ATLANTIS PRESS
Advances in Engineering Research, volume 166
3rd International Conference on Automation, Mechanical Control and Computational Engineering (AMCCE 2018)

A Hierarchical Neural Abstractive Summarization with Self-Attention Mechanism

WeiJun Yang, ZhiCheng Tang AND XinHuai Tang*

ABSTRACT

Recently, the attentional seq2seq model had made a remarkable progress on the abstractive summarization. But most of these models do not considers the relation between original sentences, which is the important feature in extractive method. In this work, we proposed a Hierarchical Neural model to address problem. First, we use a self-attention to discovers the relation between original sentences. Secondly, we use a copy mechanism to solve the OOV problem. The experiment demonstrates that our model achieves state-of-the-art ROUGE scores on LCSTS dataset.

INTRODUCTION

Text summarization is a task about producing a short text from a document that contains the main information of the original text. According to the current implementations, it can be divided into extractive and abstractive. While the extractive method is forming summaries by copying parts of the original text, the abstractive method generates new phrases, possibly rephrasing or using words that were not in the original text. Since 2015, neural network models, based on the attentional seq2seq model, had made a remarkable progress on the abstractive summarization [1][2][3]. However, it also still challenged by: (1) Most these neural models just use the attention mechanism, which only considers the relation between original text and decode state, but not the relation between original sentences. (2) the out of vocabulary(OOV) problem, it means that some important token of the original text is not in the vocabulary.

To address the above challenge, some excellent methods have been proposed one after another. For example, COPYNET [4] and Pointer-Generator Networks(PGN) [5] solve the OOV problem by copying words that were in the original text. And Graph-Based Attention mechanism [6] discovers the relation between original sentences by using the adjacent matrix.

In this paper, we propose a hierarchical neural model with self-attention to expresses the salient information of sentences and discovers the relation between original sentences. Also, we use a copy mechanism that combining the ideas of COPYNET and PGN to solve the OOV problem. We show that our model achieves state-of-the-art ROUGE result on the LCSTS dataset [7].

WeiJun Yang, School of Software, Shanghai Jiao Tong University, Shanghai, China
ZhiCheng Tang, Department of Computer Science, Hohai University, Jiangsu, China
XinHuai Tang*, School of Software, Shanghai Jiao Tong University, Shanghai, China, tang-xh@cs.sjtu.edu.cn, corresponding author

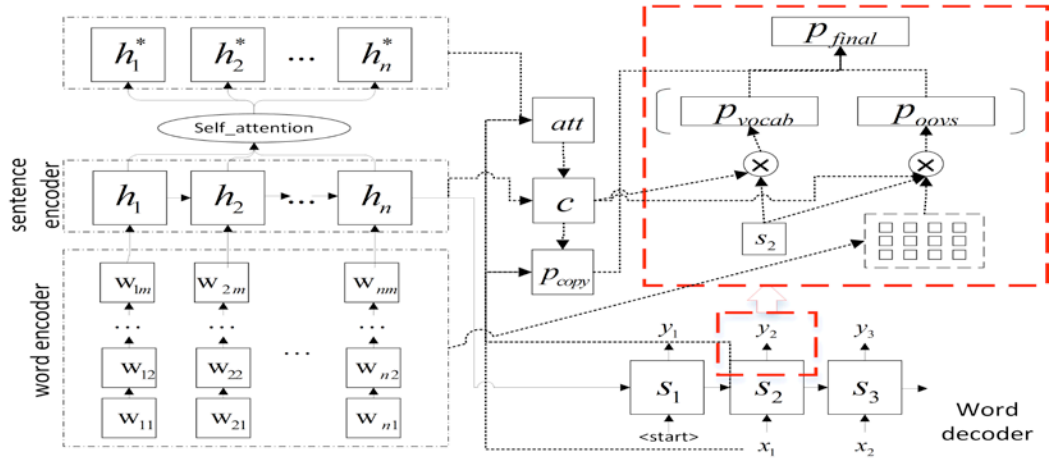

Figure 1. the framework of the Hierarchical neural model.

