IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received November 29, 2019, accepted January 1, 2020, date of publication January 10, 2020, date of current version January 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2965575

Text Summarization Method Based on Double Attention Pointer Network

ZHIXIN LI[1], ZHI PENG[1], SUQIN TANG[1], CANLONG ZHANG[1], AND HUIFANG MA[2]
[1]Guangxi Key Laboratory of Multi-Source Information Mining and Security, Guangxi Normal University, Guilin 541004, China
[2]College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China

Corresponding author: Zhixin Li (lizx@gxnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61966004, Grant 61663004, Grant 61967002, Grant 61762078, and Grant 61866004, in part by the Guangxi Natural Science Foundation under Grant 2016GXNSFAA380146, Grant 2017GXNSFAA198365, and Grant 2018GXNSFDA281009, in part by the Research Fund of Guangxi Key Lab of Multi-Source Information Mining and Security under Grant 16-A-03-02, Grant MIMS18-08, and Grant MIMS19-02, in part by the Guangxi Bagui Scholar Teams for Innovation and Research Project, and in part by the Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing.

ABSTRACT A good document summary should summarize the core content of the text. Research on automatic text summarization attempts to solve this problem. The encoder-decoder model is widely used in text summarization research. Soft attention is used to obtain the required contextual semantic information during decoding. However, due to the lack of access to the key features, the generated summary deviates from the core content. In this paper, we proposed an encoder-decoder model based on a double attention pointer network (DAPT). In DAPT, the self-attention mechanism collects key information from the encoder, the soft attention and the pointer network generate more coherent core content, and the fusion of both generates accurate and coherent summaries. In addition, the improved coverage mechanism is used to address the repetition problem and improve the quality of the generated summaries. Simultaneously, scheduled sampling and reinforcement learning (RL) are combined to generate new training methods to optimize the model. Experiments on the CNN/Daily Mail dataset and the LCSTS dataset show that our model performs as well as many state-of-the-art models. The experimental analysis shows that our model achieves higher summarization performance and reduces the occurrence of repetition.

INDEX TERMS Attention mechanism, neural networks, pointer network, text summarization.

I. INTRODUCTION

Automatic text summarization is a technology that has evolved to conform to the development of the information age. The information explosion has led to a rapid increase in the amount of available text. We need to refine and summarize this massive data content and generalize the main content of the user's attention so that users can quickly understand and browse the content. When browsing recent research, we find that existing text summarization techniques are mainly divided into two categories, i.e., the extractive methods and the abstractive methods. The extraction methods generate a summary by extracting important information from the document, while the abstract methods generate a summary by rewriting the content.

The emergence of modern neural networks, which mimics the mechanism of the human brain to explain the characteristics of data, has led to the development of abstractive summarization. In particular, the sequence-to-sequence model [1], [2] solves the sequence problem between text and summary. Then, the idea of the attention mechanism is incorporated into the model [3], and the result is better than the previous model under the non-neural network. In recent years, new technologies based on pointer networks and coverage mechanisms have emerged, ranging from pointer networks and coverage mechanisms to reinforcement learning (RL) and spawning networks, to the latest deep communication agents [4]–[7]. These improvements have led to a significant increase in the scores of the evaluation indicators. However, in the summaries generated by these models, there is still substantial room for improvement in terms of accuracy (focusing on the core content of the text) and reducing repetition.

The existing model training generally uses the ‘‘teacher forcing’’ algorithm [8] and the cross entropy loss function. Consequently, there is an ‘‘exposure bias’’ [9] between training and testing. During training, the input of each time step

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

comes from the reference word of the previous time step. During testing, however, the prediction of each word tested is based on the words generated by the model at the previous time step. Once a word is poorly generated, errors will accumulate, and the generation of subsequent words will deviate. Since the model uses the cross entropy loss function for training, the ROUGE [10] evaluation index is generally used to evaluate the quality of the generated statement in testing. One issue here is that the training loss and the evaluation index do not match. This ''teacher forcing'' algorithm and cross-entropy loss will cause the model to generate a summary of the same pattern as in the reference abstract. This is not as flexible as the ROUGE evaluation indicator in that the generated summary can be generated by arranging the vocabulary in other ways.

In this paper, we present a dual-attention pointer network (DAPT) model, using the self-attention mechanism to obtain the key features of the text from the encoder. In this process, the key information of the text will be well preserved. However, since the context of the currently predicted vocabulary is not considered, only the internal information of the text is concerned. Therefore, contextual semantic information will be lost. In the study of existing models, we found that the pointer network [4] can reproduce the details of the facts well, and generate more coherent and accurate summaries through the context vector and the attention pointer. To obtain contextual semantic information while acquiring key features, we use the gate mechanism to construct a dual-attention pointer network (DAPT) architecture. In addition, because the coverage mechanism's corrections to repetition problems are global, non-repeating parts are also affected. Therefore, we improve the coverage mechanism [4] to make the determination of repetition problems more accurate. To solve the ''exposure bias'' and training loss evaluation index mismatch problems, this paper uses the RL method to optimize the model. The two contributions of this paper are as follows:

- We propose a joint self-attention and pointer network generation mechanism. Soft attention and attention pointer were introduced to construct a dual-attention pointer network (DAPT) model. As a result, the summary generated by this model contains the key content of the text.
- We propose an improved coverage mechanism to address the repetition problems in the summary generated by the model, so that the summary become more readable and accurate.

