Difficulty-Controllable Neural Question Generation for Reading Comprehension using Item Response Theory

Masaki Uto and **Yuto Tomikawa** and **Ayaka Suzuki**
The University of Electro-Communications
Tokyo, Japan
{uto, tomikawa, suzuki_ayaka}@ai.lab.uec.ac.jp

Abstract

Question generation (QG) for reading comprehension, a technology for automatically generating questions related to given reading passages, has been used in various applications, including in education. Recently, QG methods based on deep neural networks have succeeded in generating fluent questions that are pertinent to given reading passages. One example of how QG can be applied in education is a reading tutor that automatically offers reading comprehension questions related to various reading materials. In such an application, QG methods should provide questions with difficulty levels appropriate for each learner's reading ability in order to improve learning efficiency. Several difficulty-controllable QG methods have been proposed for doing so. However, conventional methods focus only on generating questions and cannot generate answers to them. Furthermore, they ignore the relation between question difficulty and learner ability, making it hard to determine an appropriate difficulty for each learner. To resolve these problems, we propose a new method for generating question–answer pairs that considers their difficulty, estimated using item response theory. The proposed difficulty-controllable generation is realized by extending two pre-trained transformer models: BERT and GPT-2.

1 Introduction

Automatic question generation (QG) for reading comprehension is the task of automatically generating reading comprehension questions related to given reading passages. Various QG methods have been developed in the natural language processing (NLP) research field (Zhang et al., 2021). They have also been used in various educational systems, such as intelligent tutoring systems, writing support systems, and knowledge assessment systems (Ghanem et al., 2022; Kurdi et al., 2020; Le et al., 2014; Rathod et al., 2022; Zhang et al., 2021).

Early QG methods have relied on rule-based or template-based approaches, which use handcrafted rules or templates to generate an interrogative question text from a declarative text (Zhang et al., 2021). However, preparing those QG methods for a target application is time-consuming and labor-intensive because achieving high-quality QG requires well-designed rules and templates for each application (Chen et al., 2021; Zhang et al., 2021). End-to-end QG methods based on deep neural networks have received wide attention as a means of overcoming this limitation (Chan and Fan, 2019; Du et al., 2017; Ushio et al., 2022; Yu et al., 2023; Zhang et al., 2021). Earlier neural QG methods were designed as sequence-to-sequence (seq2seq) models based on recurrent neural networks (RNNs) and attention mechanisms (Du et al., 2017), while recent methods are based on pre-trained transformer models (Gao et al., 2019; Ghanem et al., 2022; Lee and Lee, 2022; Rathod et al., 2022; Ushio et al., 2022), including BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019), GPT-2 (Generative Pre-trained Transformer 2) (Radford et al., 2019), BART (Bidirectional and Auto-Regressive Transformers) (Lewis et al., 2020), and T5 (Text-to-Text Transfer Transformer) (Raffel et al., 2022). Those methods have succeeded in generating fluent questions that are pertinent to given reading passages.

A representative application of how QG can be used for educational purposes is a reading tutor that automatically offers reading comprehension questions related to various reading materials (Kurdi et al., 2020; Le et al., 2014; Rathod et al., 2022; Zhang et al., 2021). This helps to focus learners' attention on the reading materials and offers the opportunity to observe any misconceptions they might have (Kurdi et al., 2020), which supports the development of reading comprehension skills. To enhance such learning, it is useful to provide questions with difficulty levels appropriate for each

Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023), pages 119–129
July 13, 2023 ©2023 Association for Computational Linguistics

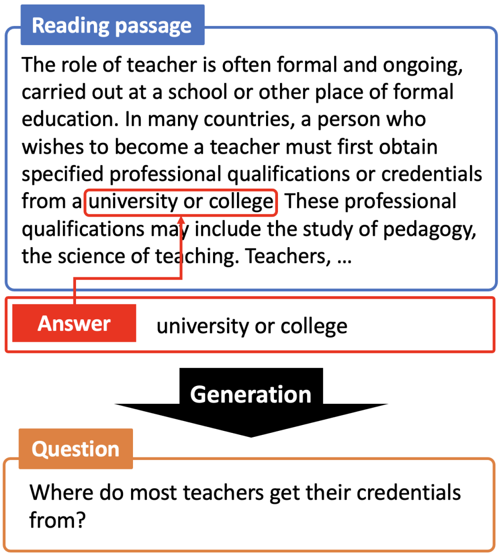
Reading passage
The role of teacher is often formal and ongoing, carried out at a school or other place of formal education. In many countries, a person who wishes to become a teacher must first obtain specified professional qualifications or credentials from a university or college. These professional qualifications may include the study of pedagogy, the science of teaching. Teachers, …
Answer
university or college
Generation
Question
Where do most teachers get their credentials from?

Figure 1: Conventional QG task.

