Neural Question Generation with Semantics of Question Type

Xiaozheng Dong, Yu Hong(✉), Xin Chen, Weikang Li, Min Zhang, and Qiaoming Zhu

School of Computer Science and Technology of Jiangsu Province,
Soochow University, Suzhou 215006, Jiangsu, China
{xzdong,xchen121,wkli88}@stu.suda.edu.cn
{hongy,minzhang,qmzhu}@suda.edu.cn

Abstract. This paper focuses on automatic question generation (QG) that transforms a narrative sentence into an interrogative sentence. Recently, neural networks have been used in this task due to its extraordinary ability of semantics encoding and decoding. We propose an approach which incorporates semantics of the possible question type. We utilize the Convolutional Neural Network (CNN) for predicting question type of the answer phrases in the narrative sentence. In order to incorporate the question type semantics into the generating process, we classify the question type which the answer phrases refer to. In addition, We use Bidirectional Long Short Term Memory (Bi-LSTM) to construct the question generating model. The experiment results show that our method outperforms the baseline system with the improvement of 1.7% on BLEU-4 score and beyonds the state-of-the-art.

Keywords: Question generation · Question type · Answer phrases

1 Introduction

The goal of automatic question generation is to create natural questions from answer phrases in the narrative sentence, where the generated questions can be answered by them. Normally, the answer phrases are short texts in the sentence. Listed below are two questions generated by the same narrative sentence, where the S_{nar} is an original narrative sentence and the S_{que} 1 and S_{que} 2 are the questions generated respectively based on the answer phrases AP 1 and AP 2.

(1) $\mathbf{S}_{nar}$: *maududi founded the jamaat–e–islami party in 1941 and remained its leader until 1972.*
$\mathbf{S}_{que}$ **1**: *when did maududi found the jamaat-e-islami party?*
AP 1: *in 1941 and remained its leader until 1972*
$\mathbf{S}_{que}$ **2**: *who found the jamaat-e-islami party?*
AP 2: *maududi*

M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11109, pp. 213–223, 2018.
https://doi.org/10.1007/978-3-319-99501-4_18

Question generation system is widely applied to many areas, such as reading comprehension, healthcare administration, knowledge-based question answering (KB-QA), and so on. For example, we can build a QA knowledge base by the generated questions and the raw narrative sentences. Incorporating the existing information retrieval technique into the knowledge base, we can create a practical QA system easily. In this case, the S_{nar} can serve as the answer, while the S_{que} 1 or S_{que} 2 serve as the questions.

It is challenging to generate questions in an automatic way. Not only the narrative sentence and the generated question share similar semantics, but also the generated question type should be correct. At the same time, the generated question ought to be a natural quesiton. As we can see in the example 1), S_{nar} and S_{que} 1 share similar semantics, "a person creates a party at some time", but are represented in different ways. Furthermore, QG can boil down to a translation problem. Therefore, existing works usually apply the translation models to handle this task [1], due to their brilliant ability of semantic encoding and decoding, especially reordering the sentence.

In this paper, we propose to utilize the question type prediction for phrases to improve the existing translation model. Adding the question type to the encoding and decoding process provides extra information to specify which type of question to generate, thus improving the performance of the translation model. In the beginning, we predict the question type which those ground truth answer phrases refer to. As shown in example 1), the AP 1 refers to a when type question and the AP 2 refers to a who type question. Then the question type is incorporated into the translation model based on Bi-LSTM [11], with the aim to provide more information for the encoding and decoding process.

Experiments are conducted on the Stanford Question Answering Dataset (SQuAD) [16], and the results show that even using such a classification model, with a precision of about 67%, for question type prediction, our translation model outperforms the baseline by increasing 1.7% on BLEU-4.

In the section below we discuss related work (Sect. 2), the details of our approach (Sect. 3) and describe our experiment setup (Sect. 4). We analyze the results in Sect. 5. Lastly, we conclude the paper in Sect. 6.

2 Related Work

Question generation has attracted the attention of the natural language generation (NLG) community, since the work of Rus et al. [17]

Heilman et al. [10] use the drafting rules to transform the declarative sentences and reorder the generated questions through the logistic regression (LR) model. They rank the generated questions and obtain the former 20% as the generation results, which nearly doubles the percentage of the questions rated as acceptable by annotators up to 52%. In addition, Liu et al. [13] apply a similar method to generate Chinese questions.

