A Neural Question Generation System Based on Knowledge Base

Hao Wang, Xiaodong Zhang, and Houfeng Wang(✉)

MOE Key Lab of Computational Linguistics,
Peking University, Beijing 100871, China
{hhhwang,zxdcs,wanghf}@pku.edu.cn

Abstract. Most of question-answer pairs in question answering task are generated manually, which is inefficient and expensive. However, the existing work on automatic question generation is not good enough to replace manual annotation. This paper presents a system to generate questions from a knowledge base in Chinese. The contribution of our work contains two parts. First we offer a neural generation approach using long short term memory (LSTM). Second, we design a new format of input sequence for the system, which promotes the performance of the model. On the evaluation of KBQG of NLPCC 2018 Shared Task 7, our system achieved 73.73 BLEU, and took the first place in the evaluation.

Keywords: Question answering · Generation · Knowledge base

1 Introduction

Question answering (QA) is a practical and challenging problem of artificial intelligence (AI). QA contains several tasks, like text based QA, knowledge based QA and table based QA. To train a QA system, it is important to get labeled data with high quality. The performance of a QA system is highly depend on the quality of the training QA pairs. However, most of labeled QA pairs, such as [2,15], are labeled manually, which is inefficient and expensive. Because of that, size of the training sets are limited.

Automatic question generation has already been a challenge task for expanding size of data for QA systems. Zhou et al. [18] worked on generating questions from raw text, which has rich information. However, structured knowledge contains less text information, which make it harder to generate questions from them. Although there are some work on question generation with multiple facts [4], questions that only contain one fact are much more common than those with several facts. Therefore, we work on generating questions with only one fact.

Most of previous works, like Rus et al. [12], are based on hand-crafted rules. However, designing a well-defined rules takes much human effort, and it performs terrible on new situations, which means the rules need to be updated frequently. Besides, limited to the form of templates, the questions generated by those methods are lack of diversity.

There are also some previous works based on seq2seq model [3], which is widely used by machine translation [6,10], summarization [7,11] and dialog generation [14,17]. Serban et al. [13] works on the SimpleQuestions [2] dataset in English, and release an English question generation dataset based on SimpleQuestions. Liu et al. [9] works on generating Chinese questions by using a template-based manner, but the generated questions is still different from those asked by human. As we can see from Fig. 1, conjunctions and question words in the output of T-seq2seq are generated improperly.

In this paper, we propose a new system to generate questions from a Chinese structured knowledge base. In NLPCC 2018 Shared Task 7, our system won the first place on the KBQG subtask. Our contributions contain three parts: (1) we improve the seq2seq model, which performs better on question generation task; (2) our system has different input with the previous work, which helps the model to generate questions with proper conjunction words; and (3) we evaluate our model by comparing with other different models, and show the contribution of each part in our model.

Fact	Input Triple	T-seq2seq	Ours
#1	张力 ||| 儿子 ||| 张量 Zhang Li ||| Son ||| Zhang Liang	张力的儿子**是什么**？ **What** is Zhang Li’s son?	张力的儿子**叫什么**？ **What’s the name** of Zhang Li’s son?
#2	董学升 ||| 祖籍 ||| 潮州市潮安区彩塘镇 Dong Xuesheng ||| Ancestral Home ||| Caitang Town	董学升的祖籍**是什么**？ **What** is Dong Xuesheng’s ancestral home?	董学升的祖籍**在哪**？ **Where** is Dong Xuesheng’s ancestral home?
#3	广州富力足球俱乐部 ||| 成立时间 ||| 2011年6月25日 Fuli Football Club ||| Time of foundation ||| June 25th, 2011	广州富力足球俱乐部是**什么时间**成立的？ **What is the foundation time** of Fuli Football Club?	广州富力足球俱乐部是**什么时候**成立的？ **When** was Fuli Football Club founded?

Fig. 1. The questions generated by Template-based seq2seq learning (T-seq2seq) [9] and our system. The questions our model generated are more similar to those asked by human. The differences of generation are shown in black.

