Study on Academic Documents –Oriented Automatic Summarization of Short Texts

Chunxiao Gao¹, Bo Guan¹, Xiaoyue Zhang¹, Hao Liu¹,² and Zhiqiang Wei¹,²,*

¹Department of Computer Science and Technology, Ocean University of China, Qingdao, China
²Pilot National Laboratory for Marine Science and Technology(Qingdao), Qingdao, China

*Corresponding author e-mail: weizhiqiang@ouc.edu.cn

Abstract. Traditional automatic text summarization relies heavily on the original text information, and the extensibility is limited. However, generation-style abstractive methods attempt to generate the corresponding summarization by understanding the original semantics. We set out to set up a sequence-to-sequence model for academic document summarization generation. For purpose of reducing the detail loss of input sequence information, we put forward the attention mechanism to assign the weight of each input word. We trained this model on Chinese literature data set. It generated a reliable document summary. Our test shows that the approach has good adaptability to Chinese academic literature and has good performance in text summarization.

1. Introduction
Automatic document summarization technology abstracts and refines the text for the user through the computer, and provides the general information of the text. Automatic text summarization is to automatically summarize the main content of a document without changing the original idea of the document. There are many scenarios for automatic summarization, such as the generation of news headlines, the generation of scientific and technical literature summaries, the generation of search results fragments, and the reviews of commodity. In the Information explosion era of big data, using short text to express the main content of information, and it will undoubtedly help us ease the problem of information overload. Extracting the key information in the document and the information which people are interested in is an important aspect and main purpose of the text summarization technology research, and has very important practical significance.

Traditional methods try to get important sentences from the text, count the word frequency, or score important sentences. The result is that the resulting abstract is heavily dependent on the original and the scalability is limited. For purpose of working out these problems, we tried the abstract method and generated the summarization according to the original content. We conducted experiments using a short-text document dataset of academic documents. We suppose that the attention model of Bahdanau could be more data efficient and work better than other methods, and find that the sequence-to-sequence model with attention mechanism is indeed the case. We combined the copy mechanism and the attention mechanism, and make it more suitable for Chinese corpus. On the basis of this method, we propose an automatic summarization model for Chinese academic literature. The experimental results show that the method has achieved a higher score and can be applied to Chinese documents.
2. Related work

Majority of the early auto-summarization studies were based on extractive methods. Luhn [1] proposed a method based on word frequency text summary generation. Based on Luhn's work, Baxendale [2] analyzed the relationship between sentence position and topic. Edmundson [3] expanded the measure of sentence importance by dividing a variety of parts of speech. Later, Salon [4] proposed the TFIDF algorithm, which combines large-scale corpus to measure the weight of words and reduces the effect of invalid words such as stop words on the results [5]. Morris et al. have designed a summary method based on lexical chains [6], and Barzilay has proposed a more specific method based on WordNet's lexical chain model [7]. With the application of machine learning methods in the field of natural language processing, automatic abstraction technology has been rapidly developed. Kupec et al. [8] used the unsupervised method for abstract extraction, considered the abstract extraction as a bi-classification problem, and designed a classifier for evaluating summary sentences, which provided a reference for subsequent research. In addition, Mihalce et al. [9] proposed the TextRank algorithm [10] based on the PageRank algorithm.

Generated abstracts are often found in the single-document abstracts field. The early FRUMP system [11] can be seen as a prototype of a generative abstract. Woodsend et al. [12] used the context analysis algorithm to generate the abstract and achieved good results. Recently, great breakthroughs have been made in machine translation by the neural network model. In particular, the sequence-to-sequence learning method improves the efficiency, and the algorithm can also be applied to abstractive summaries [13]. Later, some scholars introduced this method to the automatic abstraction field. As a result, the generating digest technology has been greatly improved. Bahdanau introduced an attention mechanism [14] in 2015 that effectively improved the quality of generated sequences. Rush et al. [15] combined Bahdanau's research with the introduction of an attention mechanism in the Encoder-Decoder framework to generate abstract sentences and achieved good results on two different English data sets. Sumit et al. [16] used the RNN model to implement the task of generating abstracts, while Oriol et al. [17] combined the LSTM model with the attention mechanism to improve the quality of abstract sentences. Based on the Seq2Seq model, Gu et al. [18] proposed a coping mechanism to work out the problem of unrecognized words garbled in summary generation.

