A Hierarchical Hybrid Neural Network Architecture for Chinese Text Summarization

Yunheng Zhang[1], Leihan Zhang[1,✉], Ke Xu[1], and Le Zhang[2]

[1] State Key Laboratory of Software Development Environment, Beihang University, Beijing, China, zhangleihan@gmail.com
[2] School of Economics and Management, Beijing University of Posts and Telecommunications, Beijing, China

Abstract. Using sequence-to-sequence models for abstractive text summarization is generally plagued by three problems: inability to deal with out-of-vocabulary words, repetition in summaries and time-consuming in training. The paper proposes a hierarchical hybrid neural network architecture for Chinese text summarization. Three mechanisms, hierarchical attention mechanism, pointer mechanism and coverage mechanism, are integrated into the architecture to improve the performance of summarization. The proposed model is applied to Chinese news headline generation. The experimental results suggest that the model outperforms the baseline in ROUGE scores and the three mechanisms can improve the quality of summaries.

Keywords: Abstractive Text Summarization · Hierarchical Attention Mechanism · Pointer Mechanism · Coverage Mechanism

1 Introduction

Text summarization is to generate a brief and coherent summary to represent the key ideas of the text. The methods of text summarization can be broadly classified into two categories: extractive summarization and abstractive summarization. The models of extractive summarization extract the segments from the original to compose the summary. In contrast, the abstractive models generate a compressed paraphrase of the main ideas of texts, potentially using words which don't exist in the source text. The extractive models can guarantee the grammaticality of summarization, while the abstractive models have more sophisticated abilities such as paraphrasing and generalization [11].

Many researches have been concentrating on abstractive summarization and the sequence-to-sequence models have been successfully introduced into the abstractive summarization [9]. Based on the framework of sequence-to-sequence models, some mechanisms, such as attention mechanism [3, 10], pointer mechanism [11] and coverage mechanism [11], have been proposed to improve the quality of summarization. Attention mechanism is usually used to solve the sequence-to-sequence tasks [3, 10]. The weaknesses of these models are that they can't deal with out-of-vocabulary (OOV) words and usually repeat words in

summaries. Pointer mechanism was proposed to copy words from the original [4, 11]. Inspired by the coverage model of Tu et al. [14], See et al. [11] proposed coverage mechanism to reduce the repetition in the output.

The above models and mechanisms can improve the quality of abstractive summarization, but usually require much time for training. In order to improve the efficiency of summarization, a hierarchical hybrid neural network architecture is proposed. The hierarchical structure can shorten the input for the encoder and reduce the need for time. Attention mechanism, pointer mechanism and coverage mechanism are modified to integrate into the hierarchical structure. Then we apply the proposed model to generate headlines for the Chinese news from Sina society news. Experimental results show that the proposed hierarchical hybrid neural network is remarkably effective in Chinese text summarization.

2 Related Work

Abstractive text summarization is a challenging problem and a few distinguished works have been achieved. Sequence-to-sequence model have been successful in many problems like machine translation [1] and abstractive text summarization [10]. Sutskever et al. [12] used encoder-decoder model to solve the sequence-to-sequence tasks. Bahdanau et al. [1] proposed the attention mechanism for the machine translation. Then, the attention-based model was introduced into the sentence summarization [10] and Chopra et al. [3] extended the model with LSTM. Guo et al. [5] used the encoder-decoder model for headline generation. Ma et al. [7] proposed a model for Chinese social media text summarization and the length of these texts is usually less than 140 characters.

Based on the encoder-decoder model, Vinyals et al. [15] proposed the pointer networks. The decoder picks the elements from the input sequence and copies them to form an output sequence. The traditional encoder-decoder model can't generate the elements which don't appear in the training data, but the pointer networks can handle the input sequence with OOV words. The advantage of the pointer networks is very helpful for abstractive text summarization. A few researchers [4, 9, 11] applied the pointer mechanism from the pointer networks to abstractive text summarization and proved that the models are able to copy the rare or unseen key words from the original instead of generating the imprecise words. Coverage is usually applied to dampen repeated attention. See et al. [11] proposed coverage mechanism to alleviate the repetition in summaries.