In addition, we have done two valuable work:

- We apply the RL method to the evaluation metric optimization of the model, and combine the scheduled sampling to improve the quality and readability of the generated summary.
- We conduct a series of experiments on the CNN/Daily Mail dataset and the LCSTS dataset. The experimental results demonstrate the effectiveness of the proposed method

II. RELATED WORKS

A. ABSTRACTIVE SUMMARIZATION OF DEEP LEARNING

Deep learning is widely used in many research fields such as natural language processing and image processing [11]. In the abstractive summarization, Rush *et al.* [3] first used modern neural networks in the generation of text summarization. According to the sequence-to-sequence architecture [2], they use the convolution neural network(CNN) as the encoded part and employ the attention feedforward neural network between contexts to generate the abstract. Good results are achieved on the DUC-2004 and Gigaword datasets. On the basis of this work and the machine translation method proposed by Bahdanau *et al.* [12], Nallapati *et al.* [13] provided a seq2seq+attention baseline model and constructed a CNN/Daily Mail text summarization dataset. The task of evaluating multiple-sentence summaries provides data protection for a large number of related works in the future. Recently, given the prevalence of RL, Paulus *et al.* [5] proposed an improved attention mechanism and a training method for reinforcement learning. Pasunuru and Bansal [14] introduced RL into the traditional seq2seq+attention model and improved the performance of the model by using multiple reward methods. It should be noted that the soft attention mechanism is generally used in the traditional seq2seq+attention baseline model. When generating a word of the summary, the soft attention mechanism can focus on the location of the most relevant information set in the source text. In general, the soft attention mechanism picks out the most useful information from the source text for the currently generated word. However, the core content of the source text in the process of abstract generation cannot be ignored. So we have introduced a self-attention mechanism for this problem. The self-attention mechanism does not consider the generated words, only considers the information of the source text. Therefore, it can learn the word dependence within the source text, and capture the key information of the source text.

Gehring *et al.* [15] successfully applied CNN to text summarization, enabling models to calculate and discover structural information in sentences in parallel. Lin *et al.* [16] used CNN to propose a gated convolution unit to extract global information and reduce duplication. Zhao *et al.* [17] made a deep research on the problem of unstructured content in the abstractive meeting summarization, and proposed the adaptive segmental encoder networks, which made the abstractive meeting summarization get new progress. In addition, in order to avoid generating false facts in the summary, Cao *et al.* [18] use the open information extraction and dependency parse tools to extract the actual fact description from the source text. In the next work [19], they used existing summary as soft templates to supplement the input, and extended the seq2seq framework. In order to generate a more credible abstract, Yang *et al.* [20] proposed a new hybrid learning model using a hierarchical human-like strategy to simulate human processing of text summarization tasks.

B. POINTER NETWORK

The pointer network [21] is a variant of the seq2seq model. Instead of performing a sequence conversion, this network produces a series of pointers to the input sequence elements. The pointer network is applied to text summarization, mainly to solve the problem of sparse words and out-of-vocabulary words. When a model containing a pointer network generates a summary, it usually generates two probabilities, i.e., the probability of generation in the existing vocabulary and the probability of copying at the pointer. The CopyNet model proposed by Gu *et al.* [22] directly superimposes the generation probability and the copy probability. The switching generator-pointer model proposed by Nallapati *et al.* [13] and the pointer softmax model proposed by Gulcehre *et al.* [23] are independent of each other and do not attempt fusion. The method behind the pointer-generator network proposed by See *et al.* [4] uses a network to learn the weight between these two probabilities. The probability of weighted sums produces a better generated summary. The model achieved the most advanced results on the CNN/Daily Mail dataset in that year.

C. COVERAGE MECHANISM

The coverage mechanism first used in text summarization comes from a task in machine translation-addressing under-translation and over-translation problems [24]–[26]. Two main models, the coverage model [26] and the coverage penalty [25], are used to address these problems. The coverage model guides the attention model to focus on non-repeating words by covering the vector. The coverage penalty is used as a reranking method to select the less repetitive summary in the beam search process. See *et al.* [4] improved on the coverage model of Tu *et al.* [24] to address the repetitive problem in text summarization by setting a coverage vector whose value is the sum of the attention distributions computed by all previous prediction steps. The model has already paid attention to the words of the original text. Simultaneously, the loss function is used to punish repeated attention to reduce repetition. We have found that although this method achieves a reduction in repetition, the performance is unstable. Because the attention mechanism in the model is global, so when the loss function is punished, other non-mainly attention words will also receive a punishment. However, although the punishment is not large, it may interfere with the generation of other targets. Therefore, we use a truncation mechanism to improve the coverage mechanism, making the loss function more accurate, reducing the number of repetitions and increasing reliability.

D. REINFORCEMENT LEARNING

In the text summarization, the "exposure bias" and loss-evaluation mismatch problems common in the model can be solved by introducing the ROUGE indicator during training. However, because ROUGE is not microscopic, it cannot be directly optimized by backpropagation. The same problem exists in not only text summarization but also other sequence generation tasks because the evaluation indicators in these tasks are also not diminable; thus, people began to consider using RL as a solution. Ranzato *et al.* [9] first proposed a sequence training method based on RNNs. The RL algorithm was used to train various RNN-based sequence generation task models. However, they needed to train another linear regression model as the baseline improved solution. Addressing the shortcomings of early high-variance gradient estimations in the model, Rennie *et al.* [27] proposed a self-critical sequence training (SCST), which uses the sentences generated during testing as a training baseline to further improve the performance. Subsequently, Paulus *et al.* [5] introduced SCST to text abstracts and integrated the "teacher forcing" algorithm to improve the quality of the generated abstract while solving the "exposure bias" and loss-evaluation mismatch problems. Celikyilmaz *et al.* [7] also adopted SCST. However, unlike the summary rewards of Paulus *et al.* [5], they used sentence-based rewards as optimization goals.

III. MODEL

Our improved model is based on the sequence-to-sequence + attention model, as shown in Fig. 1. In the model, we built a bidirectional LSTM encoder for processing input text and a unidirectional LSTM decoder for outputting summaries. A self-attention mechanism, a soft attention mechanism and a pointer structure are also created in the model. The bidirectional LSTM encoders are shared and can generate key information or context information by matching a self-attention mechanism or a soft attention mechanism. The improved coverage mechanism will reduce repetition by participating in soft attention calculations.

