Kashihara, A. et al. (Eds.) (2024). Proceedings of the 32nd International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

Enhancing Diversity in Difficulty-Controllable Question Generation for Reading Comprehension via Extended T5

Teruyoshi GOTO*, Yuto TOMIKAWA & Masaki UTO*
The University of Electro-Communications, Japan
*{goto, uto}@ai.lab.uec.ac.jp

Abstract: Recently, automatic generation of reading comprehension questions with controllable difficulty levels has attracted growing interest for educational purposes. The latest method for difficulty-controllable question generation employs a two-stage mechanism utilizing two independent large language models. Specifically, given a reading passage and a difficulty level as inputs, it first produces a reference answer using BERT and then generates a corresponding question using GPT-2. However, this two-stage approach has the limitation that the questions generated depend strongly on the reference answers produced beforehand, restricting the diversity of questions. To overcome this limitation, we propose an end-to-end method that enables the simultaneous generation of questions and reference answers by extending T5, a large language model with an encoder mechanism equivalent to BERT and a decoder mechanism equivalent to GPT-2. In our method, T5 is extended to generate answers from its encoder and questions from its decoder, with the encoder's output vector passed to the decoder. Experiments using a benchmark dataset demonstrate that our method significantly improves the diversity of both questions and answers compared with the conventional method while maintaining difficulty controllability.

Keywords: Question generation, reading comprehension, item response theory, deep neural networks, natural language process, large language models

1. Introduction

A typical method for developing reading comprehension skills involves having learners read a large volume of text and then answer various reading comprehension questions. However, manually creating such questions for various reading passages is costly and time-consuming. To address this issue, automatic question generation (QG) for reading comprehension has been gaining attention in recent years (Heilman & Smith, 2010; Labutov et al., 2015).

To efficiently support learning through the use of automatic QG systems, it is desirable to be able to adjust the difficulty of generated questions according to the reading comprehension ability of learners. Accordingly, several difficulty-controllable QG methods have been proposed in recent years (Cheng et al., 2021; Gao et al., 2019; Uto et al., 2023). One of the recent methods (Uto et al., 2023) is designed to generate answers and questions in two steps. First, the answer is generated by inputting a reading passage and a desired difficulty value into the large language model (LLM) BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019). Second, the reading passage, difficulty value, and generated answer are input into another LLM, GPT-2 (Generative Pre-trained Transformer 2) (Radford et al., 2019), to generate a question corresponding to the answer. However, this two-stage approach is limited in that the generation of questions depends heavily on the reference answers produced beforehand. Furthermore, because the answers and questions are generated by distinct models, it is not possible to associate the difficulty value with the pair of questions and answers. These limitations can restrict the diversity of the generated questions and answers.

To overcome this limitation, we propose an end-to-end difficulty-controllable QG method that can generate questions and answers simultaneously in a single deep neural model. Specifically, our method extends T5 (Text-to-Text Transfer Transformer) (Raffel et al., 2020), a LLM that incorporates an encoder mechanism equivalent to BERT and a decoder mechanism equivalent to GPT-2. In our approach, T5 is extended to generate answers using its encoder and questions using its decoder, with the encoder's output vector passed to the decoder. This method enables the simultaneous generation of questions and answers while considering the difficulty of each pair.

In this study, we evaluate the effectiveness of the proposed method by conducting experiments using the SQuAD dataset (Rajpurkar et al., 2016), a benchmark dataset widely used for reading comprehension QG tasks. As a result, we confirmed that our method significantly improves the diversity of both questions and answers compared with the conventional method while maintaining difficulty controllability.

2. Task Definition

This study aims to generate pairs of reading comprehension questions and corresponding reference answers from a reading passage and an arbitrary difficulty value. Here, we assume that the answer to each question consists of a segment of text from the corresponding reading passage, as in typical answer-aware QG tasks. The detailed task definition is as follows.

