Personalized Cloze Test Generation with Large Language Models: Streamlining MCQ Development and Enhancing Adaptive Learning

Chin-Hsuan Shen, Yi-Li Kuo, Yao-Chung Fan*,
Department of Computer Science and Engineering,
National Chung Hsing University, Taiwan
yfan@nchu.edu.tw

Abstract

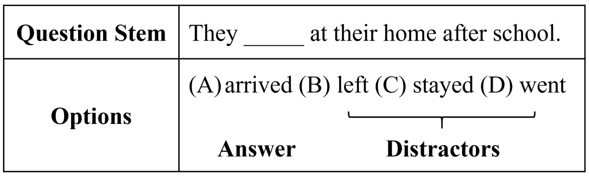
Cloze multiple-choice questions (MCQs) are essential for assessing comprehension in educational settings, but manually designing effective distractors is time-consuming. Addressing this, recent research has automated distractor generation, yet such methods often neglect to adjust the difficulty level to the learner's abilities, resulting in non-personalized assessments. This study introduces the Personalized Cloze Test Generation (PCGL) Framework, utilizing Large Language Models (LLMs) to generate cloze tests tailored to individual proficiency levels. Our PCGL Framework simplifies test creation by generating question stems and distractors from a single input word and adjusting the difficulty to match the learners proficiency. The framework significantly reduces the effort in creating tests and enhances personalized learning by dynamically adapting to the needs of each learner.

1 Introduction

Cloze multiple-choice questions are a prevalent form of assessment in educational settings. As depicted in Figure 1, a typical cloze test consists of a sentence with a blank and four answer choices: one correct answer and three distractors. Test-takers are required to select the correct answer to fill in the blank.

While high-quality distractors are crucial for accurately assessing students' comprehension levels, manually designing such distractors can be time-consuming and labor-intensive. Consequently, recent years have seen a surge in research focused on automating the task of distractor generation for cloze tests (Chiang et al., 2024; Ren and Zhu, 2021; Wang et al., 2023; Yu et al., 2024).

Despite the advancements in automated distractor generation, current methods produce non-personalized cloze tests that do not adjust to the difficulty based on a learner's abilities, overlooking the nuances of personalized learning as mentioned in (Shemshack and Spector, 2020).



Question Stem	They _____ at their home after school.
Options	(A) arrived (B) left (C) stayed (D) went **Answer** **Distractors**

Figure 1: Cloze Test example

Moreover, existing approaches typically require both a question stem and an answer as inputs. However, limited research has been conducted on generating a cloze test starting solely from a given answer, which includes creating both the corresponding question stem and distractors, as illustrated in Figure 2.

This study addresses these gaps by introducing the Personalized Cloze Test Generation (PCGL) framework. Using LLMs, the PCGL framework generates both the question stem and distractors from a single input answer, tailoring MCQs to match the user's difficulty level.

The contributions of this study are as follows:

- **Simplified Test Creation:** The PCGL Framework streamlines the process of cloze test creation by allowing users to generate a complete test from a single input word. This eliminates the need for manual preparation of question stems and distractors, thus reducing the time and effort typically required in test design.

- **Adjustable difficulty:** The PCGL is designed to adjust the difficulty level for MCQ generation, catering to the individual needs of each learner based on the desired difficulty level.

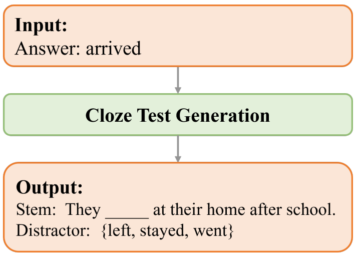

Input:
Answer: arrived
Cloze Test Generation
Output:
Stem: They _____ at their home after school.
Distractor: {left, stayed, went}

Figure 2: this study aims to generate a cloze test that includes both the corresponding question stem and appropriate distractors for a given answer.

2 Related Work

Recent methods for generating distractor options in cloze tests can be categorized into two main types: Candidate Generation and Ranking (CGR) framework (Ren and Zhu, 2021; Chiang et al., 2024), and the generative Text2Text framework (Wang et al., 2023).

In the CGR framework, CDGP (Chiang et al., 2024) is considered state-of-the-art. It employs a Candidate Selection Generator (CSG) to create multiple candidate distractors and a Distractor Selector (DS) to choose the three most suitable words as distractors, based on lexical and contextual relevance. Conversely, the Text2Text generation architecture, as described by (Wang et al., 2023), approaches distractor generation as a Text2Text task, where the question stem is concatenated with the answer before inputting into a generative language model (e.g., T5 or GPT) to train the model to produce a set of distractors.

Despite their advances, the CGR and Text2Text methods face significant limitations: they cannot adjust distractor difficulty levels and require a complete question stem with an answer. These constraints limit the adaptability of assessments and complicate the DG process. Our study aims to address these shortcomings.

3 Methodology

This study introduces a personalized cloze test generation framework, termed the PCGL Framework, which leverages LLMs for generating MCQs tailored to the difficulty experienced by individual users.