Du et al. [8] attempt to apply a Bi-LSTM network to generate questions. They respectively generate questions for sentences and paragraphs and get the best performance in automatic evaluation method, and the acceptability is also higher than the rule-based method by human evaluation. Duan et al. [9] utilize neural network model based on CNN and Bidirectional Gated Recurrent Unit (Bi-GRU) [4] model to generate question templates and then transform them into questions. Furthermore they exploit the generated question to assist QA [19] and get a better answer. Zhou et al. [20] add the syntactic feature and part-of-speech feature to the model based on Bi-GRU.

3 Approach

This paper proposes a method which merges question type prediction model and neural translation model. The former one is a CNN model, which is designed to predict the possible question type of the answer phrase in the sentence. The later one is a sequence-to-sequence model based on Bi-LSTM that aims to generate target questions. The structure of the entire system is shown in Fig. 1. Two modules are kept intact within a single pipeline stage. AP is an answer phrase in a sentence and also the focus of the artificial question. We replace the answer phrase with its interrogative pronoun that fetched from the question type prediction. After processing, a new sentence is used as the input of generate question model, whose structure is shown in Fig. 2.

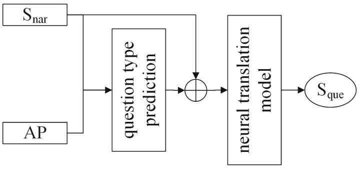


Fig. 1. The structure of the entire system

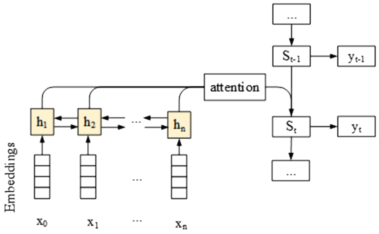


Fig. 2. Neural translation model

As shown in example 1), a “who” label can be assigned for AP 2. Then we replace AP 2 with “who” to form a new sentence $\mathrm{S}_{nar}*$ for the S_{nar}, which is listed below. $\mathrm{S}_{nar}*$ contains the semantics of question type and will be used for question generation.

> $\mathbf{S}_{nar}*$: ***who*** *founded the jamaat−e−islami party in 1941 and remained its leader until 1972.*

3.1 Question Type Prediction

As shown in Fig. 3, the question type prediction model is a slight variant of the CNN architecture. The input of the model contains a sentence S_{nar} and an answer phrase AP. The output is one of the following 13 labels, including

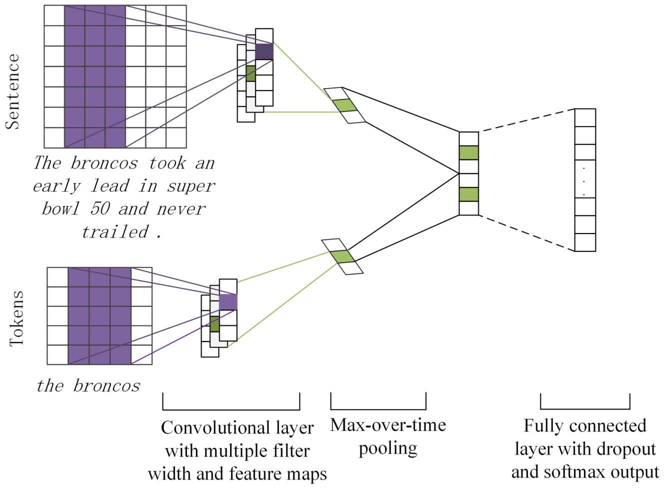

Sentence
The broncos took an early lead in super bowl 50 and never trailed .
Tokens
the broncos
Convolutional layer with multiple filter width and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Fig. 3. The question type prediction model.

"how much, how many, what, how long, which, where, how often, when, why, whose, who, how, other.". Not only the meaning of the answer phrase, but also the effect of the sentence needs to be considered. Given a sentence with n words $\mathrm{S}_{nar} = [x_1^c, x_2^c, ..., x_n^c]$ and an answer phrase with m words AP $= [t_1, t_2, ..., t_m]$, AP $\in \mathrm{S}_{nar}$. We use x_i^c and t_j to denote embedded vector of i-th word in S_{nar} and j-th word in AP.