Let a given reading passage be a word sequence $\boldsymbol{r} = \{r_o \,|o \in \{\,1, \ldots, O\}\}$, where r_o represents the o-th word in the passage, and O is the passage text length. Similarly, let a question text $\boldsymbol{q}$ and an answer text $\boldsymbol{a}$ be word sequences $\boldsymbol{q} = \{q_v \,|v \in \{\,1, \ldots, V\}\}$ and $\boldsymbol{a} = \{a_k \,|k \in \{\,1, \ldots, K\}\}$, respectively, where q_v is the v-th word in the question text, a_k is the k-th word in the answer text, V is the question text length, and K is the answer text length. Note that the answer text $\boldsymbol{a}$ is a subset of the word sequence in the reading passage $\boldsymbol{r}$, namely, $\boldsymbol{a} \subset \boldsymbol{r}$. Using this notation, our task is to generate a question text $\boldsymbol{q}$ and an answer text $\boldsymbol{a}$ given a reading passage $\boldsymbol{r}$ and a target difficulty value b, where the difficulty value b is assumed to be quantified based on item response theory (IRT).

IRT is a statistical method for analyzing the latent ability of examinees and the characteristics of questions (i.e., difficulty and discriminatory power) based on examinees' responses to test questions. IRT has been widely used in various educational and psychological tests because it offers many unique advantages over classical test theory for improving and analyzing various aspects of test properties, a simple and traditional framework based on basic statistics such as mean and variance. The conventional difficulty-controllable QG method (Uto et al., 2023) utilizes IRT to quantify question difficulty because the capability of IRT to model the relationship between question difficulty and learner ability helps in selecting a difficulty level appropriate for each learner's ability. Specifically, the conventional method uses the Rasch model, the simplest IRT model, which represents the probability p that a test-taker with ability value θ answers a question with difficulty b correctly as follows:

$$p = \frac{1}{1 + exp\big(-(\theta - b)\big)}. \tag{1}$$

Following this, we also assume the use of the Rasch model to quantify question difficulty, although the proposed method is applicable to other difficulty quantification methods.

3. Proposed Method

As mentioned in the introduction, the conventional difficulty-controllable QG method (Uto et al., 2023) generates answers questions using BERT and GPT-2, respectively. Our objective is to combine these processes by extending the T5 model, which has encoder and decoder mechanisms equivalent to BERT GPT-2, respectively. Figure 1 shows the architecture of the proposed model.

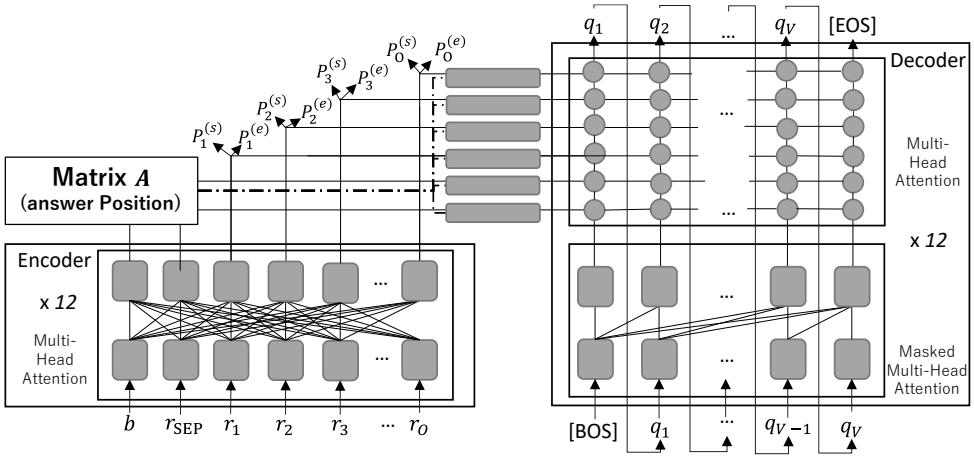

Figure 1. Architecture of the proposed model

The proposed model receives a reading passage $\boldsymbol{r}$ and a difficulty value b in the form of $b, r_{\mathrm{SEP}}, r_1, \ldots, r_O$ as input to the encoder, where r_{SEP} is a special token that represents the boundary between the difficulty and the reading passage. The encoder then outputs the start and end positions of the answer in the reading passage. Specifically, letting the BERT output vector corresponding to the o-th input word be $\boldsymbol{T}_o$, and the output sequence of the encoder be $\boldsymbol{T} = (\boldsymbol{T_1}, \boldsymbol{T_2}, \ldots, \boldsymbol{T}_{(O+2)})$, the probability $P_o^{(s)}(b, \boldsymbol{r})$ that the o-th word of the reading passage $\boldsymbol{r}$ will be the start position of the answer is calculated as