A. SELF-ATTENTION

If the encoder encodes an excessive amount of much unimportant information, the generation of the summary may be affected during decoding, and the core content of the text cannot be obtained. Therefore, we need to highlight the salient features in the source text. This paper uses the self-attention mechanism [28], [29] to match the encoder with itself to dynamically collect key information in the text. The input sequence of the source text is converted to word embedding $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, and the bidirectional LSTM encoder is used for processing, thereby obtaining the sequence of the hidden states of the encoder. The hidden state $\boldsymbol{h}$ generated by connecting the two-way hidden state before and after, will participate in the calculation of the key information vector $\boldsymbol{z}$ as input:

$$\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n\} \tag{1}$$

$$f_i^j(\boldsymbol{h}_i, \boldsymbol{h}_j) = \boldsymbol{v}^T tanh(\boldsymbol{W}_1\boldsymbol{h}_i + \boldsymbol{W}_2\boldsymbol{h}_j) + b_{attn1} \tag{2}$$

$$\boldsymbol{e}_i = \sum\nolimits_n f_i^n; \boldsymbol{a}' = softmax(\boldsymbol{e}) \tag{3}$$

$$\boldsymbol{z} = \sum \boldsymbol{a}'\boldsymbol{H}^T \tag{4}$$

where $\boldsymbol{v}$, $\boldsymbol{W}_1$, $\boldsymbol{W}_2$ and b_{attn1} are learnable parameters, and $\boldsymbol{H}$ represents a collection of all hidden states $\boldsymbol{h}$. f_i^j is the

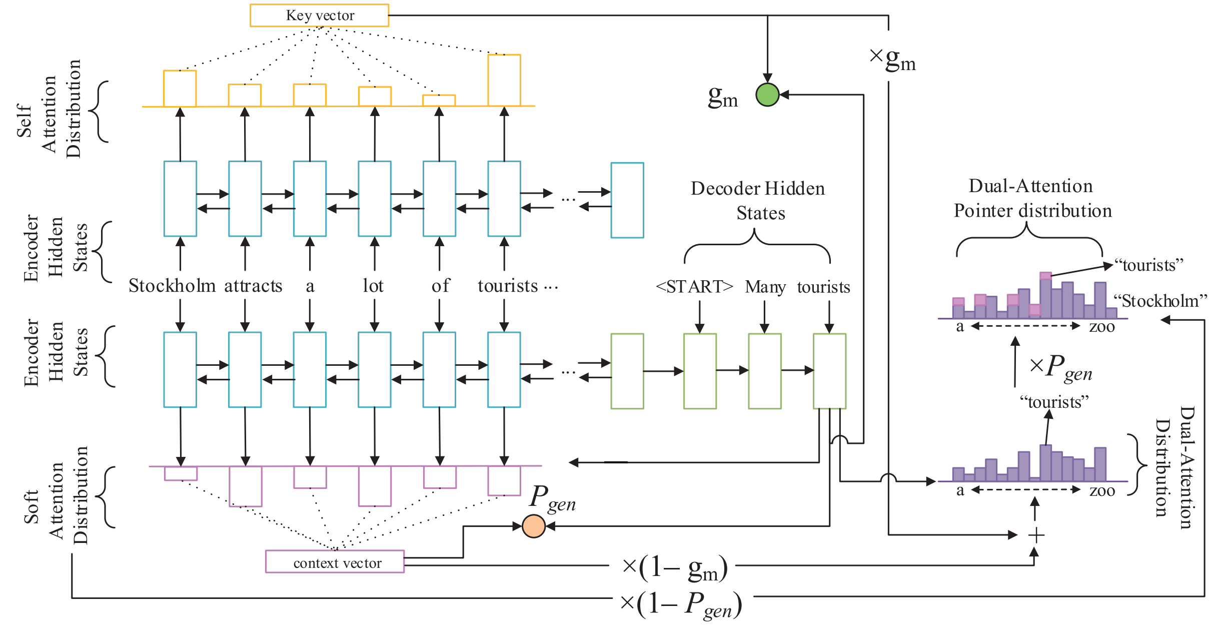


FIGURE 1. Dual-attention pointer network (DAPT) model, using the same encoder for different attention mechanisms.

most important part of the computation, representing the similarity between the i-th hidden state and the j-th hidden state. $\boldsymbol{e}_i$ represents the importance of the i-th hidden state, and finally, $\boldsymbol{a}'$ contains the attention weights of all hidden states, and can be used to determine key information according to the weight. Normalization using the softmax function makes the attention probability distribution of $\boldsymbol{a}'$ clear. We obtain the key information weight vector $\boldsymbol{a}'$, and then a weighted sum of the hidden state $\boldsymbol{H}$ is calculated according to the probability distribution provided by $\boldsymbol{a}'$ to obtain the key information vector z.

B. DUAL-ATTENTION POINTER NETWORK

The attention is fixed because the self-attention mechanism only focuses on the internal information of the text, which makes it impossible to generate a prediction with context semantics during decoding. Only repetitive and scattered words will appear in the generated summary. Therefore, we introduce the pointer network proposed by See *et al.* [4]. The pointer network is a variant of the sequence-to-sequence model with an attention mechanism. Instead of translating one sequence into another, the network produces a series of pointers to the elements of the input sequence. The soft attention mechanism [12] is combined with the self-attention mechanism. Through the context semantics used to generate the relationship between words, we supplement the missing context content and maintain the consistency of the generated summary. Moreover, if we encounter out-of-vocabulary words during decoding, we can point to and copy words according to the attention to improve the accuracy of the summary and alleviate the problem introduced by out-of-vocabulary words.

In pointer networks, the attention at each moment needs to tell the model which words in the text are more important in the prediction process of the decoder. Therefore, when generating the weight distribution at a certain moment, the soft attention adds the decoding state $\boldsymbol{s_t}$ of the current moment for calculation:

$$\boldsymbol{e}_i^t = \boldsymbol{v}_1{}^T tanh(\boldsymbol{W}_3\boldsymbol{h_i} + \boldsymbol{W}_4\boldsymbol{s}_t + b_{attn2}) \tag{5}$$

$$\boldsymbol{a}^t = softmax(\boldsymbol{e}^t) \tag{6}$$

$$\boldsymbol{c}_t = \sum\nolimits_i a_i^t \boldsymbol{h}_i \tag{7}$$

where $\boldsymbol{v}_1$, $\boldsymbol{W}_3$, $\boldsymbol{W}_4$ and b_{attn2} are learnable parameters, $\boldsymbol{a}^t$ is the attention distribution for the current moment, and $\boldsymbol{c}_t$ is the weighted sum of the hidden state of the encoder, which represents the content read from the text, and this vector is called the context vector.