In general, let $x_{i:i+h-1}^c$ refer to the concatenation of word embeddings x_i^c, x_{i+1}^c, ..., x_{i+h-1}^c and $t_{j:j+l-1}$ refer to the concatenation of word embeddings t_j,t_{j+1},...,t_{j+l-1}. A convolution operation involves two filters $W_1 \in \mathbb{R}^{hk}$ and $W_2 \in \mathbb{R}^{lk}$. W_1 is applied to a window of h words in a sentence and W_2 is applied to a window of l words in an answer phrase. c_i^x and c_i^t are generated by:

$$c_i^x = f(W_1 * x_{i:i+h-1}^c + b_x) \ \ and \ \ c_i^t = f(W_1 * t_{i:i+l-1} + b_t) \tag{1}$$

here $b_x \in \mathbb{R}$ and $b_t \in \mathbb{R}$ are bias term and f is non-linear function such as the hyperbolic tangent. W_1 filter is applied to produce a feature map. Similarly, the answer phrase is also manipulated by the filter W_2.

$$c_1 = [c_1^x, c_2^x, ..., c_{n-h+1}^x] \ \ and \ \ c_2 = [c_1^t, c_2^t, ..., c_{m-l+1}^t] \tag{2}$$

We then apply a max-over-time pooling operation [5] over the feature maps and take the maximum value $c = [max\ c_1; max\ c_2]$ as the feature corresponding to particular filters. The process is to capture the most feature, one with the highest value, for each feature map. The pooling scheme naturally deals with variable lengths of sentence and answer phrases. Upon the hidden layer, we stack a *softmax* layer for interrogative pronoun determination:

$$y_{label} = g(W * c + b) \tag{3}$$

where g is a *softmax* function, W is a parameter matrix and b is a bias term.

3.2 Neural Translation Model

The neural translation model is constructed based on Bi-LSTM. The encoder reads a word sequence of an input sentence $S_{nar}* = \{x_1, x_2, ..., x_s\}$, which contains the semantics of the possible question type. Let x_i refers to i-th word in a narrative sentence. The decoder predicts a word sequence of an output question $S_{que} = \{y_1, y_2, ..., y_q\}$, let y_i refer to i-th word in a question. The attention mechanism used in this model is adopted from Du et al.'s [8] work. The probability of generating a question Q in the decoder is computed as:

$$P(S_{que}) = \prod_{i=1}^{|S_{que}|} P(y_i|y_{<i}, c_i) \tag{4}$$

$$P(y_i|y_{<i}, c_i) = softmax(W_s tanh(W_i[h_i; c_i])) \tag{5}$$

the $softmax$ denotes a non-linear function that outputs the probability of generating y_i. h_i is computed as:

$$h_i = LSTM(y_{i-1}, h_{i-1}) \tag{6}$$

here, $LSTM$ [11] generates the new state h_i by the representation of previously generated word y_{i-1} (obtained from a word look-up table), and previous state h_{i-1}. c_i denotes the context vector, which is computed as:

$$c_i = \sum_{i=1,...,|x|} a_{i,t} b_i \quad and \quad a_{i,t} = \frac{exp({v_a}^T W_b b_i)}{\sum_j exp({v_a}^T W_b b_j)} \tag{7}$$

where ${v_a}^T$ and W_b are weights. b_i denotes the i^{th} hidden state of the encoder, which is the concatenation of the forward hidden state $\overrightarrow{b_i} = \overrightarrow{LSTM}(x_i, b_{i-1})$ and the back forward state $\overleftarrow{b_i} = \overleftarrow{LSTM}(x_i, b_{i+1})$.

4 Experimental Setup

Our method is experimented on the processed SQuAD dataset. In this section, we firstly describe the corpus. Then we give implementation details of our processing and the baselines to compare.

4.1 Dataset and Evaluation Methods

The SQuAD corpus is annotated by crowd-workers, we train the prediction model and the translation model through the processed data. Our data division refers to Du et al. [8]. Table 1 provides some statistics on the processed dataset.

The SQuAD corpus are used for training question type prediction model and neural translation model and testing. For the former, we can determine the interrogative labels of answer phrases.

We use simple and useful rules to construct the data set for the prediction model. In order to fetch the ground truth question type labels which the answer phrases refer to, we detect the interrogatives in the questions of the SQuAD to determine whether they contains the former 12 interrogative labels in Table 1. The matching order is from left to right. Once the question contains a label, we will commit the label to the answer phrase and terminate the matching. If not, we will set "*other*" label.

