IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 14, 2021, accepted August 1, 2021, date of publication August 5, 2021, date of current version August 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3102741

HG-News: News Headline Generation Based on a Generative Pre-Training Model

PING LI[1], JIONG YU[1], JIAYING CHEN[1], AND BINGLEI GUO[2]
[1]College of Information Science and Engineering, Xinjiang University, Ürümqi, Xinjiang 830000, China
[2]School of Computer Engineering, Hubei University of Arts and Science, Xiangyang 441053, China

Corresponding author: Binglei guo (binglei@hbuas.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61862060.

ABSTRACT Neural headline generation models have recently shown great results since neural network methods have been applied to text summarization. In this paper, we focus on news headline generation. We propose a news headline generation model based on a generative pre-training model. In our model, we propose a rich features input module. The headline generation model we propose only contains a decoder incorporating the pointer mechanism and the n-gram language features, while other generation models use the encoder-decoder architecture. Experiments on news datasets show that our model achieves comparable results in the field of news headline generation.

INDEX TERMS Generation model, headline generation, text summarization, neural network.

I. INTRODUCTION

The aim of the text summarization is to condense a document while the condensed content retains the core meaning of the original document. The text summarization methods consist of extractive summarization and abstractive summarization. Headline generation is an abstractive summarization subtask, which is also called sentence summarization. In order to generate the headlines that compress the information in a longer text or a short text, we need to do the research about headline generation.

We focus on the task of neural headline generation (NHG). An artificial neural network is used to solve the text generation task. Approaches using neural networks have shown promising results on the headline generation task which using an end-to-end model to encode a source document and then to decode it into a news headline. Most of the previous works are concerned with single document summarization, while this paper is only concerned with headline generation. The pioneering work of neural headline generation is [1], which uses the encoder-decoder framework to generate sentence-level summarization. With the development of the recurrent neural network (RNN) [2], [3] employed the attentive encoder-decoder model to sentence summarization. To represent the semantic better, a transformer was proposed [4]. Then, the transformer was employed for abstractive summarization [5], but the result was not improved compared with the attentive sequence-to-sequence model. OpenAI demonstrates that language models begin to learn the natural language processing tasks without any explicit supervision when trained on a new dataset. The model proposed by OpenAI is called GPT-2 [6]. Rothe developed a transformer-based sequence-to-sequence model with the pre-training BERT [7], GPT-2 and RoBERTa [8] checkpoints for sequence generation task [9]. In order to demonstrate the efficacy of the GPT-2 for the headline generation task, we don't leverage the checkpoints of the pre-training model but just use the structure of the GPT-2 model. Because most of the text summarization datasets are written in English, [10] proposed a large-scale short text summarization dataset. Currently, the whole summarization generation models use the encoder-decoder architecture to generate the summarization. We will attempt to only use the decoder to solve the headline generation task. In this paper, we conduct experiments on news datasets. We only employ the decoder model and the pointer mechanism for the headline generation task and incorporate the n-gram language information into the decoder. In our model, we propose a rich features input module. Moreover, we compare the experimental results based on the attentive sequence to the sequence model with our model.

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaochun Cheng.

II. BACKGROUND

In the task of headline generation, the input is represented as X, and the headline of the news article is represented as Y. X is equal to $\{x_1, x_2 \ldots, x_m\}$, and Y is equal to $\{y_1, y_2, \ldots, y_n\}$, where $n \ll m$. Specifically, the Y is only one sentence. Throughout this paper, x_i represents the word in the article and y_j represents the word in the summarization. The vocabulary list is represented as $V = \{w_1, w_2, \ldots, w_p\}$, and w_k also represents a word. The p is limited to 50,000 in this paper.

In our headline generation model, we model

$$p(Y|X) = \prod_{t=1}^{T} p\left(y_t|X, y < t, \theta\right), \tag{1}$$

where the θ is the model parameter.

III. MODEL

A. TRADITIONAL FRAMEWORK OF NEWS HEADLINE GENERATION

The traditional news headline generation framework consists of five parts, which are shown in figure 1. In contrast to English news headline generation, the difference of the other language news headline generation is the tokenization. Given the input document, the tokenizer isolates the news document word by word. The encoder obtains the distributed representation of the document with the input word list X. Through the attention module, the distributed representation is recalculated. The decoder calculates the probability based on the word vocabulary list. With the greedy search or beam search method, we obtain the headline.

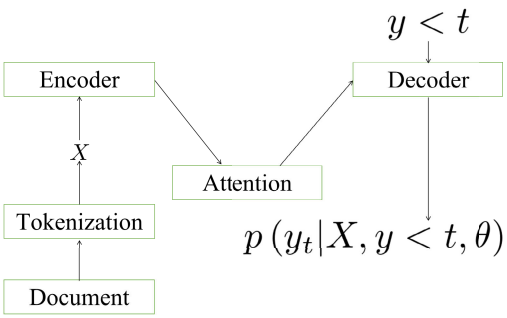

FIGURE 1. The framework of the news headline generation.