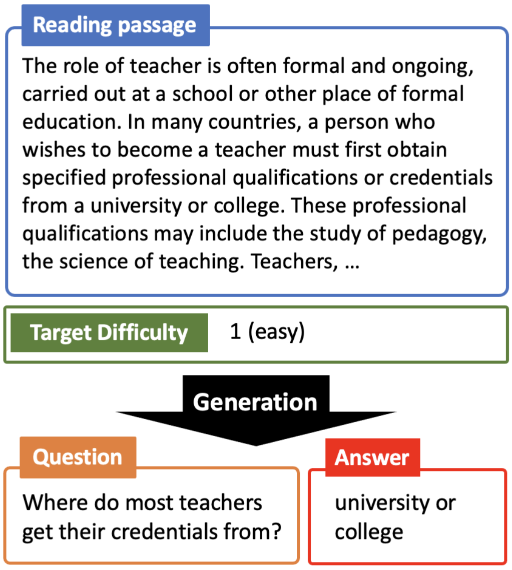
Reading passage
The role of teacher is often formal and ongoing, carried out at a school or other place of formal education. In many countries, a person who wishes to become a teacher must first obtain specified professional qualifications or credentials from a university or college. These professional qualifications may include the study of pedagogy, the science of teaching. Teachers, …
Target Difficulty
1 (easy)
Generation
Question
Where do most teachers get their credentials from?
Answer
university or college

Figure 2: Our QG task.

learner's reading ability. Such adaptivity is a core component of recent AI-based intelligent tutoring systems.

Difficulty control of QG is a relatively new task (Cheng et al., 2021; Kurdi et al., 2020), and thus previous research on difficulty-controllable QG for reading comprehension is still limited (Chen et al., 2021; Cheng et al., 2021; Gao et al., 2019). There are currently only two conventional methods; the first uses an RNN-based seq2seq model in which hidden states before its encoder are modified to receive a difficulty as input that is categorized as either easy or hard (Gao et al., 2019), and the second is a multi-hop QG (Cheng et al., 2021) that takes the question difficulty to be the number of inference steps required to answer a question and aims to generate questions while controlling the number of required inference steps. However, both methods have the following limitations that prevent them from generating questions appropriate for a learner's ability.

1. They ignore the relation between question difficulty and learner ability, making it difficult to determine an appropriate difficulty for each learner.

2. They are answer-aware QG methods, which generate questions given a reading passage and an answer text, as illustrated in Fig. 1, and thus cannot generate question–answer pairs. Without correct answers, systems cannot score learners' answers automatically, meaning adaptive systems will not work efficiently. Furthermore, controlling difficulty in answer generation is also important because difficulty is a property that generally depends on both questions and answers.

To resolve these problems, we propose a new method for generating question–answer pairs that considers the difficulty associated with learners' ability. A unique feature of our method is that it uses item response theory (IRT) (Lord, 1980), a test theory based on mathematical models, to quantify the difficulty of each question–answer pair. IRT is based on statistical models that define the relation between question difficulty and learner ability, and thus it helps us to select a difficulty appropriate for each learner's ability. For these reasons, we aim to generate question–answer pairs while considering their difficulty, quantified by IRT. For our QG method, we first propose a method for constructing a training dataset consisting of quadruplets (reading passage, question text, answer text, and IRT-based difficulty), based on the SQuAD dataset, which is the most popular benchmark dataset for the reading comprehension QG task. Then, we propose a difficulty-controllable generation method for question–answer pairs that can be trained using this dataset. Our generation method consists of two pre-trained transformer-based models, which are extended to take IRT-based difficulty values as input: *a difficulty-controllable answer extraction model using BERT*, and *a difficulty-controllable answer-aware QG model using GPT-2*.

To our knowledge, this is the first difficulty-controllable QG method aimed at generating question–answer pairs corresponding to IRT-based difficulty.

2 Task Definition

The task tackled in this study is to generate a reading comprehension question and a corresponding correct answer, given a reading passage and a target difficulty value. Here, we assume that a correct answer to each question consists of a segment of text from the corresponding reading passage, as in typical answer-aware QG tasks (Rajpurkar et al., 2016). Fig. 2 shows an outline of our task.

The detailed task definition is as follows. Let a given reading passage be a word sequence $\boldsymbol{r} = \{r_i \mid i \in \{1, \ldots, I\}\}$, where r_i represents the i-th word in the passage, and I is the passage text length. Similarly, let a question text $\boldsymbol{q}$ and an answer text $\boldsymbol{a}$ be word sequences $\boldsymbol{q} = \{q_j \mid j \in \{1, \ldots, J\}\}$ and $\boldsymbol{a} = \{a_k \mid k \in \{1, \ldots, K\}\}$, respectively, where q_j is the j-th word in the question text, a_k is the k-th word in the answer text, J is the question text length, and K is the answer text length. Note that the answer text $\boldsymbol{a}$ must be 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 IRT, as explained in the introduction.

3 Item Response Theory

IRT (Lord, 1980) is a statistical framework used in psychometrics and educational measurement to analyze examinees' responses to test items (*items* corresponds to *questions* in our study). One of the unique characteristics of IRT is that it estimates two types of latent factors from response data: examinee ability and item characteristics. Examinee ability refers to the latent trait or ability that the test is intended to measure, such as reading comprehension ability in our context. Item characteristics refer to the properties of test items, including their difficulty level and their ability to discriminate examinee ability. IRT uses probabilistic models, called IRT models, to estimate examinees' abilities and item characteristics from response data that typically consist of a binary variable taking one if an examinee answers an item correctly and zero otherwise.

