Computer Science and Information Systems 15(3):487–499 https://doi.org/10.2298/CSIS171121018Z

A Novel Framework for Automatic Chinese Question Generation Based on Multi-Feature Neural Network Model

Hai-Tao Zheng[1], Jinxin Han[1], Jinyuan Chen[1], and Arun Kumar Sangaiah[2]

[1] Graduate School at Shenzhen Tsinghua University
China, Shenzhen 518055
zheng.haitao@sz.tsinghua.edu.cn,
{hanjx16,cj-chen13}@mails.tsinghua.edu.cn
[2] School of Computing Science and Engineering VIT University
India, Tamil Nadu Vellore-632014
sarunkumar@vit.ac.in

Abstract. Automatic question generation from text or paragraph is a great challenging task which attracts broad attention in natural language processing. Because of the verbose texts and fragile ranking methods, the quality of top generated questions is poor. In this paper, we present a novel framework Automatic Chinese Question Generation (ACQG) to generate questions from text or paragraph. In ACQG, we use an adopted TextRank to extract key sentences and a template-based method to construct questions from key sentences. Then a multi-feature neural network model is built for ranking to obtain the top questions. The automatic evaluation result reveals that the proposed framework outperforms the state-of-the-art systems in terms of perplexity. In human evaluation, questions generated by ACQG rate a higher score.

Keywords: Chinese Question Generation, TextRank, Multi-Feature Neural Network Model

1. Introduction

Question generation aims to create natural questions from a given text or paragraph, and there are a lot of demands in the field of natural language processing, such as reading comprehension [8,5,6], developing as a chatbot component to request feedback [4], and improving mental health [16]. Besides, question generation systems can aid in the development of annotated datasets for question answering [19]. The existing question generation approaches can be categorized as template-based [17,10,8], semantic-based [9,13,12], and sequence-to-sequence-based [16,21,5]. The success of these approaches hinges critically on the existence of well-designed rules for declarative-to-interrogative sentence transformation or a powerful labeled dataset.

However, the main points or events of text only contain in few key sentences. Authors express their emotions or points indirectly, so the texts always exhibit complex structure and long content. The verbose texts make existing approaches generate many unimportant questions, which are unacceptable for users. In addition, the fragile ranking methods do not take enough features into account, such as the n-gram score in question, answer's location position in original sentence and so on. As a result, the key questions do not always appear in the front rank of outputs.

To address these issues, we introduce an Automatic Chinese Question Generation (ACGQ) approach. Since the verbose text content is not a good input for question construction, we firstly extract key sentences from text, and locate the main points. Next, a template-based question construction model is built to generate questions from declarative-to-interrogative sentence. Finally, a multi-feature neural ranking model is proposed to rank the generated questions.

The main contributions of our work are listed as follows:

1. We develop an adapted TextRank model to identify the key sentences of text and filter verbose sentences from source.
2. We design a multi-feature neural network model to rank all generated questions and obtain more acceptable questions in the front rank.
3. We conduct extensive experiments and the results show that our framework achieves a better performance compared to the state-of-the-art systems in terms of perplexity and human evaluation measurement.

The remainder of the paper is organized as follows. In section 2, we review related work on question generation. Section 3 introduces an overview of our model and a detailed description of each component. Section 4 details the experiments setup and their results. Section 5 concludes this paper and puts forward future work.

2. Related Work

The concept of question generation was presented dating back to 1976 by Wolfe [24], which has attracted attention of the natural language generation community in recent years since the work of Rus et al. [20].

Most works tackle this task with a template-based approach. Generally, they transform the input sentence into its syntactic representation, which then are used to generate an interrogative sentence. A lot of researches have focused on manually constructing question templates, and then applying them to generation questions [17,10,8]. Mostow et. al [17] proposed a self-questioning strategy for reading comprehension, in this strategy, three templates are firstly built to generate questions about *what,how,why.* Lindberg et al.[10] adopted a templated-based method, using predominately semantic information. This method bases on semantic patterns, which cast a wide syntactic net and a narrow semantic net. Then they construct questions according to the parser and templates. Heilman and Smith [8] introduced an overgenerate-and-rank approach. Their framework can be viewed as a two-step process for question generation. In the first step, it transforms the input sentence into a simpler sentence, which is transformed into a more succinct question. In the second step, the declarative sentence is transformed into sets of questions by a sequence of well-defined syntactic and lexical transformations. It identifies the answer phrases which may be targets for WH-movement and converts them into question phrases. Yao et al. [25] and Becker et al.[2] both followed this strategy. However, this kind of approaches does not work well when the input text is verbose, and the result is rough and redundant.