The method of constructing a double attention network model is simple. We introduce a gate mechanism into the network to obtain the probability $g_m \in [0, 1]$ of key information required for each step in the decoding. This is calculated using the key information vector $\boldsymbol{z}$, the decoding state $\boldsymbol{s}_t$ and the decoder input $\boldsymbol{x}_d^t$:

$$g_m = \sigma(\boldsymbol{W}_z\boldsymbol{z} + \boldsymbol{W}_s\boldsymbol{s}_t + \boldsymbol{W}_x\boldsymbol{x}_d^t + b_{cor}) \tag{8}$$

in which the range of g_m is [0,1], where $\boldsymbol{W}_z$, $\boldsymbol{W}_s$, $\boldsymbol{W}_x$ and b_{cor} are learnable parameters, and σ is the sigmoid function. $\boldsymbol{g}_m$ is used to select the key information vector or the context vector when generating the probability distribution. The resulting distribution of the vocabulary $\boldsymbol{P}_{vocab}$ is as follows:

$$\boldsymbol{o}_t = (1 - g_m)\boldsymbol{c}_t + g_m\boldsymbol{z} \tag{9}$$

$$\boldsymbol{P}_{vocab} = softmax(\boldsymbol{V}_4(\boldsymbol{V}_3[\boldsymbol{s}_t, \boldsymbol{o}_t] + b_3) + b_4) \tag{10}$$

where $\boldsymbol{V}_3$, $\boldsymbol{V}_4$, b_3 and b_4 are learnable parameters. $\boldsymbol{o}_t$ is a mixture vector, and $\boldsymbol{P}_{vocab}(w)$ provides a probability distribution of all word w predictions in the vocabulary. To address out-of-vocabulary words, we need to point to and copy words off the vocabulary from the source text. Therefore, we re-introduce a gate mechanism to determine whether the word is generated or copied in the current step. Based on the mixed vector $\boldsymbol{o}_t$, the decoding state $\boldsymbol{s}_t$ and the decoded input $\boldsymbol{x}_d^t$ are added to participate in the calculation of the generation probability p_{gen}:

$$p_{gen} = \sigma(\boldsymbol{U}_o\boldsymbol{o}_t + \boldsymbol{U}_s\boldsymbol{s}_t + \boldsymbol{U}_x\boldsymbol{x}_d^t + b_{ptr}) \tag{11}$$

$$\boldsymbol{P}(w) = p_{gen}\boldsymbol{P}_{vocab}(w) + (1 - p_{gen})\sum\nolimits_{i:w_i=w} a_i^t \tag{12}$$

The range of the generation probability p_{gen} is [0,1], where $\boldsymbol{U}_o$, $\boldsymbol{U}_s$, $\boldsymbol{U}_x$ and b_{ptr} are learnable parameters, and σ is a sigmoid function. Finally, we use p_{gen} to choose whether to generate words from the vocabulary or copy words from the source text. At the prediction stage, a probability distribution $\boldsymbol{P}(w)$ is output, the dimension of which is the vocabulary length plus the number of words in the source text that do not appear in the vocabulary. The loss function of timestep t and the overall loss of the entire sequence are:

$$loss_t = -\log p(y_t) \tag{13}$$

$$L = \frac{1}{T}\sum\nolimits_{t=0}^{T} loss_t \tag{14}$$

C. IMPROVED COVERAGE MECHANISM

Repetition is a common problem in many natural language processing tasks and is a common problem for all generation models, especially in neural machine translation tasks, which often use sequence-to-sequence as a benchmark architecture [26], [30], [31]. Because the neural machine translation task is similar to the text summarization task and was developed earlier, the neural machine translation optimization model [4], [32] is used in many text summarization optimization schemes. Similar to See *et al.* [4], we adopt the coverage model proposed by Tu *et al.* [24] to solve the repetition problem. However, we improved it for better performance. In the process of generating the coverage vector, the weight of the existing attention is no longer completely copied, but filtered by adding the truncation parameter. Therefore, it can prevent the weight of unrepeated words in the coverage vector to be too high.

First, in the prediction process, a coverage vector $\boldsymbol{k}^t = \sum_{t'=0}^{t-1}\boldsymbol{a}^{t'}$ is maintained, which is the sum of all the attentions of the previous steps in the decoder. It should be noted that self-attention is not included in this calculation because it is fixed. $\boldsymbol{k}^t$ records which words the model has focused on in the original text, in the case where the predicted words are not repeated, the attention weight coefficients are different at different moments, and the coefficients can be stored separately in a vector. We let this coverage vector influence the attention computation of the current step:

$$\boldsymbol{e}_i^t = \boldsymbol{v}_1{}^T tanh(\boldsymbol{W}_3\boldsymbol{h}_i + \boldsymbol{W}_4\boldsymbol{s}_t + \boldsymbol{w}_k k_i^t + b_{attn2}) \tag{15}$$

where $\boldsymbol{w}_k$ is the learning parameter, the length of which being the same as $\boldsymbol{v}_1$. The goal is to tell the model what words it has previously focused on as it performs the current step of the attention calculations, therein hoping to avoid the situation of focusing consecutive attentions onto a few words.

Second, the coverage vector is improved. Because the attention mechanism in the decoder is global, to improve the accuracy of the coverage, the truncation parameter β is added to the coverage vector to filter other non-primary words of concern, and a minimum number ε is used to replace the weight of these words to avoid the influence on the generation of primary targets after continuous accumulation. The improved algorithm for calculating the coverage weights of the i-th hidden state is shown in Algorithm 1.

Algorithm 1 Calculation of the Coverage Weight of the i-th Hidden State

Input: previous i-th hidden state attention weight $\{a_i^0, a_i^1, \cdots, a_i^{t-1}\}$, ε, β.