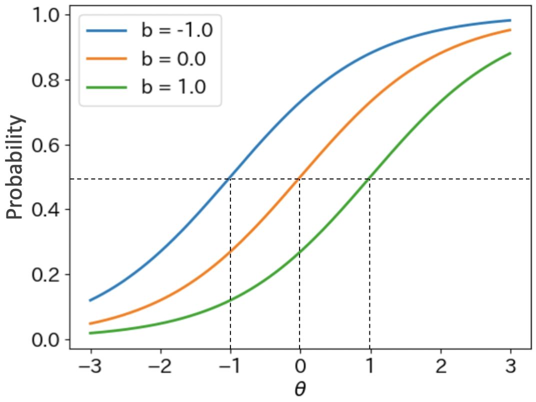

Figure 3: Item response curves for a Rasch model with different item difficulty values.

IRT has been widely used in various educational and psychological tests because it has the following typical benefits (Uto and Ueno, 2020) compared with classical test theory (a simple and traditional framework based on basic statistics such as mean, variance, and correlation coefficients): 1) IRT provides detailed information about item properties, including difficulty and discrimination, which helps test developers identify problematic items and improve test quality. 2) IRT provides accurate estimates of examinee ability and item properties. 3) The abilities of examinees who take different tests can be estimated on the same scale because examinee ability is estimated considering the effects of the items' characteristics. 4) IRT is the basis for computerized adaptive testing (CAT), which can reduce test length and increase measurement precision by selecting appropriate items for a target examinee's ability (van der Linden and Glas, 2010).

This study uses the Rasch model (a one-parameter logistic model), which is the most traditional and well-known IRT model. The Rasch model defines the probability that the m-th examinee correctly answers the n-th item as

$$p_{nm} = \frac{\exp(\theta_m - b_n)}{1 + \exp(\theta_m - b_n)}, \qquad (1)$$

where b_n represents the difficulty of the n-th item and θ_m represents the latent ability of the m-th examinee.

To explain the relationship between the latent ability θ and the difficulty parameter b in the Rasch model, Fig. 3 depicts item response curves (IRCs) of the Rasch model, which are drawn by plotting the probability p_{nm}, for three different difficulty values. In the figure, the horizontal axis shows θ, the vertical axis shows the probability p_{nm}, and three solid curves show the IRC for three items with different difficulty values.

These IRCs show that examinees with higher θ have a higher probability of responding correctly to each item. We can also see that the IRC shifts to the right as the item difficulty value increases, reflecting the fact that higher ability is required to correctly answer items with high b. Furthermore, under the Rasch model, the probability that an examinee with ability θ correctly answers the question with difficulty b becomes 0.5 when $\theta = b$.

The IRT model parameters are generally estimated in two phases, namely, *item calibration* and *ability estimation*, in order to guarantee asymptotic consistency. Item calibration estimates the item parameters from response data by marginalizing the examinee ability θ from the likelihood in order to ensure the asymptotic consistency of the item parameter estimates. Specifically, marginal maximum likelihood (MML) estimation using an expectation-maximization (EM) algorithm has been widely used for item calibration (Baker and Kim, 2004). Given calibrated item parameters, the ability estimation phase calculates the examinee's ability θ. An expected a posteriori (EAP) estimation, a type of Bayesian estimation, is generally used for the ability estimation (Fox, 2010; Uto et al., 2023).

This study aims to quantify question difficulty based on the IRT. The next section explains how to prepare the dataset with IRT-based difficulty, which is required to train our QG model.

4 Creating a Dataset with IRT-based Question Difficulty

We require an appropriate dataset to construct our QG method for solving the difficulty-controllable QG task defined in Section 2. While several popular datasets have been developed for general reading comprehension QG tasks (Zhang et al., 2021), the most popular is SQuAD (Rajpurkar et al., 2016), which consists of over 100,000 question–answer pairs from Wikipedia articles. Specifically, SQuAD is a collection of triplets $(\boldsymbol{r}, \boldsymbol{q}, \boldsymbol{a})$, where each answer $\boldsymbol{a}$ is a text fragment from a corresponding reading passage $\boldsymbol{r}$ and each reading passage $\boldsymbol{r}$ corresponds to a paragraph of a Wikipedia article. However, to construct a difficulty-controllable QG method, we require a dataset consisting of quadruplets $(\boldsymbol{r}, \boldsymbol{q}, \boldsymbol{a}, b)$. Thus, we first propose a method for extending the SQuAD dataset by appending the IRT-based difficulty values for each question–answer pair. The details for doing so are as follows.

