An Automatic Abstractive Text Summarization Model based on Hybrid Attention Mechanism

Zhe Wang
1

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China

*Corresponding author’s e-mail: wang_zhe@bupt.edu.cn

Abstract. Attentional sequence-to-sequence models based on RNN have achieved promising performances in the automatic abstractive summarization technology. However, there are still some shortcomings, including the inaccuracies and lack of key information. In this paper, an abstractive text summarization model based on the hybrid attention mechanism is presented. The model aims at adopting sentence-level attention mechanism to guide the word-level attention distribution. Moreover, this model modulates the weight of sentence-level attention value to alleviate the adverse effect of high variance on word-level attention distribution for short documents. The experimental results on the LCSTS dataset show that the presented model can effectively improve the ROUGE scores and can better summarize the source document while retaining the important information. An example in this paper shows that the model can generate summaries that are similar to human-written ones.

1. Introduction

In recent years, the explosion of internet text information has expedited the rapid development of automatic text summarization technology. Text summarization is the task of automatically condensing a long text to a shorter version while retaining the important contents. Automatic summarization technology can aid many downstream applications, such as news digest, headline generation and recommendation. This task has received much more attention in the natural language processing community than before.

There are two broad types of summarization algorithms: extractive and abstractive. Extractive summarization methods commonly assemble the words, phrases or sentences from the source text, typically selecting one whole long sentence at a time. Most of the automatic summarization work focused on extractive summarization methods in early times. In the case of abstractive summarization methods, target summary may contain words or phrases that do not exist in the source text, which is closer to human way of thinking. It’s well accepted that extractive summarization methods always work better than abstractive summarization methods, the most important reason being that extractive methods have a better ability to retain the key information of source text in the condensing process. It inspires us that the extractive summarization methods can be used to guide the abstractive summarization methods.

Neural network based models have achieved well performances in the automatic text summarization task. Among the extractive methods, Nallapati et al. used an interpretable recurrent neural network to select the sentences in the source document[1]. Zhou et al. proposed a neural network framework with jointly learning to score and select sentences[2]. Zhang et al. proposed a latent variable extractive summarization model using a sentence compression model[3]. Liu et al. came up with an extractive model based on the pre-trained language model BERT[4]. On the other hand, abstractive approaches
also made some progress. See et al. proposed a pointer-generator model with coverage to reduce inaccuracies and repetition[5]. Gu et al. proposed a model incorporating a copying mechanism to handle out-of-vocabulary words[6]. Hsu et al. combined the extractive and abstractive summarization methods by using inconsistency loss[7]. Paulus et al. adopted the reinforcement learning technology to alleviate the exposure bias[8]. Gehrmann et al. presented a content selection model for abstractive text summarization[9].

In this paper, a new abstractive summarization model based on the hybrid attention mechanism is presented, which contains word-level and sentence-level attention. Our model adopts sentence-level attention mechanism to guide the word-level attention distribution, and then modulates the weight of sentence-level attention value to alleviate the adverse effect of high variance on the word-level attention distribution for short documents during decoding. Our model achieves the highest ROUGE scores on the LCSTS dataset.

2. Methodology

In this section, a hybrid-attentional model based on the encoder-decoder network is presented, in which two recurrent neural networks work together to transform one sequence to another. An encoder network condenses an input article into a vector, and a decoder network unfolds that vector into a summary. To improve upon this network, the most popular way is to use the attention mechanism to let decoder learn to focus on the specific range of source text. A large amount of evidence shows that the attention mechanism is efficient for NLP tasks.

![Architecture of encoder-decoder network with attention mechanism.](image)

2.1. Pointer-generator network

Pointer-generator network is a well-performed model based on encoder-decoder attentional network[5], which can generate a word from the fixed vocabulary or copy a word from the source text at each time step. The model contains a bi-directional LSTM encoder to compute hidden states $h_i$ from the embedding vectors of input words, and a unidirectional LSTM decoder to receive the current ground-truth word embedding vector to compute the hidden state $s_t$ at every time step, the word-level attention distribution $\alpha^t$ is calculated as follows:

$$e_i^t = v^T \tanh(W_h h_i + W_s s_t + b_w)$$  \hspace{1cm} (1)

$$\alpha^t = \text{softmax}(e^t)$$  \hspace{1cm} (2)
where \( v, W_h, W_s \) and \( b_w \) are learnable parameters. The word-level attention distribution is used to produce context vector \( c_t \), which is the weighted sum of encoder hidden states:

\[
c_t = \sum \alpha_i t h_i
\]  

