Adaptive Question–Answer Generation With Difficulty Control Using Item Response Theory and Pretrained Transformer Models

Yuto Tomikawa, Ayaka Suzuki, and Masaki Uto

Abstract—**The automatic generation of reading comprehension questions, referred to as question generation (QG), is attracting attention in the field of education. To achieve efficient educational applications of QG methods, it is desirable to generate questions with difficulty levels that are appropriate for each learner's reading ability. Therefore, in recent years, several difficulty-controllable QG methods have been proposed. However, conventional methods generate only questions and cannot produce question–answer pairs. Furthermore, such methods ignore the relationship between question difficulty and learner ability, making it challenging to ascertain the appropriate difficulty levels for each learner. To address these issues, in this article, we propose a method for generating question–answer pairs based on difficulty, defined using a statistical model known as item response theory. The proposed difficulty-controllable generation is achieved by extending two pretrained transformer models: bidirectional encoder representations from transformers and text-to-text transfer transformer. In addition, because learners' abilities are generally not knowable in advance, we propose an adaptive QG framework that efficiently estimates the learners' abilities while generating and presenting questions with difficulty levels suitable for their abilities. Through experiments involving real data, we confirmed that the proposed method can generate question–answer pairs with difficulty levels that align with the learners' abilities while efficiently estimating their abilities.**

Index Terms—**Adaptive learning, adaptive testing, automated question generation (QG) for reading comprehension, deep neural networks, item response theory (IRT), natural language processing.**

I. Introduction

AUTOMATIC question generation (QG) for reading comprehension is the task of creating questions related to given reading passages without human intervention, where reading passages are arbitrary natural language texts comprising a sequence of words, a sentence, a paragraph, or an entire document. The field of natural language processing has produced numerous methods for QG [1], which are being utilized in various educational applications, including intelligent tutoring systems, writing-assistance tools, and knowledge-evaluation platforms [1], [2], [3], [4], [5], [6], [7].

Early QG methods were based on rule-based or template-based approaches, which generated questions by converting declarative texts into interrogative questions, using handcrafted rules or templates [1], [8], [9], [10], [11]. However, preparing well-designed rules and templates for specific applications is time consuming and labor-intensive [1], [12]. To address this limitation, end-to-end QG methods based on deep neural networks have been used [1], [13], [14], [15], [16], [17], [18], [19], [20]. Initial neural QG methods were designed as sequence-to-sequence (seq2seq) models based on recurrent neural networks (RNNs) and attention mechanisms [14], [21], whereas more recent methods have employed pretrained transformer models [2], [5], [13], [15], [22], [23], such as bidirectional encoder representations from transformers (BERT) [24], generative pretrained transformer 2 (GPT-2) [25], bidirectional and autoregressive transformers (BART) [26], and text-to-text transfer transformer (T5) [27]. These approaches have successfully generated fluent questions that are relevant to the given reading passages.

A notable application of QG in education is reading tutors [1], [3], [4], [5], which provide reading-comprehension questions for diverse reading materials. Offering questions helps direct learners' attention to the content and helps them identify misconceptions, thereby improving their reading-comprehension skills [3]. To enhance the efficiency of such learning, offering questions with difficulty levels tailored to each learner's reading ability is beneficial. For this reason, several difficulty-controllable QG methods have recently been proposed.

Difficulty-controllable QG for reading comprehension is a relatively new area of research, and thus, the literature is scant [22], [28]. One method for realizing difficulty-controllable QG is to use an RNN-based seq2seq model in which the hidden states from the encoder are adjusted to accept the difficulty levels categorized as either easy or hard [22]. Another approach is a multihop method [28], which defines the question difficulty according to the number of inference steps required to answer the question and generates questions by controlling the number of these inference steps. However, both methods face the following limitations that make it difficult to generate questions suitable for the learner's abilities.

Received 5 January 2024; revised 13 May 2024, 29 July 2024, 1 October 2024, and 29 October 2024; accepted 1 November 2024. Date of publication 4 November 2024; date of current version 15 November 2024. This work was supported by JSPS KAKENHI under Grant 24H00739, Grant 21H00898, and Grant 19H05663. *(Corresponding author: Masaki Uto.)*

The authors are with the Graduate School of Informatics and Engineering, University of Electro-Communications, Chofu 182-8585, Japan (e-mail: tomikawa@ai.lab.uec.ac.jp; suzuki_ayaka@ai.lab.uec.ac.jp; uto@ai.lab.uec.ac.jp).

Digital Object Identifier 10.1109/TLT.2024.3491801

© 2024 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

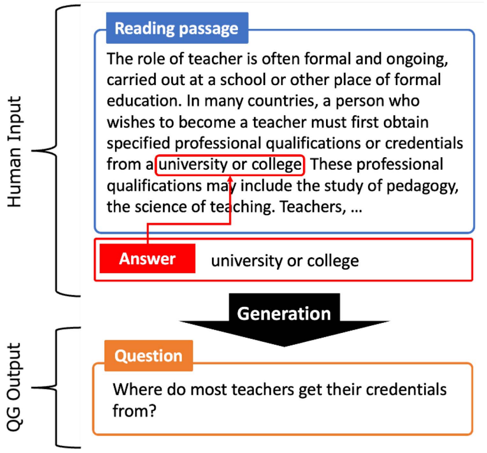
Human Input
Reading passage
The role of teacher is often formal and ongoing, carried out at a school or other place of formal education. In many countries, a person who wishes to become a teacher must first obtain specified professional qualifications or credentials from a university or college. These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

Fig. 1. Conventional QG task: answer-aware QG.

1) The relationship between the difficulty of the questions and the learner's ability is ignored, making it difficult to determine the appropriate difficulty for each learner.
2) The methods are answer-aware, which means a reading passage and an answer text must be input to generate questions, as shown in Fig. 1. For this reason, they cannot generate question–answer pairs. Generating answers along with questions is essential to realize automatic assessment of learners' responses to generated questions. Furthermore, controlling the difficulty of the generated answers is also crucial because both questions and answers generally affect the overall difficulty.

To address these issues, we introduce a novel method for generating question–answer pairs while considering the difficulty associated with the learners' ability. A unique feature of our method is the use of item response theory (IRT) [29] to quantify the question difficulty. The IRT is based on statistical models that define the relationship between question difficulty and learner ability, thereby facilitating the selection of an appropriate difficulty level for each learner. For this reason, our method is designed to generate question–answer pairs while controlling IRT-based difficulty.

For our QG method, we first propose a method to create a training dataset composed of quadruplets (reading passage, question text, answer text, and IRT-based difficulty) by extending the SQuAD [30] dataset, the most utilized benchmark dataset for the reading comprehension QG task. Subsequently, we propose a difficulty-controllable generation method for question–answer pairs that can be trained using this extended dataset. Our generation method consists of two pretrained transformer-based models that are modified to incorporate IRT-based difficulty values as inputs: *a difficulty-controllable answer-extraction model using BERT*, and *a difficulty-controllable answer-aware QG model using T5*.

Moreover, to generate questions with appropriate difficulty levels tailored to each learner, we need to know the learner's ability in advance, a requirement that is not satisfied in practice. To overcome this limitation, we propose leveraging the framework of computerized adaptive testing (CAT) [31], a well-known test administration method. This method repeats a cycle of sequentially presenting questions of a difficulty level suited to a learner's ability and estimating their ability from their responses. We expect this method to enable efficient ability estimation while tuning the difficulty levels of provided questions.

To our knowledge, this is the first QG method that enables difficulty control aimed at generating question–answer pairs quantified with IRT-based difficulty.

II. Research Questions and Contributions

The research questions of this study are summarized as follows.

1) *RQ1:* Is it possible to generate question–answer pairs by specifying arbitrary IRT-based difficulty values?
2) *RQ2:* Can the adaptive QG strategy, based on the CAT framework, enhance the selection of more suitable difficulty levels for each learner through its efficient ability estimation?

The proposed method has the potential to make the following contributions, which would be beneficial in various educational applications.

1) It enhances learning efficiency in intelligent tutoring systems by adaptively providing questions and answers at difficulty levels appropriate for each learner.
2) The CAT-inspired adaptive QG framework enables more efficient measurement of learners' abilities compared with question presentations that are not difficulty-aware.
3) It can assist in the expansion of item banks, a task often crucial for managing standardized tests, although this aspect is not the primary focus of this study.