1. **Collecting response data for each question–answer pair:** We collect answers from multiple respondents to each question in the SQuAD dataset and grade those answers as correct or incorrect. Ideally, we should gather responses from a population of target learners, but this is highly expensive and time-consuming. Thus, we substitute actual learner responses with automated question–answering (QA) systems, in the same way that several previous difficulty-controllable QG studies have done (Chen et al., 2021; Gao et al., 2019).

2. **Difficulty estimation using IRT:** Using the collected response data, we estimate the question difficulty by using the Rasch model and the item calibration procedure introduced in Section 3. Note that the difficulty value generally depends on the contents of both the question and the answer.

3. **Creating a dataset with difficulty estimates:** We construct a dataset consisting of quadruplets $(\boldsymbol{r}, \boldsymbol{q}, \boldsymbol{a}, b)$ by appending the estimated difficulty values b into the triplets $(\boldsymbol{r}, \boldsymbol{q}, \boldsymbol{a})$ of the SQuAD dataset.

5 Proposed Method

Our difficulty-controllable QG method, which is trained using the extended SQuAD dataset, is realized by performing the following two tasks in sequence: (1) *difficulty-controllable answer extraction* that extracts an answer text from a given reading passage while considering a target difficulty value, and (2) *difficulty-controllable answer-aware QG* that generates a question given a reading passage, an answer text, and a target difficulty value. Details of each are provided in the following sections.

5.1 Difficulty-Controllable Answer Extraction

We perform the difficulty-controllable answer extraction using BERT (Devlin et al., 2019). BERT is a pre-trained multilayer bidirectional transformer with 340M parameters, a transformer being a neural network architecture based on self-attention mechanisms. BERT is pre-trained on large amounts

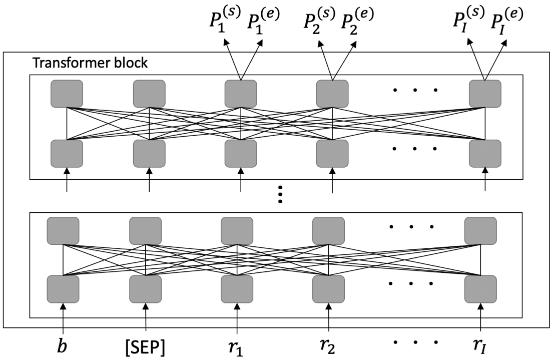

Figure 4: Difficulty-controllable answer extraction using BERT.

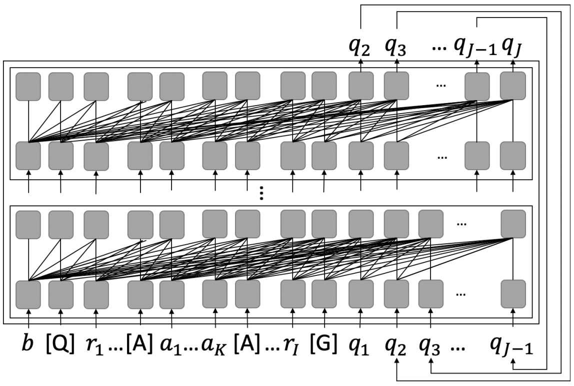
q_2 q_3 ... q_{J-1} q_J
b [Q] r_1...[A] a_1...a_K [A]...r_I [G] q_1 q_2 q_3 ... q_{J-1}

Figure 5: Difficulty-controllable answer-aware question generation using GPT-2.

of text data over two unsupervised learning tasks, *masked language modeling* and *next-sentence prediction*. The pre-trained BERT can be applied to various downstream tasks by fine-tuning the model with a task-specific supervised dataset after adding task-specific output layers. We use fine-tuned BERT for the answer extraction task because BERT has been widely used before in various text extraction tasks (Srikanth et al., 2020).

To perform answer extraction using BERT, we add output layers that predict the start and end positions of the answer text within a given reading passage. Specifically, we add two dense layers with softmax activation to transform each BERT output vector, which correspond to the words within a given reading passage, into probability values for whether the word is at the start or end position of the answer text. By extracting the word sequence within the start and end positions, which take the maximum probabilities, we can extract an answer text from a given reading passage.

We control the difficulty of the answer extraction by inputting a difficulty value with the reading passage. Specifically, the input for our model is defined as

$$b, [\mathrm{SEP}], r_1, r_2, r_3, \ldots, r_I, \tag{2}$$

where [SEP] is the special token used to separate the difficulty value and the reading passage. This input is what enables the model to extract an answer text from a reading passage while considering the input difficulty value. Fig. 4 shows an outline of the answer extraction model.

We can fine-tune the answer extraction model by using a collection of triplets $(\boldsymbol{r}, \boldsymbol{a}, b)$, which can be obtained from the extended SQuAD dataset explained in Section 4. This fine-tuning is performed by minimizing cross-entropy loss between the predicted positions of the start and end of an answer text and their true positions.

5.2 Difficulty-Controllable Answer-Aware Question Generation

We use GPT-2 to perform difficulty-controllable answer-aware QG. GPT-2 is a transformer-based language model with more than 1.5 billion parameters, and it is pre-trained on more than 8 million documents using an unsupervised learning process called language modeling, which sequentially predicts the next word from the current word sequence. We use GPT-2 for the QG tasks because it has been widely used before in various text generation tasks.