MODEL

In this section, we describe our neural model, whose base framework is similar to that of Tan et al. [6], and shown in Figure 1. The main difference is that our decoder is just a word decoder, but not a hierarchical decoder. Besides, we use a self-attention mechanism to discover the relation between original sentences, and present a copy mechanism to deal OOV words.

hierarchical encoder-decoder framework

Shown in Figure 1, we use a hierarchical encoder to map the original text to a token matrix W, a list of sentence vector and a test vector h. This hierarchical encoder consists of the word encoder and sentence encoder. In word encoder, each sentence of the original text is an input sequence and we use three layers of LSTM to encode token sequence into a sentence representation w_{im}, where i is the index of sentence. Then we sample the last word encoder output of each input sequence as the input of sentence encoder. In sentence encoder, we use one layer of LSTM to encode sentence representation to document representation h_n.

And in the decoder, we use three layers of LSTM as a word decoder. On each step t, the decoder receives x_{t-1} as input, where $x_{t-1}$is the representation of the preious word. But to address the OOV problem, in our model, x_{t-1} contains the word embedding and attentional sampling z_{word} of the preious word. The $z_{word}$is calculated by:

$$z_{word} = \begin{Bmatrix} \sum a_i * w_{ik}, w_{ik} = word \\ 0 \qquad\qquad , \text{ otherwise} \end{Bmatrix} \tag{1}$$

Then the decoder will decode the input x_{t-1} and the previous output s_{t-1}into s_t.

attention mechanism

Since the attention mechanism using in neural model, its main function usually is calculating the important score of each encode output according to decode state. But in

text summarization, the relation between original sentences is also the important feature, which is often used in extractive method. We need to know which sentence is the important or the summary sentence. And these sentences often play an important role in text summarization.

In our model, we use the Scaled Dot-Product Attention [8] to discovers the relation between original sentences, scilicet, to find out which sentence is an important one.

$$self\_attention(q_t, K, V) = \sum_{s=1}^{n} \frac{1}{Z} \exp(\frac{(q_t W_Q)(k_s W_K)^T}{\sqrt{d_k}})(v_s W_V)$$
$$h_i^* = self\_attention(h_i, H, H) \tag{2}$$

Here, we define $H = \{h_1, h_2, \cdots, h_n\}$ and d_k is the last dimension of K. Besides, the $W_Q \in R^{d_q \times d_q}$, $W_K \in R^{d_k \times d_k}$ and $W_V \in R^{d_v \times d_v}$ are learnable parameters. Then we caculate the temporary attention scores by LuongAttention [9].

$$a_{ti}^* = \frac{\exp(s_t^T M_a h_i^*)}{\sum_l \exp(s_t^T M_a h_l^*)} \tag{3}$$

where M_a is a learnable parameter. Because of our asymmetric framework, we apply an attentional forgotten gate that decides whether to keep the attention information(context) of the previous step. Then we compute the attention distribution and context by following equations:

$$f\_att_t = sigmoid(w_{att\_s} s_t + w_{att\_c} c_{t-1} + w_{att\_x} x_{t-1} + b_{f\_att}) \tag{4}$$

$$a_t = f\_att_t * a_{t-1} + (1 - f\_att_t) * a_t^* \tag{5}$$

$$c_t = \sum_i a_{ti} h_i \tag{6}$$

where w_{att_s}, w_{att_c}, w_{att_x} and b_{f_att} are learnable parameters.

prediction with copy mechanism

In this part, we assume a vocabulary V and use [unk] for any OOV word. To predict which token to generate, we should firstly calculate the final token distribution that contains the word of V and the OOV words. So similar to PGN, we also use a copying probability as a switch to choose between generating a token by sampling from V or copying a OOV token from the original source. So for each decode step t , the copying probability is computed by:

$$p\_copy_t = sigmoid(w_{copy\_s} s_t + w_{copy\_c} c_t + w_{copy\_x} x_{t-1} + b_{copy}) \tag{7}$$