Table 1. Dataset (processed) statistic

# pairs(Train)	70484
# pairs(Dev)	10570
# pairs(Test)	11877
S_{nar}: avg.tokens	32.9
S_{que}: avg.tokens	11.3
AP: avg tokens	3.4

For the neural translation model, we utilize the ground truth question type labels for changing the raw sentence in training process (Sect. 3). While in the test process, we use the question type labels produced by the CNN classifier. The target is still an artificial question.

We adopt the micro-averaged precision (P), recall (R) and F1 score to evaluate the performance of the prediction model. The evaluation package released by Chen et al. [3] serves as the evaluation measures for question generation, which was originally used to score image captions in the generation task. The package includes BLEU-1, BLEU-2, BLEU-3, BLEU-4 [14], METEOR [7] and ROUGE$_L$ [12] evaluation scripts.

4.2 Implementation Details

We will describe the experimental parameters of the prediction model and the translation model. The output of the prediction model is a label while that of the translation model is a natural question. We use 300 dimensional word embedding pre-trained by the glove.840B.300d [15] for initialization, and fix the word representations during training. The experimental parameters of the prediction model and the translation model will be described respectively, and the parameters of two models are shown in Table 2.

Table 2. Hyperparameters used in our experiments.

Question Type Prediction Model				Neural Translation Model			
Parameters	Values	Parameters	Values	Parameters	Values	Parameters	Values
S_{nar} filter size	3	S_{nar} length	100	sentence max-length	100	dropout rate	0.3
Ap filter size	3	AP length	50	source vocabulary	40k	learning rate	0.5
dropout rate	0.5	batch size	64	target vocabulary	28k	hidden size	600
hidden size	100	-	-	batch size	64	layers	2

For the prediction model, the loss function is categorical-crossentropy [6], the optimizer is Ada [18]. For the translation model, the number of LSTM layer is 2 for both encoder and decoder. It uses SGD [2] for optimization, with an initial learning rate of 1.0. We start halving the learning rate at epoch 8, and fix the gradient as 5 when it beyonds 5. During decoding process, we do beam search with a size of 5. Finally, the decoding process stops when every beam in the stack generates the EOS token.

All hyperparameters of our models are tuned in the development set. The results are reported on the test set.

4.3 Experiment Setup

To prove the effectiveness of our method, we compare it with several competitive systems. Now we briefly introduce their approaches.

DirectIn is an intuitive yet meaningful baseline in which the longest subsentence is *d*irectly taken as predicted question. To split the sentence into subsentences, we use a set of splitters, *i.e.*, {"?", "!", ",", ".", ";"}.

H&S [10] is a rule-based overgenerate-and-rank system. When running the system, we set the parameter *just*-wh "*false*" and set max-length equal the longest sentence in training set. We take the top question in the ranked list.

NQG-LSTM [8] is a basic encoder-decoder learning system for question generation. Bi-LSTM is used for the encoder and LSTM is used for decoder. The system uses the raw question-sentence pairs.

NQG-GRU makes a slight change in NQG-LSTM model.In the model, we replace LSTM network with GRU network for question generation.

NQG++ [20] is different with NQG-GRU. The copy mechanism is added the model, and the encoder and decoder share the pre-train vectors. In addition, we only report the paper's score without model.

5 Result and Analysis

The experiment report contains the results of the question type prediction model and the neural translation model. The performance of the former has a direct impact on the later. We select the best model on the development set.

The prediction model achieves score with 67.78% P, 66.80% R and 60.58% F1 on the test set. According to performance, the labels determined by the model are used to replace the answer phrases in the sentences, and new sentences are produced as the source input for the neural translation model to generate questions. In order to verify the impact of the performance of question type prediction, we use the same question generation model based on correct labels. We name the generation system using correct label as "CL-QG". Table 3 shows the results of our method and some comparative systems.

According to the results, our performance achieves the state-of-the-art Comparing with the rule-based system H&S and DirectIn. The BLEU-4 score is increased about 2.6%, and the ROUGE$_L$ and METEOR value are respectively 17.96% and 54.74%.

