Abstractive Document Summarization via Neural Model with Joint Attention

Liwei Hou[1], Po Hu[1], and Chao Bei[2]

[1]School of Computer Science, Central China Normal University, Wuhan 430079,China
[2]Global Tone Communication Technology Co., Ltd., Beijing 100043,China
houliwei@mails.ccnu.edu.cn
phu@mail.ccnu.edu.cn
beichao202@163.com

Abstract. Due to the difficulty of abstractive summarization, the great majority of past work on document summarization has been extractive, while the recent success of sequence-to-sequence framework has made abstractive summarization viable, in which a set of recurrent neural networks models based on attention encoder-decoder have achieved promising performance on short-text summarization tasks. Unfortunately, these attention encoder-decoder models often suffer from the undesirable shortcomings of generating repeated words or phrases and inability to deal with out-of-vocabulary words appropriately. To address these issues, in this work we propose to add an attention mechanism on output sequence to avoid repetitive contents and use the subword method to deal with the rare and unknown words. We applied our model to the public dataset provided by NLPCC 2017 shared task3. The evaluation results show that our system achieved the best ROUGE performance among all the participating teams and is also competitive with some state-of-the-art methods.

Keywords: Abstractive summarization; Attentional mechanism; Encoder-decoder framework; Neural Network

1 Introduction

Document summarization is a task of automatically generating a fluent and condensed summary for a document, while keeping the most important information of it as possible.

Efforts on document summarization can be roughly classified into two categories: extractive and abstractive method. Extractive methods usually generate a summary by simply selecting the most salient sentences from the original document and then directly concatenate them to compose the summary. This kind of summarization approach may produce a summary with good effectiveness and efficiency. However, the summary generated by them have some obvious drawbacks such as inevitable information redundancy within each summary sentence and higher incoherence across summary sentences. Furthermore, pure extractive way is also far from the method that human experts write summaries.

On the contrary, abstractive methods are expected to have more chance to generate better summaries by using more flexible expressions such as paraphrasing, compression or fusion with different words which do not belong to the original document. However, generating a high-quality abstractive summary is much more difficult in practice. Fortunately, the development of neural sequence-to-sequence techniques has made abstractive summarization approaches viable, and a set of recurrent neural network models based on attention encoder-decoder framework are becoming increasingly popular. It is worth noting that most of these models typically focus on summarizing documents with short input sequences and generate shorter summaries like news headlines or one-sentence summary. When generating a summary for a longer document, they usually suffer from the problems that we mentioned earlier such as the appearance of rare words and repeated phrases in the final summary.

In this paper, we present an encoder-decoder based neural abstractive model with joint attention for single document summarization task. And our contributions are as follows:

- We adopt the subword model to deal with the issue of rare and OOV words via segmenting Chinese words into subword units(more in Section 3.1), which has the advantage of simplifying the summarization process and reducing the training efforts at the same time with its accuracy as good as those using a large-vocabulary.
- We also add the attentional mechanism on the output sequence (more in Section 3.3) to address the issue of repeated phrases in the summary. By looking back at the previous decoding steps, this mechanism helps our model to make more structured prediction and avoid repeating the same content significantly.

2 Related Work

Document summarization has drawn much attention for a long time and has seen considerable progress over the years. Existing summarization systems are largely extractive by extracting a certain number of salient sentences from original document in verbatim to form the summary. They have traditionally employed linguistic and statistical features to rank sentences via the combination of unsupervised models (e.g. centroid-based method[1], graph-based method[2,3], LDA-based method[4]) or supervised models (e.g. SVR-based method[5], CRF-based method[6]) and diverse optimization strategies (e.g. integer linear programming [7], submodular function maximization[8,9]).