$$P_o^{(s)}(b, \boldsymbol{r}) = \frac{exp(\boldsymbol{S} \cdot \boldsymbol{T}_o)}{\sum_{o'=1}^{O+2} exp(\boldsymbol{S} \cdot \boldsymbol{T}_{o'})}, \tag{2}$$

where $\boldsymbol{S}$ is the trainable parameter. Similarly, the probability $P_o^{(e)}(b, \boldsymbol{r})$ that the o-th word is the end position of the answer is defined by replacing $\boldsymbol{S}$ with another trainable parameter $\boldsymbol{E}$.

Our model passes the information created by the encoder to the decoder to generate questions while considering answer information along with the reading passage and difficulty level. Specifically, using the matrix $\boldsymbol{A}$ that reflects the answer position information and the encoder output sequence $\boldsymbol{T}$, the input for the decoder is defined as

$$\boldsymbol{Z} = \boldsymbol{T} + \boldsymbol{U} \cdot \boldsymbol{A}, \tag{3}$$

where $\boldsymbol{U} \in \mathbb{R}^{\mathrm{H} \times 2}$ ($\mathrm{H} = 768$ is the dimension of the vector $\boldsymbol{T}_o$) is a trainable parameter. Furthermore, $\boldsymbol{A}$ is a matrix with two rows and $(O + 2)$ columns, whose components are defined as follows, given the start position o_s and the end position o_e of the answer:

$$A_{1o} = \begin{cases} 1 & (o_s \le o \le o_e) \\ 0 & (\text{otherwise}) \end{cases}, A_{2o} = \begin{cases} 0 & (o_s \le o \le o_e) \\ 1 & (\text{otherwise}) \end{cases}. \tag{4}$$

Here, A_{jo} corresponds to the j-th row and o-th column of the matrix $\boldsymbol{A}$.

Calculating equation (3) produces the hidden vector sequence $\boldsymbol{Z}$, which embeds the information of the answer positions into the output sequence of the encoder $\boldsymbol{T}$. The decoder of the proposed model generates a question text from the input $\boldsymbol{Z}$ in the same manner as the standard T5 decoder.

3.1 Model Training and Inference

The proposed model can be trained by minimizing the following loss function L, which combines the loss functions of answer generation and question generation.

$$L = \frac{1}{n} \sum_{i=1}^{n} L_{\mathrm{ae}}^{(i)} + L_{\mathrm{qg}}^{(i)} \tag{5}$$

Here, n is the total number of training data. Furthermore, $L_{\mathrm{ae}}^{(i)}$ and $L_{\mathrm{qg}}^{(i)}$ are the loss functions for answer generation and question generation for the i-th training data defined as follows.

$$L_{\mathrm{ae}}^{(i)} = -\sum_{o=1}^{O_i+2} s_o^{(i)} \log P_o^{(s)}(b^{(i)}, \boldsymbol{r}^{(i)}) - \sum_{o=1}^{O_i+2} e_o^{(i)} \log P_o^{(e)}(b^{(i)}, \boldsymbol{r}^{(i)}) \tag{6}$$

$$L_{\mathrm{qg}}^{(i)} = -\log P\left(q_1^{(i)}, \ldots, q_{v-1}^{(i)}, q_{\mathrm{EOS}} | \mathbf{Z}_i\right) = -\sum_{v=1}^{V_i} \log P\left(q_v^{(i)} \middle| \mathbf{Z}_i, \boldsymbol{q}_{<v}^{(i)}\right) - \log P\left(q_{\mathrm{EOS}} \middle| \mathbf{Z}_i, q_1^{(i)}, \ldots, q_{V_i}^{(i)}\right) \quad (7)$$

In equation (6), $b^{(i)}$ and $\boldsymbol{r}^{(i)}$ are the difficulty value and reading passage corresponding to the i-th training data. Furthermore, $s_o^{(i)}$ and $e_o^{(i)}$ are dummy variables that take 1 when the o-th word of the reading passage $\boldsymbol{r}^{(i)}$ is the start/end position of the answer and 0 otherwise. O_i is the number of words in $\boldsymbol{r}^{(i)}$. In equation (7), $q_v^{(i)}$ is the v-th word of the question corresponding to the i-th training data, $\boldsymbol{q}_{<v}^{(i)} = \left(q_1^{(i)}, \ldots, q_{v-1}^{(i)}\right)$, and q_{EOS} is a special token that represents the end of the question. Furthermore, V_i is the number of words in the question, and $\mathbf{Z}_i$ is the vector sequence calculated from equation (3) for the i-th training data.

