Automatic Question Generation for the Japanese National Nursing Examination Using Large Language Models

Yûsei Kido[1], Hiroaki Yamada[1][a], Takenobu Tokunaga[1][b],
Rika Kimura[2][c], Yuriko Miura[2][d], Yumi Sakyo[2][e] and Naoko Hayashi[2][f]

[1]*School of Computing, Tokyo Institute of Technology, Japan*

[2]*Graduate School of Nursing Science, St. Luke's International University, Japan*

Keywords: Large Language Models, National Nursing Examination, Distractor Generation, Multiple-Choice Questions, Automatic Question Generation.

Abstract: This paper introduces our ongoing research project that aims to generate multiple-choice questions for the Japanese National Nursing Examination using large language models (LLMs). We report the progress and prospects of our project. A preliminary experiment assessing the LLMs' potential for question generation in the nursing domain led us to focus on distractor generation, which is a difficult part of the entire question-generation process. Therefore, our problem is generating distractors given a question stem and key (correct choice). We prepare a question dataset from the past National Nursing Examination for the training and evaluation of LLMs. The generated distractors are evaluated with compared to the reference distractors in the test set. We propose reference-based evaluation metrics for distractor generation by extending recall and precision, which is popular in information retrieval. However, as the reference is not the only acceptable answer, we also conduct human evaluation. We evaluate four LLMs: GPT-4 with few-shot learning, ChatGPT with few-shot learning, ChatGPT with fine-tuning and JSLM with fine-tuning. Our future plan includes improving the LLMs' performance by integrating question writing guidelines into the prompts to LLMs and conducting a large-scale administration of automatically generated questions.

1 INTRODUCTION

Automatic question generation (AQG) is one of the applications of natural language processing (NLP) that has been actively studied in the educational domain. It is expected to significantly reduce the burden on question writers. There have been a series of comprehensive surveys in this area (Alsubait et al., 2015; Kurdi et al., 2020; Faraby et al., 2023). Alsubait et al. (2015) covered 81 papers published up to 2014 and analysed them from nine aspects: purpose, domain, knowledge source, additional input, question generation method, distractor generation method, question format, answer format and evaluation. Concerning the domain, the number of studies targeting specific domains is slightly larger than that for a general domain. Language learning is a dominant domain among domain-specific studies, followed by mathematics and medicine, but they are few.

Kurdi et al. (2020) followed Alsubait et al. (2015) to systematically collected 93 papers on AQG published from 2015 to 2019 and analysed the trend. The domain distribution is similar to Alsubait et al. (2015), i.e. the numbers of studies on domain-specific and those on non-specific domains are comparable, and the language learning domain remains dominant in the domain-specific studies. Studies for new domains, such as sports, appeared in their survey.

The studies reviewed by Alsubait et al. (2015) and Kurdi et al. (2020) utilised template-based, rule-based (Liu et al., 2010) or statistical-based (Kumar et al., 2015; Gao et al., 2019) approaches. In contrast, Faraby et al. (2023) collected 224 neural network-based AQG papers published between 2016 and early 2022. Since neural network approaches require a large amount of training data, the availability of large datasets for Question-Answering (QA) systems, such

[a] https://orcid.org/0000-0002-1963-958X
[b] https://orcid.org/0000-0002-1399-9517
[c] https://orcid.org/0000-0001-9660-4471
[d] https://orcid.org/0009-0003-5270-603X
[e] https://orcid.org/0000-0001-9519-5792
[f] https://orcid.org/0000-0002-7058-692X

Kido, Y., Yamada, H., Tokunaga, T., Kimura, R., Miura, Y., Sakyo, Y. and Hayashi, N.
Automatic Question Generation for the Japanese National Nursing Examination Using Large Language Models.
DOI: 10.5220/0012729200003693

In *Proceedings of the 16th International Conference on Computer Supported Education (CSEDU 2024) - Volume 1*, pages 821-829
ISBN: 978-989-758-697-2; ISSN: 2184-5026

as SQuAD (Rajpurkar et al., 2016), NewsQA[1], has facilitated neural question generation (NQG). These datasets were initially designed and constructed for developing QA systems but are also usable for AQG. Faraby et al. (2023) focused on NQG in the educational context and introduced several datasets usable for NQG. Also, they mentioned the difficulty of evaluating AQG, which comes from multiple potentially appropriate questions and a shortage of evaluation datasets considering this characteristic.