Various extractive summarization methods have been proposed and achieved the state-of-the-art performance. However, people tend to write a summary using their own words based on their understanding of the content and discourse-level semantics of the article, and abstractive summarization is closer to the way yet more challenging. The recent success of sequence-to-sequence frameworks has made abstractive summarization viable, in which a set of recurrent neural network

models based on attention encoder-decoder have achieved promising performance on short-text summarization tasks (Sumit et al.[10]; Wang and Ling[11]). To apply these models to more natural language processing tasks including summarization, machine translation and so on, word embedding(Pennington et al.[12]) is first used to convert each word to a fixed vector which can be considered as the inputs of these models. To make these models more scalable, attention mechanism(Bahdanau et al.[13]) is also applied into RNN encoder-decoder architecture to focus on different input information in different timesteps. However, these models often prevent themselves from learning good representations for those new words because of a fixed input and output vocabulary. In order to solve this problem, copy mechanism (Gu et al.[14]) and pointer (Gulcehre et al.[15]) are designed to reduce the rare and unknown words by locating a certain segment of input sentence and putting them into the output sequence appropriately. Although these mechanisms are good ways to solve this problem at a certain degree, they inevitably increase the complexity in both structure and time space, which will to be addressed in our current model.

Additionally, due to the challenges of compressing an original document in a lossy manner and preserving the key concepts of it in the summary, Ramesh et al.[16] proposes to use a hierarchical attention in the encoder model to consider key sentences as well as keywords at the same time. To produce higher quality summary via reducing repeated phrases of it, Paulus et al.[17] presents a deep reinforced model which combines intra-decoder attention and a reinforcement learning-based algorithm.

3 Our Method

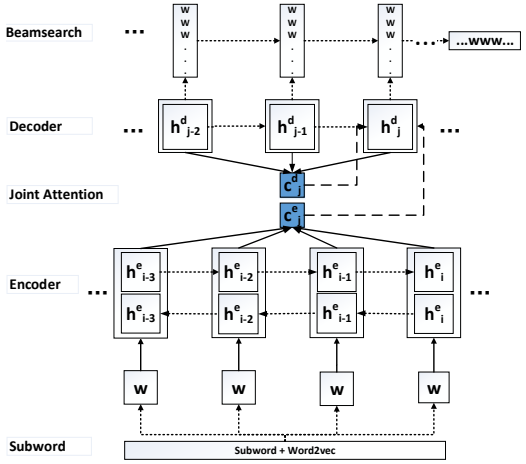

Fig. 1. System architecture

Existing abstractive summarization approaches often suffer from the disadvantages of generating repeated words and inability to deal with out-of-vocabulary

words appropriately. To address these, in this work we propose to add an attention mechanism on output sequence to avoid repetitive contents and use the subword method to deal with the rare and unknown words. We develop an abstractive neural summarization system with joint attention which consists of several important parts as shown in Figure 1: subword part, RNN encoder-decoder, joint attention. Next we will illustrate each part of our system in detail.

3.1 Subword Part

The main motivation is that some words of document can be recognized by a competent writer even if they are novel to him or her, based on subword units such as morphemes or phonemes. So we propose a hypothesis that a segmentation of rare words into appropriate subword units is sufficient to allow for the RNN encoder-decoder model to generate unknown words in abstractive summarization. And the evaluation results have shown that this hypothesis is feasible (more details will be explained in Section 4).

The subword method is initially proposed by Byte Pair Encoding(BPE) which is an effective data compression technique. Sennrich et al.[18] then adopted BPE for word segmentation in neural machine translation(NMT) task which they called subword translation and solved the problem of rare and unknown words. So in this paper, for the similar problem like rare and unknown words in the summary generation, we choose to apply this algorithm to our task.

