GLEU-Guided Multi-resolution Network for Short Text Conversation

Xuan Liu and Kai Yu(✉)

Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, SpeechLab, Department of Computer Science and Engineering, Brain Science and Technology Research Center, Shanghai Jiao Tong University, Shanghai, China
liuxuan0526@gmail.com, ky219.cam@gmail.com

Abstract. With the recent development of sequence-to-sequence framework, generation approach for short text conversation becomes attractive. Traditional sequence-to-sequence method for short text conversation often suffers from dull response problem. Multi-resolution generation approach has been introduced to address this problem by dividing the generation process into two steps: keywords-sequence generation and response generation. However, this method still tends to generate short and dull keywords-sequence. In this work, a new multi-resolution generation framework is proposed. Instead of using the word-level maximum likelihood criterion, we optimize the sequence-level GLEU score of the entire generated keywords-sequence using a policy gradient approach in reinforcement learning. Experiments show that the proposed approach can generate longer and more diverse keywords-sequence. Meanwhile, it achieves better scores in the human evaluation.

Keywords: Short text conversation · Sequence-to-sequence
Multi-resolution · Policy gradient

1 Introduction

With the emergence of social media, more and more available conversation data makes data-driven approaches for conversation possible. Short text conversation is a simplified conversation problem: one round of conversation formed by two short texts, with the former being an initial post from a user and the latter being a comment given by the computer. This problem is the route towards solving the conversation problem.

Recently, sequence-to-sequence models with attention mechanisms show promising results on machine translation and machine summarization [1,2], this model is also used in short text conversations. One of the apparent advantages of the sequence-to-sequence approach over the retrieval-based approach is its ability to generate responses that are not in the corpus.

However, the sequence-to-sequence model cannot generate informative and diverse responses and tends to reply dull responses [3,4], such as "I think so",

J. Tao et al. (Eds.): NCMMSC 2017, CCIS 807, pp. 13–23, 2018.
https://doi.org/10.1007/978-981-10-8111-8_2

"Where is it?" and so on. This phenomenon has various explanations. The traditional sequence-to-sequence model is trained according to the maximum likelihood criterion (MLE), which optimizes the Kullback-Leibler divergence (KLD) between the true distribution and the distribution given by the model. Minimizing the KLD avoids assigning an extremely small probability to any data point but assigns a lot of probability mass to the non-data region [5]. For short text conversation tasks, the generated responses only depend on the mode of the distribution given by the model. However, there is no guarantee that the true probability density in the mode of this distribution is high by minimizing the KLD. So it is likely that the model will generate dull responses, which is rare in the corpus. Meanwhile, the high perplexity of the responses given the posts also indicates that the posts do not provide much useful information.

The previous observations analyze the weakness of the model. However, the fundamental reason why the traditional sequence-to-sequence model generates dull responses is related to the mechanism of conversation. Unlike machine translation, which transforms the same content from one representation to another, responding a post contains following several steps. The first step is to understand the content of the post. Then, combined with personal experiences, to decide what to reply. Finally, in the form of natural language to express our meaning. A successful short text conversation system also should follow these steps. The sequence-to-sequence model generates dull responses since it does not explicitly model the second step.

To generate diverse and rich responses, it is necessary to imitate the decision process of the conversation, and additional information should be provided to the generation step. The keywords in the responses are most likely to be treated as additional information. [6] proposes a content-introducing approach to generate responses in a two-step fashion. First, it predicts a single keyword which is a noun reflecting the semantics of the response. Then it uses a modified encoder-decoder framework to generate the response, explicitly making sure that the predicted keyword is in the response. Although this approach improves the richness and diversity of the responses, it is not enough for a single keyword to summarize what the response is talking about. Considering this issue, multi-resolution recurrent neural network [7] regards a sequence of keywords as the additional information, which extends the model as two parallel discrete stochastic processes: a sequence of high-level coarse tokens and a sequence of natural language tokens. In practice, this model first generates a sequence of nouns, then taking the generated noun sequence as the additional input to another sequence-to-sequence network to generate the natural language response. However, the keywords-sequence generation network encountered the similar problem as the traditional sequence-to-sequence framework for short text conversation. The generated keywords-sequence tends to be short and dull. This phenomenon is also related to MLE.