In order to handle the long documents better, the hierarchical document structure is used to the document representation. The hierarchical model can integrate the information on the word level and the sentence level. Tang et al. [13] fed the word representations through a CNN or LSTM to get the sentence representation, and then fed the sentence representations through a gated RNN to get the document representation. Yang et al. [16] applied the two-level attention mechanisms to the construction of the sentence representation and the document representation. Nallapati et al. [9] used the sentence-level attention to re-scale

the corresponding word-level attention and further redefined the importance of the words according to the importance of the sentence.

3 Proposed Architecture

3.1 Vanilla Sequence-to-Sequence Model with Vanilla Attention Mechanism

The vanilla sequence-to-sequence model [15] contains an encoder and a decoder. Generally, the encoder can understand the input sequence and represent the sequence with a vector. The decoder can produce the output sequence according to the input representation.

The recurrent neural network (RNN) is used to process the input sequence or generate the output sequence as an encoder or a decoder. The gated recurrent unit (GRU) is a gated RNN variant [2] which outperforms the traditional RNN. A GRU contains a reset gate and an update gate. At the encoding time t, the reset gate r_t and the update gate z_t are computed as follows:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{1}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{2}$$

where x_t is the input, h_{t-1} is the previous hidden state, σ is the sigmoid function and W_r, U_r, b_r, W_z, U_z and b_z are learnable parameters. The new hidden state h_t is computed as follows:

$$\widetilde{h}_t = \tanh(W_h x_t + r_t \odot (U_h h_{t-1}) + b_h) \tag{3}$$

$$h_t = z_t h_{t-1} + (1 - z_t)\widetilde{h}_t \tag{4}$$

where W_h, U_h and b_h are learnable parameters. x_t and h_{t-1} are deemed as the inputs for each iteration of GRU. And let $h_t = f(x_t, h_{t-1})$ denote the whole process of equation (1), (2), (3) and (4) to simplify the below description.

We use the GRU as the encoder in the sequence-to-sequence model. $d = \{w_1, w_2, \ldots, w_n\}$ represents the input sequence of the tokens in the whole document, and h_n represents the last hidden state of the encoder GRU. In addition, we use the '[UNK]' token to represent any OOV word. The document representation d_e is computed as shown below:

$$d_e = W_e h_n + b_e \tag{5}$$

where W_e and b_e are learnable parameters.

In the decoder, another GRU generates the tokens one by one to form the output sequence, namely the summary. At the decoding time t, the hidden state h'_t is computed by the previous hidden state h'_{t-1} of the GRU, the last output y_{t-1} in the generation step and the context vector c_t, and h'_t is fed through a linear layer to produce the vocabulary distribution P_v:

$$e_{ti} = g(h'_t, h_i) \tag{6}$$

$$\alpha_t = \text{softmax}(e_t) \tag{7}$$

$$c_t = \sum_i \alpha_{ti} h_i \tag{8}$$

$$h'_t = f([y_{t-1}, c_t], h'_{t-1}) \tag{9}$$

$$P_v = \text{softmax}(W_v[h'_t, c_t] + b_v) \tag{10}$$

where W_v and b_v are learnable parameters. The function g is the combine of the linear function and the tanh function. At the first decoding time step, h'_0 is the final output of the encoder, namely the document representation d_e.

Furtherly, the final probability of the word y in the fixed vocabulary is the corresponding probability in the vocabulary distribution:

$$P(y) = P_v(y) \tag{11}$$

The architecture of vanilla sequence-to-sequence model with vanilla attention mechanism is displayed in Figure 1.

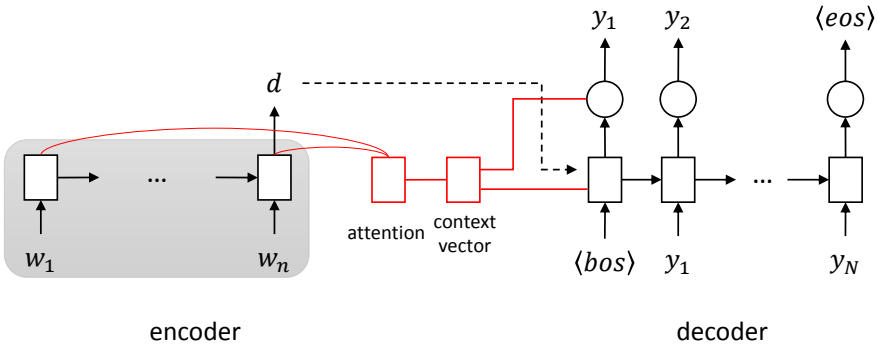

Fig. 1. The architecture of vanilla sequence-to-sequence model with vanilla attention mechanism.