Firstly, we preprocess all the documents with python word segmentation tool (i.e. jieba toolkit) and then segment each Chinese words into corresponding Chinese characters with symbol '**' in the end, such as '意想不到' to ('意**', '想**', '不**', '到'). Next, we iteratively count all the symbol pairs and replace each occurrence of the most frequent pair('A','B') with a new symbol 'AB', and for efficiency, we do not consider those pairs that cross Chinese words boundaries. So each merge operation will produce a new symbol which represents a character n-gram. In this way, it reduces a lot of redundant Chinese words. Before the subword method is used, the vocabulary size is 200,004 and after that it becomes 28,193. Actually, it not only can reduce the size of vocabulary, but also can enhance the ability of vocabulary to express some new words. Since it is used in our data preprocessing step, the complexities of the following model construction and training time will not increase. The results of the two segmentation methods are showed in Table1.

Table 1. Examples of two word segmentation methods

The result of jieba tool	The result of subword method
广受 关注 的 “ 瑞安 孕妇 重度 烧伤 ” 一事 有 了 新进展 ： 家属 称 ， 王芙蓉 现阶段 治疗 急需 型 血小板	广** 受 关注 的 “ 瑞安 孕妇 重度 烧伤 ” 一事 有 了 新** 进展 ： 家属 称 ， 王 芙** 蓉 现** 阶段 治疗 急需 型 血 ** 小** 板

3.2 RNN Encoder-Decoder

The RNN encoder-decoder framework is the fundamental for many current NLP models and is widely used in machine translation, dialog systems and automatic document summarization. The goal of this framework is to estimate the conditional probability $p(y_1, y_2,..., y_{t'} | x_1, x_2,..., x_t)$, where $\{x_1, x_2,..., x_t\}$ is an input sequence and $\{y_1, y_2,..., y_{t'}\}$ is the corresponding output sequence.

In practice, it is found that a gated RNN such as LSTM or GRU generally performs better than a basic RNN, and bidirectional gated RNN also perform better than unidirectional one. In our model, we adopt a bidirectional neural network with LSTM in encoder and a unidirectional neural network with LSTM in decoder.

Encoder. the LSTM-based encoder maps a set of input sequence vectors $X = \{x_1, x_2,..., x_t\}$ to a set of LSTM state vectors $H^e = \{h_1^e, h_2^e,..., h_t^e\}$:

$$h_t^e = f(x_t, h_{t-1}^e) \tag{1}$$

where f is the dynamic function of bidirectional LSTM. x_t is a 256 dimension vector training by word2vec. h_{t-1}^e is a 256 dimension vector produced by the bidirectional LSTM states.

The basic RNN encoder-decoder framework just uses a fixed size vector C called context vector to represent the input information:

$$C = \emptyset(\{h_1^e, h_2^e, \dots, h_t^e\}) \tag{2}$$

where $\emptyset$ integrates the bidirectional LSTM states H^e into a context vector C.

Decoder. The decoder LSTM is used to predict a target sequence $Y = (y_1, y_2,..., y_{t'})$ by unfolding the context vector C into a set of decoder state vectors $H^d = \{h_1^d, h_2^d, \dots, h_{t'}^d\}$ through the following dynamic function and prediction model:

$$h_{t'}^d = f(y_{t'-1}, h_{t'-1}^d, C) \tag{3}$$

$$p(y_{t'} | y_{<t'}, X) = g(y_{t'-1}, h_{t'}^d, C) \tag{4}$$

where $h_{t'}^d$ is the decoder LSTM state at t'-timestep, $y_{t'}$ is the predicted target symbol at t'-timestep and $y_{<t'}$ donates the history output $\{y_1, y_2,..., y_{t'-1}\}$. The prediction model is a typical classifier over a very large vocabulary with a softmax layer after the decoder LSTM. In our current model, the softmax layer produces beamsize words on each decoder timestep.

However, basic RNN encoder-decoder framework still has limitations, even if it is classic. Firstly, it requires a fixed length context vector C to act as the representation for the whole input sequence, which may make the information of C insufficient or make the dimension of C higher. Secondly, the semantic vector C

might lose the information of the whole sequence and the former information of inputs may be diluted or covered by the latter since all decoder timesteps share one common context vector C. In order to solve these problems, Bahdanau et al.[13] proposes an attention mechanism.