B. HEADLINE GENERATION MODEL

1) THE SEQUENCE-TO-SEQUENCE GENERATION MODEL BASED ON ATTENTION

The sequence-to-sequence generation model is shown in figure 2, which contains two parts. The first part is the encoder, and the second part is the decoder based on the attention mechanism. The input of the encoder is X. The encoder is a bi-directional long short-term memory(biLSTM), which generates the hidden state representation based on the one-hot representation of the input words. The biLSTM output is

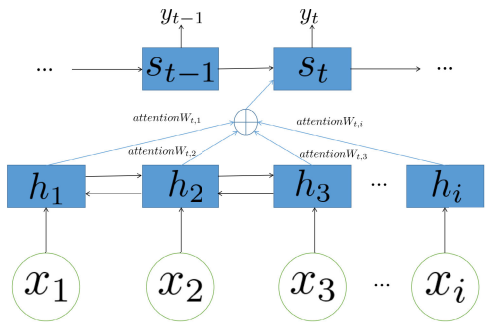

FIGURE 2. The sequence-to-sequence generation model based on an attention mechanism.

represented as $(h_1, h_2, \cdots, h_i)$. The first input word of the decoder is the <START> label. The decoder consists of a uni-directional long short-term memory(LSTM). In the training phase, the input of the decoder is the referenced headline. In the testing phase, the input of the model is the word generated from the last step.

The attentive distribution computation is the same as [2]. The equations are listed as equation 2 and equation 3, where *Score* is the function to get the attention feature between the target word and the source article words. At step t, the context representation is computed as equation 4. The *attentionW* is denoted as the attention weight. The $\boldsymbol{h}_m$ is the output of the encoder, where $\boldsymbol{h}_m \in \mathbb{R}^{d_e}$. The final distribution of the vocabulary at step t is computed as equation 5 and equation 6, where g is LSTM and f is constructed by LSTM and softmax. The $\boldsymbol{s}_t$ represents the current hidden states of the decoder, where $\boldsymbol{s}_t \in \mathbb{R}^{d_e}$. The symbol ; denotes the concat operation.

$$q_{tm} = Score\left(\boldsymbol{s}_t, \boldsymbol{h}_m\right) \tag{2}$$

$$attentionW_{tm} = \frac{\exp\left(q_{tm}\right)}{\sum_{k=1}^{i} \exp\left(q_{tk}\right)} \tag{3}$$

$$\boldsymbol{c}_t = \sum_{k=1}^{i} attentionW_{tk} \boldsymbol{h}_k \tag{4}$$

$$\boldsymbol{s}_t = g\left(\boldsymbol{s}_{t-1}, [\boldsymbol{y}_{t-1}; \boldsymbol{c}_{t-1}]\right) \tag{5}$$

$$p\left(\boldsymbol{y}_t \mid \boldsymbol{y}_1, \cdots, \boldsymbol{y}_{t-1}, \boldsymbol{X}\right) = f\left([\boldsymbol{y}_{t-1}; \boldsymbol{s}_t; \boldsymbol{c}_t]\right) \tag{6}$$

C. HEADLINE GENERATION BASED ON A GENERATIVE PRE-TRAINING MODEL

The transformer model proposed by [4] achieved state-of-the-art performance on machine translation. Some studies have employed the transformer on English single document summarization, such as [5] and [11]. However, there is no research employing generative pre-training model (GPT2) [6], [12] on the news headline generation task. Therefore, in this paper, we gain insight into news headline generation by using GPT2. Our whole model is showned as figure 3 and figure 4. The news headline generation model consists of four parts: the input module of the news headline generation model, the convolution operation on the output of the input module, the GPT2 decoder model and a pointer-generator

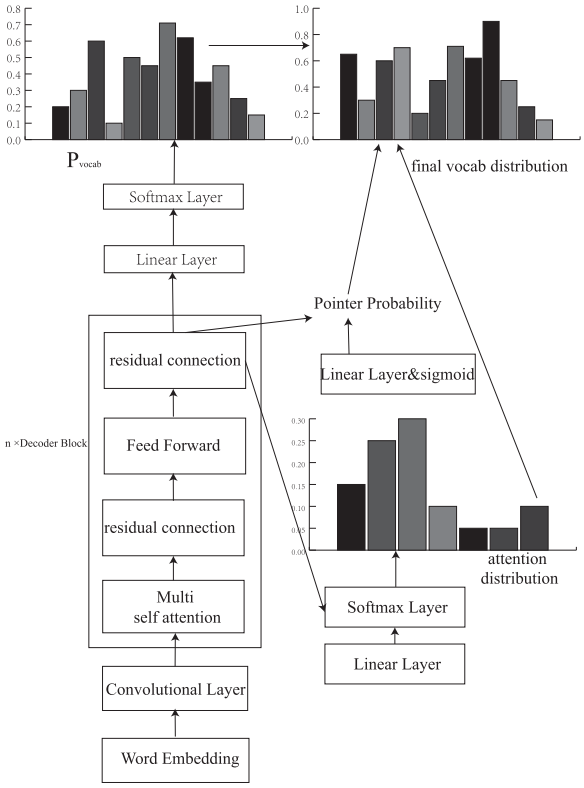

FIGURE 3. News headline generation model based on decoder only.

