Hybrid Non-Local Network for Abstractive Summarization

Zepeng Hao\(^1\), Yupu Guo\(^1\), Cheng Gong\(^1\) and Honghui Chen\(^1\)

\(^1\) Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, China, 410073.

\(^{\ast}\)hao3184045@gmail.com

Abstract. In the field of abstractive automatic summarization, the Seq2Seq model constructed by the RNN network is widely used and has achieved perfect results. However, due to the Seq2Seq model and the internal structure of the RNN network, only short-term and local information can be referenced in the process of generating summary, resulting the generated summary contains less information, and may even happen that the generated summary does not match the corresponding document. There have been many previous attempts to improve the model by using the attention mechanism, but only to use long-term information within a certain range. In our paper, we present a abstractive summarization model that enhances the global feature, which called Hybrid Non-Local Network model for single document summarization task, consisting of a non-local block and a traditional Seq2Seq module. Our model uses the non-local network block to transform the Seq2Seq model to better capture global feature. For example, the relationship between words and words, sentences and sentences in a large span of the original document, making the generated summary of higher quality, more informative, more accurate and rich representation of the original document. The experiments in our paper are performed on the CNN/DailyMail corpus. The proposed model was improved by 1.37, 0.47, and 1.16 on R-1 score, R-2 score, and R-L score, separately, compared to the baseline model.

1. Introduction
With the rapid growth in the amount of data, the demand for the retrieval, understanding and analysis of a large amount of information is constantly increasing, which continues to promote the development of the automatic text summarization field, and the automatic text summarization model is constantly improving, in order to generate more informative, smoother and more similar to human-generated summaries. The automatic text summarization methods can be separated into two categories: the extractive summarization method, and the abstractive summarization method. The extractive summarization method is to judge, score and sort the importance of words and sentences in the source document, and then extract words or sentences from the source document according to the order of importance to form the summary. The abstractive summarization method is to encode and understand the original document through the model, and then generate words and phrases by decoding to form the summary.

The extractive summarization method\(^1\) is very mature, and the effect obtained on each dataset is remarkable. Most summarization systems also use the extractive summarization method, but with the continuous development of the deep learning network structure and the improvement of computing power, the abstractive summarization method slowly showing the advantages: smoother and closer to human-generated summaries. At present, most of the abstractive summarization models are built on the Seq2Seq\(^2\)|\(^3\) structure with attention mechanism. The attention mechanism\(^4\)|\(^5\) can help the
model to collect and utilize global information better, but the scope of the attention mechanism is relatively limited. The attention mechanism cannot capture the relationship between words or sentences that are far apart in the source document. This problem will result that the generated summary contains less information, and may even happen that the generated summary does not match the corresponding document. In the recent work in the field of imagery, the Facebook team proposed a non-local network block[6] to capture long-range dependencies in images, and non-local network block can calculate the response at a position as a weighted sum of the features at all positions. This block captures more global features and achieves significant results on image recognition and image restoration tasks[7].

Inspired by the Facebook team, we try to migrate the application of the non-local network block to the automatic text summarization field, and proposed our model in this paper: Hybrid Non-Local Network model. Our model is mainly composed of two components, namely the non-local network block and the Seq2Seq module. We mainly improve the decoder in the Seq2Seq module through the non-local network block. In the non-local network block, by analyzing and mining the information generated by the encoder, a richer global information is obtained, so that richer reference information is given when the decoder generates the summary. In the Seq2Seq module, encoding is performed according to the feature representation of the document, and then the summary is generated according to the encoded information and the information provided by the non-local network block. This model ensures that the generated summary is more informative, smoother, and closer to human-generated summaries.

We test our model on the CNN/DailyMail dataset[8]. Through experiments, we found that our model performance is 1.37, 0.47 and 1.16 higher than the state-of-the-art baseline at R-1 score, R-2 score, and R-L score, separately.

In our paper, we summarize our significant contributions follow:

- We present a hybrid model consisting of the non-local network block and the Seq2Seq module called Hybrid Non-Local Network model.
- We migrate non-local network block from the image domain to the text domain. The non-local network block can better capture the global features, especially the relationships between words or sentences that span a large range in the source document, so that the information obtained by the model more abundant, the generated summary more informative and smoother.
- We test our model under conditions of different fixed-length summaries and find that our model is better at generating longer summaries.