3.1 Data Assumption

In our study, we assume the availability of a Cloze-style MCQ dataset. Prominent examples of such datasets include the CLOTH dataset (Xie et al., 2017) and the MCQ dataset (Ren and Zhu, 2021). We presuppose that each entry in the dataset comprises a question stem (Q), a correct answer (A), and a set of distractors ($\{d_i\}$). Each distractor d_i is designed to be contextually relevant to both the question stem Q and the correct answer A. This assumption allows our proposed model to effectively learn and generate content that is not only contextually appropriate but also challenging enough to serve as plausible distractors in the cloze tests.

3.2 Problem Assumption

We assume a learner's language proficiency level U is available. Such information can be derived from the questions that the learner has previously answered incorrectly.

3.3 PCGL Framework

The PCGL Framework leverages LLMs to train a system for personalized cloze test generation. The framework is structured into the following stages:

1. **Question Sentence Generation (QSG) Model:** In this stage, the QSG model generates a sentence that includes the answer, forming the basis of the question stem.

2. **Distractor Generation (DG) Model:** The final stage utilizes the sentence from the QSG model to produce corresponding distractors.

Each component is designed to ensure that the generated sentence, answer, and distractors align with the assessed level of the learner, thereby facilitating targeted educational support.

3.4 Initial Model Training

The initial training phase configures the QSG and DG models with a comprehensive MCQ dataset to establish baseline capabilities for generating question stems and distractors:

- **QSG Model Training:** The QSG model is trained to transform a given answer A into a potential question stem Q. The training objective is to minimize the loss function $\mathcal{L}_{QSG}$, defined as the negative log-likelihood of the true question stem given the generated question stem:

$$\mathcal{L}_{QSG} = -\sum_{(Q,A)\in\mathcal{D}} \log p(Q|A) \qquad (1)$$

where $\mathcal{D}$ represents the training dataset consisting of question-answer pairs.

- **DG Model Training:** The DG model generates distractors based on the combination of a question stem Q and the correct answer A. The training objective is to minimize the loss function $\mathcal{L}_{DG}$, which is similarly defined as the negative log-likelihood of the true distractors given the generated distractors:

$$\mathcal{L}_{DG} = - \sum_{(\{d_i\},Q,A)\in\mathcal{D}} \log p(\{d_i\}|Q, A) \tag{2}$$

This equation considers the dataset $\mathcal{D}$, which now includes sets of distractors along with the question-answer pairs.

3.5 Personalized Fine-Tuning

In the personalized fine-tuning phase, we focus on aligning the training process with the learner's proficiency level. This alignment is achieved by selecting a subset $\mathcal{S}$ from the comprehensive MCQ dataset $\mathcal{D}$, tailored according to a specific difficulty criterion designed to match the learner's needs.

Difficulty Evaluation For each data entry $t = (Q, A, \{d_i\})$, the difficulty is determined using the CEFR ratings for words within the entry. The steps are:

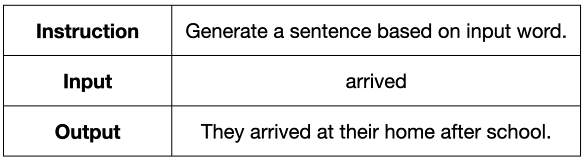
1. Extract all words from the question stem Q, correct answer A, and the set of distractors $\{d_i\}$.
2. Compute the difficulties of these words using the CEFR (Cambridge English Language Assessment for Languages) word lists (please refer to Table 2 in Appendix). Determine the overall difficulty $d(t)$ of the entry t by averaging the top-k highest word difficulties.

Subset Selection The subset $\mathcal{S}$ is selected from $\mathcal{D}$ based on how closely the difficulty of each entry aligns with the learner's assessed proficiency level U. An entry t is included in $\mathcal{S}$ if: $|U - o(t)| < 0.5$. This criterion ensures that the selected entries are challenging and relevant, promoting effective and personalized learning.

With $\mathcal{S}$, we further fine tune the QSG and DG models by the following objective functions.

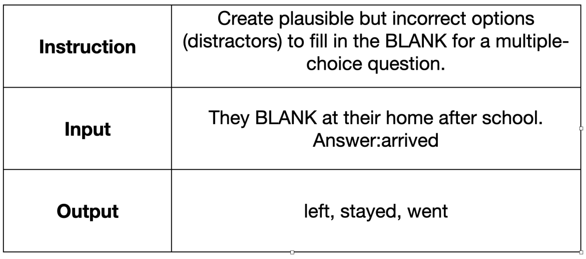
$$\mathcal{L}_{QSG} = - \sum_{(Q,A)\in\mathcal{S}} \log p(Q|A) \tag{3}$$

$$\mathcal{L}_{DG} = - \sum_{(\{d_i\},Q,A)\in\mathcal{S}} \log p(\{d_i\}|Q, A) \tag{4}$$



Instruction	Generate a sentence based on input word.
Input	arrived
Output	They arrived at their home after school.