After the proposed model is trained, it can generate pairs of questions and answers, given an arbitrary difficulty value b and a reading passage $\boldsymbol{r}$, by extracting the answer span that maximizes $P_{o_s}^{(s)}(b, \boldsymbol{r}) + P_{o_e}^{(e)}(b, \boldsymbol{r})$ $(o_s \leq o_e)$ and generating the question words one by one based on the probability $p(q_v | \boldsymbol{Z}, \boldsymbol{q}_{<v})$.

3.2 Construction of Training Data Using IRT and Its Usage

The conventional two-stage method utilized the SQuAD dataset, which consists of reading passages, questions, and answers, for QG model construction and evaluation. However, the SQuAD dataset does not include the difficulty level of questions, making it impossible to construct a difficulty-controllable generator directly. Therefore, the conventional method estimates the difficulty level of each question within SQuAD by using IRT and multiple question-answering (QA) systems with various performance levels. Specifically, this method involves administering each question in SQuAD to a variety of QA systems and collecting their correct and incorrect responses. From these data, the difficulty value for each question is estimated using the Rasch model, and these difficulty estimates are integrated into SQuAD. Following this strategy, we train the proposed model as follows:

1. Train various QA systems on the SQuAD training data.
2. Split the SQuAD test data into 90% denoted as $\boldsymbol{D}_b^{(train)}$ and 10% denoted as $\boldsymbol{D}_b^{(eval)}$.
3. Collect the responses from QA systems for the questions in $\boldsymbol{D}_b^{(train)}$ and estimate their difficulty values, using the Rasch model from the response data.
4. Train the proposed model, using the dataset integrating the difficulty values with questions in $\boldsymbol{D}_b^{(train)}$.

Note that step 2 separates $\boldsymbol{D}_b^{(eval)}$ in order to evaluate the effectiveness of the proposed method in the subsequent experiments.

For a variety of QA systems, we use 12 pre-trained transformer models from Huggingface: bert-base-uncased, bert-large-uncased, roberta-base, roberta-large, distilbert-base-uncased, albert-base-v1, albert-base-v2, albert-large-v2, microsoft/deberta-base, microsoft/deberta-large, microsoft/deberta-v3-base, and microsoft/deberta-v3-large. The QA systems are configured to predict the start and end positions of the answer within the reading passage. The input for the models comprises the concatenation of a passage and a question text, separated by the special token [SEP]. We also train each QA system with varying amounts of data to generate QA systems. Specifically, we sample random subsets of the SQuAD training dataset with 3000, 2400, 1800, 1200, and 600 data points, respectively. These procedures result in 60 QA systems with various performance levels.

4. Experiments

We conducted an experiment to evaluate the quality of the generated questions and answers by specifying various difficulties for various reading passages, using the proposed method.

4.1 Average Correct Answer Rate by Difficulty

In our experiments, we first generated questions and answers, using the proposed model for each of the reading passages in $\boldsymbol{D}_b^{(eval)}$ while changing the specified difficulty levels ranging from −3.0 to 3.0 in increments of 0.1 for each of the reading passages. Subsequently, the generated questions were answered by the 60 developed QA systems and their correct/incorrect responses were collected. Using these response data, this subsection evaluates the difficulty-controllability of the proposed model.

Figure 2 shows the average correct answer rate for each difficulty. The horizontal axis represents the specified difficulty value and the vertical axis represents the average correct answer rate for the proposed method (blue) and the conventional two-stage method (orange). The figure shows that the correct answer rate decreases as the difficulty value increases for both the proposed method and the conventional method. This indicates that the difficulty control in the proposed method is functioning as in the conventional method. Note that, although the overall correct answer rate is lower than that of the conventional method, this is attributed to the increased proportion of high-difficulty or low-quality questions generated by our method as a trade-off for enhancing question variety. A detailed analysis of this phenomenon will be addressed in future work.