III. Research Objective and Task Definition

As discussed in the aforementioned sections, our first research objective is to develop a method that generates reading comprehension questions along with their corresponding answers, based on the given reading passages and specified difficulty levels, as illustrated in Fig. 2. Our second objective is to develop an adaptive QG framework that generates questions with difficulty levels appropriate for learners while efficiently estimating their abilities each time a question is presented.

The detailed task definition for the first objective 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 ith 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 jth word in the question text, a_k is the kth 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, $\boldsymbol{a} \subset \boldsymbol{r}$, which means that our answer-generation task can be seen as a text span extraction from a reading passage. This implies that our answer-generation task is a text span extraction from the reading passage, as in typical answer-aware QG tasks [30]. Using this notation, our task is to

TABLE I
OVERVIEW OF RELATED WORKS

Method	Model	Definition of difficulty	Limitation
Traditional QG	RNN [14], [32]	(Not defined because difficulty is not considered)	Lack of capability to control difficulty
	BERT [13]		
	T5 [23]		
Difficulty-Controlable QG	RNN [22]	Correct answer ratio of two QA systems	• Lack of consideration of the relation between question difficulty and learner ability • Lack of capacity to generate difficulty-controllable question–answer pairs
	GPT-2 [28]	Number of inferential steps required for answering	

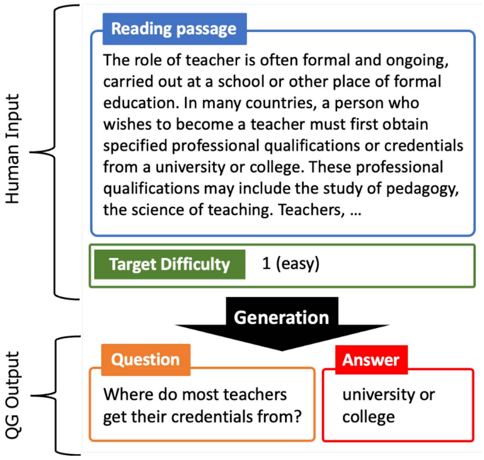
Human Input
Reading passage
The role of teacher is often formal and ongoing, carried out at a school or other place of formal education. In many countries, a person who wishes to become a teacher must first obtain specified professional qualifications or credentials from a university or college. These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

Fig. 2. Our QG task: generating a question–answer pair while controlling its difficulty.

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 the IRT, as explained in the introduction.

Furthermore, the task for the second objective is a cyclic process comprising the following subtasks.

1) Estimating the learner's ability based on their responses to previously administered questions.
2) Selecting an appropriate difficulty level for the estimated ability and generating questions using a difficulty-controllable QG method by specifying that difficulty.

For this cyclic task, we propose an efficient strategy grounded in the IRT and CAT.

IV. Technical Details and Limitations of Related Works

In this study, we employ deep neural networks as the foundational technology for enabling automated QG for reading comprehension tasks (e.g., [13], [14], and [32]). In this section, we provide an overview of conventional neural methods for generating reading comprehension questions and conventional difficulty-controllable QG methods. Table I summarizes the characteristics and limitations of related works introduced as follows.

A. Neural QG for Reading Comprehension

A representative neural QG method for reading comprehension is an RNN-based seq2seq model [14]. In this method, a reading passage and an answer are fed into an RNN encoder, and the output feature vector is given to an RNN decoder to generate a question text. In addition, Zhou et al. [32] proposed using an RNN-based QG model that can consider both the words' sequence and their part-of-speech (POS) tags.

In recent years, pretrained transformer-based models, which have outperformed RNN-based seq2seq models on many natural language processing tasks (e.g., [25], [27], [33], [34], [35], [36], [37], [38], [39], and [40]), have been used for automated QG tasks (e.g., [1], [13], [14], [41], [42], and [43]). Some examples include a QG method proposed by Chan and Fan [13] that uses BERT and a method proposed by Lee and Lee [23] that uses T5.

B. Difficulty-Controllable Neural QG for Reading Comprehension

When utilizing QG methods as a learning aid to foster reading comprehension skills, it is critical to be able to generate questions with arbitrary difficulty levels [44]. Accordingly, several recent studies have proposed difficulty-controllable QG methods [22], [28].

For example, Gao et al. [22] proposed an RNN-based seq2seq model that generates reading comprehension questions for difficulty levels categorized as either "easy" or "hard." They also proposed to construct training data for the difficulty-controllable QG task based on a standard benchmark dataset, such as SQuAD. In the method, each question in the dataset is answered by two question–answering (QA) systems. Then, the binary difficulty label is estimated for each question. Specifically, the label "easy" is assigned to a question when both QA systems correctly answer it, while the label "hard" is assigned when both incorrectly answer it. Questions for which only one QA system produces a correct answer are excluded from the dataset. An RNN-based seq2seq QG model is then trained using this training dataset, which comprises a reading passage, a question text, a corresponding answer, and a difficulty label. The model is designed to take the difficulty label as part of the input, enabling the generation of questions tailored to specific difficulty levels.

Another difficulty-controllable QG method, proposed by Cheng et al. [28], uses the number of inferential steps required for answering as a measure of difficulty. A knowledge graph rooted at the answer is first constructed from a reading passage, then questions are generated such that the number of inferential steps required for answering, as determined by the knowledge graph, is set to one. The initial question is iteratively refined by increasing the number of inferential steps required for answering, using the knowledge graph as a guide.

However, the following issues remain unaddressed in these difficulty-controllable QG methods.

1) The relation between the learner's ability level and the difficulty of the questions is not considered, making it impossible to select a difficulty level that matches the learner's ability.
2) While they can generate questions from a given reading passage and answer, they are unable to generate both a question and an answer solely from the reading passage. Controlling the difficulty in answer generation has not been investigated, although it is just as important as controlling the difficulty of questions since difficulty is a property that generally depends on both questions and answers.

To resolve these issues, we use the IRT to quantify difficulty levels and to automatically generate both questions and answers from the reading passage, based on specified difficulty values.

V. Item Response Theory (IRT)

The IRT [29] is a statistical framework that uses probabilistic models, called IRT models, to estimate two latent factors:[1] examinee's ability and item characteristics such as item difficulty and discriminative power,[2] where the *examinee* and the *item* correspond to the *learner* and the *question*, respectively, in our study. These latent factors are estimated from the response data, which generally consist of the examinee's binary correct/incorrect responses to the items. The IRT has been widely used in various educational and psychological tests because of its numerous benefits, such as offering accurate estimates of examinee's ability and item characteristics, unifying measurement scales among different tests, and facilitating CAT applications [45], [46], [47], [48].

This study uses the Rasch model [49], the most traditional and well-known IRT model, to quantify question difficulty.

A. Rasch Model

The Rasch model defines the probability that the mth examinee correctly answers the nth item as

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

where b_n denotes the difficulty of the nth item, and θ_m denotes the latent ability of the mth examinee.

For a detailed explanation of this model, Fig. 3 illustrates item response curves (IRCs) for the Rasch model, which are plots of the probability p. In this figure, the horizontal axis represents the ability θ; the vertical axis represents the probability p; and the three solid curves represent the IRCs for three items with different difficulty levels, $b = -1.0$, 0.0, and 1.0. These IRCs demonstrate that examinees with higher θ have an increased probability of responding correctly to each item, indicating that it reflects a reasonable relationship between the examinee's ability and the probability of a correct response. In addition, it is evident that the IRC shifts to the right as the b value increases, indicating that higher ability θ is necessary to correctly answer items with higher b values. This means that the parameter b reflects the difficulty of each item. Moreover, this model enables the calculation of the correct answer probability from the ability and difficulty values, which helps to determine a question with an appropriate difficulty level for an examinee with a given ability. For example, when we aim to select a question that an examinee with ability θ will answer correctly with 0.5 probability, offering a question with $b \fallingdotseq \theta$ is appropriate because the probability p becomes 0.5 when $b = \theta$, as shown in the figure.

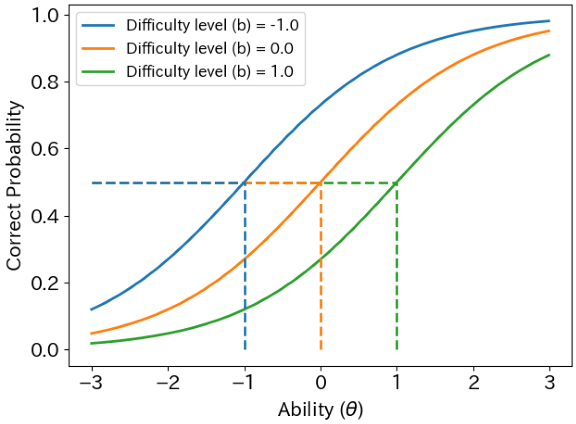

Fig. 3. IRCs for a Rasch model with different item difficulty values.