At the end of 2022, ChatGPT[2] appeared, and numerous large language models (LLMs) followed. Their versatility and high performance for various tasks without fine-tuning made a great impact on both academia and industry (Liu et al., 2023b). LLMs have already been applied to AQG. For instance, Perkoff et al. (2023) compared three types of LLM architectures: T5, BART and GPT-2, in generating reading comprehension questions and concluded that T5 was the most promising. Yuan et al. (2023) used GPT-3 to generate questions and choose better questions among automatically generated candidates. Oh et al. (2023) utilised LLM for paraphrasing references to improve the evaluation metrics for AQG. Shin and Lee (2023) conducted a human evaluation of ChatGPT-generated multiple-choice questions (MCQs) for language learners, in which 50 language teachers evaluated a mixed set of human-made and ChatGPT-made MCQs without knowing their origins. They reported both types of MCQs had comparable quality.

We follow this line of research by utilising LLMs to generate questions. This research is a part of the project funded by the Ministry of Health, Labour and Welfare (MHLW) of Japan, which aims to automate administering the National Nursing Examination. Thus, our target domain is nursing; specifically, we aim to generate questions for the Japanese National Nursing Examination. This study assesses the feasibility of using the LLM-generated questions in the Japanese National Nursing Examination. Even though the question generation could not be fully automated, automating parts of the entire generation process can reduce the workload on question writers. Unlike the prevalent language learning domain, recruiting a large number of domain experts for evaluation is difficult in our nursing domain. Therefore, the challenge includes designing an effective and efficient evaluation framework with fewer human resources.

In this paper, we report the progress of our ongoing research project that aims to generate MCQs for the Japanese National Nursing Examination using LLMs. An MCQ consists of three components: a stem, a key and distractors, as shown in Figure 1. AQG needs to generate these three components to constitute a question. Particularly, distractor generation has been actively studied (Susanti et al., 2017; Gao et al., 2019) since generating distractors is a burdensome task for question writers. Considering the result of the preliminary experiment described in section 3, we shift our focus on distractor generation by LLMs.

[1] https://www.microsoft.com/en-us/research/project/newsqa-dataset/

[2] https://chat.openai.com

Which of the following was the leading cause of death in Japan in 2020?	← **stem**
1. malignant neoplasm	← **key**
2. pneumonia 3. heart disease 4. cerebrovascular disease	} **distractors**

Figure 1: Example of multiple-choice question (MCQ). The original is in Japanese. This is our translation. The same applies to the following examples.

2 JAPANESE NATIONAL NURSING EXAMINATION

Table 1: Choice type distribution in the past ten-year examinations.

Choice type	#questions
noun phrase	309
sentence	57
numerics	96
figure & table	22
exceptional questions	16
Total	500

In Japan, one must pass the National Nursing Examination to be qualified as a registered nurse. Graduating from a college or university with a nursing curriculum is a prerequisite to the examination. The examination evaluates the knowledge and skills required to become a registered nurse. It covers a wide range of subjects to confirm the knowledge about nursing from various perspectives, including the structure and function of the human body, the origin of disease and promotion of recovery, health support and social security system, basic nursing, adult nursing, gerontological nursing, pediatric nursing, maternal nursing, psychiatric nursing, home health care nursing, and integration and practice of nursing. These subjects are organised into a three-level hierarchical structure consisting of 16 major categories, 49 middle categories and 252 minor categories.

The examination questions are in the form of MCQ and classified into three: essential, general, and situational. This study focuses on the essential questions. The essential part consists of 50 questions that assess particularly important basic knowledge. A score of 80% or higher on the essential questions is necessary to pass the examination. The questions are intended to check the examinees' knowledge of nursing and are not intended to select examinees for a certain quota. Figure 1 shows a question example in the essential questions[3]. A question consists of a stem (question sentences), a key (correct choice) and three or rarely four distractors (incorrect choices).

The choices can be words or phrases like in Figure 1, longer descriptions in clauses and sentences, numerical values, graphs and tables. Table 1 shows the distribution of the choice types in the essential questions of the examinations over the past ten years, which were provided by MHLW, the body responsible for the National Nursing Examination. This study considers only questions with choices of words, phrases and sentences. Although some recent LLMs can handle images as well as language, the number of questions with figures and tables is small in the past examinations. Questions with numerical choices are better suited to rule-based approaches, e.g. setting appropriate error offsets against the correct value will make reasonable distractors.

3 PRELIMINARY EXPERIMENT

To evaluate the potential of LLMs, we conduct a preliminary study. Table 2 shows the topic, stem and key of four questions used in the preliminary study. We analysed the questions in the past examinations regarding the correct answer rate, the discrimination index and the distractor quality. These four questions[4] were selected because our analysis of the past ten-year questions revealed that the past questions on these topics should be improved.