3. Our method

In this section, we propose an automatic summarization algorithm based on the Encoder-Decoder framework for the summarization of short texts in academic literature, which has a good effect on the processing of sequence problems. The encoding module and the decoding module use different neural network models. After the encoder encodes the file and obtains the intermediate representation, it is output in the decoding section. The predicted sequence.

3.1. Encoder

The Encoder reads each input element according to the input sequence of the word sequence. Usually, the RNN model is used to encode the sequence. In the coding process, using the tanh function as an activation function, the convergence of the iteration gradually converges, causing a problem of a gradient drop. Therefore, using the RNN model for long-sequence processing does not work well, resulting in a lot of loss of valid information in the final stage of coding. Although a suitable activation function can be replaced to avoid gradient descent, it is difficult to fundamentally solve this problem.

The current common practice is to use the GRU model or the LSTM model to solve long-term dependencies. The significant commonality of the two models is that they all contain additive components from the time t to the time t+1, which is exactly missing in the RNN model. In addition, both the GRU model and the LSTM model can retain many features through the "gate" structure, and after a trial comparison, the difference in performance between the two is not obvious. However, compared with the LSTM model, the GRU model has better convergence time and iterative efficiency than the LSTM model when using the same-sized training set. So we use a more concise GRU model to build the encoder.
As shown in Figure 1, input sequence \( \{x_1, x_2, x_3, \ldots, x_n\} \) is encoded by the GRU unit, the hidden layer state in the output sequence is represented, that is, the semantic vector sequence \( \{y_1, y_2, y_3, \ldots, y_n\} \).

![Figure 1. GRU-based Encoder structure.](image)

Inside a GRU unit, the update gate, such as Equation (1), relies on the input word vector at time \( t \) and the hidden state at the time \( t-1 \). In order to update the weight of the gate, use the sigmoid function to control its activation value in the interval \([0,1]\). The update gate is expressed as:

\[
z_t = \sigma(W_r \cdot [h_{t-1}, x_t])
\]  

(1)

The reset gate is determined by adjusting the weight of each gate, which decides how many hidden state information at time \( t-1 \) and can be represented as:

\[
r_t = \sigma(W_r \cdot [h_{t-1}, x_t])
\]  

(2)

If the reset gating value is 0, then the past hidden state information will be lost, indicating that the current memory content only has new input words, as in:

\[
h_t = \tanh(W \cdot [r_t \cdot h_{t-1}, x_t])
\]  

(3)

The update gate value is set to 1, and the final hidden state is determined by the current memory information. After the input sequence is encoded by the GRU model, the hidden state is obtained at different times. The process of encoding the text can be simplified as:

\[
h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t
\]  

(4)

3.2. Decoder

With the Encoder-Decoder framework, summarization often faced with the problem of unlisted word recognition. When a predictive word is generated at the decoder side, the corresponding word or similar word cannot be found in the dictionary to interpret and replace the input word. Missing words in the dictionary are usually marked as "UNK", unrecognized words. The usual solution is to build a huge vocabulary and maximize the scope of recognition [19]. However, this method does not fundamentally solve the problem of new word recognition. In addition, the large scale of the thesaurus will cause the layer to face computational bottlenecks and reduce the efficiency of generating prediction words. The emergence of the copy mechanism can solve the problem of unidentified words to a certain extent. The core idea is to identify common words as much as possible by building a high-frequency vocabulary. For a word that does not exist in the dictionary, the word "UNK" is not marked, but the word is copied directly from the original text to the predicted sequence and output, and this process can be regarded as a word extraction process. Combining dictionaries with word extraction, small-scale dictionaries can be used to work out the problem of unrecognized words in the digest generation.

Our work combines copying mechanism and attention mechanism and uses LSTM models to build decoders. The hidden state is determined by the hidden state at the previous time \( t-1 \), the predicted word, and the text vector at the current time \( t \). The hidden state is shown as below:

\[
h_t = f([w_{t-1}, e_t], h_{t-1})
\]  

(5)
The context vector $e_t$ is calculated by the hidden state $h_j$ output by the encoder and the attention score at the current time $t$, as in:

$$
e_t = \sum_{j=1}^{n} a_j h_j$$  \hspace{1cm} (6)

At the time of t, the attention score $\alpha_{jt}$ of the j-th word in the input sequence is expressed as $\omega_{jt}$, and normalized by the softmax function, and $\omega_{jt}$ as in:

$$\omega_{jt} = \text{att}(h_{t-1}, h_j) = \mu^T_a \tanh(W_a h_{t-1} + U_a h_j)$$  \hspace{1cm} (7)