2 Task Description

A KB, or knowledge base, is a structured database, which contains some entities and the relationships between those entities. Each relationship connects two entities. For example, *Phoenix Mountain* and *Shanxi Province* are connected by the relation *location*, which means that *Phoenix Mountain* is located in *Shanxi Province*. Those two entities with a relationship formed a triple of KB.

Given a triple $I = (\text{subject}, \text{relationship}, \text{object})$, and the goal of our system is to generate a question $q = \{c_1, c_2, ..., c_n\}$, which can be answered based on the given triple. Besides, to use those generated question to train the QA model, we hope the generation is similar with those asked by human.

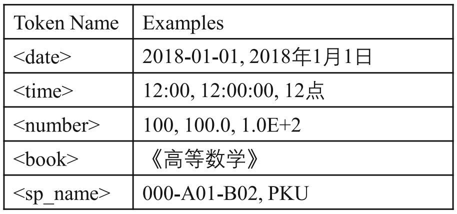
The performance of the system is evaluated by comparing the questions asked by human and machine.

3 Structure of Question Generation Model

In this section, we introduce the structure of our generation model. Our model is based on seq2seq model, which is an universal neural model on generating sequences. The model can be divided into an encoder of input triple, and a question decoder.

3.1 Construction of Input Sequence

In each triple I, we have subject, relationship and object. As we know, for most cases, subject and relationship should be shown in the generation. Although object would not be shown directly in the question, it is also important for the system to understand the target of the question. Therefore, the answer object is converted into some special tokens, which are shown on Fig. 2. <date> represents all phrases of dates, and <time> represents the specific time. For example, in the input sequence, "1月1日 12:00" (12:00, Jan.1) will be replaced by "<date> <time>". <number> represents a number is replaced here. Name of books are convert to <book>. <sp_name> represents all other phrases in western characters, numbers and some symbols, which are widely used of name of products and some foreign people. It is easy to find out those phrases by using patterns. When generating input sequence, we convert those phrases into special tokens, and the sequence is named as token_seq.



Token Name	Examples
<date>	2018-01-01, 2018年1月1日
<time>	12:00, 12:00:00, 12点
<number>	100, 100.0, 1.0E+2
<book>	《高等数学》
<sp_name>	000-A01-B02, PKU

Fig. 2. Some examples of special tokens used in our system.

However, it is hard to find Chinese name by using patterns, but it is important for the system to recognize those name. Therefore, we use HanLP toolkit[1], a framework that provides some core processing tools, to find names of people in Chinese. In HanLP, names can be found by role tagging [16]. After converting the

[1] https://github.com/hankcs/HanLP.

input sequence to token_seq, we replace words in the object of triple, excluding some conjunctions and adjectives, to their POS tags. The new sequence is called token_pos_seq.

For subjects of triples are used to replace <ent> token of the output sequence, they are not part of the input sequences. Besides, a special token <is> is inserted to split the relationship and the object. Examples of the input sequences are shown in Fig. 3. In Fig. 3, "nr" represents the token is name of a person, and all POS tags which start with "v" represent verbs, and those start with "q" represent quality words.

Original Triple	Name	Input Sequence
恩万科沃·卡努 ||| 欧冠联赛 ||| 出场55次，进5球 Nwankwo Kanu ||| UEFA Champions League ||| 55 appearance, 5 goals	token_pos_seq	欧 冠 联 赛 <is> vi **<number>** qv vf **<number>** UEFA Champions League <is> vi **<number>** qv vf **<number>**
	token_seq	欧 冠 联 赛 <is> 出 场 **<number>** 次，进 **<number>** 球 UEFA Champions League <is> **<number>** appearance, **<number>** goals
棉花枯腐病 ||| 病原拉丁学名 ||| macrophomina phaseoli ashby Cotton rot ||| Pathogenic Latin name ||| macrophomina phaseoli ashby	token_pos_seq	病 原 拉 丁 学 名 <is> **<sp_name>** Pathogenic Latin name <is> **<sp_name>**
	token_seq	病 原 拉 丁 学 名 <is> **<sp_name>** Pathogenic Latin name <is> **<sp_name>**
赵佩茹 ||| 师承 ||| 相声老前辈焦寿海先生 Zhao Peiru ||| Teacher ||| Mr. Jiao Shouhai, Senior of Xiangsheng	token_pos_seq	师 承 <is> **nr** Teacher <is> **nr**
	token_seq	师 承 <is> 相 声 老 前 辈 焦 寿 海 先 生 Teacher ||| Mr. Jiao Shouhai, Senior of Xiangsheng