To make LLMs work well on a task, appropriate prompts specialised to the task should be prepared (Liu et al., 2023a). LLMs need to create a stem, key and distractors to compose an MCQ. We prepare different prompts with varying information for LLMs. Table 3 shows variations of the prompts, in which "I" indicates the information given to LLMs in the prompt and "O" the output from LLMs. The characters in the prompt names (T, S and K) stand for the information given in the prompt. For example, the SK prompt contains a stem and key for generating distractors. The NON prompt contains only a topic shown in Talbe 3 to generate all MCQ components: a stem, key and distractors.

Additionally, we put a paragraph of the corresponding topic in nursing textbooks in the prompt, indicated in the "textbook" column and T in the prompt name, since textual information is the most popular input type to the AQG systems (Kurdi et al., 2020). Although ontology-type structured knowledge is a plausible option, we used textbook paragraphs since ontologies in our Japanese nursing domain are unavailable. We do not consider providing distractors in the prompt because creating distractors has been a crucial research theme in the past AQG research.

The eight prompt patterns in Table 3 and the four topics in Table 2 make 32 questions in total. We used ChatGPT (gpt3.5-turbo-0301) through Microsoft Azure API with the zero-shot approach, i.e. providing no exemplar. In the prompt, instead of creating a question all at once, we took an interactive approach of creating each component at each turn. For instance, an interaction with a SK prompt looks like Figure 2[5], where the USER part is our input, and the ASSISTANT part is the ChatGPT response. Surprisingly, ChatGPT generates four options, including the key, by just being instructed to "memorise" the stem. The explicit instruction to generate distractors in the second turn results in the same distractors generated in the first turn.

The generated 32 questions were assessed by four of the authors who are faculty members of a graduate school of nursing science, i.e. domain experts, and two of them have experience in writing questions of National Nursing Examination. The assessment revealed that only one question presented in Figure 2 was judged to be usable for National Nursing Examination.

After discussion with the assessors and hearing from other domain experts (non-authors) with experience in writing questions of National Nursing Examination, with showing the generated questions, we had the following findings.

- Question writers prefer manually creating question stems to manage the knowledge to assess. Creating stems is not so difficult for them in a setting like National Nursing Examination, i.e. assessing clearly defined basic domain knowledge.

[3]The actual questions are in Japanese. This is a translation by the authors.

[4]As the second topic has two keys, we can derive two questions from this topic with each as a key. We used only "dehydration" as the key in the preliminary experiment.

[5]The actual prompt is in Japanese. We translate it for readers.

Table 2: Target questions in the preliminary experiment (Translation).

Topic	Stem	Key
hair-washing assistance	Which of the following is correct in assisting with hair washing?	wash with the finger pads
repeated vomit	Which of the following is caused by repeated vomiting?	dehydration or alkalosis
standard precautions	Which of the following is considered a source of infection in the standard precautions?	saliva
basic structure of the human body and bodily fluids	Which of the following is the highest percentage of adult body weight?	intracellular fluids

Table 3: Prompt variations for the preliminary experiment.

name	topic	textbook	stem	key
NON	I		O	O
T		I	O	O
S			I	O
K			O	I
TS		I	I	O
TK		I	O	I
SK			I	I
TSK		I	I	I

USER: You will now create a question with the correct answer “intracellular fluids” from four options. Memorise the stem, which will be presented here. Stem: “Which of the following is the highest percentage of fluids in an adult’s body weight?”

ASSISTANT: Choices:

A. blood
B. lymph fluids
C. intracellular fluids
D. extracellular fluids

The correct answer is C. “intracellular fluids”.

USER: What are three distractors?
Format:
Distractors:

ASSISTANT:
Distractors:

A. blood
B. lymph fluids
D. extracellular fluids

Figure 2: Interaction with a SK prompt (Translation).

- On the other hand, creating distractors is the most difficult part of question generation. It is hard to find distractors that seem plausible but are not true. A set of distractors generated by ChatGPT is far from acceptable as is, but some of them are usable and inspiring.
- Providing an excerpt from the textbook concerning the topic in the prompt does not impact the results positively.

Based on these findings, we decided to focus on generating distractor candidates by providing a question stem and its key in the prompt.