The parameters of the IRT model are typically estimated in two stages: *item calibration* and *ability estimation*. Item calibration involves estimating the item parameters from the response data. Specifically, marginal maximum likelihood estimation is commonly employed for item calibration [50]. Once the item parameters have been calibrated, the ability estimation phase estimates the examinee's ability θ, commonly based on the expected a posteriori estimation [51], [52].

B. Computerized Adaptive Testing (CAT)

As explained earlier, the IRT is often used as a basis for CAT, which adaptively administers items appropriate for each examinee while sequentially estimating the examinee's ability from their response history. Specifically, CAT based on the Rasch model begins by initializing the ability of a target examinee, then it generally selects an item with a difficulty level that maximizes Fisher information [31]. The ability estimate is then updated with a correct or incorrect response to the offered item. By repeating these procedures, CAT offers optimal items while efficiently estimating an examinee's ability.

[1] In this context, "latent" indicates that the corresponding factors are not directly observable but lie behind the observed data.

[2] "Discriminative power" means the degree to which an item can distinguish between examinees with different levels of the underlying ability.

TABLE II
CONFIGURATION OF BERT AND T5 USED AS THE BASIS MODELS IN THIS STUDY[3]

	BERT	T5
Architecture	Transformer encoder	Transformer
No. of parameters	110 million	220 million
Pretraining tasks	Masked language model Next-sentence prediction	Denoising objective
Corpus size	3.3 billion words	750 GB text corpus

The reason for using the Fisher information for item selection is that the variance of the maximum likelihood estimate of ability converges to the inverse of Fisher information, meaning that an item with the maximum Fisher information is the most effective for accurately estimating the examinee's ability. In the Rasch model, Fisher information is maximized when the item difficulty parameter b equals the ability value θ. Repeating the ability estimation and the Fisher-information-based item selection enables us to achieve a highly accurate ability estimate with fewer items administered [31].

VI. PROPOSED METHOD

Based on conventional QG methods and IRT approach, our difficulty-controllable QG method is carried out by performing the following two tasks in sequence.

1) *Difficulty-controllable answer extraction*, which extracts an answer text from a given reading passage while considering a target IRT-based difficulty value.
2) *Difficulty-controllable answer-aware QG*, which generates a question given a reading passage, an answer text, and a target IRT-based difficulty value.

Furthermore, within the framework of CAT, we propose a methodology for sequentially estimating a learner's ability and adaptively generating questions with appropriate difficulty levels for their ability.

Because our difficulty-controllable QG method requires a dataset with IRT-based difficulty values for model training, we first propose a method for constructing it in the next section.

A. Creating a Dataset With IRT-Based Question Difficulty

While several popular datasets have been developed for general reading comprehension QG tasks [1], the most popular is SQuAD [30], which consists of over 100 000 question–answer pairs from Wikipedia articles posed by crowdworkers. 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 based on exact matching with the corresponding true answers. 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 QA systems, in the same way that several previous difficulty-controllable QG studies have done [12], [22].
2) *Difficulty estimation using IRT:* Using the collected response data, we estimate[3] the question difficulty based on the Rasch model following the item calibration procedure introduced in Section V.
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.

Using this dataset, the proposed method trains following two models:

1) a *difficulty-controllable answer-extraction model*;
2) a *difficulty–controllable answer-aware QG model*.

B. Difficulty-Controllable Answer-Extraction Model

We use BERT [24] for difficulty-controllable answer extraction. BERT is a pretrained transformer model, the details of which are summarized in Table II. BERT can be adapted for various downstream tasks by fine-tuning it with task-specific supervised datasets and incorporating task-specific output layers. We employed BERT for the answer-extraction task because of its extensive prior usage in various text-extraction applications [53].

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, letting $\boldsymbol{v}_i$ be the BERT output vector for the ith word in a passage $\boldsymbol{r}$, we add two dense layers, denoted as $f^{(s)}(\boldsymbol{v}_i)$ and $f^{(e)}(\boldsymbol{v}_i)$, for each $\boldsymbol{v}_i$, where each dense layer transforms $\boldsymbol{v}_i$ into a scalar value . The dense-layer outputs are then transformed through softmax activations as $P^{(s)} = \text{softmax}(f^{(s)}(\boldsymbol{v}_1), \ldots, f^{(s)}(\boldsymbol{v}_I))$ and $P^{(e)} = \text{softmax}(f^{(e)}(\boldsymbol{v}_1), \ldots, f^{(e)}(\boldsymbol{v}_I))$, whose ith elements $P_i^{(s)}$ and $P_i^{(e)}$ represent the probability values for whether the ith word is at the start and end positions, respectively. Thus, 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

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

where [SEP] is the special token used to separate the difficulty value and the reading passage. This configuration allows the model to consider the difficulty value when extracting the answer text from the reading passage. Fig. 4 shows an outline of the answer-extraction model.

[3] No changes were made to original BERT and T5 models in terms of architecture and training parameters.

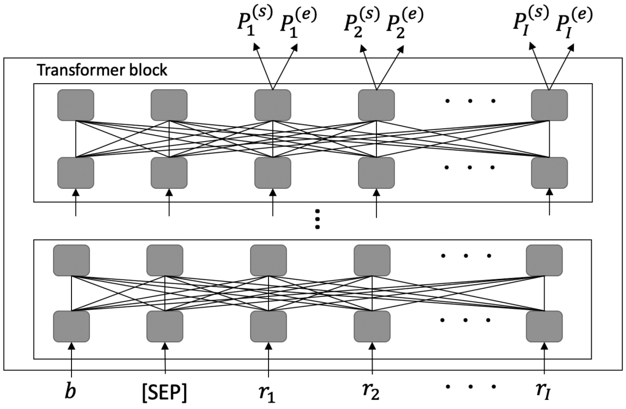
$P_1^{(s)}$ $P_1^{(e)}$ $P_2^{(s)}$ $P_2^{(e)}$ $P_I^{(s)}$ $P_I^{(e)}$
Transformer block
b
[SEP]
r_1
r_2
r_I

Fig. 4. Difficulty-controllable answer-extraction model using BERT.

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 VI-A. This fine-tuning is performed by minimizing cross-entropy loss between the true positions of the start and end of an answer text and the predicted probabilities for these positions.

C. Difficulty-Controllable Answer-Aware QG Model

We use T5 for difficulty-controllable answer-aware QG, where T5 is a pretrained transformer model, the details of which are summarized in Table II. We use T5 because it has been widely used before in various text generation tasks [54], [55], [56], [57] including QG tasks [2], [58], [59] and has achieved higher accuracy in QG compared to models, such as BART and GPT-2 [58].

Conventional answer-aware QG models [60] based on pretrained language models are implemented by designing the model's input as

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

where [A] is a special token representing an answer's start and end positions within a reading passage. The model's target, which is a question text, is designed as

$$[\mathrm{G}], q_1, \ldots, q_J, [\mathrm{E}] \tag{4}$$

where [G] and [E] are also special tokens representing the beginning of a question text and the end of a question text, respectively.

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

$$b, [\mathrm{Q}], r_1, \ldots, [\mathrm{A}], a_1, \ldots, a_K, [\mathrm{A}], \ldots, r_I \tag{5}$$

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 VI-A. Specifically, we prepare the input data following (5) format and output data following (4) format, and train T5 by maximizing the log-likelihood for target data.

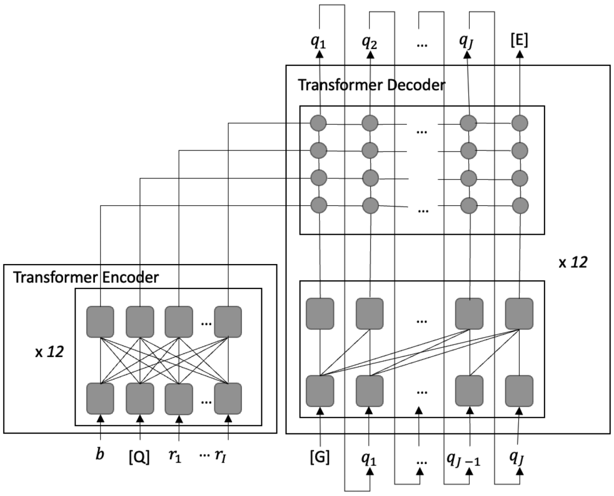

Fig. 5. Difficulty-controllable answer-aware QG model using T5.

