Keyword Provision Question Generation for Facilitating Educational Reading Comprehension Preparation

Ying-Hong Chan **Ho-Lam Chung** **Yao-Chung Fan**

Department of Computer Science and Engineering
National Chung Hsing University,
Taichung, Taiwan

Abstract

Question Generation (QG) receives increasing research attention in the NLP community. One QG motivation is to facilitate the preparation of educational reading practice and assessments. While significant advancement of QG techniques was reported, we find current QG techniques are short in terms of *controllability* and *question difficulty* for educational applications. This paper reports our studies toward the two issues. First, we report a state-of-the-art exam-like QG model by advancing the current best model from 11.96 to 20.19 (in terms of BLEU 4 score). Second, we propose a QG model that allows users to provide keywords for guiding QG direction. Human evaluation and case studies are conducted to demonstrate the feasibility of controlling question generation direction.

1 Introduction

Question generation (QG), taking a passage and an answer phrase as input and generating a context-related question as output, has received interest in recent years (Zhou et al., 2017; Zhao et al., 2018; Du et al., 2017; Chan and Fan, 2019; Dong et al., 2019; Bao et al., 2020). One motivation for developing QG is to facilitate educators in the preparation of reading comprehension assessments.

While significant QG quality was reported, we find two limitations for integrating the current QG models into educational usage scenarios.

First, the current QG model suffers from the model controllability concern. In Table 1, we show an example with a passage, an answer, and two questions (Q_1 and Q_2). The model controllability concern lies in that we have no way to control the QG direction with the model (Chan and Fan, 2019; Dong et al., 2019; Bao et al., 2020).

We note that both questions have the same answer (i.e., *Christopher Hirata*), while the models are designed to take a context and an answer span as input for QG. Thus, there are no way to control which question to generate.

Context	At the age of 12, **Christopher Hirata** already worked on college-level courses, around the time most of us were just in the 7th grade. At the age of 13, this gifted kid became the youngest American to have ever won the gold medal in the International Physics Olympiad. At the age of 16, he was already working with NASA on its project to conquer planet mars. After he was awarded the Ph.D. at Princeton University, he went back to California institute of technology. The next person with a very high IQ is Albert Einstein. With an IQ between 160 and 190, Albert Einstein is the genius behind the theory of relativity, which has had a great impact on the world of science.
Answer	Christopher Hirata
Q_1	Who once worked on the project to conquer planet mars?
Q_2	Who was the youngest American to have ever won the gold medal in the International Physics Olympiad?

Table 1: An Example for QG Model Controllability Concern: With the existing QG settings, we have no way to control which question to generate.

Second, questions generated by existing QG models are too simple (in terms of difficulty) for advanced educational reading practice assessment. Current data-driven QG models are trained with factoid QA datasets (e.g., SQuAD (Rajpurkar et al., 2016) or NewsQA (Trischler et al., 2016)), and therefore generate factoid questions, which are too simple for advanced reading practice assessment.

In this paper, we report our results toward the two limitations. First, we propose a new QG setting variant for the controllability issue, which allows users to guide the QG direction by indicating keywords (Please see Section 2). Our design, KPQG (Keyword Provision Question Generation) model, successfully enables QG controllability. Experiments are conducted using benchmark datasets to show the quality of our KPQG model. We also conduct quantitative studies to examine the controllability and feasibility of the generation in various aspects

For the issue of generating too simple questions, we investigate training QG models with exam-like datasets (e.g., RACE (Lai et al., 2017)). We investigate the employment of pre-trained language

models (LM) for exam-like QG. Our experiment results show that the LM employment significantly advances the state-of-the-art result reported by (Jia et al., 2020) from 11.96 to 20.19 (in terms of BLEU 4 score).

2 Methodology

In Subsection 2.1, we first review the existing LM architectures for QG, which are basic building blocks for QG based on LM. In Subsection 2.2, we present Keyword Provision Question Generation (KPQG) scheme for guiding QG generation.

Problem Formulation In this paper, we consider a QG setting that takes (1) a context passage, (2) answer phrase, and (3) *a set of keywords* as input and generate a question contains the keywords as output. Note that the existing QG setting takes only (1) a context passage and (2) answer phrase as input. The idea is to design QG to take additional keywords for question generation. We refer readers to the example illustrated in Figure 1.

2.1 QG Architecture

In this paper, we explore two QG architecture.

