Selective and Coverage Multi-head Attention for Abstractive Summarization

Xuwen Zhang¹ and Gongshen Liu*¹
¹ School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China
*Corresponding author’s e-mail: lgshen@sjtu.edu.cn

Abstract. Although the Transformer model has outperformed traditional sequence-to-sequence model in a variety of natural language processing (NLP) tasks, it still suffers from semantic irrelevance and repetition for abstractive text summarization. The main reason is that the long text to be summarized is usually composed of multi-sentences and has much redundant information. To tackle this problem, we propose a selective and coverage multi-head attention framework based on the original Transformer. It contains a Convolutional Neural Network (CNN) selective gate, which combines n-gram features with whole semantic representation to obtain core information from the long input sentence. Besides, we use a coverage mechanism in the multi-head attention to keep track of the words which have been summarized. The evaluations on Chinese and English text summarization datasets both demonstrate that the proposed selective and coverage multi-head attention model outperforms the baseline models by 4.6 and 0.3 ROUGE-2 points respectively. And the analysis shows that the proposed model generates the summary with higher quality and less repetition.

1. Introduction
Text summarization aims to produce a brief sentence of the salient information of a given text. And there are two ways to achieve text summarization, extractive and abstractive [10]. Unlike extractive text summarization, which is composed of same words or phrases form the source text directly, abstractive text summarization can generate more fluent and readable sentence since it uses language generation model based on semantic representation. In this work, we focus on sentence level abstractive summarization, which can be considered as a sequence-to-sequence task that the long sentence text should be mapped to the brief summary.

Many researchers have applied neural network model to abstractive text summarization. Rush et al. [14] propose a sequence-to-sequence model with a Convolutional Neural Network (CNN) encoder and a feed-forward network decoder for abstractive summarization. Follow this study, Chopra et al. [1] replace the decoder with Recurrent Neural Network (RNN) and Nallapati et al. [11] make the model a fully-RNN model by using an RNN encoder.

Recently, a new network architecture named Transformer [18] and the multi-head attention mechanism mentioned have outperformed CNN and RNN in a variety of natural language processing (NLP) tasks, such as machine translation [18,19], dialogue systems [17] and sentiment analysis [5]. This non-recurrent and solely attention-based model can easier get long-range dependencies [18] and perform better on word sense disambiguation [16]. As the summary generating process can be regarded as the encoding-decoding paradigm with source text input, the Transformer model may be more suitable for this task than RNN based sequence-to-sequence model (Seq2Seq). However, unlike machine translation,
Zhou et al. [20] point out that the main focus in abstractive text summarization is not to infer the alignment but to select highlights in the input, since there is no obvious alignment relationship between the input text and output summary. See et al. [15] mention that repetition is especially pronounced when generating long text. The Transformer can align the sequence easily and get semantic information with high quality, while it lacks the ability to filter out secondary information in the long input and keep track of what has been generated when decoding. As a result, the Transformer model can suffer from semantic irrelevance and repetition in abstractive text summarization.

To tackle this problem, we propose a Selective and Coverage Multi-head Attention Transformer (SCMAT) for sentence level abstractive text summarization. The basic idea is to use a CNN selective gate to improve semantic quality and a coverage mechanism to reduce repetition. The CNN selective gate is based on n-gram features and whole semantic representation of the input sentence to filter out secondary information. The coverage mechanism sums the attention distribution to obtain a coverage vector, and then it uses this vector to calculate new attention distribution in the following time steps. Besides, a coverage loss is added to fit the coverage mechanism. We evaluate the proposed model on Chinese LCSTS, English Gigaword and DUC 2004 test sets. Our SCMAT model achieves 31.5 ROUGE-2 F1 score, 18.3 ROUGE-2 F1 score and 13.1 ROUGE-2 recall respectively, which improves performance compared to the baseline models. And the analysis shows that the proposed model can generate the summary with higher quality and less repetition.