$\mu_a^T, W_a, U_a$ are all learnable parameters. In the foregoing, the use of fixed-dimensional eigenvectors for traditional encoders to represent the output of the entire sequence has been mentioned. At present, most of the ideas for solving unrecognized words are similar, and all unrecognized words with “UNK” mark are also characterized using fixed feature vectors. Although unrecognized words cannot find the corresponding interpretation in the dictionary, the semantics of different unrecognized words are not the same. This method of using a single vector representation will reduce the accuracy of the model. Especially in smaller corpora, the emergence of a large number of "UNK"-marked words has an extremely negative impact on the readability of the abstract. In the small-scale corpus, using traditional methods to solve the problem of unrecognized words will produce more "UNK" marks, which will greatly reduce the readability of the abstract.

In this method, we redefine the scope of recognizable words. For the word $w_{t-1}$ to be predicted, if the corresponding entries can be found in the dictionary, the word vector corresponding to the word is used to represent the text. If the word does not exist in the dictionary, but there is the same or similar definition in the input sequence, the word can be described as:

$$p_j = f(h_{t-1}|W_e, b_c)$$  \hspace{1cm} (8)

$h_j$ contains the hidden state of the input sequence encoded output, and $p_j$ is a semantic representation of the original text. $W_e, b_c$ are learnable parameters. If the input sequence and the dictionary do not contain a representation corresponding to a word, it is marked as "UNK".

The predicted word obtained from the prediction can be obtained from the dictionary and the input sequence, which reduces the probability of occurrence of the unrecognized word. As shown in Figure 2, the hidden state at the decoder at the current time $t$ is $a_t$; the hidden state at the time $t-1$ is the predicted digest word; the text vector output by the encoder is $a$, then it can be predicted by the above variables. The digest word can be output.

![Decoder structure introduced copying mechanism.](image-url)
4. Experiments

In this section, we have constructed a short-text academic literature data set, and based on this, we have trained the model, and we have experimentally proved that our method has the better readability of abstracts obtained from the Chinese-based data set.

4.1. Dataset

The summary and the title of the academic document are related to each other, we use them as a training dataset. After training, the model can be used to extract a short text summary. The equations are an exception to the prescribed specifications of this template. Our dataset covers literature in 10 disciplines and crawls about 61,000 short texts. After word segmentation and stop word preprocessing, the data still needs to perform operations such as data integrity checking, label replacement, and data alignment. Among them, the headline and short text content should correspond to each line of the line. Missing titles and missing text content are considered invalid data. For the date that may appear in the text, such as time, date, year, month, day, etc., this article uses the label to replace. For purpose of improving the quality of the generated digest, the length of the short text in the training set is limited to no more than 150 characters; the length of the title is no less than 6 characters. If one of the conditions is not satisfied or both of the conditions are not met, the data is treated as invalid data. After the final screening, effective data was 33,000. The headline and short text are respectively imported into two different files, which are left-right aligned, that is, one line headline corresponds to one line of short text.

4.2. Result

The output forecast summary consists of word sequences. According to the reading habits of Chinese, the space characters between words are removed, and the output words are combined into a sentence output. Select the part of the summary results to display, we can see that using the method proposed in this paper, the abstract obtained can extract important information in the essay, which is more consistent with the Chinese reading habits. This shows that the summary effect obtained by applying the copy mechanism and the attention mechanism prediction is better.

Three methods were used in the experiment to extract short text summaries: F-Method based on word frequency extraction method, extraction method based on the TextRank algorithm, the Sum-Method method proposed in this paper. The experiment chooses three Topic short texts for summary extraction. The ROUGE-1 evaluation standard focuses on the extent to which the abstract covers the original text topic, the score results for each method are shown in Table 1. According to the comprehensive evaluation of F1 average value, the Sum-Method method presented in this paper has the highest score, and the method based on word frequency has the lowest score.

| Method     | Topic    | score   |
|------------|----------|---------|
| Sum-Method | Topic1   | 0.3636  |
|            | Topic2   | 0.3536  |
|            | Topic3   | 0.2223  |
|            | Average  | 0.3131  |
| TextRank   | Topic1   | 0.2609  |
|            | Topic2   | 0.3195  |
|            | Topic3   | 0.2392  |
|            | Average  | 0.2372  |
| F-Method   | Topic1   | 0.2119  |
|            | Topic2   | 0.2366  |
|            | Topic3   | 0.2284  |
|            | Average  | 0.2256  |

For purpose of obtaining a more comprehensive evaluation effect, we use the ROUGE-2 standard to evaluate the three methods with six sets of short texts with different topics. The average score of recall rate scores for the three methods are as shown in Figure 3. The ROUGE-2 evaluation standard focuses on reflecting the coherence of the abstract relative to the original text. In the first group, the
second group, and the fifth group of data, the recall rate of the method proposed in this paper is not much different from the TextRank method, but the overall scores of the two methods are higher than the F-Method method based on word frequency. In the fourth and sixth experiments, the methods proposed in this paper have achieved the best results. The recall rate of the three methods is generally low, analysis of the reasons, the choice of experimental samples has a greater impact on the results. But overall, the Sum-Method method proposed in this paper is better in coherence.