Masked-LM Generation The QG model by Masked-LM Generation works as follows. A Masked-LM QG generation model $\mathbb{M}()$ takes a context paragraph C, answer A, and the previous generated tokens $q_1, ..., q_{i-1}$ and as input and output a target token q_i in an auto-regressive manner, where `[S]` and `[M]` are the sep and masked special tokens in pre-trained language models.

$$\mathbb{M}(C\,\texttt{[S]}\,A\,\texttt{[M]}) \rightarrow q_1,$$
$$\mathbb{M}(C\,\texttt{[S]}\,A\,\texttt{[S]}\,q_1\,\texttt{[M]}) \rightarrow q_2,$$
$$\mathbb{M}(C\,\texttt{[S]}\,A\,\texttt{[S]}\,q_1, q_2\,\texttt{[M]}) \rightarrow q_3,$$
$$...$$

Seq2Seq Generation A seq2seq model $\mathbb{M}()$ for QG takes a context paragraph C and an answer A as input and predicting a sequence of question tokens $\{q_1, q_2 ... q_{|Q|}\}$ as output. Specifically, we have

$$\mathbb{M}(C\,\texttt{[S]}\,A) \rightarrow q_1, q_2, ..., q_{|Q|}$$

2.2 Key Provision Question Generation

Inference Our KPQG model extends the Masked-LM Generation as follows. For a given keyword sequence [k_1, ..., k_i], a context C and an answer phrase A, the input sequence X to a LM model is to interleavely place `[M]` tokens between the keyword sequence as follows.

$$X = [C\,\texttt{[S]}\,A\,\texttt{[S]}\,\texttt{[M}_1\texttt{]}\,k_1\,\texttt{[M}_2\texttt{]} ... \texttt{[M}_i\texttt{]}\,k_i]$$

We leverage Masked-LM generation to predict the `[M]` tokens. After the prediction, we recursively insert and predict the `[M]` tokens in the same manner. At each iteration, we align the input sequence by inserting `[M]` before and after all given/generated tokens. The iteration continues till all masked tokens become `[S]`.

As a concrete example, please refer to the example shown in Figure 1 and Table 2. Two keywords (`project` and `mars`) are given in this example. At Iteration 0, we have three inserted `[M]` tokens, and the predicted results are "`Who`", "`planet`", and "`?`". And, at Iteration 1, we set the input sequence X_1 by inserting `[M]` before and after all given/generated tokens. The `[M]` placement and prediction loops until all `[M]`s becomes `[S]`.

Training to Generate Important Token First The KPQG is trained to predict a masked token before/after the input/generated keyword tokens. Under this goal, the challenge lies in which tokens should be masked for model training.

We explore the idea of learning to predict important words by employing a QA model (e.g., SQuAD) to assess the importance of tokens. Our idea is that if masking some token q_i from a question sentence [$q_1, ..., q_{|Q|}$] leads to a decreased QA model performance, then q_i shall be an important one. Therefore, for a given Q, we iteratively replace all tokens in Q with a `[PAD]` token in a one-at-a-time manner.

For example, for the question "how is the weather today?", we have the following *padded* question sentences.

- `[PAD]` is the weather today?
- how `[PAD]` the weather today?
- how is `[PAD]` weather today?
- how is the `[PAD]` today?
- how is the weather `[PAD]` ?
- how is the weather today `[PAD]`

We then post the sentences to a QA model for answer prediction, and estimate the importance of a keyword through the model's confidence in answer prediction.