2. Related Work
Since abstractive text summarization can be regarded as a sequence-to-sequence task based on language generation models and can generate novel words, this task has attracted many researchers’ attentions. Rush et al. [14] apply sequence-to-sequence model to abstractive text summarization. And the ABS model they propose produces the state-of-the-art results on Annotated English Gigaword [12] test set. Chopra et al. [1] change the ABS model with an RNN decoder, and the CNN encoder with RNN decoder model outperforms previous work. Follow this idea, Nallapati et al. [11] propose a full RNN sequence-to-sequence model by replacing the CNN encoder with an RNN encoder, which also improves performance.

Zhou et al. [20] point out that the main consideration in this task is to select highlights in the input and propose a selective encoding model based on RNN Seq2Seq. Lin et al. [8] also propose a global encoding RNN Seq2Seq to choose the core information at each encoding time step. Furthermore, Ma et al. [9] propose a supervisor model by using an auto encoder to learn a better internal representation for abstractive text summarization. Hu et al. [3] build a large-scale Chinese short text summarization dataset collected from Chinese social media, which is one of our benchmark datasets. Tu et al. [21] propose linguistic coverage and NN-based coverage to keep track of the attention history in Neural Machine Translation Task. See et al. [15] also apply a coverage mechanism to reduce repetitions in text summarization task.

3. Proposed Model
The SCMAT model is introduced in detail in this section. As shown in Figure 1, the proposed model is similar to the original Transformer which consists of two components, an encoder and a decoder. Differently, the proposed model has a selective and coverage multi-head attention sub-layer in the decoder part.

3.1. Problem Formulation
Given a text summarization dataset $\mathcal{D}$ that contains $N$ data samples, the sample $(x, y)$ consists of a source sentence $x = (x_1, x_2, ..., x_M)$, and its corresponding target sentence $y = (y_1, y_2, ..., y_L)$, where $n$ is the content length and $L \leq M$ is the summary length. $V_s$ is the source vocabulary and $V_t$ is the target vocabulary. Therefore, the abstractive text summarization problem is given by:

$$p(y|x; \theta) = \prod_{l=1}^{L} p(y_l|y_{<l}, x; \theta) \quad (1)$$
where $\theta$ is a set of model parameters and $y_{\text{cd}}$ is a partial target summary.

\begin{align*}
    y_{\text{cd}} &= \text{softmax} \left( \frac{\mathbf{Q}^T \mathbf{K}}{\sqrt{d_k}} \right) \\
    \mathbf{Q} &= \text{softmax} \left( \frac{\mathbf{Q}^T \mathbf{K}}{\sqrt{d_k}} \right)
\end{align*}

Figure 1. The overview of our model. The right part is the selective attention framework.

### 3.2. Multi-Head Attention and Transformer

In this subsection, we give a quick overview of multi-head attention mechanism and the Transformer model. Multi-head attention is to compute basic attention operation $h$ times with different dimensions in parallel. Given the input, a list of queries $Q$ and key-value pairs $(K, V)$, we can compute multi-head attention as follows:

\begin{align*}
    \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_h) \\
    \text{head}_i &= \text{Attention}(Q \mathbf{W}_i^Q, K \mathbf{W}_i^K, V \mathbf{W}_i^V) \\
    \text{Attention}(Q, K, V) &= \text{softmax} \left( \frac{Q^T K}{\sqrt{d_k}} \right)
\end{align*}

where $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V$ are parameter matrices and $d_k$ is the dimension of $K$. The attention distribution matrix $A$ between queries $Q$ and keys $K$ is:

\begin{align*}
    A &= \text{softmax} \left( \frac{Q^T K}{\sqrt{d_k}} \right) = \text{Concat}(a_0, \ldots, a_t) \\
    a_t &= \text{softmax} \left( \frac{Q^T K}{\sqrt{d_k}} \right)
\end{align*}

where $a_t$ is the attention distribution between one query and keys at each time-step $t$, which represents $t_{th}$ query in queries list.

In the Transformer model, a fully connected feed-forward network is applied after the last multi-head attention layer, it contains two linear layers with a ReLU activation in between:

\begin{align*}
    \text{FFN}(x) &= \text{max}(0, xW_1 + b_1)W_2 + b_2
\end{align*}

where $W_1$ and $W_2$ are parameter matrices; $b_1$ and $b_2$ are bias vectors and $\text{max}(0, x)$ is the ReLU activation function.