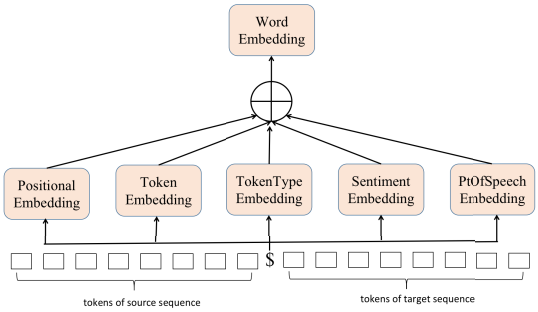

FIGURE 4. The input module of News headline generation model.

model. The input module is showed as figure 4. The output of the input module is the input of the news headline generation model which is showed as figure 3.

1) INPUT MODULE OF THE NEWS HEADLINE GENERATION MODEL

The input module is defined as equation (7) and equation (8). The input of news headline generation contains three parts: article words X, news headline words Y and a dilimiter character \$ between X and Y. We embed all of the input words into a low-dimensional real-valued vector space with embedding matrix $\mathbf{E} \in \mathbb{R}^{|V|\times d_e}$, where $|V|$ is the vocabulary size and d_e denotes the dimensionality of the words embeddings. We also embed the positional information into a low-dimensional real-valued vector space with a positional embedding matrix $\mathbf{Pos} \in \mathbb{R}^{len\times d_e}$, where *len* is the maximum length of the position and d_e denotes the dimensionality of positional embeddings. $\mathbf{T} \in \mathbb{R}^{4\times d_e}$ is the token type embedding matrix. $\mathbf{SI} \in \mathbb{R}^{3\times d_e}$ is the sentiment information embedding matrix. $\mathbf{PT} \in \mathbb{R}^{46\times d_e}$ is the part of speech feature. The $\mathbf{I}$ is the output of the input module and the input of the generation model. The $\mathbf{I} \in \mathbb{R}^{B\times L\times d_e}$ is defined as equation (8).

$$E(Z) = \{E(X); E(\$); E(Y)\} \quad \mathbf{E(Z)} \in \mathbb{R}^{B\times L\times d_e}, \tag{7}$$

where B is the batch size of the training data, L is the sequence length of the input and d_e is the dimension of the embedding representation.

$$\mathbf{I} = \mathbf{E}(Z) + \mathbf{Pos}(Z) + \mathbf{T}(Z) + \mathbf{SI}(Z) + \mathbf{PT}(Z) \tag{8}$$

2) DECODER BLOCK OF THE GENERATION MODEL

The decoder of the transformer is defined from equation (10) to equation (15). The *attention_out* is the result of the multihead attention, and H is the hidden state representation of the whole model. *FeedForward* is the one-dimensional convolution transformation.

$$\mathbf{I} = conv(\mathbf{I}) \quad \mathbf{I} \in \mathbb{R}^{B\times L\times d_e}, \tag{9}$$

where *conv* represents the convolution operation.

$$head_{\mathrm{i}} = \mathrm{Attention}(IW_i^{I_1}, I\ W_i^{I_2}, I\ W_i^{I_3}), \tag{10}$$

where $W_i^{I_1} \in \mathbb{R}^{d_e\times d_q}$, $W_i^{I_2} \in \mathbb{R}^{d_e\times d_k}$, $W_i^{I_3} \in \mathbb{R}^{d_e\times d_v}$.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{11}$$

where $Q = IW_i^{I_1}$, $K = IW_i^{I_2}$, $V = IW_i^{I_3}$.

$$\mathbf{attention\_out} = (head_1; \ldots; head_{\mathrm{h}})W, \tag{12}$$

where $W \in \mathbb{R}^{hd_v\times d_{\mathrm{e}}}$.

$$\mathbf{I_2} = \mathbf{I} + \mathbf{attention\_out} \tag{13}$$

$$\mathbf{I_3} = FeedForward(\mathbf{I_2}) \tag{14}$$

$$\mathbf{H_2} = \mathbf{I} + \mathbf{I_3} \quad \mathbf{H_2} \in \mathbb{R}^{B\times L\times d_e} \tag{15}$$

$$\mathbf{H} = \mathbf{H_2}[L_1 + 1 :] \quad \mathbf{H} \in \mathbb{R}^{B\times L_2\times d_e} \tag{16}$$

$$L = L_1 + L_2 + 1, \tag{17}$$

where L_1 is the length of the source document and L_2 is the length of the target document.

The multihead attention in the GPT2 model is differentiated from the transformer. One difference is the one-dimensional convolution of the attentive input, and the other difference is the concatenation of the current key and value with the past key and the value to avoid attenting to the repeat position of the input words.