D. Estimating the Learner's Ability and Adaptively Generating Questions

A previous study on adaptive learning [44] reported that posing questions that learners can answer correctly with a 50% probability is pedagogically effective. As discussed in Section V-A, in the Rasch model, the probability of obtaining a correct answer is 0.5 when the ability and difficulty levels are equal. Therefore, if a learner's ability level is known, using the proposed method to generate a question whose difficulty level is equal to or near the learner's ability would be effective for enhancing reading comprehensive skills. However, in typical educational settings, a learner's ability is often unknown a priori.

To address this, we are proposing an efficient method for estimating a learner's ability. We utilize the CAT framework, as mentioned in Section V-B, while generating and administering questions at appropriate levels of difficulty. Specifically, we propose the following procedure for ability estimation and QG.

1) Randomly generate and administer a few questions to the learner to obtain initial response data.
2) Utilize the obtained response data along with the Rasch model to estimate and update the learner's ability level.
3) Generate and administer a question at the estimated ability level using the proposed QG method, thereby obtaining new response data.
4) Repeat steps (2) and (3).

Using this procedure, we can effectively estimate a learner's ability while administering questions at appropriate levels of difficulty. Remember that when the ability and difficulty levels are equal, the Rasch model produces a 0.5 probability that a correct answer will be obtained, and Fischer information is maximized.

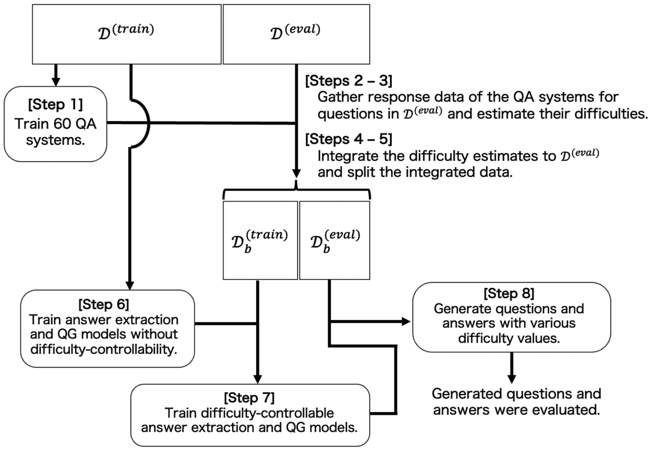
$\mathcal{D}^{(train)}$
$\mathcal{D}^{(eval)}$
[Step 1] Train 60 QA systems.
[Steps 2 – 3] Gather response data of the QA systems for questions in $\mathcal{D}^{(eval)}$ and estimate their difficulties.
[Steps 4 – 5] Integrate the difficulty estimates to $\mathcal{D}^{(eval)}$ and split the integrated data.
$\mathcal{D}_b^{(train)}$
$\mathcal{D}_b^{(eval)}$
[Step 6] Train answer extraction and QG models without difficulty-controllability.
[Step 8] Generate questions and answers with various difficulty values.
Generated questions and answers were evaluated.
[Step 7] Train difficulty-controllable answer extraction and QG models.

Fig. 6. Flow of the experimental procedures.

VII. Experiments

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

A. Experimental Procedures

We conducted the following experiment using the original SQuAD data, which was originally split into $\mathcal{D}^{(\text{train})}$ and $\mathcal{D}^{(\text{eval})}$, where the sample size of $\mathcal{D}^{(\text{train})}$ was 87 599 and that of $\mathcal{D}^{(\text{eval})}$ was 10 570. Fig. 6 illustrates the flow of the following experimental procedures.

1) Using the original SQuAD training data $\mathcal{D}^{(\text{train})}$, we constructed 60 different QA systems.[4]
   Specifically, 12 QA systems (BERT-base, BERT-large, RoBERTa-base, RoBERTa-large, DeBERTa-base, DeBERTa-large, DeBERTa-v3-base, DeBERTa-v3-large, ALBERT-base-v1, ALBERT-base-v2, ALBERT-large-v2, and DistilBERT-base), which are available on huggingface,[5] were trained with 600, 1200, 1800, 2400, and 3000 data points, which were randomly sampled from $\mathcal{D}^{(\text{train})}$, respectively. These QA systems, trained on data subsets of varying sizes, are considered to be proxies for a diverse learner population with varying abilities.
2) We subjected each question in the evaluation dataset, $\mathcal{D}^{(\text{eval})}$, to the 60 QA systems and gathered correct and incorrect binary response data based on the complete match between the system's answer and the true answer.
3) Utilizing the collected binary response data, we employed the Rasch model, expressed in (1), to estimate the difficulty level for each question in $\mathcal{D}^{(\text{eval})}$ and the ability for each of the QA systems.

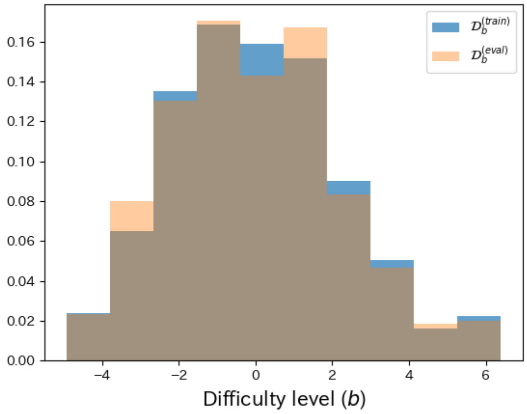

Fig. 7. Number of questions corresponding to the difficulty values in $\mathcal{D}_b^{(\text{train})}$ and $\mathcal{D}_b^{(\text{eval})}$.

4) We then integrated the estimated question difficulty with $\mathcal{D}^{(\text{eval})}$ to produce a new dataset, $\mathcal{D}_b$. The real-valued difficulty estimates were rounded to the second decimal place to facilitate processing by the language models BERT and T5.
5) The created dataset, $\mathcal{D}_b$, was further divided into two partitions that were 90% and 10%, denoted as $\mathcal{D}_b^{(\text{train})}$ and $\mathcal{D}_b^{(\text{eval})}$, respectively.
6) We first fine-tuned both the answer-extraction model and the QG model using the original SQuAD training dataset, $\mathcal{D}^{(\text{train})}$, without considering the difficulty levels. Although this step is not mandatory, we expect that pretraining these models on a large dataset without difficulty considerations can enhance their performance in answer extraction and QG.
7) Fine-tuning was then performed on $\mathcal{D}_b^{(\text{train})}$ to develop answer-extraction and QG models that consider question difficulty. This fine-tuning employed the model parameters estimated in step (6) as initial values.
8) To assess the proficiency of the developed models in controlling the difficulty, we generated question–answer pairs with various difficulties and evaluated them. Specifically, we first input each reading passage in $\mathcal{D}_b^{(\text{eval})}$ and each of the 61 difficulty values, from -3.0 to 3.0 in increments of 0.1, into the proposed answer-extraction model and generated 61 answers for each reading passage. Then, given each triplet consisting of a reading passage, difficulty value, and generated answer, we generated questions using the proposed QG model. The generated sets of questions and answers were subjected to both machine-based and human evaluations.

We now summarize the basic statistics of the datasets $\mathcal{D}_b^{(\text{train})}$ and $\mathcal{D}_b^{(\text{eval})}$, which we developed in the aforementioned procedure (5) to train and evaluate our difficulty-controllable QG method. First, the number of reading passages in $\mathcal{D}_b^{(\text{train})}$ and $\mathcal{D}_b^{(\text{eval})}$ was 1860 and 207, respectively. Next, the average number of questions per reading passage in $\mathcal{D}_b^{(\text{train})}$ and $\mathcal{D}_b^{(\text{eval})}$ was 5.21 and 4.28. Furthermore, Fig. 7 is a histogram of the difficulty

[4] The rationale behind our use of 60 QA systems is based on a previous report [61], which stated that the minimum requirement of respondents for estimating difficulty based on the Rasch model is 30. However, a larger number of respondents generally contributes to improving the stability of difficulty estimation. Thus, considering the tradeoff between computational cost and estimation stability, this experiment used 60 QA systems.