MLE evaluates how the model fits the data. However, generation task follows a different operating process. First it generates a sequence of tokens, then evaluates it. In this view, the reverse KLD seems to be a better choice [8]. The reverse

KLD is the KLD between the distribution given by the model $Q(x)$ and the true distribution $P(x)$, which can be divided into two terms (1). The first term uses the negative log-level true probability density to evaluate the expected quality of the generated samples. The second term is the entropy of the distribution given by the model, which would encourage the diversity of the model. However, we cannot directly optimize this equation, because we do not know $P(x)$.

$$d_{KL}(Q|P) = E_{x\sim Q}[-\log P(x)] + E_{x\sim Q}[\log Q(x)] \tag{1}$$

However, the reverse KLD is similar to the policy gradient approach in reinforcement learning [9] if we regard the cumulative rewards in reinforcement learning as the approximated log-level true probability density. Policy gradient approach optimizes the policy to get the maximum expected cumulative rewards, which is similar to the first term of (1). This approach suffers from high variance and inefficient explorations. The entropy term prevents it from being radical [10]. The most important element of the reinforcement learning is the reward, which provides the training signal. For machine translation, we can use BLEU as the reward function. However, for short text conversation, there is no good automated evaluation method. There are two reasons that BLEU is not a good metric to evaluate the quality of the responses. First, for open domain conversation, the responses are diverse from semantic level to expression level, and several references cannot contain all the variabilities. Second, when the model is incapable of generating very good responses, it is easier for the model to focus on promoting non-essential similarities, such as stop words, tone phrases and so on, and it is not worth generating a meaningful word that is highly likely not in the references. However, if we optimize the BLEU score on the keywords-sequence level, it can compensate the second drawback of optimizing the BLEU score on the natural language responses, and avoid the disadvantage of MLE. This approach only keeps essential words left, which helps the model generate diverse responses. Meanwhile, BLEU score has some undesirable properties when used for a single sentence, since it was designed as a corpus measure. GLEU score [11] is more suitable for measuring sentence level similarity, which is consistent with BLEU score in corpus level.

In this work, a new multi-resolution generation framework is proposed. Instead of using the word-level maximum likelihood criterion, we optimize the sequence-level GLEU score of the entire generated keywords-sequence using a policy gradient approach in reinforcement learning. It successfully overcomes the drawback of MLE, generates long and more diverse keywords-sequences, thus generating better natural language responses.

2 Model Architecture

Our approach follows the framework of the multi-resolution method [7], which consists of two steps. The first is the keyword-sequence generation step, which uses a sequence-to-sequence network to generate keywords-sequence. The second is the natural language response generation step, which takes the keywords-sequence as an additional input to another sequence-to-sequence network to generate the natural language response. In the training step, the keyword-sequence,

which is the output of the first step and one of the inputs of the second step, is the ground truth extracted from the corresponding response. In the generation step, the keyword-sequence for the second step is the output of the first step.

Considering the unsatisfying result of the MLE training for the keywords-sequence generation, we train the keywords-sequence generation network with the GLEU-guided policy gradient approach.

2.1 Network Structure for Keywords-Sequence Generation

The first step is to generate keywords-sequence. The input of the model is a post, and the output of the model is a sequence of keywords. We use the sequence-to-sequence network with the attention mechanism to model the relationship between the post and the keywords-sequence.

In sequence-to-sequence generation tasks, each input X is paired with a sequence of tokens to predict: $Y = y_1, y_2, ..., y_n$. Each token is a word, and the last token y_n is a special token $<eos>$, which represents the end of a sentence. The network sequentially predicts tokens until generate $<eos>$.