3.3 Joint Attention

The joint attention is a composed attention on both input sequence and output sequence. The attention on input sequence is used to store and deliver more complete information of input on each decoder timestep[13]. And the attention on output sequence is used to avoid repeated phrases by reviewing previous output information. Next, we will introduce these two mechanisms separately, and then the integration of them. The structure of joint attention in our system is showed in Figure 2.

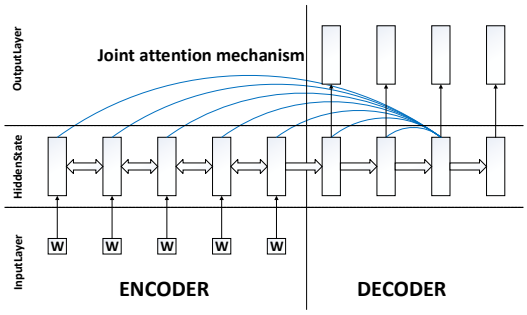

Fig. 2. Neural model of joint attention

Attention on input sequence . The attention mechanism is introduced here to solve the problems of basic RNN encoder-decoder framework mentioned in section 3.2. Instead of one vector C, our attention uses a set of dynamical changing context vectors $\{c_i, c_i, ..., c_{t'}\}$ in the decoding process. i.e.

$$e_{ij} = \eta(h_i^d, h_j^e) \tag{5}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{t'} \exp(e_{ik})} \tag{6}$$

$$c_i = \sum_{j=1}^{t'} \alpha_{ij} \, h_j^e \tag{7}$$

where η is the function that models the relationship between h_i^d and h_j^e, and it can be defined as a non-linear function. In our model, we first use 2-D convolution layers to extract features from all encoder states and then use a tanh function to get e_{ij}. The α_{ij} is an attention coefficient which indicates the contribution rate of the *i*-th original Chinese words to the generation of the *j*-th Chinese words.

Attention on output sequence. Although the attention on input sequence has solved the problems of information loss and insufficient of C which is common used in all decoder timesteps, these attention RNN encoder-decoder models still generate repeated words and phrases, because the output only depends on the latest decoder hidden states. In order to solve this problem, inspired by Paulus et al.[17] that considering the information of all input and previous outputs jointly may improve the performance, we combine the information about previously decoded outputs with the latest decoder state to generate a set of context vectors $\{ c_1^d, c_2^d, \dots, c_{t'}^d \}$.

Like above method, our model computes a new decoder context vector $c_{t'}^d$ on each decoding timestep. On the first timestep, we set the initial c_1^d to a vector of zeros. Then for $t' > 1$, we continue the method according to the following equations:

$$e_{t'j}^d = V^d * \tanh\left(h_{t'}^d W^d h_j^d\right) \tag{8}$$

$$\alpha_{t'j}^d = \frac{\exp\left(e_{t'j}^d\right)}{\sum_{k=1}^{t'-1} \exp\left(e_{t'k}^d\right)} \tag{9}$$

$$c_{t'}^d = \sum_{j=1}^{t'-1} \alpha_{t'j}^d h_j^d \tag{10}$$

First, we use tanh function to combine the previous decoder states $\{h_1^d, h_2^d, \dots, h_{t'-1}^d\}$ with the t'-timestep decoder state $h_{t'}^d$. Then we use a softmax layer to get the attention coefficients $\{\alpha_{t'1}^d, \alpha_{t'2}^d, \dots, \alpha_{t't'-1}^d\}$. Last, we take these attention coefficients and previous decoder states to obtain the output context vector $c_{t'}^d$.