Fig. 3. Some examples of input sequence.

3.2 Encoder

Given a input sequence $X = \{x_1, x_2, ..., x_n\}$, where n is the length of input sequence, and suppose we have a vocabulary V, which contains characters and tokens. Each $x_i \in \mathbb{R}^{|m|}$ is a one-hot vector, which represents a token in the vocabulary. The encoder converts those tokens first into their embeddings $\{e_1, e_2, ..., e_n\}$, by looking up the embedding matrix $E \in \mathbb{R}^{|V| \times m}$, where m is the size of embedding. Then, we use a 3-layer LSTM network to encode the input sequence x. The structure of each LSTM layer is:

$$f_t = \sigma(W_f \cdot [h_{t-1}, e_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, e_t] + b_i) \tag{2}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, e_t] + b_C) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, e_t] + b_o) \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

In Eqs.(1–6), h_t is the hidden state of the LSTM layer at time t, and $e_t, o_t \in \mathbb{R}_n$ are input and output of each layer at step t. $W_f, b_f, W_i, b_i, W_C, b_C, W_o, b_o$ are trainable parameters.

3.3 Decoder

The structure of the decoder is shown in Fig. 4. Our decoder is a neural network that models the conditional probability $p(y|H)$, while $H = \{h_1, h_2, ..., h_n\}$ is the hidden states of encoder. We use an LSTM network with attention and CopyNet [5] as our decoder.

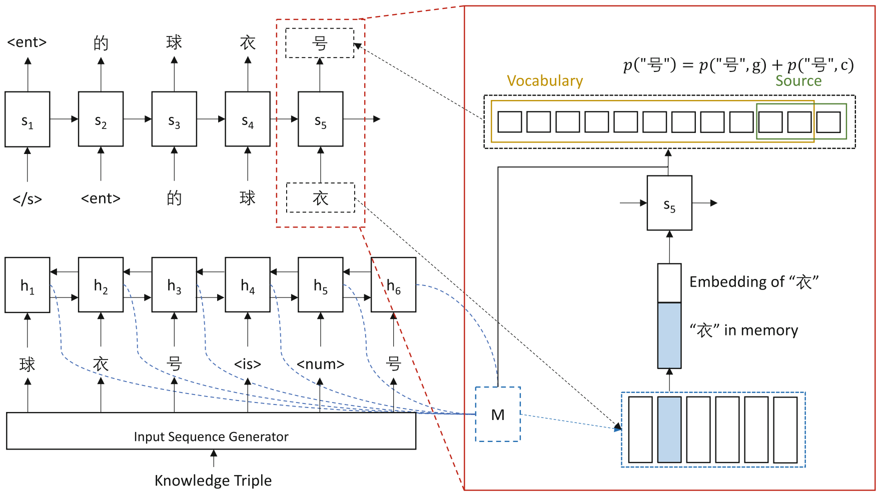


Fig. 4. Structure of the decoder. Hidden states of the encoder are used in decoding.