[5] [Online]. Available: https://huggingface.co/

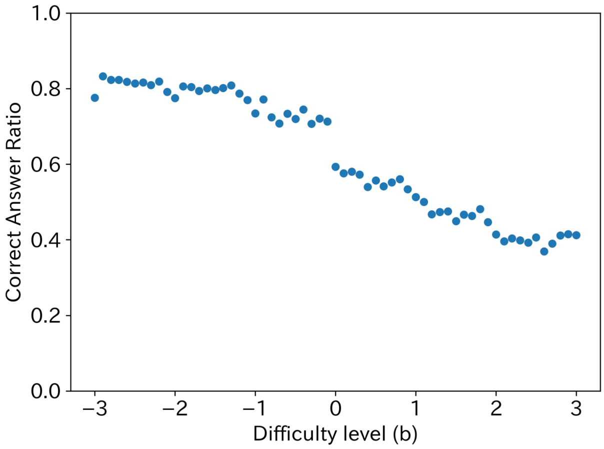

Fig. 8. Average correct answer ratio for each difficulty level.

values in each dataset. From these results, we can confirm that the difficulty distributions are similar between the two datasets, indicating that $\mathcal{D}_b$ was randomly divided into $\mathcal{D}_b^{(\text{train})}$ and $\mathcal{D}_b^{(\text{eval})}$ without bias.

B. Automatic Evaluation of Difficulty-Controllable QG Quality

We first evaluated the generated questions and answers in the aforementioned experimental procedure (8) from the following three perspectives:

1) the average correct answer ratio of QA systems for each difficulty level;
2) the average word count of the answers for each difficulty level;
3) the frequency of leading interrogative words in the generated questions for each difficulty level.

The average correct answer ratio of the 60 QA systems, which were those trained in procedure (1) of Section VII-A, for each difficulty level is illustrated in Fig. 8. The x-axis represents the specified difficulty levels, and the y-axis represents the average correct answer ratio. The figure indicates that as the specified level of difficulty increases, the average correct answer ratio of the QA systems for the generated questions tends to decrease. This suggests that the proposed method for generating questions successfully reflects the specified levels of difficulty.

Further detailed analysis is given by Fig. 9, which shows the correct answer ratio of the 60 QA systems for the generated question–answer pairs given $b = 1.0$. The x-axis represents the ability θ, the y-axis represents the correct answer ratio, each plot indicates an individual QA system, and the orange line represents the logistic regression curve fitted to the 60 data points by the least-squares method. As described before, the Rasch model gives a probability of 0.5 that a correct answer will be given when $\theta = b$. To emphasize this point, the green dashed line represents $\theta = 1.0$ and a correct answer ratio of 0.5. The figure shows that the QA systems with an ability level around $\theta = 1.0$ demonstrate a correct answer ratio of approximately 0.5, suggesting that the questions are generated as expected. A similar trend was observed for other difficulty levels, as exemplified in Figs. 10 and 11, which show the results for $b = 2.0$ and $b = 0.0$, respectively. Furthermore, Fig. 12 shows the three logistic regression curves shown in Figs. 9–11. The x-axis represents the ability θ, and the y-axis represents the correct answer ratio. The blue, orange, and green lines represent the curves for the difficulty levels $(b) = 0.0$, 1.0, and 2.0, respectively. This figure demonstrates

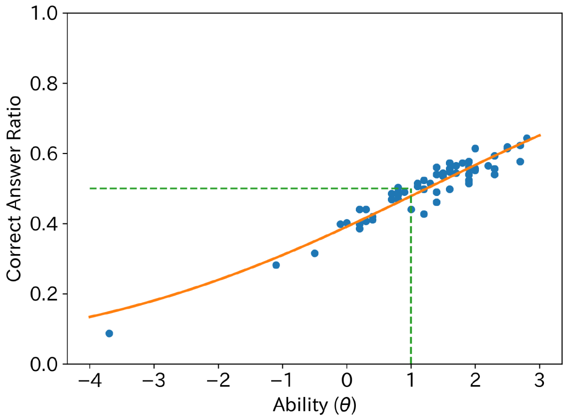

Fig. 9. Average correct answer ratio of the QA systems for generated questions given difficulty $b = 1.0$. Each dot plot represents an individual QA system.

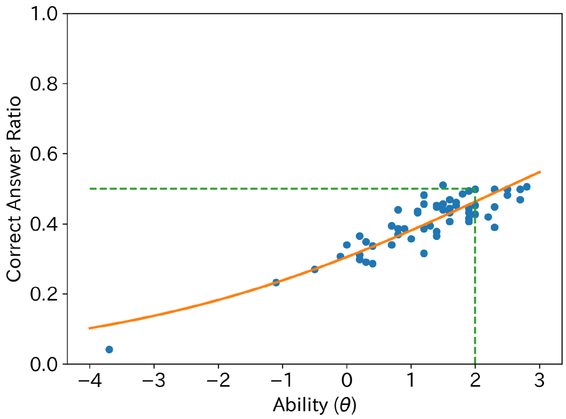

Fig. 10. Average correct answer ratio of the QA systems for generated questions given difficulty $b = 2.0$. Each dot plot represents an individual QA system.

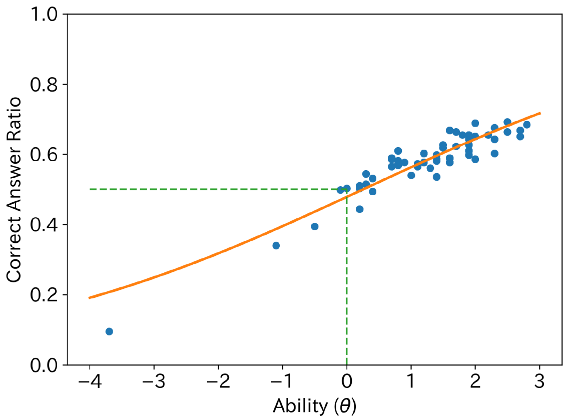

Fig. 11. Average correct answer ratio of the QA systems for generated questions given difficulty $b = 0.0$. Each dot plot represents an individual QA system.

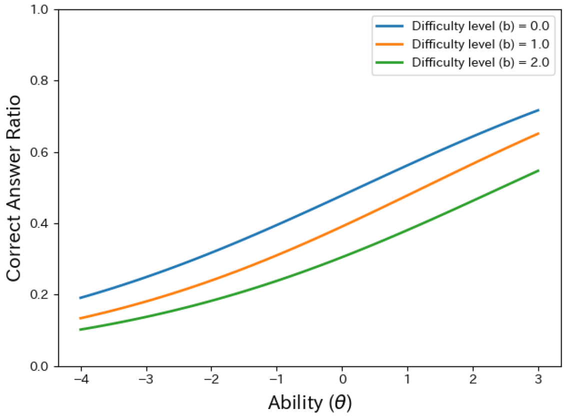

Fig. 12. Logistic regression curves for the three different difficulties shown in Figs. 9–11.

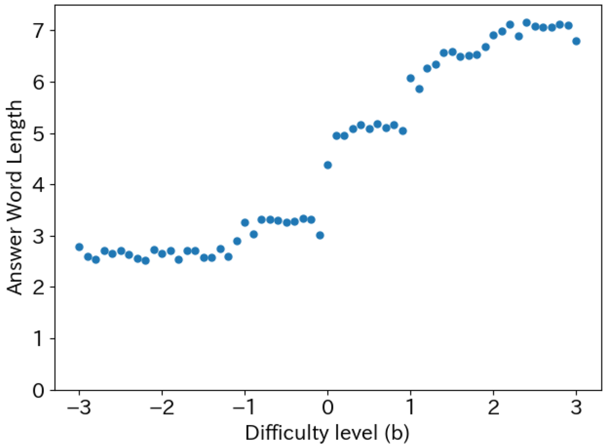

Fig. 13. Average word count of answers by difficulty level.