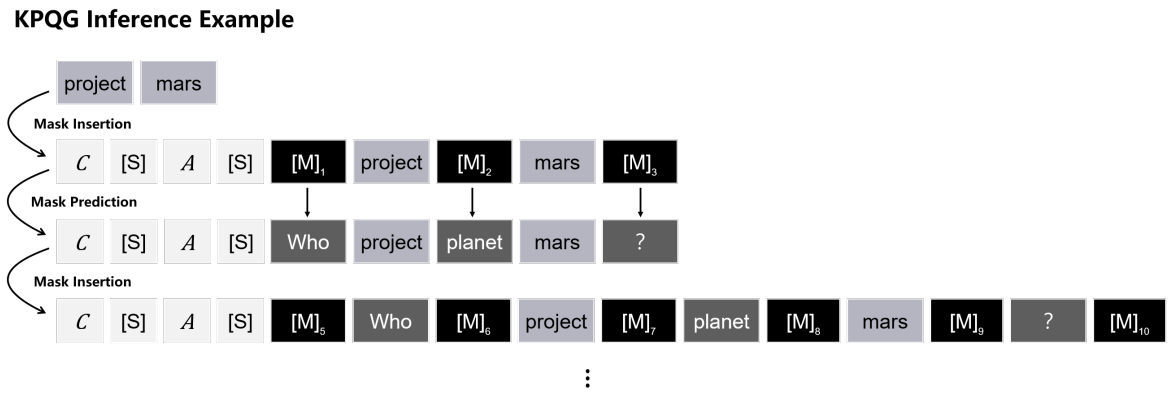

Figure 1: KPQG Mask Insertion and Prediction

After the token importance assessment, we generate training data for KPQG based on the token importance by masking important word first. In Table 3, we show an example. Assume that the importance of a question sentence $[q_1, ..., q_9]$ is $[q_4, q_6, q_2, q_5, q_3, q_1, q_9, q_7, q_8]$ (from high to low).

As shown in Table 3, six training instances are generated. The first training instance aims to instruct the KPQG model to predict the most important word (i.e., q_4) based on only C and A. That is, the label of the `[M]` token is set to q_4.

$$\mathbb{M}(C\,\texttt{[S]}\,A\,\texttt{[S]}\,\texttt{[M]}) \rightarrow q_4$$

Likewise, the second training instance is set to predict q_2 and q_6 as follows.

$$\mathbb{M}(C\,\texttt{[S]}\,A\,\texttt{[S]}\,\texttt{[M]}\,q_4\,\texttt{[M]}) \rightarrow q_2, q_6$$

Please refer to the complete training instances in Table 3.

3 Performance Evaluation

3.1 Educational QG Comparision

In this subsection, we report our results on the employment of pre-trained language models (PLM) for educational QG.

We evaluate the results on EQG-RACE (Jia et al., 2020) dataset. Table 4 summarizes statistics for the datasets. We implement the following QG models.

- Masked-LM QG architecture with BERT (Devlin et al., 2018)
- Masked-LM QG architecture with RoBERTa (Liu et al., 2019)
- Masked-LM QG architecture with DeBERTa (He et al., 2020)
- Seq2Seq QG architecture with BART (Lewis et al., 2019)

Table 5 shows the evaluation results on test data. We also list the state-of-the-art result reported by (Jia et al., 2020). We see that the PLM employment significantly improves the performance of educational QG. Among them, DeBERTa-QG advances the SOTA result from 11.96 to 20.19 (in terms of BLEU 4 score).

3.2 KPQG Performance Evaluation

3.2.1 Implementation Details

We use the **DeBERTa$_{\text{base}}$** (He et al., 2020) model for KPQG training. The KPQG model is trained by four TITAN V100 GPUs with 10 epochs for 16 hours. In addition, for the QA model for assessing token importance for training data preparation, we use the RACE QA model from (Wolf et al., 2020). This model has an accuracy of 84.9% on the RACE dataset.

3.2.2 Human Evaluation

We use human evaluation to validate the quality of the KPQG model because the premise of the KPQG model allows users to guide the QG direction by indicating keywords expected to be included in the generation result. 300 context paragraphs and the corresponding answers were randomly selected from the test set of EQG-RACE data (Jia et al., 2020). We invited 30 evaluators. Each one was given 10 contextual paragraphs and asked to use the KPQG model to provide keywords to generate questions. The evaluator is asked to compare the difference between QG and KPQG and score [0,1,2] on the Likert scale based on the following three metrics:

	M_j	Prediction for [M]
iter0	C [S] A [S] [M] project [M] mars [M]	Who, planet, ?
iter1	C [S] A [S] [M] Who [M] project [M] planet [M] mars [M] ? [M]	[S], worked, to, [S], [S], [S]
iter2	C [S] A [S] Who [M] worked [M] project [M] to [M] planet mars ?	once, the, [S], conquer
iter3	C [S] A [S] Who [M] once [M] worked [M] the [M] project to [M] conquer [M] planet mars ?	[S], [S], on, [S], [S], [S]
iter4	C [S] A [S] Who once worked [M] on [M] the project to conquer planet mars ?	[S], [S]
end	Who once worked on the project to conquer planet mars ?	