As shown in Figure 1, given the input sentence $x = (x_1, x_2, \ldots, x_M)$, the encoder reads the input and generates its dense representation $h^e$. Then given the target summary $y = (y_1, y_2, \ldots, y_L)$ and $h^e$, the decoder first calculates the dense representation $z^d$ of $y$ by the first multi-head attention layer, and calculates the attention representation between he and $z^d$ by the second multi-head attention, and generates the decoder output vector $h^d = (h^d_1, \ldots, h^d_L)$. And this vector $h^d$ is fed through the final linear layer followed by a softmax layer to computes the probability distribution of the output words at $t_{th}$ time-step:

\begin{align*}
    p(y_t | x) &= \text{softmax} (h^t W_o + b_o)
\end{align*}

where $W_o$ and $b_o$ are parameters of the output layer. The word with the highest probability is emitted as $t_{th}$ word of the generated summary.
3.3. Selective Mechanism
Zhou et al. [20] point out that the main challenge in abstractive text summarization is to filter out secondary information in the input rather than to infer the alignment between the input sentence and the output summary. However, the original Transformer doesn’t contain a mechanism to achieve this. Thus, we propose a selective gate in multi-head attention for abstractive text summarization. A selective gate is added on the attention mechanism between the encoder output and the decoder’s second sub-layer input, which is to filter out secondary information in the encoder output \( h^e \), so that the queries of input can focus on the core information in the key-value pairs of the encoder output. In this multi-head attention, the sentence dense representation \( h^e \) is used as keys and values. For each het, the selective gate generates a gate vector \( gate_t \), using \( h_t^e \) and \( h^e \). The attention is computed with the tailored keys and values, \( h' = (h'_1, ..., h'_M) \).

In detail, we implement a CNN structure which uses 1-dimension convolution to extract n-gram features \( e = (c_1, ..., c_M) \) of the input sentence as follows:

\[
c_t = \text{ReLU} \left( [h_{t-k/2}^e, ..., h_{t+k/2}^e] W_c + b_c \right)
\]

where \( W_c \) and \( b_c \) are parameters of the CNN structure; and \( k \) is the kernel size of CNN unit. The CNN structure consists of three 1-dimension convolutions with kernel size \( k = 1, 3, 3 \) respectively and concatenate the outputs into a dense vector with dimension \( d_m \).

The encoder output \( h_e \) is a dense vector to represent the meaning of the input sentence, which is also an important part of the selective gate besides the n-gram features. For each time step \( t \), the selective gate takes the n-gram features \( c_t \) and the sentence representation \( h_t \) as input to compute the gate vector \( gate_t \) and the tailored vector \( h'_t \):

\[
gate_t = \delta(h_t^eW_g + c_tU_g + b_g)
\]

\[
h'_t = h^e \odot gate_t
\]

where \( W_g \) and \( U_g \) are weight matrices; \( b_g \) is bias vector; \( \delta \) denotes sigmoid activation function and \( \odot \) is element-wise multiplication.

We obtain a new dense vector \( h' \) as the keys and values of the multi-head attention after this selective gate. Since the CNN structure can extract n-gram features and the encoder output dense vector represents the meaning of the whole sentence, this selective gate can highlight key information in the input depending on the comprehensive information. And the sigmoid function outputs a vector of value between 0 and 1 at each dimension, if \( gate_t \) is close to 1, the selective gate highlights the het at the corresponding dimension (orange bars in Figure 1), otherwise, the gate filters out the information.

3.4. Coverage Mechanism
In coverage mechanism, we use a coverage vector \( cov_t \) similar to See et al. [15]:

\[
cov_t = \sum_{i=0}^{t-1} a_i
\]

where \( a_t \) is attention distribution between query and keys at time-step \( t \). \( cov_t \) is a sum of total attention distributions before time-step \( t \). It can represent the degree of which words have been focused on from the attention mechanism. And \( cov_0 \) is a zero vector because no word has been focused on at the beginning.