Two attentions' combination. In order to get the probability of each output word, we first use a linear combination function to embed the two attentions into the t'-timestep decoder state and then use a softmax layer on it:

$$p(y_{t'}|y_{<t'}, X) = \text{softmax}(\text{linear}(h_{t'}^d, c_t^d, c_{t'})) \tag{11}$$

4 Experiments

4.1 Dataset

We conduct experiments on a public dataset provided by NLPCC2017 shared task3, which is a Chinese single document summarization task. The dataset includes 50,000 document-summary pairs for training and another 2,000 documents without corresponding summaries for testing. All data is provided by Toutiao.com. The length of the documents is between tens and tens of thousands Chinese characters and the length of the summaries is less than 60 Chinese characters.

4.2 Implementation

We convert the dataset into plain texts and save the news articles and summaries separately. First, we use subword model to process the data after conducting word segmentation by jieba[1] toolkit. Then, we retain 28,193 words in vocabulary and discard the words which are quite rare in nature. Next, we use pre-trained gensim[2] toolkit for the initialization of word vectors which will be further trained in our model. The dimension of all word vectors is 256 in this work.

We use tensorflow for implementation with one layer of bidirectional LSTM for the encoder and one layer of unidirectional LSTM for the decoder. The dimension of all hidden vectors are 128 and the batch size is set to 64 documents. Cross entropy is adopted to calculate the loss and Adam optimizer is used to optimize the loss. In the testing step, the beam size for word decoder is set equal to the batch size.

4.3 Evaluation

We adopt the widely used ROUGE-1.5.5 toolkit (Lin[19]) for evaluation. We first compare our model with various state-of-the-art extractive summarization methods provided by the PKUSUMSUM open source toolkit (Zhang et al.[20]) and UniAttention model which is a simple RNN seq2seq baseline model on 500 document-summary pairs of NLPCC2017 shared task3 dataset. The comparison results are shown in Table 2. In Table 3, we compare UniAttention model and our model on 2000 document-summary pairs. The results are the average scores of ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-4, ROUGE-L, ROUGE-SU4, ROUGE-W1.2, which are directly evaluated and provided by NLPCC2017 shared task3 organizers.

The results in Table 2 show that our proposed abstractive method outperforms traditional extractive methods considerably. It also shows superiority over some extractive methods in short summary generating task. The results in Tables2 and 3 show that our method has improvement over the neural abstractive baseline. And the problem of repeated phrases has a significant improvement.

Table 2. Comparison results on 500 document-summary pairs using F-measure of ROUGE

Method	Rouge-1	Rouge-2	Rouge-3	Rouge-4	Rouge-L
LexPageRank	0.23634	0.10884	0.05892	0.03880	0.17578
MEAD	0.28674	0.14872	0.08761	0.06124	0.22365
Submodular	0.29704	0.15283	0.08917	0.06254	0.21668
UniAttention	0.33727	0.20032	0.13176	0.10142	0.29440
Our Model	**0.34949**	**0.21172**	**0.14497**	**0.11282**	**0.30662**

[1] https://pypi.python.org/pypi/jieba/

[2] http://radimrehurek.com/gensim/

Table 3. Comparison results on more than 500 document-summary pairs: the score is provided by the track organizers of NLPCC2017 shared task3

Method	Average of ROUGE-1, 2, 3, 4, L, SU4, W-1.2
UniAttention	0.20072
Our Model	**0.22703**

Furthermore, the evaluation results of NLPCC2017 shared task3 are shown in Table4. The test data have 2000 document-summary pairs, and the scores are also the average scores of ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-4, ROUGE-L, ROUGE-SU4, ROUGE-W1.2. Each team is permitted to submit two version per day between June 5 and June 7, 2017. Table 4 shows the official evaluation results of our model and the peers and shows that our approach achieved the best performance in all participating teams. In summary, our model has a good effectiveness and performance in abstractive single document summarization.