Another view is generating question based on semantic, the semantic role labels include subject, predicate verb, object, tense and so on. The semantic roles are used to determine the interrogative pronoun for the generated question [9,13,12]. Kunichika et al. [9] carried out their work in automatically generating reading comprehension questions, the questions included both syntax and semantic questions. Mazidi and Nielsen [12] uses semantic pattern recognition to crate questions of varying depth and type for self-study. Generation patterns specify the text, verb forms and semantic arguments form the source sentence to form the question. Mazidi and Tarau [13] focused on the frequency of pattern sentence occurrences and the consistency of semantic information conveyed by the pattern to generate questions, this method could generate questions in a low dimension. Such approaches generate a wider variety of questions which are not as closely bound to original text, and there are syntax errors in some questions.

Nowadays neural network with word embedding can be effectively applied to the natural language processing task with highly accurate results [7,23]. Mostafazadeh et al. [16] introduced visual question generation to explore the deep connection between language and vision. Serban et al. [21] proposed generating simple factoid questions from logic triple (subject, relation, object). Their task tackles mapping from structured representation to natural language text, and their generated questions are consistent in terms of format. Xinya Du et.al [5] built a bigger dataset for text&question based on SQuAD [3], and trained a sequence to sequence network to generate questions on level on word and paragraph. These supervised methods need a large scale of labeled data and cost much time to train the network, which are not a good way to handle question generation problem for a rare of Chinese dataset.

To conclude, most of the existing question generation models are based on full texts, leading to generate some redundant questions. in addition, the fragile ranking methods do not take enough features into account and many useless questions are generated. In order to improve the performance of Chinese question generation, ACQG identifies the key sentences in texts and utilizes multiple features for question ranking.

3. The Framework of Automatic Chinese Question Generation

The goal of ACQG is to generate more acceptable questions from text or paragraph. The framework is shown in Fig. 1. This involves 1) identifying the key sentences of text with an adapted TextRank. 2) constructing questions according to rule-based templates. 3) ranking questions with a multi-feature neural network model.

[3] https://rajpurkar.github.io/SQuAD-explorer/

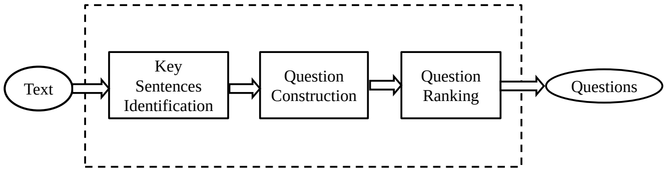

Fig. 1. Overview of Automatic Chinese Question Generation Framework

3.1. Key Sentences Identification

To identify the key sentences from text or paragraph, we use an adapted TextRank method. Graph-based ranking algorithm is a common measure to extract the key content in web [26,14]. A similar method called TextRank [15] can be applied to extract summary from document. In this work, we use an adapted TextRank to identify key sentences from text, which converts text to a graph and calculates the score of each sentence node. When the score of a sentence is greater than the set threshold θ, the sentence will be selected. The original TextRank employs the BM25[4] to compute the similarity of sequence, whereas it has many manual setting parameters leading to inefficient results. Therefore, we use the cosine similarity replaced in our method. The details are described as follows.

The input text T is divided into some sentences with different terminators (e.g. ‘。’,‘！’,‘？’). $T = \{S_0, S_1, ...S_i, ..., S_n\}$, S_i represents a sentence in T, n is the total number of sentences in T. Then we segment S_i into words. $S_i = \{w_0, w_1, ..., w_k..., w_m\}$, w_k represents a word in S_i, m is the total number of words in S_i .