The coverage vector is used to calculate new attention distribution after time-step \( t \), and we replace the dot-product attention in formulation (6) with additive attention, and the attention formulation in multi-head attention is changed to:

\[
e_{t,i} = v^T \tanh(KW_k + q_tW_q + c_tW_c + b_{\text{attn}})
\]

\[
a_t = \text{softmax}(e_t)
\]

\[
\text{Attention}(Q,K,V) = \text{Concat}(a_0, ..., a_t)V
\]

where \( v \), \( W_k \), \( W_q \) and \( W_c \) are parameter matrices; \( b_{\text{attn}} \) is bias vectors and \( t \) is the number of queries in Q. Although, the additive attention is not as fast as dot-product attention and it cannot be calculated by using highly optimized matrix multiplication code, it is necessary to use it in our coverage mechanism. Because the coverage vector \( cov_t \) is calculated by attention distribution at \( a_t \) each time-step \( t \), and at the next time-step \( t+1 \), the attention distribution \( a_{t+1} \) is calculated by \( cov_t \). This process can not be
calculated in parallel. And the additive attention uses a feed-forward network to compute the distribution, which can learn to reduce repetition with the coverage vector. We also use a coverage loss to fit the coverage mechanism:

\[ \text{covloss}_t = \sum_{i} \min(a_{t, i}, \text{cov}_{t, i}) \]  \hspace{1cm} (16)

The coverage loss is added to the loss function to penalize repeatedly attentions on the same words in K. The coverage mechanism in multi-head attention is not always used during the training stage. The selective and coverage multi-head attention layer (Figure 1) computes the attention without coverage mechanism first. That means it doesn’t add the coverage loss because the model needs to focus on learning semantic representation and reduce the main loss function first. And at the end of training stage, the coverage mechanism is turned on to reduce the repetition.

3.5. Overall Loss Function

Our goal is to maximize the generated summary probability by the given input sentence. Therefore, the learning process is to minimize the negative log-likelihood loss function, and the coverage loss is supposed to be added to prevent repetition attentions:

\[ J(\theta) = -\frac{1}{|D|} \sum_{(x, y) \in D} (\log p(y|x; \theta) + \lambda \sum_{i} \min(a_{t, i}, \text{cov}_{t, i})) \]  \hspace{1cm} (17)

where \( \lambda \) is a hyperparameter; dataset \( D \), parallel sentence pair \((x, y)\) and model parameter \( \theta \) are described in section 3.1.

4. Experiments

In the following, we introduce the datasets, the experiment settings, the baseline models and the performance of the proposed model. Also, we provide an example to demonstrate that the proposed model can obtain more core information and analyze the performance of reducing semantic irrelevance and repetition.

4.1. Datasets

The Chinese dataset is Large Scale Chinese Social Media Text Summarization Dataset (LCSTS), which is a large scale Chinese short text summarization dataset constructed by Hu et al. [3]. The dataset consists of more than 2.4 million text-summary pairs collected from a Chinese social media website. We follow the previous work [3] to split the dataset into three parts, with 2.4M pairs in training set, 8K in validation set and 0.7K in test set.

The English training set is English Gigaword, which is a text summarization dataset constructed from the Annotated English Gigaword corpus [12]. We use the data preprocessed by Rush et al. [14] to extract text-summary pairs with 3.8M for training, 189K for validation. We use the English Gigaword and DUC 2004 [13] test sets for testing. The DUC 2004 test set contains 500 sentence-summary pairs with each sentence paired with 4 different human-written reference summaries.
4.2. Experiment Settings
We implement our experiment in Tensorflow on an NVIDIA 1080Ti GPU. The number of identical layers in encoder and decoder is 6. The number of heads in multi-head attention is 8. The word embedding dimension and output dimension $d_m$ are both 512. We use Adam optimizer [4] with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$. Learning rate is varied over the course of training as Vaswani et al. [18] mentioned. Gradient clipping is applied to the maximum norm of 5.0. We use beam search to generate summaries and the beam size is set to 5 in our experiments.

Following the previous researches, we employ ROUGE [7] to evaluate the performance of our model. ROUGE score is to measure the quality of summary by calculating overlapping between output summary and reference. We use ROUGE-1, ROUGE-2 and ROUGE-L as the evaluation metrics.