Denote X_i and Y_i as the i-th post and response in the corpus. m_i and l_i are the lengths of X_i and Y_i. x_t^i and y_t^i are the t-th words in X_i and Y_i. The maximum likelihood criterion is minimizing:

$$-\sum_{i=1}^{n} \log P(Y_i|X_i) = -\sum_{i=1}^{n}\sum_{t=1}^{l_i} \log p(y_t^i|x_1^i, x_2^i, ..., x_{m_i}^i, y_1^i, y_2^i, ..., y_{t-1}^i) \tag{2}$$

The encoder is a one-layer bidirectional long short-term memory (LSTM) [12]. We concatenate the last hidden vector of each direction of the encoder as the initial hidden vector of the decoder. Traditional sequence-to-sequence model encodes the information of the post into a fixed-size vector, which cannot encode sufficient information when the post is long. To solve this issue, the attention mechanism is introduced in [13]. We also apply this method. Our decoder has two LSTM cells, which are connected in series rather than in parallel. Denote the hidden vector and the cell vector of the encoder as h_t^{enc}, c_t^{enc}. Denote the hidden vector and the cell vector of the two LSTM cells of the decoder as h_t^{dec1}, c_t^{dec1}, h_t^{dec2}, c_t^{dec2}. Denote the operations of the two LSTM cells in the decoder as f^{dec1}, f^{dec2}. The hidden vector and the cell vector of the first LSTM cell in time step t is computed according to

$$h_t^{dec1}, c_t^{dec1} = f^{dec1}(y_{t-1}, h_{t-1}^{dec2}, c_{t-1}^{dec2}) \tag{3}$$

The attention weight $a_{t,u}$ is the attention over the u-th hidden vector of the encoder at the t-th moment, which is computed by a two-layer neural network g. The input of the attention network is the hidden vector of the encoder in each time step and the current hidden vector of the first LSTM cell in the decoder (4). The attention weight $a_{t,*}$ is used for calculating the context vector ctx_t (5), which is fed to the second LSTM cell (6). The hidden vector of the second LSTM

cell h_t^{dec2} is used for predicting token and initializing the hidden vector of the first LSTM cell in the next time step.

$$a_{t,u} = \frac{\exp^{g(h_t^{dec1}, h_u^{enc})}}{\sum_{u=1}^{m} \exp^{g(h_t^{dec1}, h_u^{enc})}} \tag{4}$$

$$ctx_t = \sum_{1}^{t} a_t h_t^{enc} \tag{5}$$

$$h_t^{dec2}, c_t^{dec2} = f^{dec2}(ctx_t, h_t^{dec1}, c_t^{dec1}) \tag{6}$$

2.2 GLEU-Guided Policy Gradient Training

Usually, the sequence-to-sequence network is trained with MLE. However, as discussed before, MLE is unsuitable for generation task. Thus, we apply the policy gradient approach [9] instead.

For a given post p, there is a list of references ref_1, ref_2, ..., ref_n. Assume that the network has already generated a sequence of tokens $y_1, y_2, ..., y_{i-1}$, and is going to generate y_i. In this case, the state is the collections of p and $y_1, y_2, ..., y_{i-1}$, the action is y_i. We use GLEU score to evaluate the similarity between the references and the hypothesis. In order to avoid the sparsity of the reward signal, we do not just give the nonzero reward at the last time step. The reward of taking action y_i is designed to be the difference of the GLEU score of the hypothesis and the references before and after y_i is generated.

$$\begin{aligned} r(s_i, a_i) &= r(y_i, p, y_1, y_2, \ldots, y_{i-1}) \\ &= GLEU([ref_1, ref_2, \ldots, ref_n], [y_1, y_2, \ldots, y_i]) \\ &\quad - GLEU([ref_1, ref_2, \ldots, ref_n], [y_1, y_2, \ldots, y_{i-1}]) \end{aligned} \tag{7}$$