Output: Coverage weight k_i^t for the i-th hidden state.

```
$k_i^t = 0$
for $t' = 0, 1, \cdots, t-1$ do
    if $a_i^{t'} \leq \beta$ then
        $k_i^t = k_i^t + \varepsilon$
    else
        $k_i^t = k_i^t + a_i^{t'}$
return $k_i^t$
```

To avoid repetition, we need an additional loss function to penalize repetitive attention:

$$covloss_t = \sum\nolimits_i \min(a_i^t, k_i^t) \tag{16}$$

where β is a hyperparameter, and $covloss_t \leq \sum_i a_i^t = 1$. In this paper, the improved coverage model possesses a truncation ability, and the truncation parameter is added to make the loss penalty mainly concern the word. We must remove the loss function before the DAPT model converges. Because the attention distribution of the main target cannot be clearly determined before convergence, the loss of participation will cause the attention under the coverage mechanism to affect the generation of the main target and reduce the overall performance. Finally, the old loss generates a new loss function through a hyperparameter λ:

$$loss_t = -\log p(y_t) + \lambda\sum\nolimits_i \min(a_i^t, k_i^t) \tag{17}$$

D. MIXED LEARNING OBJECTIVES

In the introduction, we described the "exposure bias" and loss-assessment mismatch in the model training. To solve such problems, we use mixed training objectives to optimize the model, including scheduled sampling and RL.

1) SCHEDULED SAMPLING

The training of the general LSTM decoder uses a "teacher forcing" algorithm, primarily by minimizing the maximum likelihood loss in each time step. The reason for the so-called teacher compulsion is that the input for each time step comes from the reference summary of the previous step. We define $y^* = \{y_1^*, y_2^*, \ldots, y_T^*\}$ as a reference summary for a given input sequence x. We then train the goals by minimizing the following losses:

$$L_{MLE} = -\sum_{t=1}^{T} \log p(y_t^* \left| y_1^*, y_2^*, \ldots, y_{t-1}^* \right., x) \quad (18)$$

To solve the problem of "exposure bias" caused by the "teacher forcing" algorithm, in Paulus *et al.* [5] proposed a effective method. In the decoder, the input for each time step is not fully from the reference word of the previous time step; rather, the word generated by the previous time step model is selected with a probability of 25%. This reduces the "exposure bias". However, the reduction of reference words in the target sequence during early training results in the model not being able to quickly navigate from a randomly initialized state to a reasonable state.

Therefore, we adopt the scheduled sampling method [8], which is an improved version of the "teacher forcing" algorithm. During training, the input of each time step of the model selects the reference word with probability q and then selects the output word of the previous time step of the model itself with probability $1 - q$. The value of q is variable, and initially, due to inadequate training of the model, the value of q is as large as possible. With the continuous training of the model, the value of q should be reduced, and the output words of the model itself should be selected such that the model can be kept as consistent as possible during training and testing. In this article, we use the linear decay method, $q_i = \max(q, l - mi)$. where q $(0 \leq q < 1)$ is the minimum true amount to the model, l and m provide the offset and slope of the change in q, and i represents the batch. Define $y^g = \{y_1^g, y_2^g, \ldots, y_T^g\}$ as the model output sequence (generated summary) for a given input sequence x. Then, the input y_t^{ss} of the t-th step of the model has a probability of q_i of choosing y_{t-1}^*, and the probability of $(1 - q_i)$ of choosing y_{t-1}^g. We train the goals by minimizing the following losses:

$$L_{MLE(ss)} = -\sum_{t=1}^{T} \log p(y_t^{ss} \left| y_1^{ss}, y_2^{ss}, \ldots, y_{t-1}^{ss} \right., x) \quad (19)$$

2) MIXED LOSS

Policy gradient is a basic algorithm in RL. We use policy gradients to minimize the negative reward expectations and directly optimize the non-differentiable ROUGE assessment indicators. The baseline DAPT model can be seen as an agent that interacts with the external environment, and the model generates words as operations taken by the agent. After generating the complete summary sequence y, it is compared with the reference summary sequence y^* to calculate the reward $r(y)$.

Our model uses the SCST training method [27], a self-critical sequential training method that uses the rewards received by the model under the generation method used in the test as a baseline. In each training iteration, for each input sequence x, two output sequences are generated in different ways: y^s, sampled from the probability distribution $p(y_t^s \left| y_1^s, y_2^s, \ldots, y_{t-1}^s \right., x)$ at each time step, and y^b, which is the baseline output and is obtained by performing a greedy search and selecting the word with the highest probability in the probability distribution $p(y_t^b \left| y_1^b, y_2^b, \ldots, y_{t-1}^b \right., x)$ of generating a summary at each time step. We minimize the following losses to train the objective:

$$L_{RL} = -(r(y^s) - r(y^b)) \sum_{t=1}^{T} \log p(y_t^s \left| y_1^s, y_2^s, \ldots, y_{t-1}^s \right., x) \quad (20)$$

Negative expectations are minimized in the formula to use gradient descent. When the sampled y^s yield a better return than the baseline, minimizing the loss corresponds to maximizing the conditional likelihood of the y^s, thereby increasing the overall return expectation of the baseline DAPT model.

Although the optimal training of RL solves the loss-evaluation mismatch problem, it may lead to reduced readability and fluency of the generated abstract. The readability of abstracts can be obtained by "teacher forcing", and here, we obtain the readability through scheduled sampling. Similar to Paulus *et al.* [5], we also use mixed training objectives to integrate scheduled sampling and RL so that the resulting abstracts remain readable and yield higher evaluation metrics:

$$L_{MIXED} = \gamma L_{RL} + (1 - \gamma) L_{MLE(ss)} \quad (21)$$

where γ is a hyperparameter that is used to adjust the loss of the two objective functions. In general, we pre-train the model and then use the hybrid loss for training optimization.

IV. EXPERIMENTS

A. DATASET AND METRIC

We conducted experiments on two datasets: LCSTS [33] and CNN/Daily Mail [13], [34]. LCSTS is a Chinese short text summary dataset that includes more than two million pieces of news data obtained via Weibo. Each piece of data is a statement pair that includes the original text and the corresponding summary. The original text is less than 140 words, and the reference summary has only one sentence and no more than 30 words. The data set is divided into three parts. PART I contains 2.4 million pieces of data, PART II contains 10,666 pieces of data, and PART III contains 1,106 pieces of data. Each statement pair in PART II and PART III is scored 1-5 by manual scoring, and the score is used to judge the degree of relevance of the short text to the abstract. As suggested by Hu *et al.* [33], we use PART I as the training set and the 725 statement pairs with scores above 3 (including 3 points) in PART III as the test set.