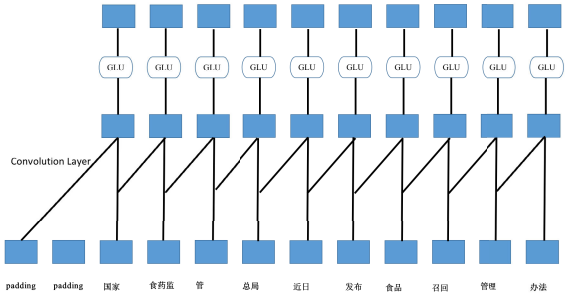

FIGURE 5. Words representations incorporating n-gram language information.

3) POINTER MECHANISM IN HEADLINE GENERATION MODEL

In the short text generation model, there will also be an out-of-vocabulary problem. To reduce the generation of out-of-vocabulary words, we employ the pointer-generator method used in [13]. We cannot use the pointer-generator model in our short text generation model, so we apply pointer mechanism defined from equation (18) to equation (21). To obtain the self-attention distribution, we transform the output hidden state H by a linear model and obtain the probability distribution by a softmax function, which is shown in equation (19). P_gen is the probability of choosing the word from the glossary, and $1 - P\_gen$ is the probability of choosing the word from the source news article.

$$\mathbf{P}_{vocab} = Softmax(\mathbf{W}_{\mathbf{p_{vocab}}}\mathbf{H} + \mathbf{b}_{\mathbf{p_{vocab}}}) \tag{18}$$
$$\mathbf{P}_{vocab} \in \mathbb{R}^{B \times vocab\_size} \tag{19}$$
$$\mathbf{P}_{attn} = Softmax(\mathbf{W}_{\mathbf{p_{attn}}}\mathbf{H} + \mathbf{b}_{\mathbf{p_{attn}}}) \tag{20}$$
$$\mathbf{P}_{attn} \in \mathbb{R}^{B \times extend\_size} \tag{21}$$
$$\mathbf{P}_{gen} = Sigmoid(\mathbf{W}_{\mathbf{p_{gen}}}\mathbf{H} + \mathbf{b}_{\mathbf{p_{gen}}}) \tag{22}$$
$$\mathbf{P}_{gen} \in \mathbb{R}^{B \times 1} \tag{23}$$
$$\mathbf{P}_{final} = \mathbf{P}_{gen} * \mathbf{P}_{vocab} + (1 - \mathbf{P}_{gen}) * \mathbf{P}_{attn}. \tag{24}$$
$$\mathbf{P}_{final} \in \mathbb{R}^{B \times extend\_vocab\_size} \tag{25}$$

4) DECODER INCORPORATING N-GRAM LANGUAGE INFORMATION

In the GPT2 decoder, the vocabulary probability is determined only by the hidden outputs of the decoder block. To obtain a more correct vocabulary distribution, we incorporate the language information into the input of the decoder block. We employ the 1-dimensional convolution method to obtain the 2-gram, 3-gram and 4-gram semantic information. The convolution layer is defined in figure 5. The three kernel sizes we use are 2, 3 and 4. Through the three different convolutional filters, we can obtain the whole words representations. When calculating the vocabulary distribution, we incorporate the language information into the hidden states.

IV. RELATED WORK

Neural headline generation conceptualizes the task as a sequence-to-sequence problem or an encode-decode problem. The encoder maps the source word sequence to a distributed representation, and the decoder generates the target headline word-by-word given the distributed representation of the source sequence and the previously generated target words.

The first work to apply the neural network to text summarization was [1]. The attention based text summary model [1] was enhanced by a recurrent neural network [3]. The work of [14] also employed the attentive sequence-to-sequence architecture. The encoder used in [14] was the bidirectional GRU-RNN [15], which incorporated the features of the part of speech, name entity and TF-IDF, and the decoder was the unidirectional GRU-RNN [15]. To solve the out-of-vocabulary problem, they also proposed a switching generator pointer model. Hu *et al*. [10] proposed a large-scale short text summarization dataset and conducted experiments on the dataset by using the attentive sequence-to-sequence model, but the experimental result was not good. To solve the problems of generating factual details inaccurately and repeatedly, [13] proposed using a pointer-generator network [16] and coverage mechanism to resolve these problems. To produce fluent summarization, [5] proposed to use the content selector to determine which part of the source document should be included in the summary. They conducted experiments by using bidirectional long short-term memory (BiLSTM) [17] and a transformer as the encoder and decoder. The work of [18] combined the abstractive method with extractive method and used a reinforcement learning method to bridge the nondifferentiable computation between these two methods. [19] first selected salient sentences by using reinforcement learning and then rewrote the selected sentence as the summary. The method of [20] used the convolutional sequence-to-sequence model and abstractive summarization achieved the state-of-the-art results in abstractive summarization field.