2. Approach

In this paper, the model we proposed consists of two components: non-local network block and Seq2Seq module. The structure of the whole model is shown in the figure 1. Next we discuss the details of the non-local network block and the Seq2Seq module.
2.1. Non-Local Network block

We migrate the non-local network block from the image domain to the text domain. Building the non-local network block requires three steps. First, define the non-local network according to the non-local operation, then select the specific instantiations, and finally define and construct the non-local network block of the text domain.

2.1.1. Non-Local Network. According to the description of non-local operations, we define the expression of the non-local network as follows:

\[ y_i = \frac{1}{C(x)} \sum_{j} f(x_i, x_j) g(x_j) \]  \hspace{1cm} (1)

In the above formula, \( i \) is the index of the computed output position, \( j \) is the index of all positions that are likely to be related with \( i \), \( x \) is the input information (here is the text representation), \( y \) is the same output information as the \( x \) dimension, \( f \) is a scalar function that computes the relationship between \( i \) and all \( j \), the \( g \) function calculates the representation of the input information at \( j \) position, and \( C(x) \) is the function used for normalization. In the traditional Seq2Seq model, only the adjacent relationship with the corresponding position is considered. The addition of the attention mechanism can make the model consider a large range of relationships, but there is no way to fully consider all positions. It can be seen from the definition of non-local network that the non-local
network considers the relationship between all locations and corresponding location, which enhances the capture of global features by the model and enriches the global information.

2.1.2. Instantiations. In order to make non-local network work, we introduce several versions of $f$ and $g$ functions. Based on previous research experience, we set the $g$ function to consist of a linear embedding function. The $g$ function is represented by the following formula:

$$g(x_j) = W_g x_j$$

(2)

Where $W_g$ is a learnable parameter matrix. Next we describe several alternative $f$ functions.

**Dot product.** The $f$ function can be defined as a dot product to calculate the similarity at different positions:

$$f(x_i, x_j) = \theta(x_i)^T \phi(x_j)$$

(3)

Here we use the embedding mode, the corresponding regularization factor $C(x) = N$, where $N$ is the number of positions of the input information $x$.

**Gaussian.** Here the $f$ function is defined as a Gaussian function to compute the similarity:

$$f(x_i, x_j) = e^{\|x_i - x_j\|^2}$$

(4)

Which is equivalent to the improvement of the dot product function, and the calculation of the Euclidean distance is applied. The normalization function corresponding to the $f$ function should be set as $C(x) = \sum_{i,j} f(x_i, x_j)$.

**Embedded Gaussian.** Embedded Gaussian function is an extension of the Gaussian function. Here we design the $f$ function as follows:

$$f(x_i, x_j) = e^{\theta(x_i)^T \phi(x_j)}$$

(5)

In the above formula, $\theta(x_i) = W_\theta x_i$ and $\phi(x_j) = W_\phi x_j$ are both composed of a learnable embedded matrix. The regularization factor corresponding to the $f$ function is $C(x) = \sum_{i,j} f(x_i, x_j)$.

Since we move the non-local network to the text domain, the input of the non-local network may be embedded text representations, so we choose the embedded Gaussian function and its corresponding normalization factor to be applied to the non-local network. The non-local network can be flexibly constructed into block, adding and improving existing model structures to enrich the global features captured by the model.

2.1.3. Non-Local block. In order to make the non-local network better fit into the existing model, we design the non-local network block as follows:

$$z_i = W_z y_i + x_i$$

(6)

Where $y_i$ is the output of the non-local network, and $+ x_i$ allows the non-local network block to be flexibly added to the pre-trained model.

2.2. Seq2Seq module

Next we introduce the Seq2Seq model that combines non-local network block. We use non-local network block to improve the decoder and capture more global features, making the information
referenced when generating the summary more abundant. Next, I will divide the model into two components (encoder and decoder) to discuss.

2.2.1. Encoder. Firstly, we use the Word2vec method to get the embedding feature representation of each word in the original document. And the embedding representation of each token as input through the Bi-LSTM (bi-directional LSTM) structure. This structure can concatenate the hidden states from the previous and next word at each time step, capturing semantic relationships from both sides of each word, resulting in a series of hidden state $h_t$. The number of neurons in the LSTM structure in each direction is half the number of neurons in the output representation, then in the same number of neurons, the both sides states can be connected.