TABLE III
EXAMPLES OF GENERATED QUESTIONS AND ANSWERS

Reading Passage	Deacons are called by God, affirmed by the church, and ordained by a bishop to servant leadership within the church. They are ordained to ministries of word, service, compassion, and justice. They may be appointed to ministry within the local church or to an extension ministry that supports the mission of the church. . . . Deacons serve supports the mission of the church. . . . Deacons serve a term of 2–3 years as provisional deacons prior to their ordination.
Difficulty	-3.0
Question	How long do provisional deacons serve as?
Answer	2–3 years
Difficulty	3.0
Question	What role are deacons assigned by the church?
Answer	Servant leadership within the church

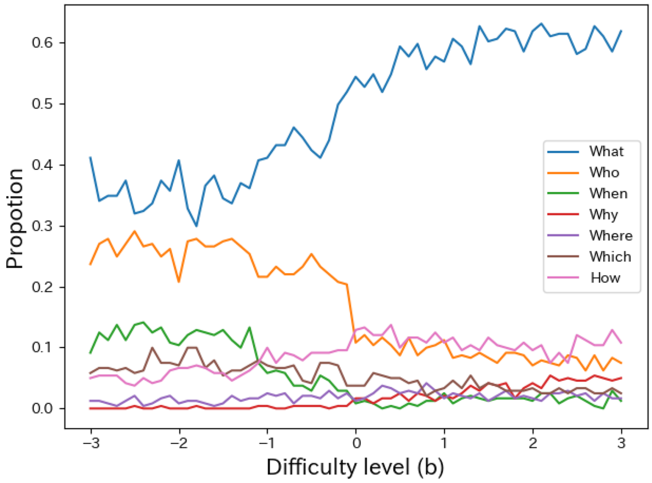

Fig. 14. Proportion of leading interrogative words.

that specifying higher difficulty values makes the generated questions more difficult across a wide range of ability levels.

Next, the average word count of the answers for each difficulty level is displayed in Fig. 13. The x-axis represents the specified levels of difficulty, while the y-axis represents the average word count of the answers. The figure shows an increasing trend in the average word count for higher levels of difficulty. Since longer answers generally make the question difficult, the proposed method could extract answers while reflecting the difficulty levels. Table III shows examples of the generated questions and answers for different difficulty levels from the same reading passage. It shows that giving a lower difficulty value induces short answer text, whereas giving a higher difficulty value induces longer and more complex answers.

Finally, we investigated how the input difficulty affects the types of generated questions. To do so, we examined the proportion of leading interrogative words, “What,” “Who,” “When,” “Why,” “Where,” “Which,” and “How” in the generated questions. The results are shown in Fig. 14. The x-axis represents the specified levels of difficulty, and the y-axis represents the proportion of each interrogative word appearing as the leading word in the generated questions, categorized by their level of difficulty. The figure shows that questions beginning with “When,” “Which,” and “Who,” which are generally answerable by a single element, are more likely to be generated at lower difficulty levels. Conversely, questions starting with “Why” and “How,” which ask for reasons or procedures, are more frequently generated at higher levels of difficulty. Questions being answerable by simple elements are generally easier than those asking for reasons or procedures, suggesting that these results reflect a reasonable difficulty-control of the proposed method. Notably, the word “What” shows a higher frequency of occurrence in high-difficulty questions. “What” can serve multiple functions, not only asking for specific elements but also asking for reasons or procedures, similar to “How” and “Why.”

C. Human Evaluation of Difficulty-Controllable Quality

To validate the quality and difficulty of the question sets generated by the methodology described in Section VII-A, we conducted a human evaluation. Specifically, we randomly selected 100 question–answer pairs with variability in difficulty levels from the generated questions. We then divided them into five sets, each containing 20 question–answer pairs, and assigned two raters to each set. The raters were crowdworkers having Test of English for International Communication (TOEIC) scores of

TABLE IV
HUMAN EVALUATION

		Frequency	Ratio
Fluency	Appropriate	128	64.0%
	Acceptable	61	30.5%
	Inappropriate	11	5.5%
Content relevance	Appropriate	181	90.5%
	Inappropriate	19	9.5%
Answerability	Appropriate	130	65.0%
	Insufficient	31	15.5%
	Excessive	21	9.0%
	Inappropriate	18	10.5%
Practicality	Already feasible	128	64.0%
	Need minor correction	52	26.0%
	Not feasible	20	10.0%

900 or higher.[6] The raters were asked to assess the assigned question–answer pairs based on the following evaluation criteria. The raters were crowdworkers having a TOEIC score of 900 or higher .

1) *Fluency:* Evaluate the grammatical correctness and fluency of the questions. Ratings were done on a three-point scale: 3. appropriate, 2. acceptable, and 1. inappropriate.
2) *Content Relevance:* Assess whether the generated questions were relevant to the reading passage. Ratings were done on a two-point scale: 2. appropriate and 1. inappropriate.
3) *Answerability:* Evaluate whether the extracted answer was indeed the correct answer for the generated question. Ratings were done on a nominal scale with four categories: a. appropriate, b. insufficient, c. excessive, and d. inappropriate.
4) *Practicality:* Assess whether the question and/or answer could become feasible with slight modifications. Ratings were done on a three-point scale: 3. already feasible, 2. need minor correction, and 1. not feasible.
5) *Question Difficulty:* Evaluate the difficulty level of the generated question. Ratings were done on a five-point scale, ranging from 1 (easiest) to 5 (most difficult).

Results for fluency, content relevance, answerability, and practicality are presented in Table IV. From this table, more than 90% of the questions were rated as either fluent or acceptable, and approximately 90% appropriately reflected the content of the reading passage. Moreover, in over 60% of cases, question and answer pairs were generated that were answerable, and nearly 90% were partially appropriate when accepting excessive or insufficient results. These findings were confirmed by the practicality ratings, which also found that nearly 90% of the questions were evaluated positively.

Next, we calculated the Spearman rank-order correlation coefficient between the human evaluations of difficulty and the difficulty levels specified for QG, obtaining a low correlation value of 0.15. To assess the reason for this outcome, we determined the agreement ratio between the two raters for each evaluation criterion. The results showed that the agreement rates for fluency, content relevance, answerability, and practicality were 0.63, 0.85, 0.53, and 0.60, respectively, while that for difficulty was 0.22. From these results, we conclude that it is hard to ensure reliable subjective evaluations for question difficulty, while the evaluations for other criteria seem acceptable.

Thus, to achieve a more reliable evaluation of question difficulty, we conducted another experiment in which the generated questions were offered to human respondents, and their responses were analyzed. The experiment was carried out through the following procedure.

1) From the questions that the two human raters judged as answerable in the aforementioned experiment, we randomly selected 30 so that their specified difficulty varied.
2) We collected ten subjects, different from those in the previous experiment, and asked them to answer 20 of those 30 questions. The set of questions administered to each subject was different, and each question was administered an equal number of times. Given that the experiment required participants with diverse language abilities, this time we recruited crowdworkers possessing English skills equivalent to a TOEIC score of 600 or higher.
3) The obtained answers were graded by one of the authors to remove the effects of superficial fluctuation in human answers.
4) Based on the correct and incorrect response data, we calculated the correct rates for each question as an index representing its difficulty.

The correlation between correct rates and the specified difficulty levels was -0.67, where a test for noncorrelation revealed that the p-value was less than .01, confirming a statistically significant correlation at the 1% level. These results suggest that the proposed method can generate questions in alignment with the difficulty levels perceived by human respondents.

VIII. EVALUATION OF THE ACCURACY OF LEARNERS' ABILITY ESTIMATION

In this section, we evaluate the efficacy of the method of adaptive QG proposed in Section V-B. We simulated the adaptive QG process for three respondents with different abilities. For the respondents, we used three QA models with the minimum and maximum θ, specifically, -3.658 and 2.766, as well as that which was closest to the average, $\theta = 1.244$. Using the three QA models, we examined the following two QG processes and analyzed the trajectory of their estimated abilities.

1) *Adaptive Generation (Proposed):* Initially, ten questions with random difficulty are generated and administered. Then, 40 questions are administered, following the adaptive procedures detailed in Section VII-B, while updating the ability estimates.

[6] The crowdworkers were recruited via the crowdsourcing platform CrowdWorks (crowdworks.jp). The only requirement was that they have a TOEIC score of 900 or above. No additional information about the raters' attributes was collected. A TOEIC score of 900, which corresponds to an effective operational proficiency in the English language, can be compared with other international English scoring systems as follows: TOEFL $>$ 79, International English Language Testing System (IELTS) $>$ 6.5, Cambridge Exam $\geq$ First Certificate in English (FCE), and Common European Framework of Reference for Languages (CEFR) $\geq$ B2.