We introduce Term Frequency–Inverse Document Frequency [1] to quantify each word, where $TF(w_k)$ is the frequency of w_k in T, $IDF(w_k) = log\frac{||T||}{||w_k \in T||}$ is an inverse word frequency in total texts, $||T||$ is the total number of texts in dataset. So a sentence can be represented to a word vector. The similar score is computed by Equation. 1.

$$sim_{ij} = sim(S_i, S_j) = \frac{\sum_{w_v \in S_i \cap S_j}(TF(w_v)IDF(w_v))^2}{|S_i||S_j|} \tag{1}$$

For a given sentence S_i, $In(S_i)$ is the previous sentences, $Out(S_i)$ is the subsequent sentences. The score of S_i is computed using the Equation. 2.

$$WS(S_i) = (1-d) + d * \sum_{S_j \in In(S_i)} \frac{sim_{ji}}{\sum_{S_k \in Out(S_j) sim_{jk}}} WS(S_j) \tag{2}$$

Where d is a parameter that is set between 0 and 1.

When the algorithm finished, a score associates with each sentence, and it expresses the importance of sentence in text. Whether the sentence is selected or not depends on this score.

[4] https://en.wikipedia.org/wiki/Okapi_BM25

3.2. Question Construction

In this section, we describe how to construct questions from sentence. In particular, we focus on the structure and relation between words to construct questions. We design different rule-based templates to construct targeted questions from the parser tree.

Firstly, we design 5 general templates. If an input sentence contains location, person, noun, or time finding them by Named Entity Recognition (NER) , we construct *Where*, *Who*, *What* or *When* type of question. If there is adjective around noun, we construct *How* question. Then we replace a question type with the named entity to generate a question by declarative-to-interrogative. A good performance of NER is important for the type of question. To ensure the quality of tokenize and NER, we expand the annotators by getting neologisms from Wikipedia[5] regularly.

Secondly, we design more templates aiming at generating targeted questions. Parser tree describes the relationship among verb, subject, object, and so on. We utilize it to analyze each sentence, and edit some templates learning from text to generate targeted questions. The templates are listed as follows.

1. *number* question : (QP < CD =number < CLP)
2. *rank* question : (QP < OD=number)
3. *if-cause* question : ((IP| PP=reason << 由于 (because of) | 因为 (because)) ..(IP| PP| VP)| << (IP| PP |VP << 所以 (so) | 于是 (so that)))
4. *relative-cause* question : (((IP | PP=front << 虽然 (though)) .IP=however)| << (IP | PP=front(IP | PP | VP=however << 但是 (but) | 但 (but))))
5. *color* question : (QP < NP=JJ < NP)

IP, PP, and VP represent different tag of words. A dot means subsequent follow and a left shaped arrow means an immediate subtree relation. These templates guide to search possible answer phrases on the parser tree. Once a subtree matches any template, a question will be constructed. In order to have a better understanding of each template, we list some examples in Table. 1.

3.3. Question Ranking

After the two processes above, we can obtain lots of questions, while these questions are sorted arbitrarily and the key questions do not always appear in the front rank of outputs. We build a multi-feature neural ranking model to select the top key questions. We import twelve features from the generated questions and text, shown in Table. 2, and design a multi-feature neural network model to rank them. This is an appropriate solution to the multiple features problem without manual intervention.

As shown in Table. 2, we select twelve features about generated questions and original text. Additionally, the features $f1 \sim f5$ are basic information of question and answer. $f6$ are the n-gram scores in question sequence, which provide rich contextual information. $f7$ is used to compute the keywords percentage in question. $f8 \sim f10$ are different tags of word distribution in question, which indicate the