4 AUTOMATIC EVALUATION METRICS

We propose automatic evaluation metrics to evaluate the quality of generated MCQs, particularly focusing on the quality of distractors. Automatic evaluation metrics are indispensable for efficient system development since human evaluation is expensive and time-consuming. ROUGE (Lin and Hovy, 2003) is a popular evaluation metric based on the overlapping ratio of units between system outputs and the references (correct outputs). ROUGE concerns the closeness of individual outputs to the references. However, we are concerned about assessing a set of generated distractors for a given stem and key, i.e. we need to consider the closeness between generated and reference distractors on a set basis instead of an individual basis. In this respect, set-theoretic metrics like recall and precision are more appropriate.

The relation among distractors in a set is also important. For instance, even though each generated distractor for a given stem and key is acceptable, they are unacceptable if they are all the same or similar.

We first consider recall and precision, well-known metrics in information retrieval. Given a question q_i in the test set Q_t, recall (R) and precision (P) are defined as in equation (1) and (2).

$$\mathrm{R_i} = \frac{|S_i \cap G_i|}{|G_i|}, \tag{1}$$

$$\mathrm{P_i} = \frac{|S_i \cap G_i|}{|S_i|}, \tag{2}$$

where S_i is an automatically generated distractor set for the stem and key of q_i and G_i is the distractors of q_i, i.e. the reference distractor set. Recall indicates how much the system can replicate the reference distractors, while precision indicates how much the generated distractors are acceptable with regard to the reference. We can average these metrics over the test set to obtain the overall quality of the generated distractors as in equation (3) and (4). They are called macro-averaged recall and precision.

$$\hat{\mathrm{R}} = \sum_{i=1}^{|Q_t|} \frac{\mathrm{R_i}}{|Q_t|}, \tag{3}$$

$$\hat{\mathrm{P}} = \sum_{i=1}^{|Q_t|} \frac{\mathrm{P_i}}{|Q_t|}. \tag{4}$$

Recall and precision are based on counting the number of generated distractors that are exactly the same as one of the reference distractors. Noun phrase distractors are highly probable to match the reference because they consist of a word or a few words. However, generated distractors in sentence form with more words hardly match the reference, meaning that we might consider generated distractors inappropriate even though they have the same meaning as the references. To remedy this problem, we introduce a similarity-based extension of recall and precision, i.e. similarity-recall ($\mathrm{R_s}$) and similarity-precision ($\mathrm{P_s}$). First, we assume a similarity metric $\mathrm{sim}(\cdot,\cdot)$ that defines a similarity value between two arguments, ranging from 0 to 1. The numerator of equation (1) and (2) can be written as

$$|S_i \cap G_i| = \frac{\sum_{j=1}^{|G_i|} \mathbb{1}(g_j \in S_i)}{|G_i|} = \frac{\sum_{k=1}^{|S_i|} \mathbb{1}(s_k \in G_i)}{|S_i|}, \tag{5}$$

where the index function $\mathbb{1}(\cdot)$ returns 1 when the argument proposition is true and 0 otherwise. Rewritten the numerator respectively, a natural extension of recall and precision from counting exact matching items to accumulating the maximum similarity values leads to equation (6) and (7).

$$\mathrm{R_s}_i = \frac{\sum_{j=1}^{|G_i|} \mathrm{sim}(\arg\max_{s \in S_i} \mathrm{sim}(s, g_j), g_j)}{|G_i|}, \tag{6}$$

$$\mathrm{P_s}_i = \frac{\sum_{k=1}^{|S_i|} \mathrm{sim}(s_k, \arg\max_{g \in G_i} \mathrm{sim}(s_k, g))}{|S_i|}. \tag{7}$$

Theoretically, these similarity-based metrics become larger values than the original recall and precision.

Our other concern is the relationship among distractors. The definition of $\mathrm{R_s}$ and $\mathrm{P_s}$ does not care about the correspondence between generated and reference distractors. For instance, $\mathrm{R_s}$ can not distinguish a situation where each generated distractor has a maximum similarity to a different reference distractor from a situation where all generated distractors have a maximum similarity to the same reference. To solve this drawback, we further extend $\mathrm{R_s}$ and $\mathrm{P_s}$ to take into account pairs of generated and reference distractors. Our idea is to find a pair set of generated and reference distractors that maximise the sum of similarities of the pairs and to use the similarity sum for the numerator to calculate recall and precision. Considering the distractor sets S_i and G_i as nodes and their correspondence as edges with the weight of their similarity, we can formulate our problem as the maximum weight matching (MWM) problem for weighted complete bipartite graphs. Efficient algorithms, e.g., the Hungarian algorithm, are known to solve this problem. Suppose we have a function $\mathrm{MWM}(V_1,V_2)$ that returns the maximum weight sum of a bipartite graph consisting of node sets V_1 and V_2, we define combinatorial similarity-based recall ($\mathrm{R_{cs}}$) and precision ($\mathrm{P_{cs}}$) by equation (8) and (9).

$$\mathrm{R_{cs}}_i = \frac{\mathrm{MWM}(S_i, G_i)}{G_i}, \tag{8}$$

$$\mathrm{P_{cs}}_i = \frac{\mathrm{MWM}(S_i, G_i)}{S_i}. \tag{9}$$

Like the original recall and precision, these extended metrics will be macro averaged over all test set questions for evaluation.