Next, the vocabulary distribution is computed by:

$$p\_vocb_t = w_{vocb}(w_o[s_t, c_t] + b_o) + b_{vocb} \tag{8}$$

At the same time, to solve the OOV problem, we should know the the OOVs distribution. However, because the OOVs of each example is different, we should find a representation to express each OOV token. Here we use the Equation 1 to do this.Then we calculate the OOVs distribution by:

$$p\_oovs_t = tanh\left(z_{word} w_{oov}\right) * (w_o[s_t, c_t] + b_o), word \in OOVs \tag{9}$$

Finally, the probability distribution over the extended vocabulary is computed by:

$$p\_final_t = soft\max([p\_copy_t * p\_vocb_t, (1 - p\_copy_t) * p\_oovs_t]) \tag{10}$$

During training, the loss function of the step t is the negative log likelihood of the target word:

$$loss_t = -\log(p\_final_t(w)) \tag{11}$$

And the overall loss is the average value of $loss_t$.

EXPERIMENTS

dataset

The published LCSTS dataset [7] is a published Chinese short text summarization dataset, which collects the pairs of (short news, summary) from Sina Weibo. Similar to the setting of [7], we select 2.08 million pairs from Part I as the training set and 9k+ pairs from Part II and Part III, which scored ⩾3. And each short news of these pair contains no less than 2 sentences.

experimental result

In training phase, we set the size of vocabulary to 25000, the embedding dimension to 128 and the dimension of hidden value to 256. Then we evaluate our model by using ROUGE [10] and compare with following methods, including attentional RNN [7] and COPYNET [4].

The experimental result is shown in TABLE I, where (+C) expresses the model based on character, and (+W) expresses that based on word.

TABLE I. EXPERIMENTAL RESULT

Methods	ROUGE-1	ROUGE-2	ROUGE-L
Attentional RNN (+C)	29.9	17.4	27.2
Attentional RNN (+W)	26.8	16.1	24.1
COPYNET (+C)	34.4	21.6	31.3
COPYNET (+W)	35.0	**22.3**	32.0
Our Method	**35.4**	11.9	**33.3**

According to result, our method achieves the better ROUGE-1 and ROUGE-L scores, which shows our method captures more important information from the original text.

However, we get a lower ROUGE-2 score. This phenomenon may be due to two reasons. One is that the structure of the generated summary is different from the target summary. Other is our method can’t capture the order characteristics of the token in the original better, such as the Equation 1.

CONCLUSION AND FUTURE WORK

In this paper, we proposed a Hierarchical Neural model to discover the relation between original sentences and address the OOV problem. The experiment shows that our method achieves competitive performance on ROUGE-1 and ROUGE-L scores.

But the ROUGE-2 score also shows that our model still has many places to be optimized, such as capturing the order characteristics of the token in the original.

REFERENCES

1. Rush, A.M., Chopra, S. and Weston, J., 2015. A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685.
2. Nallapati, R., Zhou, B., Gulcehre, C. and Xiang, B., 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv preprint arXiv:1602.06023.
3. Paulus, R., Xiong, C. and Socher, R., 2017. A deep reinforced model for abstractive summarization. arXiv preprint arXiv:1705.04304.
4. Gu, J., Lu, Z., Li, H. and Li, V.O., 2016. Incorporating copying mechanism in sequence-to-sequence learning. arXiv preprint arXiv:1603.06393.
5. See, A., Liu, P.J. and Manning, C.D., 2017. Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368.
6. Tan, J., Wan, X. and Xiao, J., 2017. Abstractive document summarization with a graph-based attentional neural model. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 1171-1181).
7. Hu, B., Chen, Q. and Zhu, F., 2015. Lcsts: A large scale chinese short text summarization dataset. arXiv preprint arXiv:1506.05865.
8. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. In Advances in Neural Information Processing Systems (pp. 6000-6010).
9. Luong, M.T., Pham, H. and Manning, C.D., 2015. Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025.
10. Lin, C.Y., 2004. Rouge: A package for automatic evaluation of summaries. Text Summarization Branches Out.