[5] https://en.wikipedia.org/wiki/Main_Page

Table 1. The Template Examples

Type	Sentence	Question
Who	刘佩英任安吉斯中国首席商务官。(Peiying Liu acts as the China chief commercial officer of Angis.)	谁任安吉斯中国首席商务官?(Who acts as the China chief commercial officer of Angis?)
When	2017 年退休人员基本养老金上调。(The basic pension of retirees will increase in 2017.)	何时退休人员基本养老金上调?(When will the basic pension of retires increase?)
Where	二战时日本曾在越南制造大饥荒。(Japan had created a famine in Vietnam during World War II.)	二战时日本曾在哪里制造大饥荒?(Where had Japan created a famine during World War II?)
rank	昆仑鸿星取关键一胜跃居东区第六位 (Kunlun Hongxing becomes the Sixth in the East by taking a key victory.)	昆仑鸿星取关键一胜跃居东区第几位?(What is the ranking of Kunlun Hongxing in the East by taking a key victory?)
number	亚泰 2000 万镑交易震惊世界！(The transaction of Yatay 20 million pound shocked the world.)	亚泰多少镑交易震惊世界？(What‘s the amount of the transaction of Yatay shocked the world?)
if-cause	因为人工智能快速发展，人们的生活方式得到巨大改变。(Because of the rapid development of artificial intelligence, people‘s lifestyle has been greatly changed.)	人们的生活方式得到巨大改变，因为什么？(What makes people’s lifestyle have been change greatly?)
relative-cause	春运将至，铁道部门虽然提前发售车票，但买票难的问题仍然存在。(As Spring Festival is coming, the railway department has sold tickets in advance, but it is still difficult to buy a ticket.)	春运将至，铁道部门虽然提前发售车票，但怎么样？(As Spring Festival is coming, what‘s happened when the railway department has sold tickets in advance?)
color	天气红色预警发布后，学校需要停课。(After the weather red warning released, classes will be suspended.)	什么颜色天气预警发布后,学校需要停课？(Classes will be suspended after which color warning released?)

structure of question sequence. $f11$ is a probability data, which represents the uncertainty of sentence. $f12$ is a numerical statistic measure that is a popular sentence-weighting scheme.

We design a three layers neural network, as Fig. 2, this model is a gradient descent method designed to minimize the total error of predicted score and human rated score. The input layer is all features of each question. The middle layer is a hidden layer with K nodes, and the last layer is the predicted score. f_{ij} is jth feature of ith question. Furthermore, the multi-feature neural model is viewed as statistical model of the under form:

$$s_i = \beta_0 + \sum_{j=1}^{K} \beta_j \Psi(w_j f_i) \tag{3}$$

Where $i = 1, 2, ..., M$, M is the total number of training questions, f is the set of features, β_j is the weight between layers, s is the predicted score, Ψ is a non-linear

Table 2. All Features to Rank

Feature
$f1$: number of tokens in question and answer
$f2$: number of named entity in question
$f3$: type of question
$f4$: score of key sentence
$f5$: answer's location position in original sentence
$f6$: the unigram, bigram and trigram score of question
$f7$: percentage of keywords in question
$f8$: stopwords density in question
$f9$: noun density in question
$f10$: verb density in question
$f11$: information gain
$f12$: TF-IDF score

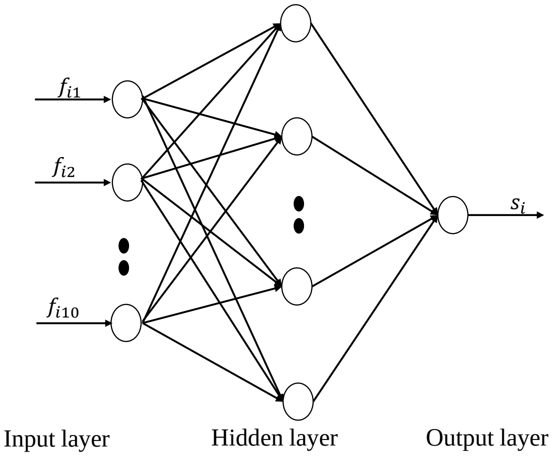

Fig. 2. The Description of Multi-Feature Neural Ranking Model

squashing function like the rectified linear unit function:

$$\Psi(z) = \frac{1}{1 + exp(-z)} \tag{4}$$

And we use the mean square error e as loss function, that is:

$$e = \frac{1}{2M} \sum_{i=1}^{M} (t_i - p_i)^2 \tag{5}$$

Where t_i is the i-th actual score by human rated. We use a gradient descent method to reduce e. When $e < \varepsilon$, the neural network will be constructed, which can be used to compute the score of each question. Besides, by descending these scores attached to questions, we can obtain the top questions which are more acceptable by users.

4. Experiments

We conduct extensive experiments to evaluate the feasibility and effectiveness of our framework, and compare it with the state-of-the-art question generation systems.