We are also concerned about the relationship among generated distractors in a question. A set of distractors representing similar concepts for a question should be avoided. We propose a metric, distractor variation (DV), representing how much distractors of a question differ from each other. Equation (10) defines the distractor variation for a question q_i. A larger DV indicates the distractors of a question are diverse.

$$\mathrm{DV}_i = 1 - \frac{\sum_{\{(s_j,s_k)|s_j,s_k \in S_i, j \neq k\}} \mathrm{sim}(s_j, s_k)}{|S_i|} \tag{10}$$

There are various ways to calculate the similarity between linguistic expressions, i.e., the implementation of $\mathrm{sim}(\cdot,\cdot)$. The ROUGE score mentioned above is one of them. In the experiment below, we use the text embedding technique (Patil et al., 2023) to calculate similarities between two distractors. Each distractor is transformed into an embedding (a dense real vector) and their cosine similarity with the minimum value rounded up to zero is used for a similarity of distractor pairs.

5 GENERATING DISTRACTOR CANDIDATES BY LLM

5.1 Dataset

As mentioned in section 2, we obtained the essential questions of the National Nursing Examination over the past ten years from MHLW. We target the questions with noun phrases and sentences as the choice, i.e. 309 (noun phrase choice) and 57 (sentence choice), 366 questions in total (cf. Table 1). These questions consist of 347 four-choice questions (95%) and 19 five-choice questions (5%). They are divided into 193, 87 and 86 questions for training, validation and test data while maintaining the distribution of subjects.

5.2 Models

We consider three language models: ChatGPT (gpt-3.5-turbo-0613), GPT-4 (gpt4-0613) (Achiam et al., 2023) developed by OpenAI, and the Japanese Stable LM instruct alpha 7B-v2 (JSLM) model (Lee et al., 2023) developed by Stability AI Japan. We used ChatGPT and GPT-4 through the Microsoft Azure API and JSLM on our GPU server. Since JSLM was created using Japanese corpora, we expect it to perform well in processing Japanese text.

We conduct fine-tuning for ChatGPT and JSLM using 193 questions in the training data. The fine-tuning is not available for GPT-4 at the time of writing this paper. The fine-tuning of ChatGPT is performed in five epochs through Microsoft Azure OpenAI API. We fine-tune JSLM using the SGD optimiser[6], batch size 1, learning rate 10^{-5} and 50 epochs. The model with the lowest loss on the validation data is adopted as the tuned model. We do not adopt any approximation techniques of parameter tuning like LoRA (Hu et al., 2021); instead, we conduct full-parameter tuning.

5.3 Prompting

In the name of prompt engineering, various techniques have been developed to create better prompts for controlling LLMs to respond successfully. The in-context learning method is the most popular technique, in which several exemplars consisting of an instruction and its corresponding appropriate response are included after instructions (Brown et al., 2020). It is also called the one/few-shot learning method according to the number of exemplars. In-context learning is particularly useful when additional model learning is difficult as in GPT-4. Our experiment adopts four-shot learning for GPT-4, as it does not allow fine-tuning; this model is named "gpt4". We also apply four-shot learning to ChatGPT (gpt3.5-turbo-0613) to see the difference in impact on performance between fine-tuning and in-context learning. We name ChatGPT with in-context learning "gpt3.5-few" and that with fine-tuning "gpt3.5-FT". For JSLM, we use no exemplar in the prompt (zero-shot).

For few-shot learning of gpt4 and gpt3.5-few, four questions are randomly selected from 193 questions in the training data for exemplars; they are used for all test questions. Figure 3 shows a zero-shot prompt, and Figure 4 shows a few-shot prompt, in which a stem and a key are embedded in the placeholders $\langle \mathrm{Q} \rangle$ and $\langle \mathrm{A} \rangle$ respectively, and N is four or five depending on the reference question[7]. In addition to filling the stems and keys, distractors are filled in the placeholder $\langle \mathrm{D}_n \rangle$ in the few-shot prompt. Most past questions have three distractors (four-choice questions), and the rest have four distractors. Since we are generating distractor candidates, our prompts instruct LLMs to generate five distractors for each question, including one or two additional candidates.