We have a vocabulary list V, which is same as the vocabulary for encoder, and use <UNK> to represent the OOV (out-of-vocabulary) word. Since there are some words that not contained in V are useful, words in the input sequence X are also included in generation candidates, which makes CopyNet to output some OOV words. To read the states from encoders, we build a memory matrix M, which is initialized by $\{h_1, h_2, ..., h_n\}$, and will be updated while decoding. For each step t, we can get the probability of generating any target word y_t as,

$$p(y_t|\mathbf{s}_t, y_{t-1}, c_t, M) = p(y_t, \mathbf{g}|s_t, y_{t-1}, c_t, M) + p(y_t, \mathbf{c}|s_t, y_{t-1}, c_t, M) \quad (7)$$

where $\mathbf{g}$ represents the generate-mode, and $\mathbf{c}$ represents the copy-mode. The two probability are calculated by

$$p(y_t, \mathbf{g}|\cdot) = \begin{cases} \frac{1}{Z} e^{\psi_g(y_t)}, & y_t \in V \\ 0, & y_t \in X \cap \overline{V} \\ \frac{1}{Z} e^{\psi_g(\text{<UNK>})} & y_t \notin X \cup V \end{cases}; \quad (8)$$

$$p(y_t, \mathbf{c}|\cdot) = \begin{cases} \frac{1}{Z} \sum_{j:x_j = y_t} e^{\psi_c(y_t)}, & y_t \in X \\ 0 & y_t \notin X \end{cases}. \quad (9)$$

In Eqs. (8–9), ψ_g and ψ_c are score functions for two modes, while Z is the normalization term shared by both two probabilities. The score of copy-mode is

$$\psi_c(y_t) = \tanh(h_j^T W_c) s_t, x_j \in X. \tag{10}$$

In Eq. (10), hidden states in M are used to represent the input sequence. Location of each token in the sequence is important for copying. ψ_g is similar with the scorer function in the basic seq2seq model. It is based on a 3-layer LSTM network with attention. Input of the first LSTM layer is embedding of the former output word. Attention of our model is inspired by [10], which gets a great improvement to seq2seq models. That is,

$$\psi_g(y_t) = \mathbf{v}_i^T W_s \tanh(W_t[s_t, a_t]) \tag{11}$$

In Eq. (11), $\mathbf{v}_i$ is the one-hot indicator vector for v_i, and s_t is the t-th hidden state of the decoder LSTM network. a_t is the attention vector calculated by hidden states:

$$a_t = \sum_{i=1}^{n} \alpha_{t_i} h_i \tag{12}$$

while α_{t_i} is the normalization weight calculated by:

$$\alpha_{t_i} = \frac{e^{g(s_t, h_i)}}{\sum_{j=1}^{n} e^{g(s_t, h_j)}}, \tag{13}$$

where $g(s_t, h_j)$ represents the relevance between the hidden state for decoder and encoder. In our model, the relevance score is measured by

$$g(s_t, h_j) = s_t^T h_j. \tag{14}$$

The input y_{t-1} of decoder is different with the original seq2seq network. It is represent as $[e(y_{t-1}); \zeta(y_{t-1})]^T$, where $e(y_{t-1})$ is the word embedding of y_{t-1}, and $\zeta(y_{t-1})$ is the weighted sum of M, i.e.

$$\zeta(y_{t-1}) = \sum_{\tau=1}^{n} \rho_{t\tau} h_\tau, \tag{15}$$

$$\rho_{t\tau} = \begin{cases} \frac{1}{K} p(x_\tau, \mathbf{c} | s_{t-1}, M), & x_\tau = y_{t-1} \\ 0, & \text{otherwise} \end{cases}. \tag{16}$$

In (16), K is the normalization term.

To avoid the improper topic words generated by seq2seq model, we use a template-based method. That is, the system is set to generate questions by replacing the subject entity to a <ent> token. Then, to get the complete question, we convert <ent> back to the subject. Subject entities in all training data and test golds are also replaced to <ent>, and those cases without <ent> are not be used to train the model.

3.4 Train

Seq2seq model with CopyNet and attention is an end-to-end model, which can be trained using back-propagation. We train the model in batches. The objective of training is to minimize the Cross-Entropy (CE) loss:

$$\mathcal{L} = -\sum_{i=1}^{n} \log p(y_i^{\text{ans}}), \tag{17}$$

where y_i^{ans} is the i-th word of the answer. The model can be learned to improve both generate-mode and copy-mode. If the target token can be found in source sequence, the copy-mode is contributed to the model and can be trained, while the copy-mode is discouraged when the token is not in the input.

