A. T. Sadiq, Y. H. Ali, and M. S. M. N. Fadhil, "Text summarization for social network conversation," in *Proc. Int. Conf. Adv. Comput. Sci. Appl. Technol.*, Dec. 2013, pp. 13–18.

[20] Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, and T. Zhao, "Neural document summarization by jointly learning to score and select sentences," 2018, *arXiv:1807.02305*. [Online]. Available: http://arxiv.org/abs/1807.02305

[21] S. Ghodratnama, A. Beheshti, M. Zakershahrak, and F. Sobhanmanesh, "Extractive document summarization based on dynamic feature space mapping," *IEEE Access*, vol. 8, pp. 139084–139095, 2020, doi: 10.1109/ACCESS.2020.3012539.

[22] L. Wang, J. Yao, Y. Tao, L. Zhong, W. Liu, and Q. Du, "A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization," 2018, *arXiv:1805.03616*. [Online]. Available: http://arxiv.org/abs/1805.03616

[23] A. Vaswani *et al.*, "Attention is all you need," 2017, *arXiv:1706.03762*. [Online]. Available: http://arxiv.org/abs/1706.03762

[24] T. Ma, H. Wang, L. Zhang, Y. Tian, and N. Al-Nabhan, "Graph classification based on structural features of significant nodes and spatial convolutional neural networks," *Neurocomputing*, vol. 423, pp. 639–650, Jan. 2021, doi: 10.1016/j.neucom.2020.10.060.

[25] T. Cai, M. Shen, H. Peng, L. Jiang, and Q. Dai, "Improving transformer with sequential context representations for abstractive text summarization," in *Proc. CCF Int. Conf. Natural Lang. Process. Chin. Comput. (NLPCC)*, vol. 11838. Cham, Switzerland: Springer, Oct. 2019, pp. 512–524, doi: 10.1007/978-3-030-32233-5_40.

[26] M.-H. Su, C.-H. Wu, and H.-T. Cheng, "A two-stage transformer-based approach for variable-length abstractive summarization," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 2061–2072, 2020, doi: 10.1109/TASLP.2020.3006731.

[27] X. Zhang, F. Wei, and M. Zhou, "HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization," 2019, *arXiv:1905.06566*. [Online]. Available: http://arxiv.org/abs/1905.06566

[28] A. Hoang, A. Bosselut, A. Celikyilmaz, and Y. Choi, "Efficient adaptation of pretrained transformers for abstractive summarization," 2019, *arXiv:1906.00138*. [Online]. Available: http://arxiv.org/abs/1906.00138

[29] H. Zhang, J. Xu, and J. Wang, "Pretraining-based natural language generation for text summarization," 2019, *arXiv:1902.09243*. [Online]. Available: http://arxiv.org/abs/1902.09243

[30] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, Nov. 2019, pp. 3728–3738, doi: 10.18653/v1/D19-1387.

[31] Q. Wang, P. Liu, Z. Zhu, H. Yin, Q. Zhang, and L. Zhang, "A text abstraction summary model based on BERT word embedding and reinforcement learning," *Appl. Sci.*, vol. 9, no. 21, p. 4701, Nov. 2019.

[32] A. Srikanth, A. S. Umasankar, S. Thanu, and S. J. Nirmala, "Extractive text summarization using dynamic clustering and co-reference on BERT," in *Proc. 5th Int. Conf. Comput., Commun. Secur. (ICCCS)*, Patna, India, Oct. 2020, pp. 1–5, doi: 10.1109/ICCCS49678.2020.9277220.

[33] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*. [Online]. Available: http://arxiv.org/abs/1607.06450

[34] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," 2016, *arXiv:1601.00670*. [Online]. Available: http://arxiv.org/abs/1601.00670

[35] M. X. Chen *et al.*, "The best of both worlds: Combining recent advances in neural machine translation," 2018, *arXiv:1804.09849*. [Online]. Available: http://arxiv.org/abs/1804.09849

[36] K. M. Hermann *et al.*, "Teaching machines to read and comprehend," 2015, *arXiv:1506.03340*. [Online]. Available: http://arxiv.org/abs/1506.03340

[37] A. Paszke *et al.*, "Automatic differentiation in PyTorch," in *Proc. NIPS Autodiff Workshop, Future Gradient-Based Mach. Learn. Softw. Techn.*, Long Beach, CA, USA, Dec. 2017.

[38] G. Klein, Y. Kim, Y. Deng, V. Nguyen, J. Senellart, and A. M. Rush, "OpenNMT: Neural machine translation toolkit," 2018, *arXiv:1805.11462*. [Online]. Available: http://arxiv.org/abs/1805.11462

[39] D. P. Kingma and B. Jimmy, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015.