USER: Give us five distractors for the N-choice question "$\langle \mathrm{Q} \rangle$" with the correct answer "$\langle \mathrm{A} \rangle$".

Figure 3: Zero-shot prompt (Translation).

USER: Give us three distractors for the four-choice question "$\langle \mathrm{Q} \rangle$" with the correct answer "$\langle \mathrm{A} \rangle$".

ASSISTANT: Distractors:

- $\langle \mathrm{D}_1 \rangle$
- $\langle \mathrm{D}_2 \rangle$
- $\langle \mathrm{D}_3 \rangle$

— three more exemplars here —

USER: Give us five distractors for the N-choice question "$\langle \mathrm{Q} \rangle$" with the correct answer "$\langle \mathrm{A} \rangle$".

Figure 4: Few-shot prompt (Translation).

5.4 Results

Table 4 shows the evaluation metric values of the models for comparison. We used the Multilingual-E5-Large model (Wang et al., 2022) to transform distractors into 1024-dimensional real vectors for calculating similarity-based metrics. Table 4 is broken down into two tables: Table 5 and Table 6, which correspond to the questions with noun-phrase choices and those with sentence choices, respectively. The boldface indicates the best values across the compared models. The distractor variation (DV) values of the references are 0.126 for the entire set, 0.125 for the noun-phrase choice set and 0.127 for the sentence-choice set, respectively.

[6] https://pytorch.org/docs/stable/generated/torch.optim.SGD.html

[7] The prompt is in Japanese. This is a translation by the authors.

Table 4: Result of the entire test set (86 questions).

	gpt4	gpt3.5-few	gpt3.5-FT	JSLM
R	0.147	0.155	**0.178**	0.101
P	0.088	0.093	**0.107**	0.060
R_s	0.903	0.906	**0.909**	0.892
P_s	0.894	0.894	**0.896**	0.877
R_{cs}	0.901	0.903	**0.905**	0.886
P_{cs}	0.540	0.542	**0.543**	0.532
DV	0.116	0.116	0.116	**0.122**

Table 5: Result of the noun-phrase choice set (75 questions).

	gpt4	gpt3.5-few	gpt3.5-FT	JSLM
R	0.169	0.178	**0.204**	0.116
P	0.101	0.107	**0.123**	0.069
R_s	0.906	0.909	**0.913**	0.895
P_s	0.895	0.896	**0.899**	0.880
R_{cs}	0.903	0.906	**0.910**	0.889
P_{cs}	0.542	0.544	**0.546**	0.534
DV	0.116	0.117	0.120	**0.124**

We also conducted a manual evaluation by the domain-expert authors, the same as the preliminary experiment. The topics in Table 2 are used to generate distractor sets for human evaluation. We give a pair of a stem and a key as input; the second topic makes two questions by using individual keys out of two (dehydration or alkalosis). Therefore, we have five distractor sets generated by each model. Table 8 shows how many generated distractors are deemed acceptable by the experts ("exp") and how many of them are the same as the reference ("ref").

6 DISCUSSION AND PROSPECTS

In-Context Learning vs Fine-Tuning

Comparing gpt3.5-few and gpt3.5-FT reveals that fine-tuning is consistently more effective than in-context learning throughout all recall/precision metrics. Furthermore, we can gain improvement by fine-tuning with only 193 training instances. Gpt3.5-FT outperforms gtp4 in the question set with noun phrase choices and the entire set. We might achieve further improvement with more training data through fine-tuning. Collecting new data or applying data augmentation techniques (Li et al., 2022) to the existing data are possible research directions.

Impact of the Parameter Size

JSLM is consistently inferior to the GPT family in the recall/precision metrics. The parameter size of JSLM we used in this experiment is seven billion

Table 6: Result of the sentence choice set (11 questions).

	gpt4	gpt3.5-few	gpt3.5-FT	JSLM
R	0.000	0.000	0.000	0.000
P	0.000	0.000	0.000	0.000
R_s	**0.887**	0.884	0.881	0.874
P_s	**0.881**	0.880	0.877	0.856
R_{cs}	**0.885**	0.882	0.877	0.862
P_{cs}	**0.531**	0.529	0.526	0.517
DV	**0.115**	0.111	0.088	0.114

Table 7: Number of different most similar references to a generated distractor.