4.1. Experimental Setup

We crawl ten topics of news data from Netease[6] and Toutiao [7], which are the popular Chinese media and website. The news contains ten topics, such as Economic (Ec), House, National, Military, Woman, Cars, Sports, Web, Entertainment (En), and Others. There are two hundred media news totally in the dataset [8]. For each news topic, 80% news are selected randomly as the training set, the rest news is used as the testing set.

There are some parameters to setup before generating. In the adopted TextRank, the parameter d is usually set to 0.85 [15], and this is the value we are also using in our implementation; the threshold θ is 1.2, which is a little greater than the independent sentence score. In neural ranking model, the parameter K, ε are set to 5, 0.01 [3] respectively. These values can ensure the model is robust. For sentences analysis, we apply the Stanford CoreNLP [11] to handle sentences including named entity recognizer and parser tree building.

We introduce three recent methods including H&S, HSKS and HSMF for comparison to evaluate the performance of each framework. The compared methods are listed as follows:

- H&S: H&S transforms the input sentence into a simpler sentence and the simpler sentence is transformed into sets of questions by a sequence of well-defined syntactic and lexical transformations.[8]
- HSKS: HSKS extracts the key sentences from texts and then constructs questions from the key sentences using templates.
- HSMF: HSMF applies a multi-feature model to rank all generated questions to make the top questions more useful.

We calculate the perplexity [18] score of generated questions, which is used to compare probability for nature language processing. Note that the lower perplexity indicates the generated questions are more likely to human writing. Besides, we also use human evaluation that is frequently adopted in recent works about question generation [5,22,10]. In human evaluation, all generated questions mixed together are evaluated by human, and each question is rated by two people. In detail, we define an evaluation criterion of question with different score values.

[6] http://news.163.com/

[7] https://www.toutiao.com/

[8] https://github.com/hanjx16/question-data.git

- *Good* represents the question is related to the text well without grammatical error and its score is 3.
- *Borderline* represents the question has no problem from the view of the question but the answer does not make much sense (e.g., feature, today) and its score is 2.
- *Bad* represents the question is redundant or the answer is a part of other question and its score is 1.

4.2. Experimental Results

Table. 3 shows the perplexity score of n-gram in each framework. Unigram is $n = 1$, Bigram is $n = 2$ and Trigram is $n = 3$. ACQG outperforms other frameworks, whose perplexity scores are 654.28 (Unigram), 418.46 (Bigram) and 166.53 (Trigram). H&S uses the over-generated questions and ranks them to get the output questions. The full text is used to generate questions, thereby the outputs are redundant and the scores are 841.39 (Unigram), 608.21 (Bigram) and 213.34 (Trigram). HSKS filters unmeaning sentences in texts, which is able to generate more targeted questions. Thus the scores decrease dramatically. Referring to HSMF system, HSMF is superior to H&S slightly. However the score of unigram is greater than H&S. The reason is that they both construct more absurd questions and the ranking method cannot improve this issue effectively. When we improve both key sentence extracting and question ranking, the performance of ACQG achieves best.

Table 3. The Perplexity Score of n-gram in Different Frameworks

Frameworks	Unigram	Bigram	Trigram
H&S	841.39	608.21	213.34
HSKS	735.42	464.62	187.25
HSMF	821.8	584.42	205.72
ACQG	**654.28**	**418.46**	**166.53**

Furthermore, we introduce human evaluation into this work. The average marking scores are shown in Table. 4 , the detailed score distribution in each topic is shown in Fig. 3.

Table 4. The Average Score of Human Evaluation in Different Frameworks

Frameworks	H&S	HSKS	HSMF	ACGQ
Average Score	1.79	2.12	1.95	**2.37**

From Table. 4, we can see ACGQ has the highest score 2.37, which indicates that the generated questions are around *Good* and *Borderline*. The questions are more likely to be acceptable. Because of the bad input and fragile ranking method, H&S has a poor performance, whose human evaluation score is only 1.79. For HSKS and HSMF, whose human evaluation score are 2.12 and 1.95 respectively. They both show better performance than H&S, but perform worse than ACGQ. Thus, the results of human evaluation are consistent with the above automatic evaluation.