For training, the training objective is the minimization of the total loss L as shown below:

$$L = -\frac{1}{N} \sum_{t=1}^{N+1} \log P(y_t^* | y_1^*, \ldots, y_{t-1}^*, x; \theta) \tag{12}$$

where $y_N^* = \{y_1^*, y_2^*, \ldots, y_t^*\}$ is the ground truth sequence, x is the corresponding input sequence and θ represents the learnable parameters in the whole model. For testing, we pick the token with maximum probability to serve as the optimal result at each decoding time step.

3.2 Hierarchical Sequence-to-Sequence Model

Hierarchical sequence-to-sequence model is a variant of the vanilla sequence-to-sequence model. The module of decoder is unchanged and the hierarchical encoder is introduced to process the Chinese documents. The advantage of this structure is that it can shorten the input length for each GRU. In addition, the structure can also help the model understand the text on the word level and the sentence level.

A Chinese document usually consists of some sentences, and each sentence consists of some words. Therefore, the hierarchical document structure mainly includes two parts. For the first part, the sequence $s_i = \{w_{i1}, w_{i2}, \ldots, w_{in}\}$ is inputted to a GRU named word encoder, where w_{ij} represents the j^{th} word in the corresponding i^{th} sentence. The sequence $h_s = \{h_{s1}, h_{s2}, \ldots, h_{sm}\}$ is fed through a linear layer to get the output s_i as the sentence representation, where h_{si} represents the last hidden state of word encoder of the i^{th} sentence:

$$s_i = W_{se} h_{si} + b_{se} \tag{13}$$

where W_{se} and b_{se} are learnable parameters. For the other part, the sequence $d = \{s_1, s_2, \ldots, s_n\}$ is inputted to another GRU named sentence encoder and the last hidden state h_{dn} is fed through a linear layer to get the document representation d_e:

$$d_e = W_{de} h_{dn} + b_{de} \tag{14}$$

where W_{de} and b_{de} are learnable parameters.

3.3 Hierarchical Attention Mechanism

Attention mechanism enables neural network models to focus on the important information of the original at each decoding time step. However, because the hierarchical encoder has two encoding parts and the encoding of the sentences in the document is separate, the traditional attention mechanism is not suitable for the hierarchical document structure. In order to utilize the two encoding processes and make the hierarchical model focus on the different word-level and sentence-level information during the decoding, we apply the hierarchical attention mechanism. The sentence-level attention is based on the hidden state of the sentence encoder. It makes the decoder have the ability to identify the key sentences and produce the relevant tokens. The sentence-level context vector c_{st} is computed as follows:

$$e_{sti} = g_s(h'_t, h_{si}) \tag{15}$$

$$\alpha_{st} = \text{softmax}(e_{st}) \tag{16}$$

$$c_{st} = \sum_i \alpha_{sti} h_{si} \tag{17}$$

where i is the ID of the sentence.

Then, the word-level attention is applied to help the decoder pay attention to the key words in the document. Because it is independent to compute the word-level context vectors for the different sentences, so the word-level attention is cooperated with the sentence-level attention. For each sentence in the document, the word-level context vector c_{wt} is computed as follows:

$$e_{wtij} = g_w(h'_t, h_{wij}) \tag{18}$$

$$\alpha_{wti} = \text{softmax}(e_{wti}) \tag{19}$$

$$\alpha_{wtj} = \alpha_{sti}\alpha_{wtij} \tag{20}$$

$$c_{wt} = \sum_{i,j} \alpha_{wtj} h_{wij} \tag{21}$$

where j is the ID of the word in the corresponding sentence. The biggest difference from the sentence-level attention is that the weight of the word-level attention is also influenced by the sentence-level attention of the corresponding sentence, which means that the importance of the word is related to the word-level attention and the sentence-level attention.

To realize the hierarchical attention mechanism, we put c_{st} and c_{wt} into equation (9) and equation (10) and respectively change them to new equations as shown below:

$$h'_t = f([y_{t-1}, c_{st}, c_{wt}], h'_{t-1}) \tag{22}$$

$$P_v = \text{softmax}(W_v[h'_t, c_{st}, c_{wt}] + b_v) \tag{23}$$

In this way, the decoder can focus on the important words and sentences when generating the tokens to form the summary of the document.