CNN/Daily Mail is a long paragraph summary dataset formed by collecting approximately one million news data. It contains online news articles (average of 781 tokens or approximately 40 sentences) and artificially generated summaries (average of 56 tokens or approximately 3.75 sentences). We obtained a non-anonymous version

TABLE 1. Data statistic after CNN/Daily Mail dataset preprocess.

Dataset	Train	Test	Val
Size	51.6M	1.3G	59.6M
Avg-ref	58.30	55.15	61.42
Avg-abs	777.27	791.7	768.29

of the data using the data processing method provided by See *et al.* [4], which included 287,226 training pairs, 13,368 verification pairs, and 11,490 test pairs. The statistics of the corpus are shown in Table 1, where Avg-ref is the average manual summary length, and Avg-abs is the average article length.

We evaluate the performance of different methods in abstract text summarization tasks through the ROUGE-N [10] and ROUGE-L [10] evaluation metrics. ROUGE-N is an n-gram recall between a generated summary and a set of reference summaries. In the field of text summarization, the values of N are generally 1 and 2. In this paper, ROUGE-1 and ROUGE-2 are used as evaluation criteria. ROUGE-L calculates the similarity between two sentences by the length of the generated sentence and the largest common subsequence of the reference sentence. The ROUGE scores are all based on F1 scores of ROUGE-1, ROUGE-2 and ROUGE-L.

B. EXPERIMENTAL SETUP

In training stage, the magnitude of the truncation parameter β is averaged by subtracting the minimum value from the maximum value of the attention weight. We also attempted higher values, as this affects the penalty for duplication, the value of λ is 1. The minimum true amount q in the scheduled sampling is 0.6, and the value of γ in the mixed training is 0.975.

The number of hidden units in the LSTM in both our encoder and decoder is set to 512. For the LCSTS dataset, to reduce the occurrence of word segmentation errors and out-of-vocabulary words, we use Chinese character-based methods to process source and target outputs. The vocabulary is limited to 4000 tokens, the size of the input article is limited to the first 140 tokens, and the digest is limited to 30 tokens. In the CNN/DailyMail data set we limit the input vocabulary and output decode layer to 50,000 tokens. We limit the size of the input article to the first 800 tokens, and the digest is limited to 100 tokens. The model was trained with cross-entropy loss using the Adam [35] optimizer with an initial learning rate of 1×10^{-3} and a batch size of 64. Finally, the training method of the mixed training is run on the trained model. At this stage, the learning rate is set to 3×10^{-4} and the batch size is 64. During the test, we set the beam size to 5 during the beam search.

C. BASELINES

1) LCSTS

In the experiments on the LCSTS dataset, because we used Chinese character-based methods to process the data, the results are compared mainly with advanced models of the same type.

- **RNN and RNN-context** [33] are based on the seq2seq model, where the difference is in the presence or absence of attention mechanisms.
- **CopyNet** [22] integrates the copy mechanism into the seq2seq model.
- **DRGN** [36] is based on the seq2seq+attention model and the VAE (variational auto-encoder) concept from the imaging field, and a deep loop generation decoder is proposed to capture the implicit structure information.
- **CGU** [16] adds a convolutional gated unit to the traditional seq2seq+attention model to control global information and reduce duplication.

2) CNN/DAILY MAIL

We compare the proposed model with six mainstream baselines. These models are abstract models, and a brief description of the model is as follows.

- **words-lvt2k-temp-att** [13] applies the seq2seq+ attention model and fuses the semantic features into the model. The CNN/Daily Mail dataset was also simultaneously proposed in that work.
- **graph-based attention** [37] introduces the graph-based attention mechanism based on the encoder-decoder framework and proposes a hierarchical decoding algorithm for improving the quality of summary generation.
- **pointer generator** [4] uses a pointer-generator network. When generating the summary, the model can also extract words from the original text to make the summary more accurate.
- **pointer generator + coverage** [4] adds a coverage mechanism based on the pointer generator model. It records the content that has been generated to reduce duplication.
- **ML+RL, with intra-attention** [5] includes an intra-attention model and uses the hybrid learning objective to train the model.
- **ML+RL ROUGE+Novel, with LM** [38] incorporates the pre-training language model into the decoder and adds new metrics to produce a new RL method.

D. RESULTS AND ANALYSIS

Tables 2 and 3 show the performance of the models on the CNN/Daily Mail and LCSTS datasets, respectively. The data in the tables show that the DAPT model outperforms the traditional baseline model in terms of ROUGE values, therein achieving the performance of mainstream methods. Among those methods, DAPT is a double attention pointer network, DAPT + imp-coverage is a double attention pointer network with improved coverage mechanism, and RL+MLE(ss) indicates that hybrid training is used for optimization. First, our DAPT model can obtain the key information of the source text and can obtain better semantic features such that the attention mechanism becomes more effective, which is very beneficial to the model. Simultaneously, the pointer network can effectively reduce the appearance of out-of-vocabulary words, which can further improve the quality of the generated

TABLE 2. Performance comparison with existing methods on the CNN/Daily Mail dataset.

Model	ROUGE-1	ROUGE-2	ROUGE-L
words-lvt2k-temp-att [13]	35.46	13.30	32.65
graph-based attention [37]	38.01	13.90	34.00
pointer generator [4]	36.44	15.66	33.42
pointer generator + coverage [4]	39.53	17.28	36.38
ML+RL, with intra-attention [5]	39.87	15.82	36.90
ML+RL ROUGE+Novel, with LM [38]	40.19	17.38	**37.52**
DAPT	36.86	16.01	33.66
DAPT + imp-coverage	40.26	17.62	36.61
pointer generator + coverage (RL+MLE(ss))	40.31	17.75	36.92
DAPT + imp-coverage (RL+MLE(ss))	**40.72**	**18.28**	37.35

TABLE 3. Performance comparison on LCSTS dataset datasets.