Conventional answer-aware QG models based on pre-trained language models (Srivastava and Goodman, 2021), including GPT-2, are implemented by designing the model's input as

$$r_1, \ldots, [\mathrm{A}], a_1, \ldots, a_K, [\mathrm{A}], \ldots, r_I, [\mathrm{G}], \tag{3}$$

where [A] is a special token representing an answer's start and end positions within a reading passage. [G] is also a special token representing the end of a reading passage. Conventional QG models receive this input and generate a question text after the special token [G].

To implement difficulty-control for the answer-aware QG model, we concatenate a target difficulty value to the conventional input form above using

$$b, [\mathrm{Q}], r_1, \ldots, [\mathrm{A}], a_1, \ldots, a_K, [\mathrm{A}], \ldots, r_I, [\mathrm{G}], \tag{4}$$

where [Q] is the special token used to separate the difficulty value and the given reading passage. Given this input, the model generates a question text based on a reading passage, an answer, and a target difficulty value. Fig. 5 presents an outline of our QG model.

We can fine-tune the answer-aware QG model by using a dataset consisting of quadruplets $(\boldsymbol{r}, \boldsymbol{q}, \boldsymbol{a}, b)$, explained in Section 4. Specifically, we prepare the following format data and train GPT-2 by maximizing the log-likelihood for question texts:

$$b, [\mathrm{Q}], r_1, \ldots, [\mathrm{A}], a_1, \ldots, a_K, [\mathrm{A}], \ldots, r_I, [\mathrm{G}], q_1, \ldots, q_J. \quad (5)$$

5.3 Determining Appropriate Difficulty based on IRT

As explained in Section 1, IRT helps us to select a difficulty appropriate for each learner's ability. Earlier studies on adaptive learning have demonstrated that offering questions with a difficulty at which the learner would have a 50% chance of answering correctly is the most effective approach for learning (Ueno and Miyazawa, 2018). As explained in Section 3, under the Rasch model, the probability that a learner with ability θ correctly answers the question with difficulty b becomes 0.5 when $\theta = b$. Thus, we can generate questions with a difficulty appropriate for each learner using the following steps inspired by the framework of CAT (van der Linden and Glas, 2010).

1. Provide some questions randomly to a learner and collect response data.
2. Estimate the learner's ability using the Rasch model and the response data.
3. Generate a question–answer pair by inputting the estimated ability value as the difficulty value into the proposed QG method.

Furthermore, by repeating procedures 2–3, we can enable adaptive QG.

6 Experiments

In this section, we demonstrate that our proposed method can generate questions and answers corresponding to target IRT-based difficulty values.

6.1 Data preparation

For our experiment, we first constructed an extended SQuAD dataset consisting of quadruplets $(\boldsymbol{r}, \boldsymbol{q}, \boldsymbol{a}, b)$ by following the procedures explained in Section 4. The original SQuAD dataset was divided into training data (90%) and test data (10%) in advance. In this experiment, we trained QA models using the training data and constructed an extended dataset using the test data. The detailed procedures were as follows.

1. **Training QA models:** Using the SQuAD training data, we trained five different QA models: two neural models, the BERT-based model (Devlin et al., 2019) and the ALBERT-based model (Lan et al., 2020), and three feature-based models, a logistic regression model using dependency-tree features (Rajpurkar et al., 2016), a logistic regression model using selected features (Rajpurkar et al., 2016), and a sliding-window model using bag-of-words features (Richardson et al., 2013).
2. **Collecting response data for each question:** We collected answers from the five QA models for all the questions in the SQuAD test data and scored those answers.
3. **Estimating IRT-based difficulty:** Using the correct/incorrect response data, we estimated the difficulty of each question using the Rasch model. Here, we conducted the estimation using the MML method with the EM algorithm. The difficulty values were estimated to be one of six values (-3.96, -1.82, -0.26, 0.88, 2.01, 3.60), where questions with lower difficulty estimates indicate that they were easier. We linearly transformed the difficulty values estimated on the real value scale (-3.96, -1.82, -0.26, 0.88, 2.01, 3.60) to positive integer values (1, 29, 49, 64, 79, 100) to make it easier for the language models to understand the numerical inputs. Table 1 shows the ability estimates $\hat{\theta}$ for the five QA systems, where the abilities were estimated by the EAP estimation using a Gaussian quadrature (Baker and Kim, 2004), given the calibrated item-difficulty parameters. The table shows that the abilities of the five QA systems differ greatly.

Table 1: Ability estimates $\hat{\theta}$ of five QA systems.

	$\hat{\theta}$
BERT-based model	2.25
ALBERT-based model	1.28
Logistic regression	0.52
Logistic regression (selected features)	-0.64
Sliding-window model	-2.84

A larger variety of respondent abilities is generally effective for clearly distinguishing the difficulty among questions, suggesting that our use of these five QA systems in our experiment is reasonable. Note that ability and question difficulty are estimated assuming a standard normal distribution, meaning that these estimates distribute approximately on a scale with a mean of 0 and a standard deviation of 1.