3.4 Pointer Mechanism

The traditional sequence-to-sequence model can only generate the summaries with the fixed generation vocabulary and it can't produce OOV words. Therefore, pointer mechanism is introduced to the decoding step to enable the decoder to copy the key words from the original as part of the summary. Inspired by the idea of pointer-generator network [11], we use the generation probability p to control the decoder to generate a word or directly copy an existing word from the source. The generation probability p seems like a switch and can be calculated by:

$$p = \sigma(w_p^T[y_{t-1}, c_{st}, c_{wt}, h'_{t-1}] + b_p) \tag{24}$$

where w_p and b_p are learnable parameters.

According to equations (22) and (23), we can get the vocabulary distribution P_v. It represents the generative probabilities of the words in the fixed vocabulary. Due to the introduction of pointer mechanism, we should consider the copying probability for the words existing in the fixed vocabulary and coming from the original. The word-level attention distribution α_{wt} means the importance of the

words in the original and can also represent the copying probability in some sense.

Therefore, the copying distribution P_c is computed by the word-level attention distribution α_{wt}:

$$P_c(y) = \sum_{j:w_j=y} \alpha_{wtj} \tag{25}$$

That is to say, for the word y in the document, the corresponding copying probability is the sum of all the word-level attention weights of the words which are the same as the word y. And the copying probability will be zero if the word y doesn't exist in the document.

Then, the vocabulary distribution P_v and the copying distribution P_c are combined to get the final probability of the word y as follows:

$$P(y) = pP_v(y) + (1-p)P_c(y) \tag{26}$$

In fact, the final probability is a linear interpolation between the vocabulary distribution and the copying distribution. In this way, the model can generate a word from the fixed vocabulary or copy a word from the original at each decoding time step. Pointer mechanism can help the decoder produce the words which are not in the vocabulary and make the summaries containing the key information from the particular documents.

3.5 Coverage Mechanism

According to See et al. [11], the coverage mechanism is helpful for the model to alleviate the repetition in the output. The main idea of the coverage mechanism is to prevent the model from focusing on the same word repeatedly. Then, the coverage mechanism is introduced into the hierarchical model. The architecture of the model is shown in Figure 2. Coverage vector c'_t reflects the all previous total word-level attentions of the model and it can affect the next word-level attention distribution:

$$c'_t = \sum_{t'=0}^{t-1} \alpha_{wt'} \tag{27}$$

$$e_{wtij} = g_w(h'_t, h_{wij}, c'_{tj}) \tag{28}$$

Generally, paying attention to the same words or phrases repeatedly can result in the repetition of words in the summaries. Therefore, it is reasonable to alleviate the repetition by restricting the repeated attention on the word-level content. And because the sentences usually consist of some words or phrases, it is not necessary to restrict the attention on the sentence-level content.

In addition, the loss L in this model should be added the coverage loss and be changed as shown below:

$$L = -\frac{1}{N}\sum_{t=1}^{N+1} \log P(y^*_t | y^*_1, \ldots, y^*_{t-1}, x; \theta) + \lambda \sum_i \min(\alpha_{wt}, c'_t) \tag{29}$$

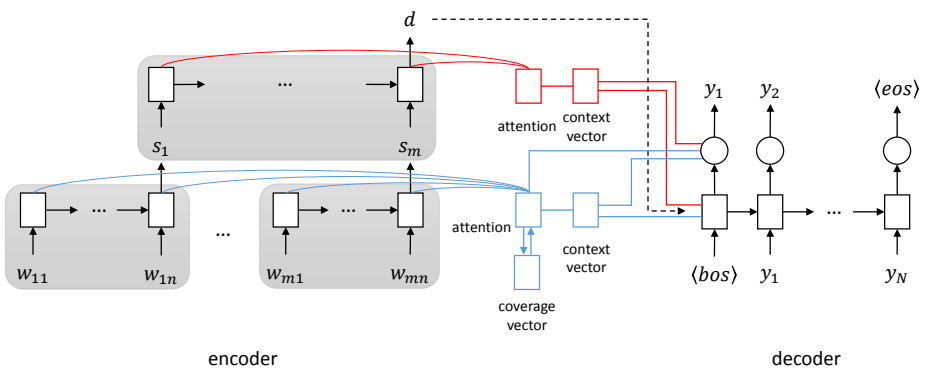

Fig. 2. The hierarchical hybrid architecture with hierarchical attention mechanism, pointer mechanism and coverage mechanism.