Table 2: KPQG Inference Example

	X_i	Labels for [M]
i=0	C [S] A [S] [M]	q_4
i=1	C [S] A [S] [M] q_4 [M]	q_2 q_6
i=2	C [S] A [S] [M] q_2, [M] q_4 [M] q_6 [M]	q_1 q_3 q_5 q_9
i=3	C [S] A [S] [M] q_1 [M] q_2 [M] q_3 [M] q_4 [M] q_5 [M] q_6 [M] q_9 [M]	[S] [S] [S] [S] [S] [S] q_7 [S]
i=4	C [S] A [S] q_1 q_2 q_3 q_4 q_5 q_6 [M] q_7 [M] q_9	[S] q_8
i=5	C [S] A [S] q_1 q_2 q_3 q_4 q_5 q_6 q_7 [M] q_8 [M] q_9	[S] [S]

Table 3: The training instance creation example: the importance of a question sentence [q_1, ..., q_9] is $[q_4, q_6, q_2, q_5, q_3, q_1, q_9, q_7, q_8]$ (from high to low). Six training instances are generated in this example.

	Train	Test	Dev
EQG-RACE	17445	950	1035

Table 4: **EQG-RACE** Dataset statistics

- **Fluency:** how grammar and structural fluency the generated sentence is.
- **Expectedness:** The extent to which the generated question are in line with expectations.
- **Answerability:** whether the generated question that can be answered.

The human evaluation results are summarized in Table 6. We have the following observations.

For fluency, the two compared models are able to generate grammatical and structural sentences. This is not a surprising result as with the help of the language model, the existing QG models are all able to generate fluent question sentences.

For expectedness, we see there is a big difference between the two compared models. This result validates the KPQG model addresses the QG controllability concern.

For answerability, we also observe improvement. We consider this is due to providing additional keywords guides QG to generate more specific questions other than general questions, which therefore the answerability measure is improved.

3.3 Qualitative Comparison

In Table 7, we show generation results. The examples are selected from the test set of EQG-RACE (Jia et al., 2020). In each example, we show the context paragraph, answer, and the gold question (the first three row of the tables). We use the gold question to simulate it as the one that the user expects to generate. We list the QG results by DeBERTa-QG and DeBERTa-KPQG with different keyword sets.

Example 1 As can be seen from Example 1, although the result of DeBERTa-QG is the correct question, the direction of the question is not the same as the expected golden question. This is because no keywords are used to guide the QG direction. However, in the results of DeBERTa-KPQG, we can see that with the given ["`mars`"] keyword, the KPQG model has successfully guided the generation toward the golden question. In addition, KPQG can also use keywords to control the generated sentence syntactical structure. For example, in this case, we prompt ["`mars`","`who`"] for KPQG. We see that "For conquering plant mars, who did he work with NASA?" is generated. The generated result not only includes the indicated keywords but also consider the order of the keywords. We consider this ability might be also helpful to improve the QG diversity in terms of different syntactical structure generation.

Example 2 In Example 2, we can also see that DeBERTa-KPQG's question on the given keyword ["`largest meat`"] is closer to the golden question. Furthermore, prompting different keywords leads to different results. For example, given the ["`rice`"] keyword, the model generates "Where dos lunch usually eat in order of rice, potatoes and vegetables?", which is a complete different question direction. This result shows that KPQG can control the generation results according to the keywords given by the user. This feature is also helpful for teachers to have inspiration for preparing reading assessment.

Model	BLEU 1	BLEU 2	BLEU 3	BLEU 4	ROUGE-L	METEOR
(Jia et al., 2020)	35.10	21.08	15.19	11.96	34.24	14.94
BERT-QG	43.37	29.53	22.25	17.54	44.26	20.47
RoBERTa-QG	46.37	32.15	24.34	19.21	46.96	22.32
BART-QG	46.78	32.30	24.53	19.39	47.00	22.22
DeBERTa-QG	**47.16**	**32.81**	**25.18**	**20.19**	**47.33**	**22.55**

Table 5: Performance Comparison

Model	Fluency	Expectedness	Answerability
DeBERTa-QG	1.60	0.86	1.20
DeBERTa-KPQG	1.60	**1.37**	**1.44**

Table 6: Human evaluation results

Example 3 Similar to the conclusion from the previous example, in Example 3, we prompt the keyword ["`Megan Smith`"] to guide the direction of the KPQG model generation. Again, we see the result is close to the golden question. In addition, KPQG can also control the sentence syntax by giving only the "`wh-`" keyword. For example, in Example 3, the answer is that a person's name usually uses the sentence structure of "`who`", but when the keyword ["`which`"] is given, KPQG can control the generated result to use "`which`" as a question syntax. This feature can provide users with the specified sentence syntax when generating questions, helping users to have variability and controllability in the application of generating questions.