(3)

The vocabulary probability distribution \( P_{vocab} \) is calculated by feeding the vector which concatenates decoder hidden state \( s_t \) and context vector \( c_t \) to two linear layers:

\[
P_{vocab} = \text{softmax}(V'(V[s_t, c_t] + b) + b')
\]  

(4)

To deal with the out-of-vocabulary words, pointer-generator network sets a generating-copying switch \( p_{gen} \) to choose between generating a word in fixed vocabulary by sampling from the vocabulary probability distribution \( P_{vocab} \) or copying a word in source text by sampling from the word-level attention distribution \( \alpha_t \). The final probability distribution \( P(w) \) is calculated as follows:

\[
p_{gen} = \sigma(w^T h_t + w^T s + w^T x + b_{ptr})
\]

(5)

\[
P(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum_{i, w_i = w} \alpha_t
\]

(6)

The overall loss for the whole sentence is the average negative log likelihood of the ground-truth target word \( w^* \) for each time step \( t \):

\[
\text{loss} = -\frac{1}{T} \sum_{t=0}^{T} \log P(w^*)
\]

(7)

2.2. Hybrid attention mechanism

Our model is partially inspired by a document classification model with hierarchical attention mechanism, which aggregates important words into sentence vectors and then aggregates important sentence vectors to document vectors. We would like to encourage the model to use dynamic sentence-level attention mechanism to increase the key word attention value and decrease the spurious word attention value on each time step. Slightly different from Yang et al.[10], to alleviate the adverse effect of excessive attention, we use a bidirectional LSTM to compute sentence-level hidden states \( h'_i \) from the average vectors of hidden states \( h_i \) rather than the weighted sum vectors, and then compute the dynamic sentence-level attention value through an attention layer. The dynamic sentence-level attention value \( \epsilon^T_t \) is calculated as follows:

\[
\epsilon^T_t = \nu^T \tanh(W' h_i + W_s s_t + b_s)
\]  

(8)

where \( \nu', W'_h, W_s \) and \( b_s \) are learnable parameters. We reuse \( W_s \) to maintain the consistency of the decoder hidden state component in both word-level and sentence-level attention distribution. In the decoding phase, we find that the sentence-level attention mechanism will bring high variance on the word-level attention distribution for the shorter texts, which causes the model to generate uninformative summaries. We set a parameter \( \lambda \) to modulate the sentence-level attention value \( \epsilon^T_t \) so that the model will pay more attention to key words for the shorter documents and informative sentences for the longer documents. Let \( l \) denote the number of sentences in a document, and \( C \) denote some constant related to the average of \( l \) in a corpus, \( \lambda \) is calculated as follows:

\[
\lambda = \begin{cases} e^{l-C}, & l < C \\ 1, & l \geq C \end{cases}
\]  

(9)

We add the weighted sentence-level attention value \( \lambda \epsilon^T_t \) to the word-level attention value \( e^T_t \) as a new hybrid attention value, and then compute the new word-level attention distribution \( \alpha^T_t \), changing equation (2) to:
\[ \alpha'_t = \text{softmax}(e^t + \lambda e^t) \]  

(10)

3. Results

In this section, we introduce the dataset and report the results of our experiments, and then analyse the reasons why our model works better. In the end, an example of summaries generated by a part of abstractive text summarization models is shown.

3.1. Dataset

We evaluate our approach on a large scale Chinese short text summarization dataset (LCSTS), which is collected from the Chinese microblogging website Sina Weibo. Each text in this dataset is paired with a human-written summary in Chinese. This dataset has 2400591 training pairs, 10666 evaluation pairs and 1106 test pairs. We follow Hu et al. and use Part I as the training set and the subset of Part III which is scored from 3 to 5 as the test set[11].

3.2. Experiments

We use PyTorch to carry out the experiments. We set the vocabulary size to 40k for both source document and target summary. Our model has 256-dimensional word embeddings and 512-dimensional hidden states. Our experiments are conducted with 1 NVIDIA 1080 Ti GPU. We use Adam as the optimizer and use early stopping to prevent overfitting. In the testing phase, we use beam search algorithm to generate the sequence of summary words, and the beam size is set to 4.