4 Evaluation

In this section, we first introduce the dataset we use. Then we show the settings of our model. After that, we compare our system with other systems.

4.1 Dataset and Evaluation Metrics

We take our evaluation on the dataset released by knowledge based question generation (KBQG) subtask of NLPCC 2018 Shared Task 7. There are 24,479 answer-question pairs in the training set, and 357 answer-question pairs in the test set. The evaluation of shared task is based on BLEU-4 score with case insensitive. All the questions are generated by human. In the train set, each case has one question. And in the test set, each case is corresponded to three answers. For each case, the BLEU score is determined by the most similar answer with the output of model.

4.2 Settings of the System

The preprocessing on input sequence of our system is described in Sect. 3.1. The input sequences are split by Chinese character, punctuations and special tokens after preprocessing. All the embeddings of tokens are 200-dimensional vectors pre-trained on Chinese Wikipedia documents by word2vec, and all of the vectors are trainable. Encoder and decoder share the same embedding and vocabulary. Based on the training set, we build a vocabulary with 15,435 words, containing special tokens. We use Adam optimizer [8] for optimization, with 1e-3 as the initial learning rate. The batch size of the training stage is 32. All the implementations of our system is based on TensorFlow framework [1]. To evaluate the system during training, we extract the training set to a training set with 20,000 pairs randomly, and rest of the pairs into a validation set. All these configurations are based on the performance of our validation set.

We offer four systems to solve the result. Three of them are single models with different input sequences. One of them is character-by-character raw input, and the other two have token_seq and token_pos_seq as their input respectively. There is also an ensemble model, whose result is constructed by three sub-models using token_seq as input sequence, and another three sub-models using token_pos_seq. Two of them are submitted to the competition, their name are shown in Table 1. All these sub-models are trained separately, and on test cases, the $p(y_t|\cdot)$ is calculated by:

$$p(y_t|\cdot) = \sum_{i=1}^{6} \frac{1}{Z} p_i(p_t|\cdot), \tag{18}$$

while Z is the normalization term of all sub-models.

4.3 Result

The BLEU-4 of the systems is shown in Table 1. To show the contribution of each part of our system, we offer three system whose settings are different with our system. Our ensemble system and single system are all much better than other systems of the task. Compared with raw input, the special designed input sequences lead to 4 BLEU of improvement. The ensemble model combines the advantage of different input sequences and stochastic learning of each sub-models.

Table 1. Comparison between other systems in KBQG task.

System name	BLEU-4
AQG, SWU	49.79
AQG-PAC_greedy_relation_predict, SWU	51.46
AQD-PAC_soft_relation_predict, SWU	51.62
AQG-question_sentence_relation_predict, SWU	53.32
CCNU-319	59.82
LPAI, Soochow Univ	63.67
unique AI group, CCNU	64.38
Ours with raw input (single model)	66.18
Ours with token_pos_seq (single model, ICL-1)	70.21
Ours with token_seq (single model)	70.81
Ours (ensemble model, ICL-2)	**73.73**

4.4 Error Analysis

To find the reason of improper generations, we analysis some cases manually. First, in most cases whose relationships are not in the training set, the generated questions are in the same pattern "<ent> 的 *relationship* 是什么/多少/谁?" (What/ How many/ Who is the *relationship* of <ent>?).

It is better than most earlier models, whose question words in their generation are "是什么" (what) only, but still needs to be improved. Besides, some of generated questions contain incomplete words. For example, for the triple "达克鲁斯效力于哪个俱乐?"(Which *Clu* does Da Cruz work for?), "部"(b) of the word "俱乐部"(Club) is missed. The reason of this fault is that the our model is character based, and it is hard to for the system to learn word information to decoder.