4 Conclusion

In this paper, we report the following two findings. First, we find that a very simple QG architecture based on pre-trained language models beats the complicated exam-like QG design (Jia et al., 2020) with or without the keyword indication. Second, by providing keyword information, we can generate results that are closer to the user's expectation. We believe that our method is more practical to educational QG system applications.

Acknowledgement

This work is supported by MOST 110-2634-F-005-006 - project Smart Sustainable New Agriculture Research Center (SMARTer) and MOST Project under grant No.109-2221-E-005-058-MY3.

References

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, et al. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. *arXiv preprint arXiv:2002.12804*.

Ying-Hong Chan and Yao-Chung Fan. 2019. A recurrent bert-based model for question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 154–162.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13042–13054.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. *arXiv preprint arXiv:1705.00106*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Xin Jia, Wenjie Zhou, Xu Sun, and Yunfang Wu. 2020. Eqg-race: Examination-type question generation. *arXiv preprint arXiv:2012.06106*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural

Example 1	
Context	At the age of 12, **Christopher Hirata** already worked on college-level courses, around the time most of us were just in the 7th grade. at the age of 13, this gifted kid became the youngest American to have ever won the gold medal in the International Physics Olympiad. at the age of 16, he was already working with NASA on its project to conquer planet mars. after he was awarded the Ph.D. at Princeton University, he went back to California institute of technology. the next person with very high IQ is Albert Einstein. with an IQ between 160 and 190, Albert Einstein is the genius behind the theory of relativity, which has had great impact on the world of science.
Answer	Christopher Hirata
Gold-Question	Who once worked on the project to conquer planet mars?
DeBERTa-QG	Who was the youngest American to have ever won the gold medal in the International Physics Olympiad?
Keywords 1	"mars"
DeBERTa-KPQG	Who helped NASA on the project to conquer planet mars?
Keywords 2	"mars", "who"
DeBERTa-KPQG	For conquering planet mars, who did he work with NASA?
Example 2	
Context	**Brazil** like the French, Brazilians usually eat a light breakfast. Lunch, the largest meal of the day, usually consists of meat, rice, potatoes, beans, and vegetables. between 6:00 p.m. and 8:00 p.m., people enjoy a smaller meal with their families. Brazilians do not mind eating a hurried or light meal and sometimes buy food from street carts. but they always finish eating before walking away.
Answer	Brazil
Gold-Question	In which country do people consider lunch the largest meal?
DeBERTa-QG	Which country has a light breakfast?
Keywords 1	"largest meal"
DeBERTa-KPQG	Which country's lunch has the largest meal of the day?
Keywords 2	"rice"
DeBERTa-KPQG	Where does lunch usually eat in order of rice, potatoes and vegetables?
Example 3	
Context	Three Central Texas men were honored with the Texas department of public safety's director's award in a Tuesday morning ceremony for their heroism in saving the victims of a fiery two car accident. the accident occurred on March 25 when a vehicle lost control while traveling on a rain-soaked state highway 6 near Baylor camp road. it ran into an oncoming vehicle, leaving the occupants trapped inside as both vehicles burst into flames. **Bonge** was the first on the scene and heard children screaming. he broke through a back window and pulled Mallory Smith, 10, and her sister, Megan Smith, 9, from the wreckage. The girls' mother, Beckie Smith, was not with them at the time of the wreck, as they were traveling with their baby sitter, Lisa Bow Bin.
Answer	Bonge
Gold-Question	Who saved Megan Smith from the damaged car?
DeBERTa-QG	Who was the first on the scene and heard children screaming?
Keywords 1	"Megan Smith"
DeBERTa-KPQG	Who saved Megan Smith from the accident?
Keywords 2	"which"
DeBERTa-KPQG	In the accident, which man was the hero of the victims?

Table 7: Results of KPQG model

language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural ques-

tion generation from text: A preliminary study. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 662–671. Springer.