ROUGE is a commonly used metric for automatic text summarization. We use the \( F_5 \) scores for ROUGE-1, ROUGE-2 and ROUGE-L to evaluate the models. The higher ROUGE scores the model achieves, the higher quality summaries are generated. Comparing our model with the baseline models that Hu et al. used[11], the RNN model is based on a standard encoder-decoder architecture without any attention mechanism, and the RNN context model has a similar context vector as the one included in the attentional architecture shown in Figure 1. We also compare our model with the pointer-generator network proposed by See et al. and Copypet proposed by Gu et al.[5][6], both adopt word-level attention and copying mechanism. The results are shown in Table 1.
3.3. Discussion
As illustrated in Table 1, our model gets the highest ROUGE-1, ROUGE-2 and ROUGE-L scores, which means the summaries generated by our model are closer to the golden reference summaries than the other models. We can also make some interesting observations from the results: 1) the copying mechanism works well in handling out-of-vocabulary words; 2) attention mechanism can help the model to grasp the key information of source text; 3) hybrid attention mechanism works better than word-level attention mechanism only, one important reason for which is that we improve the ROUGE scores on the longer texts. As shown in Table 2, our model can retain the key information from the source text and produce a coherent, readable and concise summary, which proves that the hybrid attention mechanism is effective for the abstractive summarization task.

| Model            | ROUGE-1 | ROUGE-2 | ROUGE-L |
|------------------|---------|---------|---------|
| RNN-word         | 17.70   | 6.50    | 15.80   |
| RNN-char         | 21.50   | 8.90    | 18.60   |
| RNN context-word | 26.80   | 16.10   | 24.10   |
| RNN context-char | 29.90   | 17.40   | 27.20   |
| CopyNet-char     | 34.40   | 21.60   | 31.30   |
| CopyNet-word     | 35.00   | 22.30   | 32.00   |
| Pointer generator network | 37.61 | 25.53 | 36.26 |
| Our model        | 38.70   | 26.50   | 37.65   |

Table 2. An example of generated summaries (in English).

source article:
This paper summarizes ten design principles of wearable products, which are also the most attractive aspects of this industry in the author's opinion: 1. Solve the repetitive problems for people; 2. Start with people, not machines; 3. Ask for attention, but not try; 4. Improve the ability of users, not replace them.

reference summary:
Ten design principles for wearable technology

RNN-word:
10 things the startup should know, 10 things you must know, 5 things you must know, 5 things you must know

RNN context-word:
Which of the ten design principles of wearable products are you? The design principles of [UNK] You don't know how you do it? (appendix for design principles of products)

pointer-generator network:
Design principles for wearable technology

our model:
Ten design principles for wearable technology
4. Conclusions
In this paper, we presented an automatic abstractive text summarization model based on hybrid attention mechanism. We adopted the sentence-level attention mechanism to guide the word-level attention distribution, and made the generated summaries more readable and informative. We also got the highest ROUGE scores on the LCSTS dataset. For the future work, we will continue to focus on how to generate high-quality summaries.

References
[1] Nallapati, R., Zhai, F., Zhou, B. (2017) SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents. In: the Thirty-First AAAI Conference on Artificial Intelligence. San Francisco. pp. 3075-3081.
[2] Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., Zhao, T. (2018) Neural Document Summarization by Jointly Learning to Score and Select Sentences. In: the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne. pp. 654-663.
[3] Zhang, X., Lapata, M., Wei, F., Zhou, M. (2018) Neural Latent Extractive Document Summarization. In: the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels. pp. 779-784.
[4] Liu, Y., Lapata, M. (2019) Text Summarization with Pretrained Encoders. In: the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne. pp. 132-141.
[5] Paulus, R., Xiong, C., Socher, R. (2018) A Deep Reinforced Model for Abstractive Summarization. In: the 6th International Conference on Learning Representations. Vancouver.
[6] Gehrmann, S., Deng, Y., Rush, A. M. (2018) Bottom-Up Abstractive Summarization. In: the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels. pp. 4098-4109.
[7] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E. (2016) Hierarchical Attention Networks for Document Classification. In: the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego. pp. 1480-1489.
[8] Hu, B., Chen, Q., Zhu, F. (2015) LCSTS: A Large Scale Chinese Short Text Summarization Dataset. In: the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon. pp. 1967-1972.