4 Experiments

4.1 Dataset

The proposed hierarchical hybrid architecture is designed for relative long Chinese document summarization, so a proper dataset is necessary for evaluation. However, to the best of our knowledge, there is no standard dataset for long Chinese documents. Hence, we construct a dataset by collecting the Sina society news[3]. The body texts and the corresponding headlines of Sina society from June 11, 2012 to August 31, 2017 were collected. We kept the news whose body texts are not less than 60 characters and finally got 41,792 valid pairs for our experiments. We randomly selected 33,434 pairs as the training set and remained 8,358 pairs as the test set.

4.2 Implementation

Firstly, the input documents were split into sentences according to punctuation. And the sentences and headlines were tokenized by jieba[4]. Statistically, the average length of the input is about 649 tokens and the average length of the output is about 12 tokens. Then, the word representations were derived by using word2vec model [8] and the embedding dimension was set to 128. There are 359,043 different tokens in the corpus, and the tokens with frequency less than 20 were filtered. At last, 52,614 tokens were remained to build the fixed vocabulary.

We used PyTorch[5] to implement the deep learning model. The hidden state dimension of the encoder and the decoder was set to 512. For training, we used Adam with an initial learning rate of 0.001 and a batch size of 8. We applied

[3] http://news.sina.com.cn/society/
[4] https://pypi.org/project/jieba/
[5] https://pytorch.org/

gradient clipping with a maximum gradient norm of 2. And the training samples were shuffled in each epoch. The output length was limited to 30 tokens. We set $\lambda = 1$ for equation (29). Then the model was evaluated with the F1-score of ROUGE-1, ROUGE-2 and ROUGE-L metrics [6]. In addition, the quantity of OOV words was also analyzed. At last, the running time for the proposed model was compared with the baseline.

4.3 Results

Table 1. The ROUGE scores on the test set.

Model	ROUGE-1	ROUGE-2	ROUGE-L
seq2seq + attn	14.88	1.75	13.40
h-h-attn	15.89	2.05	14.34
h-h-attn + pointer	20.87	2.65	18.17
h-h-attn + pointer + coverage	**21.97**	**3.59**	**19.27**

As shown in Table 1, the performance of our model was compared with the baseline. In detail, the vanilla sequence-to-sequence model with vanilla attention mechanism (*seq2seq + attn*) is selected as baseline. The hierarchical sequence-to-sequence model with hierarchical attention mechanism is denoted *h-h-attn*. The hierarchical sequence-to-sequence model with hierarchical attention mechanism and pointer mechanism is denoted *h-h-attn + pointer*. And the hierarchical sequence-to-sequence model with hierarchical attention mechanism, pointer mechanism and coverage mechanism is denoted as *h-h-attn + pointer + coverage*. Experimental results in Table 1 suggest the outstanding performance of all the three mechanisms in the model. Hierarchical sequence-to-sequence model with hierarchical attention mechanism outperforms the baseline by focusing on the information on the word level and the sentence level during the decoding stage. And the pointer mechanism enables the model to copy the rare or unseen words as part of the summaries. Hierarchical sequence-to-sequence model with hierarchical attention mechanism, pointer mechanism and coverage mechanism outperforms all the other models in the three metrics of ROUGE-1, ROUGE-2 and ROUGE-L, which proves the effectiveness of our model.

Table 2. The percentage of OOV words in generated summaries.

Model	OOV percentage
seq2seq + attn	13.3%
h-h-attn	13.2%
h-h-attn + pointer	8.8%
h-h-attn + pointer + coverage	8.7%

Pointer mechanism can help the model copy the rare or unseen words from the source texts and reduce the appearance of OOV words. The percentage of OOV words was computed in the output sequences and the results are shown in Table 2. As can be seen, the models with pointer mechanism can produce summaries with less OOV words, which can further improve the quality of summarization.

Table 3. The running time for various models in our experiments.