The goal of the policy gradient approach is to find a policy $\pi(a_t|s_t)$ which can maximize the expected return (8). In the sequence generation task, $\pi(a_t|s_t)$ is equal to $P(y_t|p, y_1, y_2, \ldots, y_{t-1})$.

$$\begin{aligned} J(\pi) &= E_{s_1, a_1, \cdots \sim \pi}[\Sigma_{t=1}^{\infty} r(s_t, a_t)] \\ &= \Sigma_{a_1, s_2, a_2, \ldots} \pi(a_1, s_2, a_2, s_3, \ldots | s_1) R_{s_1, a_1, s_2, a_2, \ldots} \end{aligned} \tag{8}$$

$R_{s_1, a_1, s_2, a_2, \ldots}$ is the cumulative reward of the state-action trajectory.

$$R_{s_T, a_T, s_{T+1}, a_{T+1}, \ldots} = \Sigma_{t=T}^{\infty} r(s_t, a_t) \tag{9}$$

Denote the parameters of the policy π as θ. Using the likelihood ratio trick, the gradient of the expected return J is

$$\begin{aligned} \nabla_\theta J(\theta) &= \Sigma_{a_1, s_2, a_2, \ldots} \nabla_\theta \pi(a_1, s_2, a_2, s_3, \ldots | s_1; \theta) R_{s_1, a_2, s_2, \ldots} \\ &= \Sigma_{a_1, s_2, a_2, \ldots} \pi(a_1, s_2, a_2, s_3, \ldots | s_1; \theta) \\ &\quad \nabla_\theta \log \pi(a_1, s_2, a_2, s_3, \ldots | s_1; \theta) R_{s_1, a_2, s_2, \ldots} \\ &= E_{s_1, a_1, s_2, \cdots \sim \pi} \nabla_\theta \log \pi(a_1, s_2, a_2, s_3, \ldots | s_1; \theta) R_{s_1, a_2, s_2, \ldots} \\ &\approx \Sigma_{t=1}^{\infty} \nabla_\theta \log \pi(a_t | s_t; \theta) R_{s_t, a_t, s_{t+1}, \ldots} \end{aligned} \tag{10}$$

The gradient of the expected return is estimated based on a single rollout trajectory according to the policy π. We sample the keyword one by one until sample $<eos>$. During training, we sample multiple trajectories at the same time to reduce uncertainty. Policy gradient approach suffers from high variance, slow convergence and inefficient exploration. It tends to learn an extreme policy, which is harmful for exploration. So, as introduced in [10], we add a weighted entropy term to prevent the policy from being extreme and encourage exploration, which also encourages the diversity of the generated keywords-sequence.

$$\nabla_\theta J(\theta) \approx \Sigma_{t=1}^{\infty} \nabla_\theta \log \pi(a_t|s_t;\theta) R_t - \gamma \Sigma_{t=1}^{\infty} \nabla_\theta \Sigma_a \pi(a|s_t;\theta) \log \pi(a|s_t;\theta) \quad (11)$$

2.3 Response Generation

The response generation network generates the natural language response given the post and the keywords-sequence. The network architecture is very similar to the keywords-sequence generation network with only several difference. In the response generation network, the output is the natural language response, but the keywords-sequence becomes another input besides the post. The keywords-sequence is encoded by a bidirectional LSTM. The hidden vector of the keyword sequence is concatenated with the hidden vector of the post to be the initial hidden vector of the decoder. The attention network is still focusing on the post. We still train the response generation network according to MLE.