4. **Creating a dataset with difficulty estimates:** We created a dataset $\mathcal{D}$ consisting of quadruplets $(\boldsymbol{r}, \boldsymbol{q}, \boldsymbol{a}, b)$ by integrating the obtained IRT-based difficulty values and SQuAD test data.

6.2 Experimental Procedures

We conducted the following experiment using the created dataset $\mathcal{D}$ and the original SQuAD training data.

1. Using the original SQuAD training data, we fine-tuned the proposed answer extraction model and the answer-aware QG model, ignoring the difficulty. This fine-tuning was done by removing the difficulty value from the input of the proposed models. Although this procedure is not mandatory, we applied it to improve the basic QG performance.

2. We randomly divided the dataset $\mathcal{D}$ into parts, one 90% (designated as $\mathcal{D}^{(train)}$) and the other 10% (designated as $\mathcal{D}^{(eval)}$). Then, using the 90% dataset $\mathcal{D}^{(train)}$, we fine-tuned the difficulty-controllable answer extraction model and the difficulty-controllable answer-aware QG model, where the initial model parameters were set to the values obtained in procedure 1.

3. We generated questions and answers for each reading passage in the remaining 10% dataset $\mathcal{D}^{(eval)}$, given each of the six difficulty values (1, 29, 49, 64, 79, 100). Using the generated questions and answers, we conducted both an automatic evaluation and a human evaluation, which are explained below.

We used *PyTorch* and the *Transformers* library to implement the proposed models and the neural QA systems. Furthermore, we used *R* and the *TAM* package to perform the IRT parameter estimation.

Table 2: Number of questions corresponding to the six difficulty values in $D^{(train)}$ and $D^{(eval)}$.

Difficulty	$D^{(train)}$	$D^{(eval)}$
1	662 (0.07)	90 (0.1)
29	2,739 (0.28)	269 (0.3)
49	1,623 (0.17)	144 (0.16)
64	2,362 (0.24)	195 (0.22)
79	1,389 (0.14)	107 (0.12)
100	909 (0.09)	81 (0.09)

Numbers in parentheses indicate ratios.

Here, we summarize the basic statistics of the datasets $D^{(train)}$ and $D^{(eval)}$, which we developed in the above procedure 2 to train and evaluate our difficulty-controllable QG method. First, the number of reading passages in $D^{(train)}$ and $D^{(eval)}$ was 1,860 and 207, respectively. Next, the average number of questions per reading passage in $D^{(train)}$ and $D^{(eval)}$ was 5.21 and 4.28. Furthermore, Table 2 shows the number of questions corresponding to the six difficulty values in each dataset. From these results, we can confirm that the basic statistics and the difficulty distributions are similar between the two datasets, indicating that the dataset $\mathcal{D}$ was randomly divided into $D^{(train)}$ and $D^{(eval)}$ without bias.

6.3 Automatic Evaluation

We performed an automatic evaluation by calculating the percentage of correct answers given by the neural QA systems (BERT-based and ALBERT-based QA models) to the questions generated for each difficulty. Fig. 6 shows the results, which indicate that the correct answer rate of QA systems

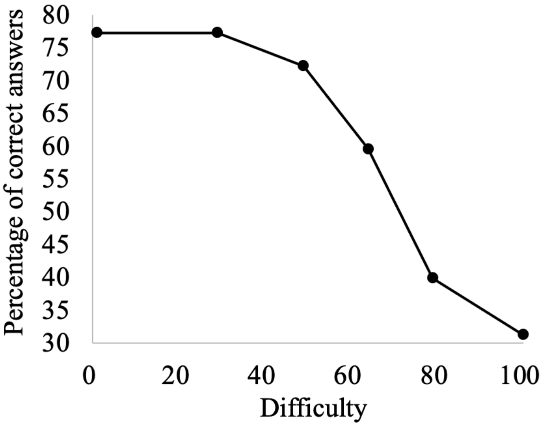

Figure 6: Percentage of correct answers by neural QA systems to questions generated for each difficulty.

Table 3: Examples of generated questions and answers for different difficulties.

Reading passage	Much of the work of the Scottish Parliament is done in committee. The role of committees is stronger in the Scottish Parliament than in other parliamentary systems, partly as a means of strengthening the role of backbenchers in their scrutiny of the government and partly to compensate for the fact that there is no revising chamber. The principal role of committees in the Scottish Parliament is to take evidence from witnesses, conduct inquiries and scrutinise legislation.
Difficulty	1 (easiest)
Question	Where is much of the work of the Scottish Parliament done?
Answer	committee
Difficulty	100 (most difficult)
Question	What is the purpose of the chairman and member of the committee?
Answer	take evidence from witnesses, conduct inquiries and scrutinise legislation

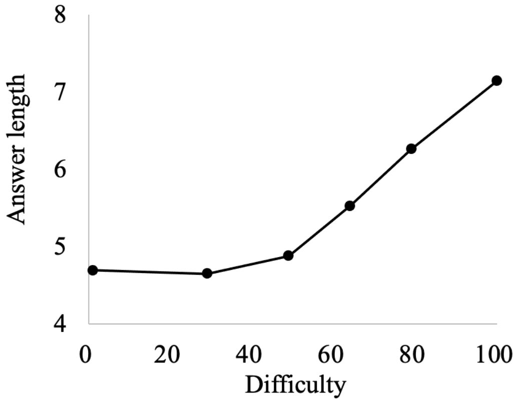

Figure 7: Average word length in generated answers for each difficulty.

decreases as the difficulty increases. This suggests that our proposed method generates questions that reflect the given difficulty.