Recently, there are many researches in the field of the news headline generation [21]–[27]. The paper [21] presented a method for Nepali News Headline Generation. In the model they used the GRU as the encoder and the decoder. But they use the blue score as the evaluation criteria. Alexey and Ilya fine-tuned two pre-training transformer-based models for Russian news headline generation task [22]. In [23], they presented a Bengali news headline generation model based on RNN. The paper [24] presented a multiple headlines generation model and proposed a multi-source Transformer decoder. The paper [25] implemented a Myanmar news headline generation model based on LSTM. In paper [26], they proposed a model transformer(XL)-CC to generate headline and conducted experiments on the NYT datasets and Chinese LSCC news datasets. Wu *et al*. [27] proposed NewsBERT model on news recommendation dataset. The [11] showcased how pre-training bert model can be usefully applied in text

summarization. All the headline generation models talked above employ the traditional encoder-decoder architecture while our model attempts to use the decoder only. The generation model with the decoder only can achieve the comparable results with the encoder-decoder models. We conduct the experiments on both the English datasets and Chinese datasets.

A. DATASETS

We evaluate the proposed approach on a news dataset [10]. This dataset is a large corpus of Chinese short text summarization dataset constructed from the Chinese microblogging website Sina Weibo. This corpus consists of over 2 million real Chinese short texts with short summaries given by the author of each text. The dataset consists of three parts. The first part is the main content of LCSTS that contains 2,400,591 training pairs. These pairs can be used to train supervised learning model for summary generation. The second part is the validation set, which contains 10,666 validation pairs. The third part is the testing set, which contains 1,106 testing pairs. In our experiments, we only use part I and part III of the datasets. Because the dataset contains empty content, we preprocess the dataset. After preprocessing, the training dataset contains 2,081,212 training pairs and 1,106 testing pairs. The average length of the article is 59, and the average length of the target title is 9.

The second dataset we use is XSum developed by Narayan *et al*. [28]. The XSum dataset consists of BBC articles and accompanying single sentence summaries. This dataset is a one-sentence summary dataset. The XSum dataset covers a wide variety of domains, such as News, Politics, Sports, Weather, Business, Technology, Science, Health, Family, Education, Entertainment and Arts. The XSum dataset is not diverse. This dataset is a single news outlet and a uniform summarization style. The summaries in this dataset are written professionally. The average length of the articles is 378, and the average length of the target title is 8. This dataset has 204,045 training pairs, 11,332 validation pairs and 11,334 testing pairs. We use the natural language toolkit to preprocess the text and get the sentiment information and the part of speech information.

To train the model better, we employ the Jieba tokenizer to tokenize Chinese articles.

B. IMPLEMENTATION DETAILS

We train our model on one machine with one Tesla V100 GPU. For all experiments, our model has 768-dimensional hidden states and 768-dimensional word embeddings. During training and testing time, the source article on the LCSTS is truncated to 80 tokens and the target title is truncated to 10 tokens. During training and testing time, the source article on XSum is truncated to 200 tokens and the target title is truncated to 10 tokens. The optimizer we use is the Adam optimizer [29] with $\beta_1 = 0.9$, $\beta_2 = 0.998$ and $\epsilon = 10^{-9}$. The vocabulary size is set to 50,000 for both source and target. The learning rate of the optimizer is set to 0.0003. The heads of self-attention are set to 8, and we employ a 6-layer decoder block.

During training, the batch size is set to 64. We train the model for 10 epochs because the dataset is too large. In the decoding process, we use the beam search method as the decoding search method. The beam size in the beam search method is set to 4.

C. EVALUATION

We evaluate the performance of our model by using the ROUGE metrics. On the datasets, we use standard ROUGE-1, ROUGE-2, and ROUGE-L [30] on full length F1. Unigram and bigram overlaps assess the informativeness, and ROUGE-L represents the headline fluency.

V. RESULTS

Our experiments results are shown from Table 1 to Table 4. Experiments are performed on the LCSTS dataset and the XSum dataset. The testing set on LCSTS consists of 1,106 sentences and 1,106 headlines. The testing set on the XSum consists of 11,334 sentences and 11,334 headlines.

TABLE 1. Experimental results of headline generation on LCSTS.

Methods	Rouge1	Rouge2	RougeL
RNN(word) [10]	4.3	2.5	4.3
RNN(char) [10]	6.1	2.8	5.7
RNN-context(word) [10]	8.7	5.4	8.5
RNN-context(char) [10]	10.8	7.3	10.7
LSTM + pointer + coverage(word) [13]	20.6	7	18.9
LSTM + pointer + coverage(char) [13]	25.08	14.7	22.9
HG-News(word)	22.79	7.7	21.36

TABLE 2. Experimental results of headline generation with different encoders on LCSTS.

Methods	Rouge1	Rouge2	RougeL
Embedding-LSTM	17.7	4.4	15.8
LSTM-LSTM	20.6	7	18.9
Transformer-LSTM(1 Layer)	21	7.2	19.2

TABLE 3. Experimental results of headline generation with different encoders on XSUM.