3 Experiments

3.1 Data Set

We evaluate our approach on a Chinese weibo corpus. We blend the training corpus of the STC1 [14] and STC2, remove similar post-response pairs (since the corpus of the STC1 and the STC2 partially coincide), and segment the posts and responses by LTP [15]. Since one post may correspond to several responses, to avoid over-emphasizing some posts, we also truncate the post-response pairs if the corresponding post appears more than 100 times. We split part of the remaining data into the training set and the validation set. The training set has 1713277 post-response pairs and 155435 distinct posts. The validation set has 86295 post-response pairs and 8674 distinct posts. The test set has 100 distinct posts. The training set, the validation set, and the test set share no posts. We construct the vocabulary independently for the post and the response. Any word that appears in more than five different posts is included in the post vocabulary. Any word that appears in more than 25 different responses is included in the response vocabulary. Others are replaced by a special symbol $<unk>$. Keywords-sequence shares the same vocabulary with the response vocabulary, although only part of the words can be used as keywords. The size of the post vocabulary is 33187, and the size of the response vocabulary is 39278.

3.2 Training Details

For the keywords-sequence generation network and the natural response generation network, the dimension of the word-embedding is 512, and the dimension of the hidden vector is 1024. We use the ADAM optimizer [16] to train the network. The learning rate is 0.0005 for the supervised learning and 0.00005 for the policy gradient approach. The validation set is used for early stopping. Different from [7], the part-of-speech (POS) of the acceptable keywords are not limited to nouns. Nouns, verbs, and adjectives can be the keyword unless it is in the stop-word list. The stop-word list has more than one thousand words. The reason that we accept words of more POS as the keyword is that nouns cannot represent the whole response. Sometimes, a good response may not contain nouns. However, it contains at least one of the nouns, verbs or adjectives. In our experiment, we also compare different POS limitation of the keywords. When calculating the GLEU score, we only count unigram and bigram overlaps. From our point, bigram can represent the relationship between continuous keywords, but trigram is unnecessary for calculating keywords-sequences similarity. Before training the keywords-sequence generation network with policy gradient approach, the parameter is initialized according to the MLE. The entropy term weight is set to 0.0002. We sample 64 trajectories for one post at the same time.

3.3 Evaluation Methods

It is still very tough to automatically evaluate the generative conversation system. Traditional metrics used to evaluate machine translation or machine summarization, such as BLEU, ROUGE, are not suitable for open domain conversation system. Given this observations, we perform the human evaluation. There are five volunteers to annotate the results of the test set, which consists of 100 distinct posts. All the volunteers are familiar with this field. We follow the evaluation criterion of the STC2 task. The basic requirement is that the response is acceptable as a natural language text and is logically connected to the original post. The advanced requirement is that the response provide new information in the eye of the originator of the post and the assessor can judge the comment by reading nothing other than the post-response pair. If the basic requirement is not met, the label is "L0". If the basic requirement is met, but the advanced requirement is not met, the label is "L1". If the basic requirement and the advanced requirement are met, the label is "L2". Meanwhile, to tackle dull response problem, we add a special label "LC" to represent the dull response. Although the dull response does not conflict with the basic requirement, it would make the conversation boring. When calculating the final score of the system, "L2" counts two points, "L1" counts one point, "L0" and "LC" counts zero.

3.4 Results

Figure 1 shows how the GLEU score of the generated keywords-sequences varies with the progress of the training. From this figure, we can see the average GLEU

score of the keywords-sequences generated by sampling on the training and validation set, and the average GLEU score of the first or the first ten keywords-sequences generated by beam search on the validation set. The starting point is the MLE system. Although the network is initialized with supervised training, the initial average GLEU score of the keywords-sequences generated by sampling on the validation set is rather low. This is because the difference in the probability between a good keywords-sequence and a bad keywords-sequence is too small to get a reasonable keywords-sequence by sampling. However, beam search can make up for this to some extent since beam search is not as random as sampling. For the average GLEU score of the first keywords-sequence generated by beam search on the validation set, it rises from 0.221 to 0.235. In the training step, we use sampling to generate samples, because it can provide randomness. However, In the generation step, we use beam search since it can generate better keywords-sequences.