[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2670313

[41] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," in *Proc. 31st AAAI Conf. Artif. Intell.* San Francisco, CA, USA. AAAI Press, Feb. 2017, pp. 3075–3081. [Online]. Available: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14636

[42] S. Narayan, S. B. Cohen, and M. Lapata, "Ranking sentences for extractive summarization with reinforcement learning," 2018, *arXiv:1802.08636*. [Online]. Available: http://arxiv.org/abs/1802.08636

[43] K. Al-Sabahi, Z. Zuping, and M. Nadher, "A hierarchical structured self-attentive model for extractive document summarization (HSSAS)," *IEEE Access*, vol. 6, pp. 24205–24212, 2018, doi: 10.1109/ACCESS.2018.2829199.

[44] Y. Liu, "Fine-tune BERT for extractive summarization," 2019, *arXiv:1903.10318*. [Online]. Available: http://arxiv.org/abs/1903.10318

[45] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proc. 55th Annu. Meeting Assoc. for Comput. Linguistics (ACL)*, vol. 1. Stroudsburg, PA, USA: Association for Computational Linguistics, Jul./Aug. 2017, pp. 1073–1083, doi: 10.18653/v1/P17-1099.

[46] S. Gehrmann, Y. Deng, and A. M. Rush, "Bottom-up abstractive summarization," 2018, *arXiv:1808.10792*. [Online]. Available: http://arxiv.org/abs/1808.10792

[47] A. Çelikyilmaz, A. Bosselut, X. He, and Y. Choi, "Deep communicating agents for abstractive summarization," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol. (NAACL-HLT)*, vol. 1. Stroudsburg, PA, USA: Association for Computational Linguistics, Jun. 2018, pp. 1662–1675, doi: 10.18653/v1/n18-1150.

[48] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, 2004, pp. 74–81.

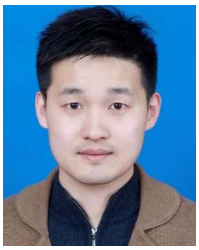
Tinghuai Ma (Member, IEEE) received the bachelor's and master's degrees from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1997 and 2000, respectively, and the Ph.D. degree from the Chinese Academy of Sciences, Beijing, China, in 2003.

He was a Post-Doctoral Associate with AJOU University, Suwon, South Korea, in 2004. From November 2007 to July 2008, he visited the Chinese Meteorology Administration, Beijing. From February 2009 to August 2009, he was a Visiting Professor with the Ubiquitous Computing Laboratory, Kyung Hee University, Seoul, South Korea. He is currently a Professor of computer sciences with Nanjing University of Information Science and Technology, Nanjing, China. He has published more than 100 journal articles/conference papers. His research interests are data mining, cloud computing, ubiquitous computing, privacy-preserving, and so on.

Qian Pan received the bachelor's degree in software engineering from the Nanjing University of Information Science and Technology, Nanjing, China, in 2021.

She is currently a Computer Professional Researcher with the Nanjing University of Information Science and Technology. Her research interest lies in data mining, especially on the text summarization task.

Authorized licensed use limited to: Sathyabama Institute of Science and Technology. Downloaded on December 20,2024 at 20:57:20 UTC from IEEE Xplore. Restrictions apply.



Huan Rong received the Ph.D. degree in computer science from the Nanjing University of Information Science and Technology, Nanjing, China, in 2020.

He is currently a Visiting Scholar with the University of Central Arkansas, Conway, AR, USA. He is also an Assistance Professor with the School of Artificial Intelligence, Nanjing University of Information Science and Technology. His research interests lie in deep learning and the application of artificial intelligence, especially in sentiment analysis and other interdisciplinary tasks. His research contributions have been published in *Information Sciences*, IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, *Soft Computing*, and so on.

Yurong Qian received the B.S. and M.S. degrees in computer science and technology from Xinjiang University, Urumqi, China, in 2002 and 2005, respectively, and the Ph.D. degree in biology from Nanjing University, Nanjing, China, in 2010.

From 2012 to 2013, she was a Post-Doctoral Fellow with the Department of Electronics and Computer Engineering, Hanyang University, Seoul, South Korea. She is currently a Professor with the School of Software, Xinjiang University. Her research interests include cloud computing, image processing, and intelligent computation, such as artificial neural networks.

Yuan Tian received the master's and Ph.D. degrees from Kyung Hee University, Seoul, South Korea, in 2009 and 2012, respectively.

She is currently an Assistant Professor with the College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. She is also an Associate Professor with the School of Computer, Nanjing Institute of Technology, Nanjing, China. Her research interests are broadly divided into privacy and security, which are related to the cloud.

Dr. Tian is also a member of the technical committees of several international conferences. She is also an active reviewer of many international journals.

Najla Al-Nabhan received the B.S. degree (Hons.) in computer applications and the M.S. degree in computer science from The George Washington University, Washington, DC, USA, in 2005 and 2008, respectively, and received the Ph.D. degree in computer science from King Saud University (KSU), Riyadh, Saudi Arabia, in 2013.

She is currently the Vice Assistant Professor with the Computer Science Department, College of Computer and Information Sciences (CCIS), KSU. Her current research interests include wireless sensor networks, multimedia sensor networks, cognitive networks, and network security.

Authorized licensed use limited to: Sathyabama Institute of Science and Technology. Downloaded on December 20,2024 at 20:57:20 UTC from IEEE Xplore. Restrictions apply.