5 Conclusion

In this paper, we propose a neural generation system to generate questions from structured knowledge base. Our system achieves a great performance in the competition of NLPCC 2018 Shared Task 7. Then, we analysis the contribution of each feature of our model by different experiments. After that, we evaluate our system manually, to know the reason of errors in generations.

In future work, it is important to improve the accuracy of generated words and fluency of sentences. Besides, it is also important to get a dataset with diverse questions, which can help exploring to improve the diversity of the question generation systems.

Acknowledgments. Our work is supported by National Natural Science Foundation of China (No. 61370117).

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M.: TensorFlow: a system for large-scale machine learning. OSDI **16**, 265–283 (2016)
2. Bordes, A., Usunier, N., Chopra, S., Weston, J.: Large-scale simple question answering with memory networks. CoRR abs/1506.02075 (2015)
3. Cho, K., van Merrienboer, B., Gülçehre, Ç., 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 2014, 25–29 October 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1724–1734 (2014)
4. Colin, E., Gardent, C., Mrabet, Y., Narayan, S., Perez-Beltrachini, L.: The WebNLG challenge: generating text from DBPedia data. In: INLG 2016 - Proceedings of the Ninth International Natural Language Generation Conference, 5–8 September 2016, Edinburgh, UK, pp. 163–167 (2016)
5. He, S., Liu, C., Liu, K., Zhao, J.: Generating natural answers by incorporating copying and retrieving mechanisms in sequence-to-sequence learning. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, 30 July–4 August, Volume 1: Long Papers, pp. 199–208 (2017)

6. Johnson, M., Schuster, M., Le, Q.V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F.B., Wattenberg, M., Corrado, G., Hughes, M., Dean, J.: Google's multilingual neural machine translation system: enabling zero-shot translation. TACL **5**, 339–351 (2017)
7. Kim, M., Moirangthem, D.S., Lee, M.: Towards abstraction from extraction: multiple timescale gated recurrent unit for summarization. In: Proceedings of the 1st Workshop on Representation Learning for NLP, Rep4NLP@ACL 2016, Berlin, Germany, 11 August 2016, pp. 70–77 (2016)
8. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. CoRR abs/1412.6980 (2014)
9. Liu, T., Wei, B., Chang, B., Sui, Z.: Large-scale simple question generation by template-based Seq2seq learning. In: Huang, X., Jiang, J., Zhao, D., Feng, Y., Hong, Y. (eds.) NLPCC 2017. LNCS (LNAI), vol. 10619, pp. 75–87. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73618-1_7
10. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, 17–21 September 2015, pp. 1412–1421 (2015)
11. Nallapati, R., Zhou, B., dos Santos, C.N., Gülçehre, Ç., Xiang, B.: Abstractive text summarization using sequence-to-sequence RNNs and beyond. In: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, 11–12 August 2016, pp. 280–290 (2016)
12. 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)
13. Serban, I.V., García-Durán, A., Gülçehre, Ç., Ahn, S., Chandar, S., Courville, A.C., Bengio, Y.: Generating factoid questions with recurrent neural networks: the 30m factoid question-answer corpus. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, 7–12 August 2016, Berlin, Germany, Volume 1: Long Papers (2016)
14. Song, Y., Yan, R., Li, X., Zhao, D., Zhang, M.: Two are better than one: an ensemble of retrieval- and generation-based dialog systems. CoRR abs/1610.07149 (2016)
15. Yin, W., Schütze, H., Xiang, B., Zhou, B.: ABCNN: attention-based convolutional neural network for modeling sentence pairs. TACL **4**, 259–272 (2016)
16. Zhang, H.P., Liu, Q.: Automatic recognition of chinese personal name based on role tagging. Chin. J. Comput. Chin. Edn. **27**(1), 85–91 (2004)
17. Zhou, H., Huang, M., Zhang, T., Zhu, X., Liu, B.: Emotional chatting machine: emotional conversation generation with internal and external memory. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, 2–7 February 2018 (2018)
18. 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

Knowledge Graph/IE