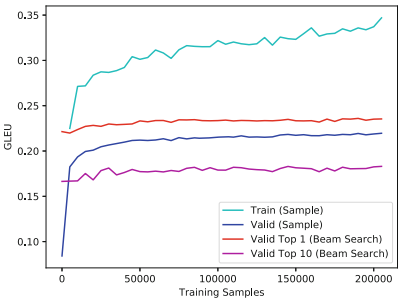

Fig. 1. Learning curve

Figure 2 shows the length and diversity statistics of the keywords-sequences on the validation set, which consists of 8674 distinct posts. The responses are generated by beam search. For each post, we count the result of the first response and the first ten responses. The x-axis is the number of training samples. The starting point of the x-axis is the MLE system. The first figure shows the number of distinct keywords-sequences. The second figure shows the number of distinct words. The last figure shows the average length of the generated keywords-sequences. Our method can generate longer and more diverse keywords-sequences. Although it seems that the word level diversity of our approach is worse than baseline, our method can utilize the combination of keywords rather than generating rare words. We think this property is good for the response generation. Longer keywords-sequences also mean the responses will contain more information.

Table 1 shows the results of the human evaluation. "S2SA" is the attention based sequence-to-sequence baseline. "MR" means the multi-resolution framework. "NOUN" means only accepting nouns as keywords while "NVA" means accepting nouns, verbs, and adjectives as the keywords. "MLE" means using supervised training method to train the keywords-sequence generation network.

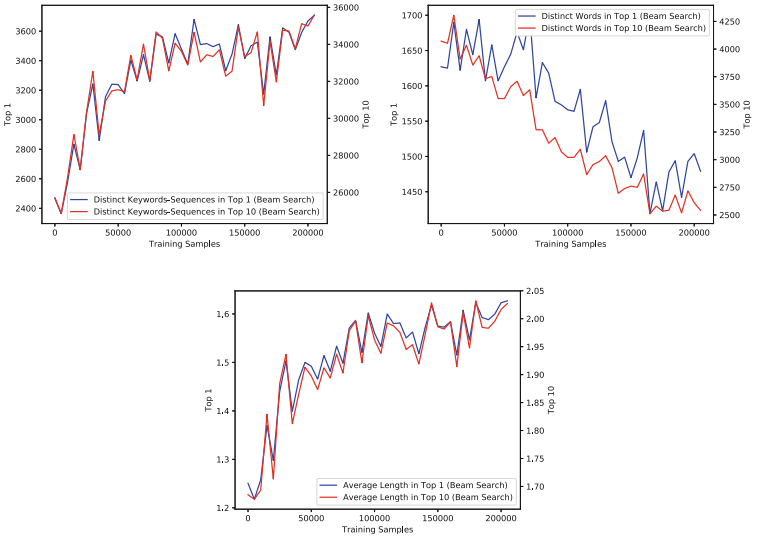

Fig. 2. Diversity and length statistics of keywords-sequences

"RL" means using policy gradient approach to train the keywords-sequence generation network. "NVA" is better than "NOUN" because it can generate less "L0" responses. This result verifies our judgment that nouns representation is not enough. Policy gradient approach does not achieve a better result on nouns level keywords-sequence since the nouns-level keywords-sequence is usually short, the GLEU-guided training method does not exert its strength. The policy gradient approach achieves a better result on "NVA" level keywords-sequence.

Table 1. Human evaluation

Model	L2	L1	L0	LC	Score
S2SA	18.8	12.2	54.4	14.6	0.498
MR+NOUN+MLE	35.4	13.8	49.0	1.8	0.846
MR+NVA+MLE	35.0	16.8	43.0	5.2	0.868
MR+NOUN+RL	35.8	12.6	48.8	2.8	0.842
MR+NVA+RL	37.0	16.8	40.2	6.0	0.908

3.5 Case Study

We provide case studies in Table 2. For each post, we show the top three keywords-sequences and the corresponding responses. MLE tends to generate short keywords-sequence. GLEU-guided policy gradient approach generates longer and more coherent keywords-sequence. This observation is consistent with the length and diversity statistics in Fig. 2.