Methods	Rouge1	Rouge2	RougeL
LSTM-LSTM	31.8	12.8	27.0
Transformer(1 Layer)-LSTM	35.89	15.39	31.70

On the LCSTS dataset, the baseline models we use include the RNN model [10] and RNN-context model [10]. The RNN model and RNN-context model experiments are conducted on the news headline generation dataset. The encoder of the RNN model is RNN, and the decoder of the RNN model is also a RNN. The RNN-context model uses the context as the input of the decoder. We re-implement the LSTM sequence to sequence generation model that incorporates the pointer mechanism and the coverage mechanism. The input of the model is the words or the characters. The experiment

TABLE 4. Experimental results of headline generation on XSUM.

Methods	Rouge1	Rouge2	RougeL
LEAD(in [28])	16.3	1.6	11.9
T-CONVS2S [28]	31.8	11.5	25.7
BertSumExtAbs [11]	38.81	16.50	31.27
Transformer-LSTM	34.5	15.2	31.0
HG-News(without conv, sentiment and partofspeech)	32.21	14.54	29.68
HG-News(with conv, without sentiment and partofspeech)	35.46	16.50	31.97
HG-News(without conv, with sentiment and partofspeech)	36.51	17.1	33.19
HG-News(with conv, sentiment and partofspeech)	37.19	17.46	33.71
LEAD(Our result)	35.8	13.7	28.8
MATCHSUM([31])	23.35	4.46	16.71

results are shown in Table 1. When the input is the characters, the vocabulary size is limited to 50,000. Our model is named HG-News. We only use the word-level input as the model input. In the HG-News model, we use the decoder only, and there is no encoder in our model. We also incorporate the pointer mechanism and n-gram language information into the HG-News model. Comparing to the pointer model [13], table 1 shows that the HG-news model achieves 2.19 points improvement on rouge-1 metric, 0.7 points improvement on rouge-2 metric and 2.46 points improvement on rouge-L metric. Because of the part of speech feature in the input data, the convolution layer in our model and the usage of the generative pre-training architecture incorporating the pointer mechanism, the news headline generation model we propose on LCSTS achieves comparable result with the other methods. Table 2 and table 3 show that the generation model with the transformer encoder is more effective than the generation models with other encoders.

On the XSum dataset, T-ConvS2S [28], which is a topic-aware convolutional variant of the sequence-to-sequence model, is one of our baseline models. The transformer(6-layer)-LSTM model significantly outperforms the T-ConvS2S model. The model BertSumExtAbs [11] is a model based on pre-training model bert. Our model doesn't beat the BertSumExtAbs model on the Rouge-1 metric but beat the BertSumExtAbs model on the Rouge-2 and Rouge-L metrics. We conduct experiments with convolution layer condition, sentiment information and the part of speech information. When using the input data with the sentiment information and the part of speech information, our model improves 1.73 points on the Rouge-1 metric, 0.96 points on Rouge-2 metric and 1.74 points on Rouge-L metric.

Table 2 shows the results with different encoders on LCSTS. Table 3 shows the results with different encoders on XSUM. Table 4 shows the results of our model on XSum compared with the baseline models. Table 2 and Table 3 show that the news headline generation model using the transformer as the encoder performs better than the news headline generation model with the embedding encoder and LSTM encoder. Table 4 shows that the news generation model based on the GPT2 model achieves comparable results compared with the baselines.

VI. ANALYSIS

A. COMPARISON WITH EXTRACTIVE SUMMARIZATION

It is clear from Table 4 that the abstractive headline generation model achieves much higher ROUGE scores than the extractive system. The headline generation model HG-News we propose and the transformer-LSTM generation model beat the strong extractive lead baseline. Given these observations, we can provide some explanations as follows.

First, the first sentence in the news article contains some important information, which is proven by the lead result shown in Table 4. We found that using the first 80 tokens of the article on the xsum dataset yielded significantly higher ROUGE scores than using the first 400 tokens of the news article. The transformer encoder performs well in the short text.

Second, the abstractive headline generation model can beat the extractive lead baseline and the extractive systems because the title of the article is much shorter than other summarizations. The average length of the title of the news article is 8. The title generated by the abstractive systems achieves much higher ROUGE-2 and ROUGE-L scores than the ROUGE-1 scores. The abstractive headline generation model can produce coherent words and sentences adhering to the article.

B. THE RELATION BETWEEN THE CHINESE HEADLINE GENERATION MODEL AND ENGLISH HEADLINE GENERATION MODEL

Recently, many studies have combined the extractive method with the abstractive method. The extractive method can remove the unimportant information of the whole document, and then the abstractive method paraphrases the important words. In our headline generation model, we only paraphrase the news document. In contrast to other abstractive summarization models, we only use the decoder to train our headline generation model, while others simultaneously contain the encoder and decoder.