Table 4. Official ROUGE evaluation results for the formal runs of all participating teams

Method	Average of ROUGE-1, 2, 3, 4, L, SU4, W-1.2
Our Model(NLP_ONE)	**0.22102**
ICDD_Mango	0.22093
NLP@WUST	0.21648
CQUT_AC326	0.19138
HIT_ITNLP_TS	0.19133
DLUT_NLPer	0.17537
AC_Team	0.17090
ECNU_BUAA	0.15988
ccnuSYS	0.15790

5 Conclusion

In this paper we tackle the challenging task of abstractive document summarization, which is still less investigated to date and very challenging. We study the difficulty of the existing attention RNN encoder-decoder summarizers, and address the need of producing rare or new words and reduce repeated phrases in the final summary. We adopt subword units and joint attention mechanism to improve the performance of traditional models. Extensive experiments have verified the effectiveness of our proposed approach. Our method also achieved the best performance in the competition of single document summarization track held by NLPCC2017. There is still lots of future work to do. An appealing direction is to combine keywords and key sentences in this neural abstractive model or investigate the neural abstractive method on the multi-document summarization task.

Acknowledgments

This work was supported by the National Natural Science Foundation of China

(No. 61402191), the Specific Funding for Education Science Research by Self-determined Research Funds of CCNU from the Colleges' Basic Research and Operation of MOE (No. CCNU16JYKX15), and the Thirteen Five-year Research Planning Project of National Language Committee (No.WT135-11). We also thank Zhiwen Xie for helpful discussion. Po Hu is the corresponding author.

References

1. Dragomir R. Radev, Hongyan Jing, Małgorzata Stys, and Daniel Tam. 2004. Centroid-based Summarization of Multiple Documents. *Information Processing & Management* 40(6): 919-938.
2. Günes Erkan, and Dragomir R. Radev. LexPageRank: Prestige in Multi-Document Text Summarization. *EMNLP* 2004.
3. Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. Manifold-Ranking Based Topic-Focused Multi-Document Summarization. *IJCAI* 2007.
4. Ivan Titov, and Ryan McDonald. A Joint Model of Text and Aspect Ratings for Sentiment Summarization. *ACL* 2008.
5. Sujian Li, Ouyang You, Wei Wang, and Bin Sun. Multi-document Summarization Using Support Vector Regression. *DUC* 2007.
6. Hitoshi Nishikawa, Kazuho Arita, Katsumi Tanaka, Tsutomu Hirao, Toshiro Makino, and Yoshihiro Matsuo. Learning to Generate Coherent Summary with Discriminative Hidden Semi-Markov Model. *COLING* 2014.
7. Dan Gillick, and Benoit Favre. A Scalable Global Model for Summarization. *ACL* 2009.
8. Jingxuan Li, Lei Li, and Tao Li. 2012. Multi-document Summarization via Submodularity. *Applied Intelligence* 37(3): 420-430.
9. Hui Lin, and Jeff Bilmes. Multi-document Summarization via Budgeted Maximization of Submodular Functions. *NAACL* 2010.
10. Sumit Chopra, Michael Auli, and Alexander M. Rush. Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. *NAACL* 2016.
11. Lu Wang, and Wang Ling. Neural Network-Based Abstract Generation for Opinions and Arguments. *NAACL* 2016.
12. Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global Vectors for Word Representation. *EMNLP* 2014.
13. Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv preprint arXiv:1409.0473*.
14. Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. Incorporating Copying Mechanism in Sequence-To-Sequence Learning. *ACL* 2016.
15. Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the Unknown Words. *arXiv preprint arXiv:1603.08148*.
16. Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. Abstractive Text Summarization Using Sequence-to-Sequence Rnns and Beyond. *CoNLL* 2016.
17. Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A Deep Reinforced Model for Abstractive Summarization. *arXiv preprint arXiv:1705.04304*.
18. Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. *ACL* 2016.
19. Chin-Yew Lin. Rouge: A Package for Automatic Evaluation of Summaries. *ACL* 2004.
20. Jianmin Zhang, Tianming Wang, and Xiaojun Wan. PKUSUMSUM: A Java Platform for Multilingual Document Summarization. *COLING* 2016.