Table 2. Generation examples

Post	吃素第一天, 坚持住, 崔朵拉。 The first day of vegetarianism, insisted, Cui Dora.	
MLE	吃素 vegetarian	是吃素吗? Is it vegetarian?
	吃 eat	吃饱了吗? Are you full?
	吃素 好 vegetarian good	吃素好了吗? Is vegetarian well?
RL	吃 减肥 eat lose weight	吃饱了再减肥 Eat enough, then lose weight
	减肥 lose weight	我也要减肥! I have to lose weight!
	吃 水果 eat fruit	吃水果了吗? Have you eaten fruit?
Post	每个人都在努力都在奋不顾身, 不是只有你受尽委屈 Everyone is hard at work, Not only you suffer grievances	
MLE	人 person	我就是这样的人 I am such a person
	人 努力 person work hard	每个人都在努力 Everyone is working hard
	说 sound	说的太对了! That sounds right!
RL	人 努力 person work hard	每个人都在努力 Everyone is working hard
	人 person	我就是这样的人 I am such a person
	努力 work hard	我在努力中··· I'm working hard ...
Post	台风要袭击香港了 Typhoon will attack Hong Kong.	
MLE	北京 Beijing	这是北京吗? Is this Beijing?
	深圳 Shenzhen	这是深圳吗? Is this Shenzhen?
	下雨 raining	这是下雨了吗? Is it raining?
RL	香港 台风 Hong Kong typhoons	香港也有台风了 There are typhoons in Hong Kong
	下雨 raining	这是下雨了吗? Is it raining?
	香港 下雨 Hong Kong raining	香港下雨了吗? Is it raining in Hong Kong?

4 Conclusion

Multi-resolution approach splits the generation task into two steps, the keywords-sequence generation step, and the natural language generation step. This approach was introduced to solve the dull response problem. Although this approach relieves the pressure of response generation and adds to the diversity of responses, it still tends to generate short and dull keywords-sequence. To tackle this problem, we apply the GLEU-guided policy gradient training, which overcomes the drawbacks of the maximum likelihood criterion and generates long and diverse keywords-sequence. The proposed method achieves better results in the human evaluation.

References

1. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
2. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685 (2015)

3. Shang, L., Lu, Z., Li, H.: Neural responding machine for short-text conversation. arXiv preprint arXiv:1503.02364 (2015)
4. Vinyals, O., Le, Q.: A neural conversational model. arXiv preprint arXiv:1506.05869 (2015)
5. Theis, L., van den Oord, A., Bethge, M.: A note on the evaluation of generative models. arXiv preprint arXiv:1511.01844 (2015)
6. Mou, L., Song, Y., Yan, R., Li, G., Zhang, L., Jin, Z.: Sequence to backward and forward sequences: a content-introducing approach to generative short-text conversation. arXiv preprint arXiv:1607.00970 (2016)
7. Serban, I.V., Klinger, T., Tesauro, G., Talamadupula, K., Zhou, B., Bengio, Y., Courville, A.C.: Multiresolution recurrent neural networks: an application to dialogue response generation. In: AAAI, pp. 3288–3294 (2017)
8. Yu, L., Zhang, W., Wang, J., Yu, Y.: Seqgan: sequence generative adversarial nets with policy gradient. In: AAAI, pp. 2852–2858 (2017)
9. Sutton, R.S., McAllester, D.A., Singh, S.P., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. In: Advances in Neural Information Processing Systems, pp. 1057–1063 (2000)
10. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning, pp. 1928–1937 (2016)
11. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al.: Google's neural machine translation system: bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
13. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
14. Shang, L., Sakai, T., Lu, Z., Li, H., Higashinaka, R., Miyao, Y.: Overview of the NTCIR-12 short text conversation task. In: NTCIR (2016)
15. Che, W., Li, Z., Liu, T.: LTP: a Chinese language technology platform. In: Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations, pp. 13–16. Association for Computational Linguistics (2010)
16. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)