Table 2. ROUGE Scores on LCSTS Test Set (on character-level).

| model     | Gigaword R-1 | Gigaword R-2 | Gigaword R-L | DUC2004 R-1 | DUC2004 R-2 | DUC2004 R-L |
|-----------|--------------|--------------|--------------|-------------|-------------|-------------|
| ABS       | 29.6         | 11.3         | 26.4         | 26.6        | 7.1         | 22.1        |
| ABS+      | 29.8         | 11.9         | 27.0         | 28.2        | 8.5         | 23.8        |
| Feats2s   | 32.7         | 15.6         | 30.6         | 28.4        | 9.5         | 24.6        |
| CA2s      | 33.8         | 16.0         | 31.2         | 29.0        | 8.3         | 24.1        |
| SEASS     | 36.2         | 17.5         | 33.6         | 29.2        | 9.6         | 25.5        |
| DRGD      | 36.3         | 17.6         | 33.6         | 31.8        | 10.8        | 27.5        |
| CGU       | 36.3         | 18.0         | 33.8         | -           | -           | -           |
| seq2seq   | 33.4         | 15.9         | 31.2         | 28.5        | 9.1         | 24.6        |
| transformer | 35.8       | 17.1         | 32.9         | 31.3        | 10.3        | 26.7        |
| SCMAT     | **36.9**     | **18.3**     | **34.2**     | **35.2**    | **13.1**    | **31.6**    |

4.3. Baselines
Since we compare the results of the proposed model with the baseline models reported in their original papers, the evaluation on Chinese and English datasets has different baselines. For LCSTS, we compare our model with the following baselines. RNN and RNN-context [3] are two Seq2Seq models with GRU encoder and decoder, and RNN-context has attention mechanism. CopyNet [2] is an attention based Seq2Seq with copy mechanism to allow parts of output are copied from input. DRGD [6] is a Seq2Seq with a variational autoencoder and a deep recurrent generative decoder. CGU [8] is a conventional sequence-to-sequence model with a global encoding over the encoder output.

For English test sets, ABS and ABS+ [14] are the CNN based models with attention and handcrafted features. CA2s [1] is an extension of ABS model with an RNN decoder. Feats2s [11] is a full RNN based sequence-to-sequence model. SEASS [20] is a sequence-to-sequence model with a selective mechanism to produce a second sentence representation. DRGD and CGU are also baselines for English Gigaword test set.

We also implement an RNN based sequence-to-sequence model and an original Transformer as our baselines and denote them as seq2seq and transformer.

4.4. Results and Discussion
We report ROUGE F1 scores of our model and baselines for LCSTS and English Gigaword test sets. And we report ROUGE recall scores for DUC 2004 test set following Zhou et al. [20]. The experimental results show our model outperforms the recent abstractive text summarization system (reported in the original articles) and our implement baselines on these test sets.

For LCSTS test set, Table 1 shows that our SCMAT model achieves 44.7 ROUGE-1, 31.5 ROUGE-2 and 41.7 ROUGE-L F1 score respectively. The results of our implement seq2seq and transformer shows that multi-head attention mechanism is more effective for abstractive text summarization. However, the original transformer has lower scores than superAE and CGU for lacking the ability to obtain core information from long source input. As for the proposed model, selective and coverage
multi-head attention helps it own a gain of 5.3 ROUGE-1, 4.6 ROUGE-2 and 5.2 ROUGE-L F1 scores compared with CGU model.

Table 2 shows the evaluations on the Gigaword and DUC2004 test sets. For the popular ROUGE-2 metric on Gigaword test set, the implement transformer model has a lower score than SEASS, DRGD and CGU. While the proposed model SCMAT achieves 0.3 ROUGE-2 F1 score compared to the highest CGU model. And for DUC2004 test set, SCMAT owns an advantage of 3.4 ROUGE-1, 2.3 ROUGE-2 and 4.1 ROUGE-L recall scores compared with DRGD model.