Figure 3. Comparison of recall rates for three different methods.

5. Conclusion
Based on the Encoder-Decoder framework, we propose a short text auto-summORIZATION method and obtain an auto-summary model through training. First, the encoder design was performed. By comparing and analyzing the LSTM and GRU models, the GRU unit was selected to construct the encoder. In addition, the copy mechanism and attention mechanism are introduced in the decoder to improve the quality of the generated abstract. Experimental results show that our method has achieved good results in the experimental dataset.

In this work, a short text data set was constructed for academic literature, but the size of the data set needs to be improved and the data acquisition framework needs to be optimized. The method proposed by us is limited by the length of the sequence, and the effect of abstracting the long text is not good, and the readability of the abstract needs to be enhanced. Therefore, the focus of our work in the future is to optimize the automatic summarization algorithm and improve the readability of short text summaries.

Acknowledgments
This work was financially supported by The Aoshan Innovation Project in Science and Technology of Qingdao National Laboratory for Marine Science and Technology(No.2016ASKJ07), Key R&D plan of Shandong province (2016ZDJS09A01) and Qingdao Science and technology plan(17-1-1-3-jch). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References
[1] Luhn H P. The automatic creation of literature abstracts[J]. IBM Journal of research and development, 1958, 2(2): 159-165.
[2] Baxendale P B. Machine-made index for technical literature—an experiment[J]. IBM Journal of Research and Development, 1958, 2(4): 354-361.
[3] Edmundson H P. New methods in automatic extracting[J]. Journal of the ACM (JACM), 1969, 16(2): 264-285.
[4] Salton G, Yu C T. On the construction of effective vocabularies for information retrieval[C]//ACM SIGIR Forum. ACM, 1973, 9(3): 48-60.
[5] Suqin Z B S H M. An Improved Text Feature Weighting Algorithm Based On TFIDF [J]. Computer Applications and Software, 2011, 2: 007.
[6] Morris J, Hirst G. Lexical cohesion computed by thesaural relations as an indicator of the structure of text[J]. Computational linguistics, 1991, 17(1): 21-48.
[7] Barzilay R, Elhadad M. Using lexical chains for text summarization[J]. Advances in automatic text summarization, 1999: 111-121.
[8] Sharan A, Imran H, Joshi M L. A trainable document summarizer using bayesian classifier approach[C]//Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on. IEEE, 2008: 1206-1211.
[9] Page L, Brin S, Motwani R, et al. The PageRank citation ranking: Bringing order to the web[R]. Stanford InfoLab, 1999.
[10] Mihalcea R, Tarau P. Textrank: Bringing order into text[C]//Proceedings of the 2004 conference on empirical methods in natural language processing. 2004.
[11] DeJong G. An overview of the FRUMP system[J]. Strategies for natural language processing, 1982, 113: 149-176.
[12] Zhong S, Deng Z. A web crawler system design based on distributed technology[J]. Journal of Networks, 2011, 6(12): 1682.
[13] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]//Advances in neural information processing systems. 2014: 3104-3112.
[14] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.
[15] Rush A M, Chopra S, Weston J. A neural attention model for abstractive sentence summarization[J]. arXiv preprint arXiv:1509.00685, 2015.
[16] Chopra S, Auli M, Rush A M. Abstractive sentence summarization with attentive recurrent neural networks[C]//Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016: 93-98.
[17] Vinyals O, Kaiser Ł, Koo T, et al. Grammar as a foreign language[C]//Advances in Neural Information Processing Systems. 2015: 2773-2781.
[18] Gu J, Lu Z, Li H, et al. Incorporating copying mechanism in sequence-to-sequence learning[J]. arXiv preprint arXiv:1603.06393, 2016.
[19] Parada C, Dredze M, Filimonov D, et al. Contextual information improves OOV detection in speech[C]//Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2010: 216-224.