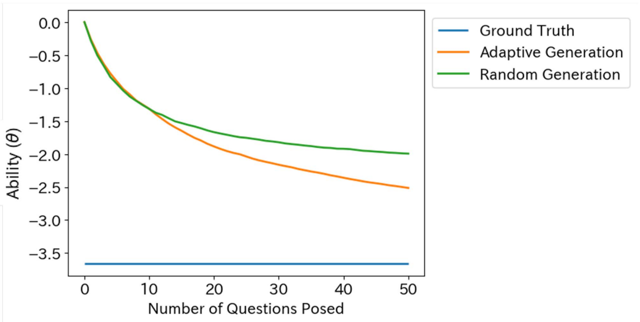

Fig. 15. Trajectory in ability estimates for a QA system with ability -3.658 (Minimum θ).

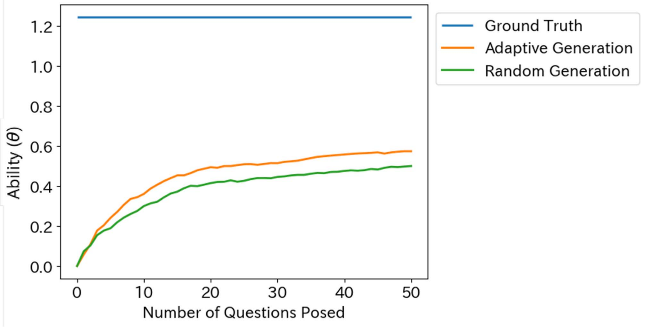

Fig. 16. Trajectory in ability estimates for a QA system with ability 1.244 (average θ).

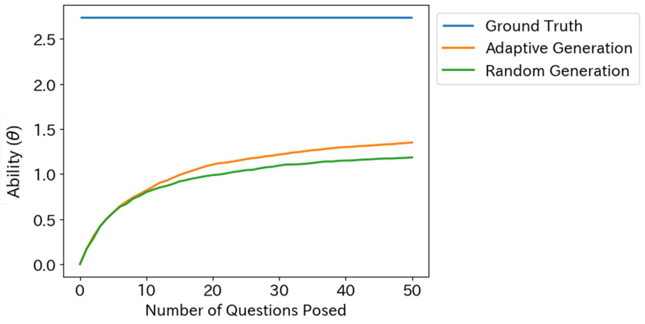

Fig. 17. Trajectory in ability estimates for a QA system with ability 2.766 (Maximum θ).

2) *Random Generation:* In total, 50 questions with random difficulty are generated and administered.

The trajectories in estimated θ values obtained using these methods are presented in Figs. 15–17. The horizontal axis represents the number of questions administered, and the vertical axis denotes the value of θ. The blue line signifies the true value, whereas the orange and green lines represent the trajectory in estimated abilities due to adaptive and random question posing, respectively. Note that these graphs are constructed by averaging the results from 500 iterations of the experiment. The results show that the proposed adaptive strategy estimated the respondents' true ability more efficiently than the random strategy.

For further analysis, we conducted a performance evaluation targeting all the QA systems. Specifically, we ran the the aforementioned experiment 500 times for each of the 60 QA models and calculated the mean absolute error (MAE) between the true ability values and the estimated abilities after posing 50 questions, as given by

$$\mathrm{MAE} = \frac{1}{60 \times 500} \sum_{d=1}^{60} \sum_{k=1}^{500} |\hat{\theta}_{dk} - \theta_d| \tag{6}$$

where θ_d is the true ability of the $d \in \{1, \ldots, 60\}$th model, and $\hat{\theta}_{dk}$ is the estimated ability of the dth model obtained in the $k \in \{1, \ldots, 500\}$th experiment. We found that the MAE was 0.707 for the proposed method and 0.744 for random generation. This confirms that even when considering all QA systems, our method enables ability estimation with higher accuracy than the random method.

More efficient ability estimation enables the generation of questions with a more appropriate difficulty level for each learner, suggesting that the proposed adaptive strategy is an important component for the difficulty-controllable QG.

IX. Conclusion

In this study, we proposed a method for automatically generating question-and-answer pairs with specified difficulties based on the IRT, as well as an adaptive QG technique that leverages this method. Through machine-based evaluation experiments using the SQuAD dataset, we demonstrated that the proposed method is capable of generating questions with appropriately adjusted difficulty levels. In addition, human-based evaluation experiments supported the result that the generated questions were fluent and contextually relevant. These experiments also confirmed that questions can be generated with difficulty levels aligned with human abilities. Moreover, we showed that even in general scenarios where the ability parameter θ of learners is unknown, adaptive generation enables more efficient estimation of θ compared to random question posing. This results in the effective generation of questions tailored to individual abilities.

Finally, we summarize the conclusions based on our two research questions. Our first research question was, "Is it possible to generate question–answer pairs by specifying arbitrary IRT-based difficulty values?" We conclude that our method makes this possible because our experiments demonstrated that the specified IRT-based difficulty values correlate well with the correct answer rates of both QA systems and human respondents. Our second research question was, "Can the adaptive QG strategy, based on the CAT framework, enhance the selection of more suitable difficulty levels for each learner through efficient ability estimation?" We also answer this question affirmatively, given that our experiments in Section VIII demonstrated that the proposed method increases ability estimation accuracy more quickly than the random method as the number of questions increases, indicating that the proposed method enables more efficient ability estimation. The accurate ability estimation allows for the selection of difficulty levels that more closely align with each learner's true ability, leading to more appropriate difficulty selection.

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 an educational setting, it is not only important to adjust the difficulty of the questions according to the learner's ability but also to select reading passages with appropriate difficulty levels. Therefore, future investigations will also consider methods for selecting reading passages based on learner ability and techniques for QG that take into account the difficulty of these passages.

References

[1] R. Zhang, J. Guo, L. Chen, Y. Fan, and X. Cheng, "A review on question generation from natural language text," *ACM Trans. Inf. Syst.*, vol. 40, no. 1, pp. 1–43, Sep. 2021.

[2] B. Ghanem, L. L. Coleman, J. R. Dexter, S. von der Ohe, and A. Fyshe, "Question generation for reading comprehension assessment by modeling how and what to ask," in *Proc. Findings Assoc. Comput. Linguistics*, 2022, pp. 2131–2146.

[3] G. Kurdi, J. Leo, B. Parsia, U. Sattler, and S. Al-Emari, "A systematic review of automatic question generation for educational purposes," *Int. J. Artif. Intell. Educ.*, vol. 30, pp. 121–204, Nov. 2019.

[4] N.-T. Le, T. Kojiri, and N. Pinkwart, "Automatic question generation for educational applications—The state of art," in *Proc. Adv. Comput. Methods Knowl. Eng.*, 2014, vol. 282, pp. 325–338.

[5] M. Rathod, T. Tu, and K. Stasaski, "Educational multi-question generation for reading comprehension," in *Proc. 17th Workshop Innov. Use NLP Build. Educ. Appl.*, 2022, pp. 216–223.

[6] M. Liu and R. A. Calvo, "Using information extraction to generate trigger questions for academic writing support," in *Proc. Int. Conf. Intell. Tutoring Syst.*, 2012, pp. 358–367.

[7] M. Liu, R. A. Calvo, and V. Rus, "G-Asks: An intelligent automatic question generation system for academic writing support," *Dialogue Discourse*, vol. 3, no. 2, pp. 101–124, Mar. 2012.

[8] J. Mostow and W. Chen, "Generating instruction automatically for the reading strategy of self-questioning," in *Proc. Int. Conf. Artif. Intell. Educ.*, 2009, pp. 456–472.

[9] H. Kunichika, T. Katayama, T. Hirashima, and A. Takeuchi, "Automated question generation methods for intelligent English learning systems and its evaluation," in *Proc. Int. Conf. Consum. Electron.*, 2004.

[10] Y. Huang and L. He, "Automatic generation of short answer questions for reading comprehension assessment," *Natural Lang. Eng.*, vol. 22, no. 3, pp. 457–489, May 2016.

[11] M. Heilman and N. A. Smith, "Good question! Statistical ranking for question generation," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2010, pp. 609–617.

[12] F. Chen, J. Xie, Y. Cai, T. Wang, and Q. Li, "Difficulty-controllable visual question generation," in *Proc. Conf. Web Big Data*, 2021, pp. 332–347.

[13] Y.-H. Chan and Y.-C. Fan, "A recurrent BERT-based model for question generation," in *Proc. 2nd Workshop Mach. Reading Question Answering*, 2019, pp. 154–162.