Furthermore, we calculated the average word length in the generated answer texts for each difficulty. Fig. 7 shows the results, and these indicate that the average word length in the generated answer texts increases as the target difficulty values increase. Considering that questions with longer and more complex answers are generally difficult to correct perfectly, this result suggests that the proposed method extracts answers that reflect the specified difficulty.

Table 3 shows examples of the generated question–answer pairs when given the same reading text but different difficulty values, demonstrating that higher difficulty values correspond to longer answers.

6.4 Human Evaluation

For the human evaluation, we randomly selected ten reading passages from $\mathcal{D}^{(eval)}$ and extracted question–answer pairs for the six difficulty values corresponding to each reading passage from the generated data obtained in experimental procedure 3. Then, the 60 question–answer pairs were evaluated by four human judges according to the following four evaluation metrics.

1. *Difficulty*: The subjective difficulty evaluation for each question–answer pair, graded on a scale from one to five, where smaller grades mean the question was easier.

2. *Fluency*: Evaluation of the grammatical correctness of generated questions, graded on a three-point scale: Yes, Acceptable, and No.

3. *Relevance*: Evaluation of the content relevance between generated questions and reading passages, graded on a binary scale: Yes and No.

4. *Answerability*: Evaluation of the answerability of each generated question–answer pair from a given reading passage, graded on a four-point scale: Yes, Partially, and No. Here, “Partially” indicates that the generated answer does not entirely match the correct answer for the generated question but partially includes the correct answer.

Fig 8 shows the relation between the input difficulty values and the averaged scores in the human difficulty evaluation for the generated questions. They indicate that the human subjects judged the questions generated with higher difficulty values to

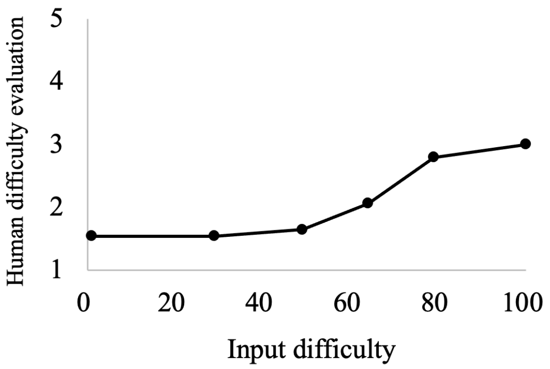

Figure 8: Human difficulty evaluation of generated question–answer pairs for each difficulty.

Table 4: The fluency, relevance, and answerability of generated questions and answers.

Fluency	Yes	Acceptable	No
	76.0%	16.3%	7.6%
Relevance	Yes	No	
	87.8%	12.2%	
Answerability	Yes	Partially	No
	67.4%	17.4%	15.3%

be more difficult. This indicates that the proposed method can appropriately control the difficulty of generated question–answer pairs.

Table 4 gives the results for *Fluency*, *Relevance*, and *Answerability*. It shows that more than 90% of the questions were generated with correct or acceptable grammar, and about 90% appropriately reflected the content of the given reading passages. Furthermore, about 70% of generated question–answer pairs were completely answerable, and about 85% were partially appropriate. These results indicate that fluency and relevance are acceptable but further improvement might be required in terms of answerability, which is planned for future work.

7 Conclusion

In this study, we proposed a new neural QG method that generates question–answer pairs while considering their difficulty, estimated using IRT. We also evaluated the effectiveness of this method through experiments using SQuAD.

One limitation of this study is that we used only the SQuAD dataset in our experiments. The SQuAD dataset has often been criticized because it is overly dependent on the similarity of question/answer sentences rather than on human-type reasoning, meaning it requires only superficial reading skills. Thus, examining the effectiveness of our proposed method by applying it to various other datasets will be an important future task.

Furthermore, in the human evaluation experiment presented in Section 6.4, we examined only 60 question–answer pairs generated through the proposed model from ten randomly selected reading passages. The relatively small scale of the experiment is due to the high workload required for people to carefully evaluate the various properties of a large number of questions. However, in the future, we aim to conduct a larger-scale human evaluation in order to increase the reliability of the experimental results.

Although the present study used only five QA systems, the use of a larger number of QA systems with different characteristics is expected to improve the accuracy of question-difficulty estimation and provide difficulty estimates with finer granularity. Therefore, examining the effects of increasing the number and variability of QA systems will be another future direction of this research.