Model	ROUGE-1	ROUGE-2	ROUGE-L
RNN [33]	21.50	8.90	18.60
RNN-context [33]	29.90	17.40	27.20
CopyNet [22]	34.40	21.60	31.30
DRGN [36]	36.99	24.15	34.21
CGU [16]	39.40	26.90	36.50
pointer generator	38.91	25.16	34.75
pointer generator + coverage	39.37	25.98	36.04
pointer generator + coverage (RL+MLE(ss))	41.29	26.97	36.92
DAPT	37.27	25.32	34.97
DAPT + imp-coverage	39.53	26.11	36.26
DAPT + imp-coverage (RL+MLE(ss))	**41.52**	**27.28**	**37.05**

abstract. The addition of the coverage mechanism greatly reduces the number of repeated fragments in the generated sentence. As is shown in Fig. 2, we can observe that the problem of repetition is almost completely eliminated in the CNN/Daily Mail dataset. The improved coverage mechanism does not affect the generation of the primary target and improves the penalty effect.

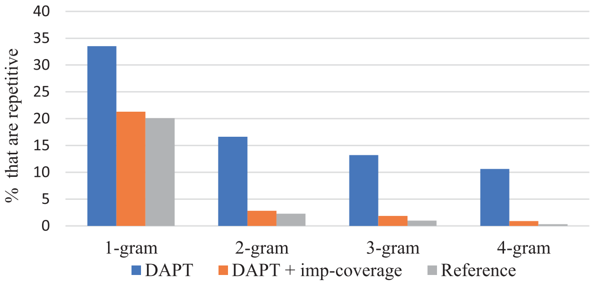


FIGURE 2. The repetition percentage in the summary, and the improvement in the coverage model in percent is similar to the reference summary(in the CNN/Daily Mail dataset).

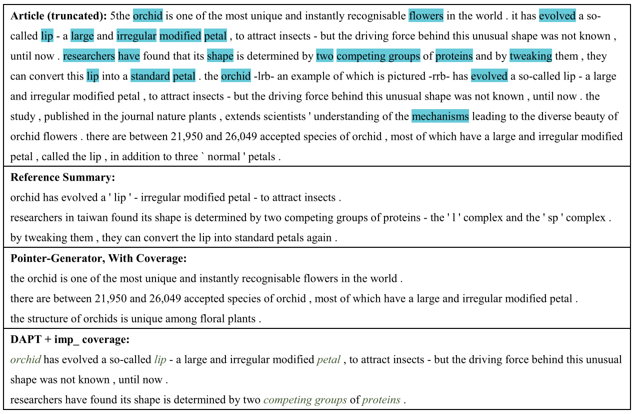
In order to better reflect the significance of model improvement, we added the experiment of pointer generator + coverage (RL+MLE(ss)) in the CNN/Daily Mail dataset, which is to add reinforcement learning to the model of See *et al.* [4]. At the same time, in the LCSTS dataset, the pointer generator model of See *et al.* [4] was tested. As can be seen in Tables 2 and 3, our improved model is superior to the original model in the two datasets, but our model does not improve much in the LCSTS dataset relative to the original model. Because the LCSTS dataset is a short text summary dataset, the encoder is more likely to get the main information, and the accumulation of the coverage vector is not excessive. Therefore, the effect of our model cannot be fully embodied.

Both the improved coverage mechanism and the traditional coverage mechanism can reduce repetitions in the summary, and this degree of reduction is similar; however, the improved coverage mechanism can better improve the quality of the summary. To better show this performance, based on DAPT, we compared the improved coverage mechanism with the traditional coverage mechanism. The results are shown in Table 4. We can see that the improved coverage mechanism yields a higher ROUGE score.

TABLE 4. Introducing a different coverage mechanism ROUGE score in DAPT (taking CNN/Daily Mail dataset as an example).

Model	ROUGE-1	ROUGE-2	ROUGE-L
DAPT	36.86	16.01	33.66
+ coverage	39.98	17.47	36.37
+ imp-coverage	**40.26**	**17.62**	**36.61**

After the model converges, we optimizes the model's ROUME evaluation metric by a hybrid training approach. This approach can improve the overall evaluation metric of the model by 2%-10%, which has a very large performance improvement for the model. At last, in Fig. 3 we present a summary generated by our model and compare it with the pointer-generator model and reference summary. The source text describes the cause of the orchid's lip formation. Obviously, the main point of the article is the orchid's lip petals. However, the summary generated by the pointer-generator model does not contain information about the lips but instead focuses on other trivial details. This is because too little attention is paid to key information in the model, and the



Article (truncated): 5the orchid is one of the most unique and instantly recognisable flowers in the world . it has evolved a so-called lip - a large and irregular modified petal , to attract insects - but the driving force behind this unusual shape was not known , until now . researchers have found that its shape is determined by two competing groups of proteins and by tweaking them , they can convert this lip into a standard petal . the orchid -lrb- an example of which is pictured -rrb- has evolved a so-called lip - a large and irregular modified petal , to attract insects - but the driving force behind this unusual shape was not known , until now . the study , published in the journal nature plants , extends scientists ' understanding of the mechanisms leading to the diverse beauty of orchid flowers . there are between 21,950 and 26,049 accepted species of orchid , most of which have a large and irregular modified petal , called the lip , in addition to three ` normal ' petals .

Reference Summary:

orchid has evolved a ' lip ' - irregular modified petal - to attract insects .

researchers in taiwan found its shape is determined by two competing groups of proteins - the ' l ' complex and the ' sp ' complex .

by tweaking them , they can convert the lip into standard petals again .

Pointer-Generator, With Coverage:

the orchid is one of the most unique and instantly recognisable flowers in the world .

there are between 21,950 and 26,049 accepted species of orchid , most of which have a large and irregular modified petal .

the structure of orchids is unique among floral plants .

DAPT + imp_ coverage:

orchid has evolved a so-called *lip* - a large and irregular modified *petal* , to attract insects - but the driving force behind this unusual shape was not known , until now .

researchers have found its shape is determined by two *competing groups* of *proteins* .

FIGURE 3. A qualitative example and comparison with the pointer-generator model; blue indicates the final value of the key information weight at the end of the final model generation summary (main part), and green italics represent the main part of the key information vector involved in the model.

unimproved coverage mechanisms can affect the generation of the main target. In contrast, the abstracts of our models are more coherent and contain more prominent information. Our model focuses on the lip rather than the just orchid, and it provides the most important identifying information. Because self-attention is responsible for selecting important information from the encoder output to improve the quality of the attention score, the trivial details can be reduced well.