[14] X. Du, J. Shao, and C. Cardie, "Learning to ask: Neural question generation for reading comprehension," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1342–1352.

[15] A. Ushio, F. Alva-Manchego, and J. Camacho-Collados, "Generative language models for paragraph-level question generation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2022, pp. 670–688.

[16] J. Yu, Q. Su, X. Quan, and J. Yin, "Multi-hop reasoning question generation and its application," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 725–740, Jan. 2023.

[17] Y. Zhao, X. Ni, Y. Ding, and Q. Ke, "Paragraph-level neural question generation with maxout pointer and gated self-attention networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3901–3910.

[18] J. Li, Y. Gao, L. Bing, I. King, and M. R. Lyu, "Improving question generation with to the point context," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 3216–3226.

[19] P. Nema, A. K. Mohankumar, M. M. Khapra, B.V. Srinivasan, and B. Ravindran, "Let's ask again: Refine network for automatic question generation," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 3314–3323.

[20] X. Ma, Q. Zhu, Y. Zhou, and X. Li, "Improving question generation with sentence-level semantic matching and answer position inferring," in *Proc. AAAI conf. Artif. Intell.*, 2020, vol. 34, pp. 8464–8471.

[21] L. Song, Z. Wang, and W. Hamza, "A unified query-based generative model for question generation and question answering," 2017, *arXiv:1709.01058*.

[22] Y. Gao, L. Bing, W. Chen, M. Lyu, and I. King, "Difficulty controllable generation of reading comprehension questions," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 4968–4974.

[23] S. Lee and M. Lee, "Type-dependent prompt CycleQAG: Cycle consistency for multi-hop question generation," in *Proc. 29th Int. Conf. Comput. Linguistics*, 2022, pp. 6301–6314.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2019, pp. 4171–4186.

[25] A. Radford, J. Wu, R. Child, D. Luan, and D. A. I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[26] M. Lewis and et al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7871–7880.

[27] C. Raffel and et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, Jan. 2020.

[28] Y. Cheng et al., "Guiding the growth: Difficulty-controllable question generation through step-by-step rewriting," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics/11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 5968–5978.

[29] F. M. Lord, *Applications of Item Response Theory to Practical Testing Problems*. Evanston, IL, USA: Routledge, 1980.

[30] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000 questions for machine comprehension of text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 2383–2392.

[31] W. J. van der Linden and C. A. Glas, *Elements of Adaptive Testing*. New York, NY, USA: Springer, 2010.

[32] Q. Zhou, N. Yang, F. Wei, C. Tan, H. Bao, and M. Zhou, "Neural question generation from text: A preliminary study," in *Proc. Nat. CCF Conf. Natural Lang. Process. Chin. Comput.*, 2018, pp. 662–671.

[33] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017.

[34] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Proc. Neural Inf. Process. Syst.*, 2019.

[35] P. He, X. Liu, J. Gao, and W. Chen, "DeBERTa: Decoding-enhanced BERT with disentangled attention," in *Proc. Int. Conf. Learn. Representations*, 2021.

[36] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf Empirical Methods Natural Lang. Process./9th Int. Joint Conf Natural Lang. Process.*, 2019, pp. 3982–3992.

[37] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, "CTRL: A conditional transformer language model for controllable generation," 2019, *arXiv:1909.05858*.

[38] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[39] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.

[40] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[41] S. Subramanian, T. Wang, X. Yuan, S. Zhang, A. Trischler, and Y. Bengio, "Neural models for key phrase extraction and question generation," in *Proc. Workshop Mach. Reading Question Answering*, 2018, pp. 78–88.

[42] Y. Kim, H. Lee, J. Shin, and K. Jung, "Improving neural question generation using answer separation," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 6602–6609.

[43] X. Sun, J. Liu, Y. Lyu, W. He, Y. Ma, and S. Wang, "Answer-focused and position-aware neural question generation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3930–3939.

[44] M. Ueno and Y. Miyazawa, "IRT-based adaptive hints to scaffold learning in programming," *IEEE Trans. Learn. Technol.*, vol. 11, no. 4, pp. 415–428, Oct.–Dec. 2018.

[45] M. Uto and M. Ueno, "A generalized many-facet Rasch model and its Bayesian estimation using Hamiltonian Monte Carlo," *Behaviormetrika*, vol. 47, pp. 469–496, May 2020.

[46] M. Uto and M. Ueno, "Empirical comparison of item response theory models with Rater's parameters," *Heliyon*, vol. 4, no. 5, pp. 1–32 May 2018.

[47] M. Uto, "A Bayesian many-facet Rasch model with Markov modeling for rater severity drift," *Behav. Res. Methods*, vol. 55, pp. 3910–3928, Oct. 2022.

[48] M. Uto, "A multidimensional generalized many-facet Rasch model for rubric-based performance assessment," *Behaviormetrika*, vol. 48, pp. 423–457, Jul. 2021.

[49] G. Rasch, *Probabilistic Models for Some Intelligence and Attainment Tests*. Chicago, IL, USA: Univ. Chicago Press, 1981.

[50] S.-H. K. Frank and B. Baker, *Item Response Theory: Parameter Estimation Techniques*. Boca Raton, FL, USA: CRC Press, 2004.

[51] J.-P. Fox, *Bayesian Item Response Modeling: Theory and Applications*. New York, NY, USA: Springer, 2010.

[52] M. Uto, I. Aomi, E. Tsutsumi, and M. Ueno, "Integration of prediction scores from various automated essay scoring models using item response theory," *IEEE Trans. Learn. Technol.*, vol. 16, no. 6, pp. 983–1000, Dec. 2023.

[53] A. Srikanth, A. S. Umasankar, S. Thanu, and S. J. Nirmala, "Extractive text summarization using dynamic clustering and co-reference on BERT," in *5th Int. Conf. Comput., Commun. Secur.*, 2020, pp. 1–5.

[54] E. Tavan and M. Najafi, "MarSan at SemEval-2022 task 11: Multilingual complex named entity recognition using T5 and transformer encoder," in *Proc. 16th Int. Workshop Semantic Eval.*, 2022, pp. 1639–1647.

[55] M. Guo et al., "LongT5: Efficient text-to-text transformer for long sequences," in *Proc. Findings Assoc. Comput. Linguistics, North Amer. Chapter Assoc. Comput. Linguistics*, 2022, pp. 724–736.

[56] K. Grover, K. Kaur, K. Tiwari, and R. P. Kumar, "Deep learning based question generation using T5 transformer," in *Proc. Int. Adv. Comput. Conf.*, 2021, pp. 243–255.

[57] J. Ni et al., "Sentence-T5: Scalable sentence encoders from pre-trained text-to-text models," in *Proc. Findings Assoc. Comput. Linguistics*, 2022, pp. 1864–1874.

[58] E. M. Perkoff, A. Bhattacharyya, J. Cai, and J. Cao, "Comparing neural question generation architectures for reading comprehension," in *Proc. 18th Workshop Innov. Use NLP Building Educ. Appl.*, 2023, pp. 556–566.

[59] K. Vachev, M. Hardalov, G. Karadzhov, G. Georgiev, I. Koychev, and P. Nakov, "Leaf: Multiple-choice question generation," in *Proc. Eur. Conf. Inf. Retrieval*, 2022, pp. 321–328.

[60] M. Srivastava and N. Goodman, "Question generation for adaptive education," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 692–701.

[61] J. M. Linacre, "Sample size and item calibration stability," *Rasch Meas. Trans.*, vol. 7, 1994, Art. no. 328.

Yuto Tomikawa received the B.S. degree in engineering from the University of Electro-Communications, Chofu, Japan, in 2023, where he is currently working toward the M.S. degree in engineering.

His research interests include educational and psychological measurement, machine learning, and natural language processing.

Ayaka Suzuki received the B.S. degree in engineering from the University of Electro-Communications, Chofu, Japan, in 2022, where she is currently working toward the M.S. degree in engineering.

Her research interests include educational and psychological measurement, machine learning, and natural language processing.

Masaki Uto received the Ph.D. degree in engineering from the University of Electro-Communications (UEC), Chofu, Japan, in 2013.

He has been an Associate Professor with the Graduate School of Informatics and Engineering, UEC, since 2020. His research interests include educational and psychological measurement, Bayesian statistics, machine learning, and natural language processing.

Dr. Uto was the recipient of the Best Paper Runner-Up Award at the 2020 International Conference on Artificial Intelligence in Education.