Model	Time spent
seq2seq + attn	37.2 h
h-h-attn	35.3 h
h-h-attn + pointer	82.0 h
h-h-attn + pointer + coverage	95.2 h

Table 4. The summaries generated by the proposed models for one document in the test dataset.

source text: ...刘师傅说，...，他接到了“巡视组”的电话，...，并说要查账核对...刘师傅的儿子说，父亲两次被骗转账1.79万余元... ...Mr. Liu said, ..., he picked the phone from “inspection group”, ..., and they claimed to audit his account ...Mr. Liu’s son said, his father was cheated to transfer more than 17.9 thousand yuan ...
ground truth: 男子轻信“巡视组查账” 手机操作被骗1.79万元 A man believed in that “inspection group audited his account”, and he was cheated out of 17.9 thousand yuan by operating the phone ***seq2seq + attn***: 男子被冒名贷款后转走走走走走走走走走走走走走走 An imposter of a man got a loan and the money was transferred transferred transferred ***h-h-attn***: 男子网购[UNK]万元现金被骗 骗子称[UNK][UNK] A man was cheated out of [UNK] yuan in cash when he shopped online, and cheater claimed that [UNK] [UNK] ***h-h-attn + pointer***: 男子手机被骗 男子骗损失损失 A man was cheated on phone, a man cheated loss loss ***h-h-attn + pointer + coverage***: 男子轻信“巡视组” 被骗近万 被骗近万 A man believed in “inspection group”, and was cheated out of about ten thousand yuan, and was cheated out of about ten thousand yuan

In order to evaluate the efficiency of the hierarchical mechanism, we also compared the running time for the above models. As shown in Table 3, the time represents the sum of the preprocessing time, the training time and the test

time. In general, the hierarchical model with hierarchical attention mechanism is a little faster than the vanilla sequence-to-sequence model with vanilla attention mechanism.

At last, we also made case analysis to investigate the contributions of each mechanism. As shown in Table 4, the results show that the hierarchical sequence-to-sequence model and three mechanisms can obviously improve the quality of the summaries. Hierarchical attention mechanism makes the model focus on the source text and generate more related content. Pointer mechanism reduce the appearance of the '[UNK]' token. And coverage mechanism alleviates the repetition in the output.

5 Conclusion and Future Work

In this work, we proposed a hierarchical hybrid neural network architecture for Chinese text summarization. The hierarchical document structure was utilized to shorten the length of the input sequences for the encoder, and the hierarchical attention mechanism, pointer mechanism and coverage mechanism were incorporated to generate the rational and informative summaries. And experiments on Chinese news with long texts proved the efficiency of the proposed model. In the future, we will try to incorporate the extractive and abstractive methods to deal with the summarization for long texts.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv e-prints **abs/1409.0473** (Sep 2014)
2. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1724–1734. Association for Computational Linguistics (2014)
3. 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. Association for Computational Linguistics (2016)
4. Gu, J., Lu, Z., Li, H., Li, V.O.: Incorporating copying mechanism in sequence-to-sequence learning. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1631–1640. Association for Computational Linguistics (2016)
5. Guo, Y., Huang, H., Gao, Y., Lu, C.: Conceptual multi-layer neural network model for headline generation. In: Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. pp. 355–367. Springer International Publishing, Cham (2017)
6. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Marie-Francine Moens, S.S. (ed.) Text Summarization Branches Out: Proceedings of the ACL-04 Workshop. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (July 2004)

7. Ma, S., Sun, X., Xu, J., Wang, H., Li, W., Su, Q.: Improving semantic relevance for sequence-to-sequence learning of chinese social media text summarization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 635–640. Association for Computational Linguistics (2017)
8. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. pp. 3111–3119. NIPS'13, Curran Associates Inc., USA (2013)
9. Nallapati, R., Zhou, B., dos Santos, C., Gulcehre, C., Xiang, B.: Abstractive text summarization using sequence-to-sequence rnns and beyond. In: Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning. pp. 280–290. Association for Computational Linguistics (2016)
10. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 379–389. Association for Computational Linguistics (2015)
11. See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1073–1083. Association for Computational Linguistics (2017)
12. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. pp. 3104–3112. NIPS'14, MIT Press, Cambridge, MA, USA (2014)
13. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1422–1432. Association for Computational Linguistics (2015)
14. Tu, Z., Lu, Z., Liu, Y., Liu, X., Li, H.: Modeling coverage for neural machine translation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 76–85. Association for Computational Linguistics (2016)
15. Vinyals, O., Fortunato, M., Jaitly, N.: Pointer networks. In: Advances in Neural Information Processing Systems 28, pp. 2692–2700. Curran Associates, Inc. (2015)
16. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the NAACL: Human Language Technologies. pp. 1480–1489. Association for Computational Linguistics (2016)