Furthermore, the experimental score of NQG-GRU is the lower than NQG-LSTM, because the representation of sentence is Inadequate. We adopt Bi-LSTM in neural translation model for question generation. In these methods using neural translation model, our method performs better than the NQG++ system

Table 3. Results of generating questions. (n/a: the paper didn't list results of this task)

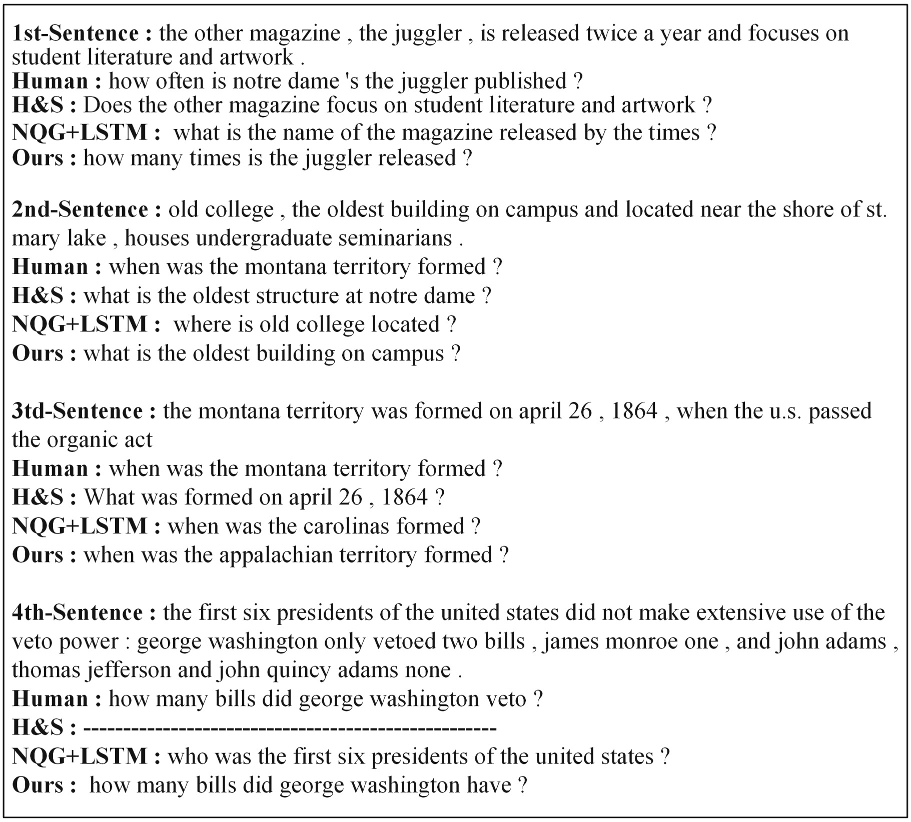
Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE$_L$
DirectIn	0.3171	0.2118	0.1511	0.1120	0.1495	0.2247
H&S	0.3850	0.2280	0.1552	0.1118	0.1595	0.3098
NQG-GRU	0.2563	0.0990	0.0518	0.0310	0.0779	0.2846
NQG-LSTM	0.4288	0.2570	0.1728	0.1210	0.1644	0.3967
NQG++	n/a	n/a	n/a	0.1329	n/a	n/a
ours	**0.4572**	**0.2826**	**0.1934**	**0.1376**	**0.1796**	**0.4245**
CL-QG	0.4837	0.3079	0.2151	0.1556	0.1934	0.4574

with the highest performance and far exceeds the baseline NQG-LSTM system. This shows the question type prediction is helpful for question generation.

Although the recall rate of the prediction model is only about 67%, the promotion has a significant effect on the question generation performance. Comparing with "CL-QG", there is still room for growth in our method. The performance of the NQG-GRU system is lower, and NQG++ model are unknown here. So the generation results are shown in Fig. 4. For our qualitative analysis, we examine the sample outputs generated by H&S and our method. There exists a large gap between our results and H&S's. In the first two samples, the H&S only performs some syntactic transform over the input without paraphrasing, but our generated questions are "*wh*"-question and have higher reasoning. In the third sample, our model can successfully pay attention in "*april* 26, 1864". For the last sentence, the H&S system can not generate a question in that the sentence's length beyond its ability. These show that our method is better than rule-based system.

Even though NQG-LSTM utilizes the semantics of the sentence, the generated question labels are hardly similar to that of the ground truth questions. Such as the second sample, NQG-LSTM produces a question of "*where*" rather than "*what*". Our method easily detects the correct question type. Furthermore, NQG-LSTM model creates a good question by focusing on the answer phrase "*george washington*" in the fourth sentence, but it is wrong. We assign "*how many*" label to "*two bills*" through the prediction model and generate more similar question to artificial question.

Our system has the following advantages in generating questions: (1) Different from the systems based on rules, the generated questions are more reasonable in ours method. (2) Compared with some models based on neural network model, our method generated questions which are more fitting to the artificial questions. (3) our method outperforms the state-of-the-art.