Table 3. Ablation Test Results with Each Mechanism.

| model               | LCSTS | Gigaword | DUC2004 |
|---------------------|-------|----------|---------|
|                     | R-1   | R-2      | R-L     | R-1   | R-2      | R-L     | R-1   | R-2      | R-L     |
| transformer         | 38.0  | 24.5     | 35.0    | 35.8  | 17.1     | 32.9    | 28.5  | 9.1      | 24.6    |
| transformer+coverage| 38.2  | 26.1     | 35.5    | 36.3  | 17.4     | 33.6    | 29.8  | 9.8      | 27.0    |
| transformer+selective| 44.3  | 30.9     | 41.1    | 36.4  | 17.7     | 33.9    | 34.1  | 11.1     | 30.6    |
| SCMAT               | 44.7  | 31.5     | 41.7    | 36.9  | 18.3     | 34.2    | 35.2  | 13.1     | 31.6    |

Table 3 shows the ablation test results with coverage mechanism and selective mechanism. For Chinese test set LCSTS, the selective mechanism helps the implement transformer own an advantage of 6.4 ROUGE-2 score, while the coverage mechanism only helps it own a gain of 1.6 ROUGE-2 score. For English test sets, our implemented transformer with coverage mechanism achieves 17.4 and 9.8 ROUGE-2 scores on Gigaword and DUC2004 respective, and the model with selective mechanism gets a little higher score with 17.7 and 11.1 ROUGE-2 scores. It shows that the two mechanisms both help the base transformer get summaries with higher quality and the selective mechanism is more effective one in the sentence level text summarization.

Table 4. A summarization example of our model compared with that of baselines and the reference.

**Source:** Last night, several people were caught to smoke on a flight of China United Airlines from Chengdu to Beijing. Later the flight temporarily landed on Taiyuan Airport. Some passengers asked for a security check but were denied by the captain, which led to a collision between crew and passengers. At present, China United Airlines is contacting the crew for verification.

**Reference:** Several people were smoked on a flight from Chengdu to Beijing which led to a collision between crew and passengers.

**seq2seq:** A flight of China United Airlines temporarily landed on Taiyuan Airport.

**transformer:** A flight of China United Airlines from Chengdu to Beijing temporarily landed on Taiyuan Airport.

**SCMAT:** Several people were smoked on a flight from Chengdu to Beijing which led to a collision between crew and passengers.

Table 5. ∆ ROUGE F1 scores on different groups of sentences in terms of their length for SCMAT and transformer on LCSTS test set. ∆ means the difference between SCMAT and transformer, e.g., ∆R-1 equals R-1 of SCMAT minus R-1 of transformer.

| Length | ∆ R-1 | ∆ R-2 | ∆ R-L |
|--------|-------|-------|-------|
| <90    | 1.28  | 1.38  | 1.70  |
| 90-100 | 5.90  | 7.05  | 6.48  |
| 100-110| 8.17  | 8.15  | 7.85  |
| >110   | 7.00  | 6.99  | 6.73  |

Table 4 shows a summarization example of SCMAT model, compared with our implement baseline models (seq2seq and transformer) and the reference. The main idea of the source text contains the departure and destination of the flight and the collision between crew and passengers. The summary generated by seq2seq model doesn’t contain any core information, which leads to a completely unrelated summary. The transformer model generates a better result than seq2seq, which contains the departure...
and destination, but still has an irrelevant meaning. The two baselines both focus on the secondary information of the airport which the flight temporarily landed on and miss the real core information. However, with selective and coverage multi-head attention, the SCMAT obtains the core information of the source context and generate the summary which is very close to the reference.

To demonstrate the effectiveness of the proposed model, we further test SCMAT and transformer with different sentence lengths on LCSTS test set. The sentences can be grouped into 4 different groups considering the length and total numbers of the sentences in each group. Table 5 shows the increment of ROUGE F1 score between SCMAT and transformer on different groups of input sentences. The results show that the proposed model has an advantage on all the sentence groups compared with transformer, and the increment becomes more obvious with the growth of the sentences length.

![Figure 2. The percentage of the duplicates in the summary of the Gigaword test set.](image)

Besides the semantic quality, we calculate the percentage of the duplicates in the summary to evaluate the degree of repetition. Figure 2 shows duplicates of 1-gram to 3-gram on the English Gigaword test. We remove the English stop words and masked tokens in the summary, which leads to low percentages of 3-gram duplicates. The result proves that the selective and coverage multi-head attention can reduce repetition compared with seq2seq and transformer.