We conduct experiments on two datasets: a Chinese news dataset and an English news dataset. The average length of the article is 59 and the average length of the target title is 9. The average length of the articles is 378, and the average length of the target title is 8. We find that the length of the news article on the XSum dataset is much larger than that on the LCSTS dataset and that the length of the title is nearly the same on both datasets. We need to process the document with Chinese word segmentation in Chinese articles. Then, the training processing of the Chinese text is the same as that of the English text. Based on the lengths of the news articles and news headlines, we encode the articles with different steps and generate the headlines with different word length. Figure 6 shows the headline examples generated by the transformer-LSTM model on the two datasets.

article on xsum	Poland's conservative Law and Justice party won enough votes in Sunday's parliamentary elections to govern alone, final results show.The party won 37.58% of the vote, giving it a majority in the lower house of 235 out of 460 seats.Civic Platform, which led Poland's coalition government for the last eight years, ...
headline on xsum	Poland's conservative Law and Justice party shares jump in Sunday's
reference headline on sum	Poland elections: Law and Justice party can govern alone
article on lcsts	受众在哪里，媒体就应该在哪里，媒体的体制、内容、技术就应该向哪里转变。媒体融合关键是以人为本，即满足大众的信息需求，为受众提供更优质的服务。这就要求媒体在融合发展的过程中，既注重技术创新，又注重用户体验。
headline on lcsts	媒体融合是以人为本
reference headline on lcsts	媒体融合关键是以人为本

FIGURE 6. Headline generation examples with transformer encoder.

Examples of system summaries are shown in figure 6. We randomly select 2 documents from the testing dataset and compare the two kinds of results on the two models.

C. THE ABSTRACTIVE LEVEL OF OUR HEADLINE GENERATION MODEL

We have shown that our headline generation model is effective on a Chinese news dataset and an English news dataset. However, does the ease of the decoder mechanism make our system any less abstractive?

Table 5 and Figure 7 show the proportion of novel n-grams that do not appear in the article for transformer-LSTM, T-CONVS2S and the HG-News model on the XSum test dataset. From Table 5, we can find that our model's titles and the transformer-LSTM titles contain a much lower rate of novel n-grams than the reference article titles, indicating a lower degree of abstraction. The gold result of the novel n-gram proportion is different from [28], which may be caused by the difference in the text preprocessing process. Our HG-News model has a comparable degree of abstraction, which proves that our model is effective. Our model is not comparable to T-CONVS2S because the result of T-CONVS2S is much more different from ours.

TABLE 5. Proportion of novel n-grams in summaries generated by various models on the XSum test set.

Methods	1-gram	2-gram	3-gram	4-gram
T-CONVS2S [28]	30.73	79.18	94.10	98.03
Transformer-LSTM	12.5	49.5	74.8	74.7
HG-News	9.7	61.4	85.2	83.4
GOLD(our result)	14.2	67.1	89.2	87.1
GOLD(in [28])	35.76	83.45	95.50	98.49

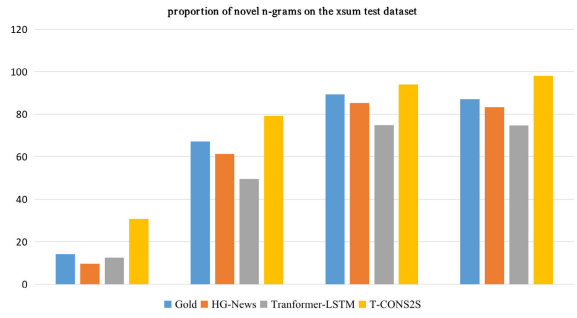

FIGURE 7. Proportion of novel n-grams on the xsum test dataset.

VII. CONCLUSION

We propose a news headline generation model in this paper. The generation model is no longer a framework with an encoder-decoder structure. Our generation model has a decoder only. The attention mechanism in our model is multi-head attention, which can obtain the semantic representation of the input tokens and obtain the attention distribution on the input tokens. In our news headline generation model, there is a rich feature input module which incorporates the sentiment feature and the part of speech feature into our model. We also present a pointer generation model to solve the out-of-vocabulary problem in the short text generation task. We also incorporate the n-gram language features into the hidden states. When generating a new word in the encoder-decoder model, the last token of the target words only focuses on the source tokens. In the model with decoder only, the current token of the target words cannot only focus on the source tokens but also focus on the generated tokens. The decoding process in our model is just like the human reading process which makes our model effective. The experimental results on the news headline generation datasets show that the model we propose achieves comparable results. However, there are also some problems in the news headline generation task; for example, the out-of-vocabulary problem cannot be avoided completely, and the word generated by the model sometimes is not correct. In the future, we will improve the capability of the feature representation and the accuracy probability of the word generation.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable feedback.

REFERENCES

[1] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 379–389.

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 5998–6008.

[3] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 93–98.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. U. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.

[5] S. Gehrmann, Y. Deng, and A. Rush, "Bottom-up abstractive summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4098–4109.