1st-Sentence : the other magazine , the juggler , is released twice a year and focuses on student literature and artwork .
Human : how often is notre dame 's the juggler published ?
H&S : Does the other magazine focus on student literature and artwork ?
NQG+LSTM : what is the name of the magazine released by the times ?
Ours : how many times is the juggler released ?

2nd-Sentence : old college , the oldest building on campus and located near the shore of st. mary lake , houses undergraduate seminarians .
Human : when was the montana territory formed ?
H&S : what is the oldest structure at notre dame ?
NQG+LSTM : where is old college located ?
Ours : what is the oldest building on campus ?

3td-Sentence : the montana territory was formed on april 26 , 1864 , when the u.s. passed the organic act
Human : when was the montana territory formed ?
H&S : What was formed on april 26 , 1864 ?
NQG+LSTM : when was the carolinas formed ?
Ours : when was the appalachian territory formed ?

4th-Sentence : the first six presidents of the united states did not make extensive use of the veto power : george washington only vetoed two bills , james monroe one , and john adams , thomas jefferson and john quincy adams none .
Human : how many bills did george washington veto ?
H&S : --
NQG+LSTM : who was the first six presidents of the united states ?
Ours : how many bills did george washington have ?

Fig. 4. Sample output questions generated by human, our system, NQG-LSTM and H&S system.

6 Conclusion

In the paper, we propose a novel method for question generation which integrates the question type into the generating process. Only in this way can we acquire the representation of sentence which contains the semantic of question types. This makes question generation model preform better in that it accesses more semantic information. In the future, we will improve the performance of the question type prediction to generate better questions.

Acknowledgements. This work was supported by the national Natural Science Foundation of China via Nos. 2017YFB1002104, 61672368 and 61672367.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Lechevallier, Y., Saporta, G., (eds.) Proceedings of COMPSTAT 2010, pp. 177–186. Springer (2010). https://doi.org/10.1007/978-3-7908-2604-3_16
3. Chen, X., Fang, H., Lin, T.Y., Vedantam, R., Gupta, S., Dollár, P., Zitnick, C.L.: Microsoft coco captions: data collection and evaluation server. arXiv preprint arXiv:1504.00325 (2015)
4. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
5. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**(Aug), 2493–2537 (2011)
6. De Boer, P.T., Kroese, D.P., Mannor, S., Rubinstein, R.Y.: A tutorial on the cross-entropy method. Annal. Oper. Res. **134**(1), 19–67 (2005)
7. Denkowski, M., Lavie, A.: Meteor universal: Language specific translation evaluation for any target language. In: Proceedings of the Ninth Workshop on Statistical Machine Translation, pp. 376–380 (2014)
8. Du, X., Shao, J., Cardie, C.: Learning to ask: Neural question generation for reading comprehension. arXiv preprint arXiv:1705.00106 (2017)
9. Duan, N., Tang, D., Chen, P., Zhou, M.: Question generation for question answering. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 866–874 (2017)
10. Heilman, M., Smith, N.A.: Good question! statistical ranking for question generation. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 609–617. Association for Computational Linguistics (2010)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
12. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. Text Summarization Branches Out (2004)
13. Liu, M., Rus, V., Liu, L.: Automatic chinese factual question generation. IEEE Trans. Learn. Technol. **10**(2), 194–204 (2017)
14. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting on association for computational linguistics, pp. 311–318. Association for Computational Linguistics (2002)
15. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
16. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250 (2016)
17. Rus, V., Wyse, B., Piwek, P., Lintean, M., Stoyanchev, S., Moldovan, C.: The first question generation shared task evaluation challenge. In: Proceedings of the 6th International Natural Language Generation Conference, pp. 251–257. Association for Computational Linguistics (2010)
18. Schwarz, B., Kirchgässner, W., Landwehr, R.: An optimizer for ADA-design, experiences and results. In: ACM SIGPLAN Notices, vol. 23, pp. 175–184. ACM (1988)

19. Voorhees, E.M., et al.: The TREC-8 question answering track report. In: Trec, vol. 99, pp. 77–82 (1999)
20. Zhou, Q., Yang, N., Wei, F., Tan, C., Bao, H., Zhou, M.: Neural question generation from text: a preliminary study. In: Huang, X., Jiang, J., Zhao, D., Feng, Y., Hong, Y. (eds.) NLPCC 2017. LNCS (LNAI), vol. 10619, pp. 662–671. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73618-1_56