2.2.2. Decoder. In this part, we improve the decoder by adding non-local network block to capture more global features, solve the problem that the attention mechanism obtains the limitation of the relationship, and make the generated summary information more abundant.

First, we use the sequence of hidden states obtained by the encoder as the input of the non-local network block, and capture more potential global features through the non-local network block, so that the content vector information generated in the next step is more abundant.

Next, we input the embedding representation of the previous word into the LSTM to get the hidden state of the decoder, and then calculate the attention distribution according to the output of the non-local network block and the hidden state of the decoder:

$$e'_i = v^T \tanh(W_s h^*_i + W_s s_{t,i} + b_a)$$ \tag{9}

$$a' = \text{soft max}(e')$$ \tag{10}

In the above formula, $v, W_s, W'_s$ and $b_a$ are learnable parameters. The attention distribution depends on the relationship between the hidden state of the decoder and the hidden state of the encoder with enhanced global features. Next, we can compute the content vector $c_i$ by the following formula:

$$c_i = \sum a'_i h^*_i$$ \tag{11}

We incorporate the content vector $c_i$ with the hidden state of the decoder $s_{t,i}$ and then calculate the probability distribution of the vocabulary $P_{t,i}$ through a two-layer linear network:

$$P_{t,i} = \text{soft max}\left(W'(W[s_{t,i}, c_i] + b) + b'\right)$$ \tag{12}

where $W, W', b$ and $b'$ are learnable parameters. Simultaneously, we add the pointer-generator network[9][10] to the decoder to solve the out-of-vocabulary (OOV) problem. Then the probability distribution of the final predictive word is as follows:

$$P(w) = p_s P_{t,i}(w) + (1 - p_s) \sum_{a'_i \neq w} a'_i$$ \tag{13}
is an influence factor, it decides whether to sample a word from the probability distribution or copy a word from the original document. $p_s$ is calculated by the content vector $c$, and the hidden state of the decoder $s_t$:

$$p_s = \sigma(w_c^T c_t + w_s^T s_t + b_p)$$

(14)

Where $w_c$, $w_s$ and $b_p$ are parameters obtained through training and $\sigma$ is the sigmoid function. Next, we use the sum of the negative log likelihood of the target word $w_t$ in each time step in the whole sequence as the TotalLoss. In the training process, we minimize the TotalLoss by updating the model parameters $\lambda$:

$$\min_{\lambda} \sum_{t=0}^{T} - \log P(w_t^*)$$

(15)

At the same time, we added the beam-search mechanism to our model to solve the optimization problems in model training and testing, so that we can better train and test our model. Finally, we can get the generated summary through the model.

3. Experiments

In this section, we describe the dataset and evaluation metrics for the automatic text abstractive summarization task. Also introduced the baseline models compared to our model, research questions as well as the details of the parameters used in the experiment.

3.1. Datasets and Evaluation metrics

Here, we apply the CNN/DailyMail corpus that is commonly used in the domain. The dataset has 286,817 training data pairs, 13,368 validation data pairs and 11,487 test data pairs. At the same time, the source document in the training set have an average of 766 words, with an average of 29.74 sentences, while the summary has an average of 53 words with an average of 3.72 sentences. The uniqueness of this training set is that there are many long documents, and the summaries are mostly composed of multiple sentences in an order, which brings certain challenges to the abstractive summarization model. We use the ROUGE metrics to evaluate the summary generated by the model: ROUGE-1 and ROUGE-2 mainly reflect the amount of information, and ROUGE-L mainly reflects fluency.

3.2. Baseline Models and Research questions

We select two abstractive summarization models to be compared to our hybrid non-local network model, which are ABS and PGC, respectively. ABS is a traditional Seq2Seq model with attention mechanism. PGC is a Seq2Seq model with pointer-generator coverage network. In the experiments we mainly discuss two research questions: (RQ1) Can our model perform better than baselines on dataset? (RQ2) When the generated summary length is fixed to 75 bytes, 175 bytes, and full length, what is the quality of summaries generated by models?