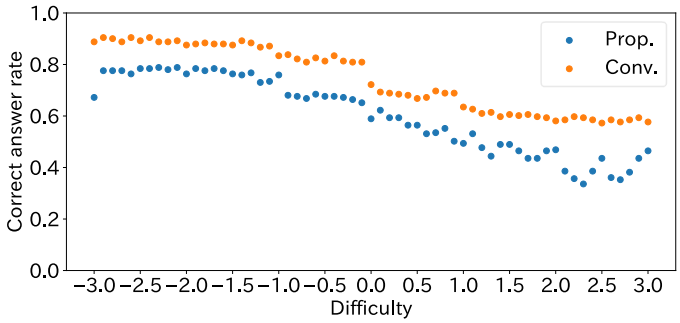

Figure 2. Average correct answer rate for each difficulty level.

4.2 Diversity of Generated Questions and Answers

This subsection evaluates the variety of questions and answers generated by the proposed model and the conventional method for each of the reading passages in $\boldsymbol{D}_b^{(eval)}$ while changing the specified difficulty levels ranging from −3.0 to 3.0 in increments of 0.1. Table 1 presents the number of patterns generated by the conventional and proposed methods. This table shows that the proposed method greatly increased the variety of both questions and answers.

For further analysis, we investigated the percentage of generated questions in which the first word is an interrogative word: *what, who, when, how, where, why,* or *which*. The second and third columns of Table 2 show the results. Focusing on interrogative words other than “what,” the proportion of appearances in the proposed method is more uniform compared with the conventional method. To confirm this, we calculated the kurtosis of the appearance rate of the six interrogative words excluding “what,” with the results shown in the last row of Table 2. The kurtosis is significantly smaller for the proposed method than for the conventional method. Furthermore, we also confirmed the number of patterns of the second word following each interrogative word, with the results shown in the last two columns of Table 2. The results show that, for all interrogative words, the variety of second-word patterns is higher in the proposed method than in the conventional method. These results suggest that the proposed method increases the diversity of questions by generating a wider variety of questions.

Table 1. Number of patterns of generated questions and answers

	Questions	Answers
Conventional	885	757
Proposed	2464	941

Table 2. The rate of interrogative words appearing as the first word and the number of patterns of the second word following each interrogative word

	Rate of first words		No. of second words	
	Prop.	Conv.	Prop.	Conv.
what	62.89	56.93	110	93
who	11.84	17.28	58	52
when	9.09	7.02	7	3
how	7.65	9.50	14	10
where	3.82	1.54	8	6
why	1.37	1.37	5	5
which	0.54	2.18	23	13
Kurtosis	−1.806	0.788	-	-

5. Conclusions

In this study, we proposed a new difficulty-controllable QG method that enhances the variety of generated answers and questions. The proposed method is designed as an extension of T5. The experimental results show that the proposed method significantly improves the diversity of both questions and answers compared with the conventional method while maintaining difficulty controllability. This study requires future work, such as comparing the performance details between the proposed method and previous QG methods, designing appropriate evaluation metrics for QG diversity, and discussing theoretically why the proposed method could improve diversity. Additionally, it is important to demonstrate how the proposed method can be applied to the field of education.

References

Cheng, Y., Li, S., Liu, B., Zhao, R., Li, S., Lin, C., & Zheng, Y. (2021). Guiding the Growth: Difficulty-Controllable Question Generation through Step-by-Step Rewriting. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 5968–5978.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.

Gao, Y., Bing, L., Chen, W., Lyu, M., & King, I. (2019). Difficulty Controllable Generation of Reading Comprehension Questions. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 4968–4974.

Heilman, M., & Smith, N. A. (2010). Good Question! Statistical Ranking for Question Generation. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 609–617.

Labutov, I., Basu, S., & Vanderwende, L. (2015). Deep Questions without Deep Understanding. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 889–898.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. *OpenAI blog*.

Raffel, C., Shazeer, N., Roberts, A., & Others. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140), 1–67.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2383–2392.

Uto, M., Tomikawa, Y., & Suzuki, A. (2023). Difficulty-Controllable Neural Question Generation for Reading Comprehension using Item Response Theory. *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications*, 119–129.