#ref	gpt4	gpt3.5-few	gpt3.5-FT	JSLM
1	34	29	32	25
2	45	47	42	40
3	7	10	12	21

(7B), which is far smaller than those of GPTs. The parameter sizes of GPT 3.5-turbo and GPT-4 are not officially published. Still, considering that the parameter size of their predecessor, GPT3, is 175 billion, we can estimate that GPT 3.5-turbo and GPT-4 have a parameter size of three or more orders of magnitude larger than JSLM (7B). We should adopt a larger JSLM model to see the impact of parameter sizes on performance. However, interestingly, JSLM shows a larger variation value (DV) for noun-phrase distractors than other models. The fact that JSLM has been trained on Japanese corpora might be the reason.

Noun-Phrase Choices vs Sentence Choices

The original recall and precision metrics (R and P) do not work for evaluating sentence distractors (Table 6). In contrast, the proposed similarity-based recall and precision metrics work for both noun-phrase and sentence distractors. To investigate the effectiveness of combinatorial similarity-based metrics (R_{cs} and P_{cs}), we counted how many most similar reference distractors to a generated distractor are there. The correspondence is shown in Table 7. In more than half of the cases in all models, multiple reference distractors are most similar to the same generated distractor. Such a situation is not favourable, particularly for measuring recall. Our combinatorial extension should remedy this problem. We presume that the larger number in the second and third rows of JSLM in Table 7 partially explains a larger drop from R_s to R_{cs} for JSLM than other models.

Human Evaluation vs Automatic Evaluation

Table 8 shows how many generated distractors are judged acceptable by the experts ("exp") and are the same as the reference ("ref"). We can see that gpt3.5-

Table 8: Result of manual evaluation.

	gpt4		gpt3.5 -few		gpt3.5 -FT		JSLM	
Topic	exp	ref	exp	ref	exp	ref	exp	ref
hair-washing	1	0	1	0	0	0	0	0
vomit/dehydration	3	1	3	1	5	2	2	1
vomit/alkalosis	1	0	3	0	1	0	1	0
std. precautions	3	2	1	1	1	1	0	0
body fluids	5	1	3	1	5	3	2	1
Total	13	4	11	3	12	6	5	2

FT reproduces the reference distractors most, which can be considered as an effect of fine-tuning. Another notable observation is that gpt4 generates many distractors acceptable by human experts, although it reproduces fewer reference distractors than gpt3.5-FT. Table 9 shows distractor examples generated by each model. We notice that gpt4 generates unique acceptable distractors compared to other models. In this respect, gpt4 might be more creative. As Faraby et al. (2023) pointed out, the reference distractor set is not the only appropriate set, which is also supported by our human evaluation. Gpt4 can assist the question writers by suggesting inspiring distractors they may not have considered. Reference-based automatic evaluation metrics can not appropriately evaluate generated distractors in this respect. Similarity-based metrics might remedy this problem since they calculate the similarity of generated distractors to the references. We need to verify the effectiveness of our automatic evaluation metrics by comparing them with human evaluation results on a large scale. Building an evaluator model using reinforcement learning techniques is another possible approach. Reinforcement learning based on human feedback is a common technique for tuning LLM (Christiano et al., 2023). The evaluator model can be free from the reference, and it can also be used to tune the distractor generation model.

Open vs Proprietary LLMs

This study employed an open LLM (JSLM) and proprietary LLMs (ChatGPT and GPT-4). Only a few research institutes can train huge language models, such as ChatGPT. The size of open LLMs currently available, such as JSLM, is still limited. It is unclear how close the performance of open LLMs can reach that of huge proprietary LLMs. However, open LLMs have advantages in their transparency, tuneability, reproducibility and security. Security is particularly crucial for our case, the national examination, considering the confidentiality of information. We will continue to work on both open and proprietary LLMs, balancing the aforementioned advantages of open LLMs and the high performance of proprietary LLMs.

Table 9: Example of generated distractors (Translation).

Stem: Which of the following is the highest percentage of body fluids to adult body weight?
Key: intracellular fluids

Model	Generated distructors
gpt4	**plasma**, *urine*, *sweat*, *bile*, *cerebrospinal fluids* (0.929, 0.909, 0.925, 0.555, 0.136)
gpt3.5-few	blood, adipose tissue, **lymph fluids**, *digestive fluids*, *urine* (0.940, 0.906, 0.931, 0.558, 0.137)
gpt3.5-FT	**plasma**, **interstitial fluids**, **lymph fluids**, *blood cell*, *platelet* (1.000, 0.963, 1.000, 0.600, 0.130)
JSLM	blood, somatic cell, extracellular fluids, **plasma**, **interstitial fluids** (0.953, 0.928, 0.950, 0.570, 0.127)
Reference	**plasma**, **interstitial fluids**, **lymph fluids**

Bolds are the same as the reference; italics are deemed acceptable by the experts. The values in the parentheses are metrics (R_s, P_s, R_{cs}, P_{cs}, DV).