[6] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[7] 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 Associat. Comput. Linguist.*, 2019, pp. 4171–4186.
[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*. [Online]. Available: http://arxiv.org/abs/1907.11692
[9] S. Rothe, S. Narayan, and A. Severyn, "Leveraging pre-trained checkpoints for sequence generation tasks," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 264–280, Dec. 2020.
[10] B. Hu, Q. Chen, and F. Zhu, "LCSTS: A large scale Chinese short text summarization dataset," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1967–1972.
[11] 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)*, 2019, pp. 3728–3738.
[12] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," unpublished.
[13] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics (Long Papers)*, vol. 1, 2017, pp. 1073–1083.
[14] R. Nallapati, B. Zhou, C. dos Santos, C. Gulcehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn.*, 2016, pp. 280–290.
[15] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. Deep Learn. Represent. Learn. Workshop (NIPS)*, 2014, pp. 1–9.
[16] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Proc. 28th Int. Conf. (NIPS)*, 2015, pp. 2692–2700.
[17] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. 27th Adv. (NIPS)*, 2014, pp. 3104–3112.
[18] Y.-C. Chen and M. Bansal, "Fast abstractive summarization with reinforce-selected sentence rewriting," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (Long Papers)*, vol. 1, 2018, pp. 675–686.
[19] S. Bae, T. Kim, J. Kim, and S.-G. Lee, "Summary level training of sentence rewriting for abstractive summarization," in *Proc. 2nd Workshop New Frontiers Summarization*, 2019, pp. 10–20.
[20] Y. Zhang, D. Li, Y. Wang, Y. Fang, and W. Xiao, "Abstract text summarization with a convolutional Seq2seq model," *Appl. Sci.*, vol. 9, no. 8, p. 1665, Apr. 2019.
[21] K. Raj Mishra, J. Rathi, and J. Banjara, "Encoder decoder based nepali news headline generation," *Int. J. Comput. Appl.*, vol. 175, no. 20, pp. 1–4, Sep. 2020.
[22] A. Bukhtiyarov and I. Gusev, "Advances of transformer-based models for news headline generation," 2020, *arXiv:2007.05044*. [Online]. Available: http://arxiv.org/abs/2007.05044
[23] A. K. M. Masum, M. M. Islam, S. Abujar, A. K. Sorker, and S. A. Hossain, "Bengali news headline generation on the basis of sequence to sequence learning using bi-directional RNN," in *Soft Computing Techniques and Applications*. New York, NY, USA: Springer, 2021, pp. 491–501.
[24] D. Liu, Y. Gong, Y. Yan, J. Fu, B. Shao, D. Jiang, J. Lv, and N. Duan, "Diverse, controllable, and keyphrase-aware: A corpus and method for news multi-headline generation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 6241–6250.
[25] Y. Thu and W. P. Pa, "Myanmar news headline generation with sequence-to-sequence model," in *Proc. 23rd Conf. Oriental COCOSDA Int. Committee Co-Ordination Standardisation Speech Databases Assessment Techn. (O-COCOSDA)*, Nov. 2020, pp. 117–122.
[26] Y. Liao, K. Meng, J. Zhang, and G. Liu, "Unleashing the potential of attention model for news headline generation," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.
[27] C. Wu, F. Wu, Y. Yu, T. Qi, Y. Huang, and Q. Liu, "NewsBERT: Distilling pre-trained language model for intelligent news application," 2021, *arXiv:2102.04887*. [Online]. Available: http://arxiv.org/abs/2102.04887
[28] 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," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 1797–1807.
[29] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
[30] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proc. ACL Workshop*, 2004, pp. 74–81.
[31] M. Zhong, P. Liu, Y. Chen, D. Wang, X. Qiu, and X. Huang, "Extractive summarization as text matching," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6197–6208.

PING LI was born in Zhuzhou, Hunan, China, in 1989. She received the B.S. degree in software engineering from Xinjiang University, in 2011, and the M.S. degree in software engineering from Peking University, in 2014. From 2014 to 2017, she was a Lecture with Xinjiang Normal University. Her research interests include information extraction and natural language generation.

JIONG YU received the M.S. degree from the Key Laboratory of Materials Medication by Laser, Ion and Electron Beams, Ministry of Education, Dalian University of Technology, China, in 1995, and the Ph.D. degree from the School of Computer Science and Technology, Beijing Institute of Technology, China, in 2009. He was a Visiting Professor with the National Research Council of Canada, in 2003. He served as the Vice-Dean of the School of Computer Science, Beijing University of Technology, in 2005. He is currently working as a Full Professor with the School of Information Science and Engineering, Xinjiang University, and the Dean of the Graduate School of Xinjiang University, China. He has hosted several funded projects from the National Science Foundation of China (NSFC) and published papers in several international journals, including eight papers in SCI and six papers in ISTP. His research interests include high-performance computing, network security, and computer networks. He is a Senior Member of the China Computer Federation (CCF).

JIAYING CHEN received the B.E. degree from Northwest A and F University, in 2011, and the M.E. degree from Xinjiang University, in 2017. She is currently pursuing the Ph.D. degree with the School of Information Science and Engineering, Xinjiang University, China. Her current research interests include machine learning and recommender systems.

BINGLEI GUO received the bachelor's degree, the master's degree in software engineering, and the Ph.D. degree in computer application technology from Xinjiang University, China, in 2014, 2016, and 2020, respectively. She is currently a Lecturer with the School of Computer Engineering, Hubei University of Arts and Science. Her current research interests include distributed database systems and green computing.