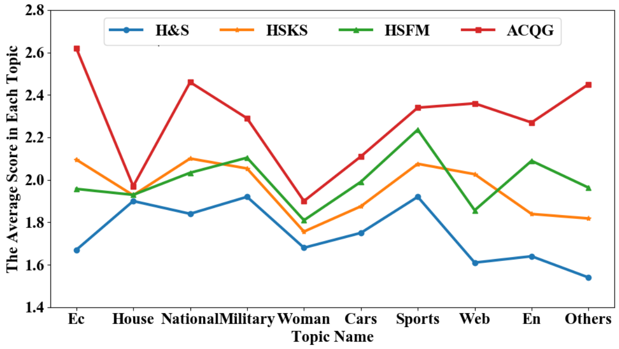

Fig. 3. The Human Evaluation in Each Topic

Fig. 3 shows the detailed human evaluation score distribution in each topic. We can observe that all frameworks have a low satisfaction score in the topic of House. The reason is that the collected news is almost advertisements in House, which cannot generate meaningful questions. For other topics, H&S always shows a lower score in each topic, while ACQG always shows a higher score in each topic. For other frameworks, the results fluctuate within H&S and ACQG. In detail, the upper scores are 2.62 (Ec), 1.97(House), 2.46 (National), 2.29 (Military), 1.90 (Woman), 2.11 (Cars), 2.34 (Sports), 2.36 (Web), 2.27 (En), 2.45 (Others) respectively. HSKS pays attention to the main points of text, which filters some noisy sentences from verbose text, so the generated questions almost match texts well. HSMF takes enough features into account, which makes the ranking model effective, thus the top questions are more likely acceptable. ACGQ includes both improved components, so we can see the average score of ACQG by human evaluation is the highest. Therefore, ACQG works effectively by extracting key sentences instead of full text and improving ranking method.

5. Conclusion and Future Work

We have proposed an automatic Chinese question generation framework (ACQG), a novel framework that generates more meaningful questions automatically to help readers obtain the main content of Chinese text. We use key sentences rather than the full content to generate questions, and then a template-based method is built to construct more variant questions. Finally, these questions are ranking by an efficient ranking method that leverages a multi-feature neural model to solve the multiple features problem. Extensive experiments are conducted on a Chinese dataset from the popular Chinese website and media, and the results show a significant improvement in terms of perplexity and human evaluation comparing with the baseline methods. We believe the proposed method will play an important role in question generation.

There still exist limitations in ACGQ. For example, the types of generated questions are limited, which is caused by the limited templates. In the future work, we will design more useful templates from texts to generate more meaningful questions. In addition, we will construct enough pairs of text and questions to train an end-to-end neural model to generate high-quality Chinese questions directly.

Acknowledgments This research is supported by National Natural Science Foundation of China (Grant No. 61773229), Natural Science Foundation of Guangdong Province (Grant No. 2014A030313745), Basic Scientific Research Program of Shenzhen City (Grant No. JCYJ20160331184440545), and Cross fund of Graduate School at Shenzhen, Tsinghua University (Grant No. JC20140001).

References

1. Aizawa, A.: An information-theoretic perspective of tf–idf measures. Information Processing & Management 39(1), 45–65 (2003)
2. Becker, L., Basu, S., Vanderwende, L.: Mind the gap: learning to choose gaps for question generation. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 742–751. Association for Computational Linguistics (2012)
3. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: Proceedings of the 22nd international conference on Machine learning. pp. 89–96. ACM (2005)
4. Colby, K.M., Weber, S., Hilf, F.D.: Artificial paranoia. Artificial Intelligence 2(1), 1–25 (1971)
5. Du, X., Shao, J., Cardie, C.: Learning to ask: Neural question generation for reading comprehension. CoRR abs/1705.00106 (2017), `http://arxiv.org/abs/1705.00106`
6. Duke, N.K., Pearson, P.D.: Effective practices for developing reading comprehension. The Journal of Education 189(1/2), 107–122 (2008)
7. He, D., Xia, Y., Qin, T., Wang, L., Yu, N., Liu, T., Ma, W.Y.: Dual learning for machine translation. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp. 820–828. Curran Associates, Inc. (2016), `http://papers.nips.cc/paper/6469-dual-learning-for-machine-translation.pdf`
8. 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)
9. Kunichika, H., Katayama, T., Hirashima, T., Takeuchi, A.: Automated question generation methods for intelligent english learning systems and its evaluation (2001)
10. Lindberg, D., Popowich, F., Nesbit, J., Winne, P.: Generating natural language questions to support learning on-line (2013)

11. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The stanford corenlp natural language processing toolkit. In: ACL (System Demonstrations). pp. 55–60 (2014)
12. Mazidi, K., Nielsen, R.D.: Linguistic considerations in automatic question generation. In: ACL (2). pp. 321–326 (2014)
13. Mazidi, K., Tarau, P.: Infusing nlu into automatic question generation. In: INLG. pp. 51–60 (2016)
14. Mihalcea, R.: Graph-based ranking algorithms for sentence extraction, applied to text summarization. In: Proceedings of the ACL 2004 on Interactive poster and demonstration sessions. p. 20. Association for Computational Linguistics (2004)
15. Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: EMNLP. vol. 4, pp. 404–411 (2004)
16. Mostafazadeh, N., Misra, I., Devlin, J., Mitchell, M., He, X., Vanderwende, L.: Generating natural questions about an image. arXiv preprint arXiv:1603.06059 (2016)
17. Mostow, J., Chen, W.: Generating instruction automatically for the reading strategy of self-questioning. In: AIED. pp. 465–472 (2009)
18. Popel, M., Mareček, D.: Perplexity of n-gram and dependency language models
19. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250 (2016)
20. 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)
21. Serban, I.V., García-Durán, A., Gulcehre, C., Ahn, S., Chandar, S., Courville, A., Bengio, Y.: Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. arXiv preprint arXiv:1603.06807 (2016)
22. Susanti, Y., Tokunaga, T., Nishikawa, H., Obari, H.: Evaluation of automatically generated english vocabulary questions. Research and Practice in Technology Enhanced Learning 12(1), 11 (Mar 2017), https://doi.org/10.1186/s41039-017-0051-y
23. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. CoRR abs/1411.4555 (2014), http://arxiv.org/abs/1411.4555
24. Wolfe, J.H.: Automatic question generation from text-an aid to independent study. ACM SIGCSE Bulletin 8(1), 104–112 (1976)
25. Yao, X., Tosch, E., Chen, G., Nouri, E., Artstein, R., Leuski, A., Sagae, K., Traum, D.: Creating conversational characters using question generation tools. Dialogue & Discourse 3(2), 125–146 (2012)
26. Zhu, H., Zhang, P., Sun, C.: Deep q-learning with prioritized sampling. In: Neural Information Processing: 23rd International Conference, ICONIP 2016, Kyoto, Japan, October 16–21, 2016, Proceedings. vol. 9947, p. 13. Springer (2016)

Hai-Tao Zheng got the bachelor degree in Department of Computer Science, Sun Yat-Sen University, 2001. He got the master degree in Department of Computer Science, Sun Yat-Sen University 2004. He got the Ph.D degree in Medical Informatics Major, Seoul National University. His research interests include artificial intelligence, semantic web, information retrieval, machine learning, and medical informatics. Dr. Zheng has published more than 30 papers including 10 SCI journal papers. He has a deep understanding of the semantic web and knowledge mining fields.

Jinxin Han is a master student in Department of Computer Science and Technology, Tsinghua University. His research interests include natural language processing, machine learning and deep learning.

Jin-Yuan Chen is a doctoral student of Dr. Hai-Tao Zheng. He got his master degree in computer engineering, Graduate School at Shenzhen, Tsinghua University 2013. His research interests include artificial intelligence, information retrieval and machine Learning.

Arun Kumar Sangaiah has received his Master of Engineering (ME) degree in Computer Science and Engineering from the Government College of Engineering, Tirunelveli, Anna University, India. He had received his Doctor of Philosophy (PhD) degree in Computer Science and Engineering from the VIT University, Vellore, India. He is presently working as an Associate Professor in School of Computer Science and Engineering, VIT University, India. His area of interest includes software engineering, computational intelligence, wireless networks, bioinformatics, and embedded systems. He has authored more than 100 publications in different journals and conference of national and international repute. His current research work includes global software development, wireless ad hoc and sensor networks, machine learning, cognitive networks and advances in mobile computing and communications. Also, he was registered a one Indian patent in the area of Computational Intelligence. Besides, Prof. Sangaiah is responsible for Editorial Board Member/Associate Editor of various international journals.

Received: November 21, 2017; Accepted: June 10, 2018.