3.3. Parameters

In the experiment we apply the 128-dimensional word2vec embedded representation trained on the dataset as input, the hidden states of our model is set to 256 dimensions, for the pointer generation network we apply a 50k-word vocabulary.
4. Results and Analysis

4.1. Model performance

In order to answer RQ1, we test and evaluate the summaries generated by models through the ROUGE metrics. The evaluation results are shown in Table 1. It can be seen from the results that the PGC model is significantly higher than the ABS model in the three metrics, indicating that the summary generated by the PGC model is better than the summary generated by the ABS model in terms of information volume and fluency.

Our model is obviously higher than the PGC model on R-1 score and R-L score, and slightly higher than the PGC model on the R-2 score, indicating that the summary generated by our model smoother and more informative. Experiments show that the non-local network block can enhance the model's capture of global features and enable the model to generate a more informative summary.

Table 1. The score on the evaluation metrics when the generated summary length is fixed to the full length. (%)

| Method     | R-1 Score | R-2 Score | R-L Score |
|------------|-----------|-----------|-----------|
| ABS Model  | 36.41     | 15.43     | 32.79     |
| PGC Model  | 38.89     | 16.90     | 35.32     |
| Hybrid Model | 40.26   | 17.37     | 36.48     |

4.2. Different length summaries

To answer RQ2, we use the model to generate summaries of full length, 75 bytes, and 175 bytes. These summaries are evaluated by the ROUGE metrics, and the results of the evaluation are shown in Tables 1, 2, and 3, separately. As can be seen from the evaluation results, the performance of our model is similar to the PGC model when generating the 75-byte length summary. However, when generating the summary of 175 bytes in length, the metrics of R-1 score, R-2 score and R-L score are significantly higher than the other two models. This shows that our Hybrid Non-Local Network model can capture more global features, and can make the summary more informative when generating the longer summary. At the same time, we can see from Figure 2 that for our model itself, the model performs better when generating the longer summary, indicating that our model is more suitable for generating long summaries.

Table 2. The score on the evaluation metrics when the generated summary length is fixed to the 75 bytes. (%)

| Method     | R-1 Score | R-2 Score | R-L Score |
|------------|-----------|-----------|-----------|
| ABS Model  | 13.74     | 6.20      | 12.88     |
| PGC Model  | 21.40     | 9.66      | 20.44     |
| Hybrid Model | 21.60   | 9.80     | 20.46     |

Table 3. The score on the evaluation metrics when the generated summary length is fixed to the 175 bytes. (%)

| Method     | R-1 Score | R-2 Score | R-L Score |
|------------|-----------|-----------|-----------|
| ABS Model  | 26.77     | 11.68     | 24.52     |
| PGC Model  | 32.58     | 14.16     | 29.97     |
| Hybrid Model | 32.95   | 14.41     | 30.28     |
Figure 2. The score of our model on the evaluation metrics when the generated summary length changes.

5. Conclusions
In our paper, we present the Hybrid Non-Local Network model for single-document abstractive summarization. The model improves the traditional Seq2Seq model through non-local network block, so that the hybrid model can capture more global features and generate higher-quality, more informative summaries. The experimental results demonstrate that our model performs better on the ROUGE evaluation metrics than the baseline models. Our model is more suitable for generating long summaries.

References
[1] Moratanch N, Chitrakala S. A survey on extractive text summarization[C]// International Conference on Computer. 2017.
[2] Nallapati R, Zhou B, Gulcehre C, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond[J]. arXiv preprint arXiv:1602.06023, 2016.
[3] Kim S, Kang S G. TL; DR: Improving Abstractive Summarization Using LSTMs[J].
[4] Rush, A.M., Chopra, S., and Weston, J. A neural attention model for abstractive sentence summarization. In EMNLP, 2015.
[5] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems. 2017: 5998-6008.
[6] Wang X, Girshick R, Gupta A, et al. Non-local Neural Networks[J]. 2017.
[7] Liu D, Wen B, Fan Y, et al. Non-Local Recurrent Network for Image Restoration[J]. 2018.
[8] Hermann K M, Kocisky T, Grefenstette E, et al. Teaching machines to read and comprehend[C]//Advances in Neural Information Processing Systems. 2015: 1693-1701.
[9] See A, Liu P J, Manning C D. Get to the point: Summarization with pointer-generator networks[J]. arXiv preprint arXiv:1704.04368, 2017.
[10] Jiang X, Hu P, Hou L, et al. Improving Pointer-Generator Network with Keywords Information for Chinese Abstractive Summarization[J]. 2018.