5. Conclusion
In this work, we propose a selective and coverage multi-head attention mechanism which extends the Transformer model for abstractive text summarization. The selective gate helps the decoder focus on the core information of the input representation to improve the semantic quality. The coverage mechanism keeps track of what has been summarized to prevent the decoder from generating repetition words. Experimental results on LCSTS, Gigaword and DUC2004 show that our model outperforms the baseline models on these test sets.

Acknowledgements
This research work has been funded by the National Natural Science Foundation of China (Grant No.61772337, U1736207), and the National Key Research and Development Program of China NO. 2018YFC0830703 and 2016QY03D0604.

References
[1] Chopra, Chopra, S., Auli, M., Rush, A.M.: Abstractive sentence summarization with attentive recurrent neural networks. In: Proceedings of the 2016 Conference of the NAACL: Human Language Technologies, pp. 93-98 (2016)
[2] Gu, J., Lu, Z., Li, H., Li, V.O.: Incorporating copying mechanism in sequence-to-sequence learning. In: ACL. vol. 1, pp. 1631–1640 (2016)
[3] Hu, B., Chen, Q., Zhu, F.: Lcsts: A large scale chinese short text summarization dataset. In: EMNLP. pp. 1967–1972 (2015)
[4] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR abs/1412.6980 (2014)
[5] Letarte, G., Paradis, F., Gigüere, P., Laviolette, F.: Importance of self-attention for sentiment analysis. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. pp. 267–275 (2018)
[6] Li, P., Lam, W., Bing, L., Wang, Z.: Deep recurrent generative decoder for abstractive text summarization. In: EMNLP. pp. 2091–2100 (2017)

[7] Lin, C.Y., Hovy, E.: Automatic evaluation of summaries using n-gram co-occurrence statistics. In: Proceedings of the 2003 Conference of the NAACL on Human Language Technology-Volume 1. pp. 71–78. ACL (2003)

[8] Lin, J., Xu, S., Ma, S., Su, Q.: Global encoding for abstractive summarization. In: ACL. vol. 2, pp. 163–169 (2018)

[9] Ma, S., Xu, S., Lin, J., Houfeng, W.: Autoencoder as assistant supervisor: Improving text representation for Chinese social media text summarization. In: ACL. vol. 2, pp. 725–731 (2018)

[10] Mani, I.: Advances in automatic text summarization. MIT press (1999)

[11] Nallapati, R., Zhou, B., dos Santos, C., glar Gulcehre, C., Xiang, B.: Abstractive text summarization using sequence-to-sequence rnns and beyond. CoNLL 2016 p.280 (2016)

[12] Napoles, C., Gormley, M., Van Durme, B.: Annotated gigaword. In: Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction. pp. 95–100. ACL (2012)

[13] Over, P., Dang, H., Harman, D.: Duc in context. Information Processing & Management 43(6), 1506-1520 (2007)

[14] Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. In: EMNLP. pp. 379–389 (2015)

[15] See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. In: ACL, vol. 1, pp. 1073-1083 (2017)

[16] Tang, G., Müllner, M., Rios, A., Sennrich, R.: Why self-attention? a targeted evaluation of neural machine translation architectures. In: EMNLP. pp. 4263–4272 (2018)

[17] Tao, C., Gao, S., Shang, M., Wu, W., Zhao, D., Yan, R.: Get the point of my utterance! learning towards effective responses with multi-head attention mechanism. In: IJCAI. pp. 4418-4424 (2018)

[18] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.

[19] Zhang, J., Luan, H., Sun, M., Zhai, F., Xu, J., Zhang, M., Liu, Y.: Improving the transformer translation model with document-level context. In: EMNLP. pp. 533-542 (2018)

[20] Zhou, Q., Yang, N., Wei, F., Zhou, M.: Selective encoding for abstractive sentence summarization. In: ACL (Volume 1: Long Papers). vol. 1, pp. 1095–1104 (2017)

[21] Tu Z, Lu Z, Liu Y, et al. Modeling Coverage for Neural Machine Translation[C]//Proceedings of the 54th ACL (Volume 1: Long Papers). 2016: 76-85.