We also need to confirm in greater detail whether QA systems can be substituted for human learners. A comparison between IRT-based question difficulties calibrated from the responses of QA systems as well as human learners might be a plausible approach.

Another future goal is to develop a method of transforming the scale of the IRT-based difficulty, estimated based on QA systems, into a scale appropriate for a population of target learners. Such a scaling adjustment is expected to be achievable by using *equating*, which is a well-established technique in IRT.

Furthermore, our QG method is easily extended to adaptive QG systems based on the framework of computerized adaptive testing, as mentioned in Section 5.3. Developing and evaluating such an adaptive system using our QG method will also be our focus in future work.

Acknowledgements

This work was supported by 19H05663, and 21H00898.

References

F.B. Baker and Seock Ho Kim. 2004. *Item Response Theory: Parameter Estimation Techniques*. CRC Press, Boca Raton, FL, USA.

Ying-Hong Chan and Yao-Chung Fan. 2019. A recurrent BERT-based model for question generation. In *Proc. Workshop on Machine Reading for Question Answering*, pages 154–162.

Feng Chen, Jiayuan Xie, Yi Cai, Tao Wang, and Qing Li. 2021. Difficulty-controllable visual question generation. In *Proc. Web and Big Data: International Joint Conference*, pages 332–347. Springer-Verlag.

Yi Cheng, Siyao Li, Bang Liu, Ruihui Zhao, Sujian Li, Chenghua Lin, and Yefeng Zheng. 2021. Guiding the growth: Difficulty-controllable question generation through step-by-step rewriting. In *Proc. Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing*, pages 5968–5978.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pages 1342–1352.

Jean-Paul Fox. 2010. *Bayesian item response modeling: Theory and applications*. Springer, New York, NY, USA.

Yifan Gao, Lidong Bing, Wang Chen, Michael Lyu, and Irwin King. 2019. Difficulty controllable generation of reading comprehension questions. In *Proc. International Joint Conference on Artificial Intelligence*, pages 4968–4974.

Bilal Ghanem, Lauren Lutz Coleman, Julia Rivard Dexter, Spencer von der Ohe, and Alona Fyshe. 2022. Question generation for reading comprehension assessment by modeling how and what to ask. In *Findings of the Association for Computational Linguistics*, pages 2131–2146.

Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. 2020. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30:121–204.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *Proc. International Conference on Learning Representations*.

Nguyen-Thinh Le, Tomoko Kojiri, and Niels Pinkwart. 2014. Automatic question generation for educational applications – the state of art. In *Advanced Computational Methods for Knowledge Engineering*, pages 325–338.

Seungyeon Lee and Minho Lee. 2022. Type-dependent prompt CycleQAG : Cycle consistency for multi-hop question generation. In *Proc. International Conference on Computational Linguistics*, pages 6301–6314.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

F.M. Lord. 1980. *Applications of item response theory to practical testing problems*. Routledge, Evanston, IL, USA.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2022. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proc. Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Manav Rathod, Tony Tu, and Katherine Stasaski. 2022. Educational multi-question generation for reading comprehension. In *Proc. Workshop on Innovative Use of NLP for Building Educational Applications*, pages 216–223.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proc. Conference on Empirical Methods in Natural Language Processing*, pages 193–203.

Anirudh Srikanth, Ashwin Shankar Umasankar, Saravanan Thanu, and S. Jaya Nirmala. 2020. Extractive text summarization using dynamic clustering and coreference on BERT. In *Proc. International Conference on Computing, Communication and Security*, pages 1–5.

Megha Srivastava and Noah Goodman. 2021. Question generation for adaptive education. In *Proc. Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing*, pages 692–701.

Maomi Ueno and Yoshimitsu Miyazawa. 2018. IRT-based adaptive hints to scaffold learning in programming. *IEEE Transactions on Learning Technologies*, 11(4):415–428.

Asahi Ushio, Fernando Alva-Manchego, and Jose Camacho-Collados. 2022. Generative language models for paragraph-level question generation. In *Proc. Conference on Empirical Methods in Natural Language Processing*.

Masaki Uto, Itsuki Aomi, Emiko Tsutsumi, and Maomi Ueno. 2023. Integration of prediction scores from various automated essay scoring models using item response theory. *IEEE Transactions on Learning Technologies*, pages 1–18.

Masaki Uto and Maomi Ueno. 2020. A generalized many-facet Rasch model and its Bayesian estimation using Hamiltonian Monte Carlo. *Behaviormetrika*, 47(2):469–496.

Wim J. van der Linden and Cees A.W. Glas. 2010. *Elements of Adaptive Testing*. Springer New York.

Jianxing Yu, Qinliang Su, Xiaojun Quan, and Jian Yin. 2023. Multi-hop reasoning question generation and its application. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):725–740.

Ruqing Zhang, Jiafeng Guo, Lu Chen, Yixing Fan, and Xueqi Cheng. 2021. A review on question generation from natural language text. *ACM Transactions on Information Systems*, 40(1):1–43.