Future Plan

This feasibility study shows promising results in automatically generating distractors for Japanese National Nursing Examination using LLMs.

To achieve further improvement in distractor generation, we are considering two approaches. The first is posting a negated question stem to QA systems or LLMs to obtain answers, which should be usable as distractors of the original question stem. Logically, this seems to work, but we must verify it through experiments. The second is integrating question writing guidelines into the LLMs prompts. A different group in our project has analysed the past examination questions to compile the guidelines. There are several prohibitions in question writing for National Nursing Examination, e.g. coined words should not be an option, opposing choices can not coexist and so on. We observed non-existing words in the generated distractors of our experiments. The distractors in breach of the prohibitions can be filtered out in the post-processing. However, breach-free outputs by LLMs are more preferable. The guidelines will also include rules to make questions better. Those rules can be incorporated into the prompts for LLMs.

After improving distractor generation, we plan to administer a large-scale mock-up examination that includes questions with automatically generated distractors. The size of the participating test-takers is expected to be 1,000. We will conduct a human evaluation of a mixed set of human-made and machine-made questions, following Susanti et al. (2017) and Shin and Lee (2023). Also, we will compare the testtaker responses to both types of questions on the same topic.

REFERENCES

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). GPT-4 technical report. *CoRR*, abs/2303.08774.

Alsubait, T., Parsia, B., and Sattler, U. (2015). Ontology-based multiple choice question generation. *KI - Künstliche Intelligenz*, 30.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Christiano, P., Leike, J., Brown, T. B., Martic, M., Legg, S., and Amodei, D. (2023). Deep reinforcement learning from human preferences. *CoRR*, abs/1706.03741.

Faraby, S. A., Adiwijaya, A., and Romadhony, A. (2023). Review on neural question generation for education purposes. *International Journal of Artificial Intelligence in Education*, pages 1–38.

Gao, Y., Bing, L., Chen, W., Lyu, M., and King, I. (2019). Difficulty controllable generation of reading comprehension questions. pages 4968–4974.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Kumar, G., Banchs, R., and D'Haro, L. (2015). Automatic fill-the-blank question generator for student self-assessment. pages 1–3.

Kurdi, G., Leo, J., Parsia, B., Sattler, U., and Al-Emari, S. (2020). A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30:121 – 204.

Lee, M., Nakamura, F., Shing, M., McCann, P., Akiba, T., and Orii, N. (2023). Japanese stablelm instruct alpha 7b v2.

Li, B., Hou, Y., and Che, W. (2022). Data augmentation approaches in natural language processing: A survey. *AI Open*, 3:71–90.

Lin, C.-Y. and Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 71–78.

Liu, M., Calvo, R. A., and Rus, V. (2010). Automatic question generation for literature review writing support. In *International Conference on Intelligent Tutoring Systems*.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2023a). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., Wu, Z., Zhao, L., Zhu, D., Li, X., Qiang, N., Shen, D., Liu, T., and Ge, B. (2023b). Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, 1(2):100017.

Oh, S., Go, H., Moon, H., Lee, Y., Jeong, M., Lee, H. S., and Choi, S. (2023). Evaluation of question generation needs more references. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6358–6367, Toronto, Canada. Association for Computational Linguistics.

Patil, R., Boit, S., Gudivada, V., and Nandigam, J. (2023). A survey of text representation and embedding techniques in nlp. *IEEE Access*, 11:36120–36146.

Perkoff, E. M., Bhattacharyya, A., Cai, J. Z., and Cao, J. (2023). Comparing neural question generation architectures for reading comprehension. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 556–566.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In Su, J., Duh, K., and Carreras, X., editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Shin, D. and Lee, J. H. (2023). Can chatgpt make reading comprehension testing items on par with human experts? *Language Learning & Technology*, 27(3):27–40.

Susanti, Y., Tokunaga, T., Nishikawa, H., and Obari, H. (2017). Evaluation of automatically generated English vocabulary questions. *Research and Practice in Technology Enhanced Learning*, 12(11):1–12.

Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., and Wei, F. (2022). Text embeddings by weakly-supervised contrastive pre-training. *CoRR*, abs/2212.03533.

Yuan, X., Wang, T., Wang, Y.-H., Fine, E., Abdelghani, R., Sauzéon, H., and Oudeyer, P.-Y. (2023). Selecting better samples from pre-trained LLMs: A case study on question generation. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12952–12965, Toronto, Canada. Association for Computational Linguistics.