V. CONCLUSION

In this paper, we propose a dual-attention pointer network (DAPT) for text summarization. The self-attention mechanism is introduced to obtain the key information of the source text, and the gate mechanism is combined to control the selection of information. Based on the existing coverage mechanism, we added truncation parameters to prevent this mechanism from interfering with the generation of other targets. In addition, this paper optimizes the evaluation metrics of the model by hybrid training, which improves the overall performance of the model without negatively impacting the readability of the generated abstract. The experimental results show that our method can generate a more accurate and consistent summary and has improved the ROUGE evaluation index compared with the traditional pointer-generator model.

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers for their comments and suggestions.

REFERENCES

[1] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1700–1709.

[2] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[3] 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.

[4] 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*, 2017, pp. 1073–1083.

[5] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," in *Proc. 6th Int. Conf. Learn. Represent.*, 2018.

[6] L. Liu, Y. Lu, M. Yang, Q. Qu, J. Zhu, and H. Li, "Generative adversarial network for abstractive text summarization," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 8109–8110.

[7] A. Celikyilmaz, 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.*, vol. 1, 2018, pp. 1662–1675.

[8] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1171–1179.

[9] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," in *Proc. 4th Int. Conf. Learn. Represent.*, 2016.

[10] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proc. 42nd Annu. Meeting Assoc. Comput. Linguistics Workshop*, 2004, pp. 74–81.

[11] H. Wei, Z. Li, C. Zhang, T. Zhou, and Y. Quan, "Image captioning based on sentence-level and word-level attention," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[12] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015.

[13] 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.

[14] R. Pasunuru and M. Bansal, "Multi-reward reinforced summarization with saliency and entailment," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 2, 2018, pp. 646–653.

[15] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1243–1252.

[16] J. Lin, X. Sun, S. Ma, and Q. Su, "Global encoding for abstractive summarization," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2018, pp. 163–169.

[17] Z. Zhao, H. Pan, C. Fan, Y. Liu, L. Li, M. Yang, and D. Cai, "Abstractive meeting summarization via hierarchical adaptive segmental network learning," in *Proc. World Wide Web Conf.-WWW*, 2019, pp. 3455–3461.

[18] Z. Cao, F. Wei, W. Li, and S. Li, "Faithful to the original: Fact aware neural abstractive summarization," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 4784–4791.

[19] Z. Cao, W. Li, S. Li, and F. Wei, "Retrieve, rerank and rewrite: Soft template based neural summarization," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 152–161.

[20] M. Yang, Q. Qu, W. Tu, Y. Shen, Z. Zhao, and X. Chen, "Exploring human-like reading strategy for abstractive text summarization," in *Proc. 33rd AAAI Conf. Artif. Intell.*, vol. 33, Aug. 2019, pp. 7362–7369.

[21] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2692–2700.

[22] J. Gu, Z. Lu, H. Li, and V. O. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1631–1640.

[23] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio, "Pointing the unknown words," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 140–149.

[24] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Modeling coverage for neural machine translation," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 76–85.

[25] Y. Wu *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," Sep. 2016, *arXiv:1609.08144*. [Online]. Available: https://arxiv.org/abs/1609.08144

[26] Z. Tu, Y. Liu, L. Shang, X. Liu, and H. Li, "Neural machine translation with reconstruction," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 3097–3103..

[27] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7008–7024.

[28] Z. Lin, M. Feng, C. N. D. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017.

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

[30] Y. Li, T. Xiao, Y. Li, Q. Wang, C. Xu, and J. Zhu, "A simple and effective approach to coverage-aware neural machine translation," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2018, pp. 292–297.

[31] H. Mi, B. Sankaran, Z. Wang, and A. Ittycheriah, "Coverage embedding models for neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 955–960.

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

[33] 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.

[34] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1693–1701.

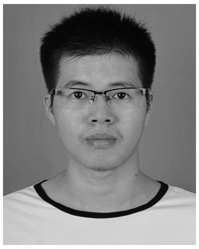
[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015.

[36] P. Li, W. Lam, L. Bing, and Z. Wang, "Deep recurrent generative decoder for abstractive text summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 2091–2100.

[37] J. Tan, X. Wan, and J. Xiao, "Abstractive document summarization with a graph-based attentional neural model," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1171–1181.

[38] W. Kryściński, R. Paulus, C. Xiong, and R. Socher, "Improving abstraction in text summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 1808–1817.

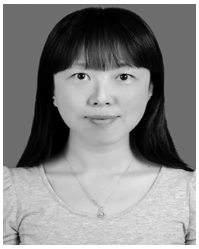
ZHIXIN LI received the Ph.D. degree in computer software and theory from the Institute of Computing Technology, Chinese Academy of Sciences, in 2010. He is currently a Professor with the College of Computer Science and Information Engineering, Guangxi Normal University. His research interests include image understanding, machine learning, and multimedia information retrieval. He has received the best doctoral dissertation award of Chinese Association of Artificial Intelligence, in 2011.



ZHI PENG received the B.E. degree from the Hunan Institute of Science and Technology, China, in 2017. He is currently pursuing the master's degree with the College of Computer Science and Information Engineering, Guangxi Normal University, China. His research interests include natural language processing and machine learning.

SUQIN TANG received the Ph.D. degree in information science and engineering from Central South University, China, in 2011. She is currently a Professor with the Department of Education, Guangxi Normal University. Her research interests include description logic and knowledge engineering.

CANLONG ZHANG received the Ph.D. degree in control technology and control engineering from Shanghai Jiao Tong University, China, in 2014. He was involved as an evaluation expert of science and technology project of Guangxi, in 2011. He is currently a Professor with the College of Computer Science and Information Engineering, Guangxi Normal University. His research interests include target tracking, pattern recognition, and multisensor data fusion.



HUIFANG MA received the B.E. degree from Northwest Normal University, China, in 2003, the M.S. degree from Beijing Normal University, China, in 2006, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2010. She is currently a Professor with the College of Computer Science and Engineering, Northwest Normal University, China. Her research interests include data mining and machine learning.