Figure 3: QSG Prompt example



Instruction	Create plausible but incorrect options (distractors) to fill in the BLANK for a multiple-choice question.
Input	They BLANK at their home after school. Answer:arrived
Output	left, stayed, went

Figure 4: DG Prompt example

3.6 Inference Process

During inference, a word A (served as answer) is inputted into the fine-tuned QSG model to generate a question stem $\hat{Q}$. This stem, along with A, is then fed into the DG model to generate the final set of distractors $\{\hat{d_i}\}$, completing the personalized question generation process.

3.7 Prompting

In the fine-tuning process of a LLM, the prompt is designed to provide clear guidance to the model. The structure of the prompt is as follows: "Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request. Instruction:$\{instruction\}$ Input:$\{input\}$ Response:$\{output\}$"

The $instruction$, $input$ and $output$ in the prompt will be different due to each model and data.

QSG In the process of fine-tuning the QSG model, the instruction remains consistent across all training data, while the input and output vary according to each specific example, as illustrated in Figure 3.

DG In the process of fine-tuning the DG model, the instruction remains consistent across all training data, while the input and output vary according to each specific example, as illustrated in Figure 4.

In summary, the fine-tuning process for both the QSG and DG models relies on a structured prompt that provides consistent instructions while allowing the input and output to adapt based on the specific training data. This approach ensures that each model is effectively guided to perform its specialized taskwhether generating sentences or creating distractorsresulting in a robust and contextually sensitive LLM capable of producing high-quality cloze tests.

4 Performance Evaluation

4.1 Dataset

CLOTH Dataset (Xie et al., 2017) The CLOTH dataset, comprising English cloze tests with sentences, missing words, answers, and distractors, serves as the benchmarking dataset in this study. For dataset pre-processing details, please refer to the appendix section.

4.2 Implementation Details

Please refer to Appendix.

4.3 Evaluation Metrics and Methodology

The effectiveness of the PCGL Framework was assessed on two main fronts: difficulty adjustment and generation quality. To ensure the stability and credibility of the results, each experiment was conducted three times.

Difficulty Adjustment This metric evaluates the ability of the PCGL Framework to generate content that aligns with pre-defined difficulty levels (CEFR A1 and CEFR B2). We compared the difficulty distribution of outputs from both the base model and the personalized PCGL models. Difficulty levels were analyzed by calculating the proportion of generated sentences and distractors that fall within target difficulty ranges.

Generation Quality The quality of the generated questions was assessed by comparing outputs from our PCGL Framework against those produced by the existing CDGP method. We used GPT-4 to evaluate the questions from both methods by presenting generated questions to the model and observing its selection preferences. Please refer to the details about the GPT evaluation in Appendix.

4.4 Findings and Discussion

4.4.1 Difficulty Adjustment

- **Turning into A1 Level:** When evaluating A1 level difficulty, the baseline model demonstrated a higher frequency of producing sentences within the targeted difficulty range (0.5 to 1.5), achieving a match rate of 50.7%. In contrast, the enhanced A1 model from the PCGL framework matched this range at a slightly lower rate of 41.3%, as indicated in Figure 1 in appendix and Table 1. Despite this, the PCGL model excelled in generating distractors suitable for A1 level difficulty, with 61.7% of distractors falling within the target range, surpassing the 52.3% achieved by the baseline model in Figure 2 in appendix. This suggests that while the PCGL model may slightly underperform in sentence generation at A1 level, it offers significant improvements in distractor quality and relevance.

- **Turning into B2 Level:** At the B2 difficulty level, the enhanced B2 model of the PCGL framework outperformed the baseline model significantly, with 83.3% of generated sentences and 27.3% of distractors accurately matching the desired difficulty range of 3.5 to 4.5. This performance represents a substantial enhancement over the baseline model, which only managed to align 37% of its sentences and 13% of its distractors with the same difficulty range. These findings, highlighted in Figure 3 4 in appendix and detailed in Table 1, underscore the PCGL framework's effectiveness in tailoring content to more challenging B2 level requirements, demonstrating its capability to adaptively generate both sentences and distractors that meet specific educational standards.

4.4.2 Model comparison

We compare the QSG model and DG model's difficult adjustment with different training data (table 4 in appendix).

- **QSG:** Due to table 4 in appendix, we know that baseline model training on 10000 entries and enhanced model fine-tuning on 2000 and 10% baseline model training entries has better performance on average. It's sentence on a1, a2, b1 and b2 level is close to target score. On the other side, the QSG model whose base line model training on 20000 entries and enhanced model fine-tuning on 2000 and 10% baseline model training entries only has good performance on b2 level.

- **DG:** The performance on two type of DG model in table 4 in appendix is similar. There is only a difference in performance on a2

Experiment	Model Configuration	Mean	Median	STD
A1 Sentence Difficulty	**Baseline Model**: Standard settings	1.88	1.67	0.886
	Enhanced A1 Model: Tuned for A1 difficulty level	2.19	2.0	0.997
A1 Distractor Difficulty	**Baseline Model**: Standard settings	1.70	1.67	0.848
	Enhanced A1 Model: Tuned for A1 difficulty level	1.52	1.17	0.818
B2 Sentence Difficulty	**Baseline Model**: Standard settings	3.05	3.0	0.714
	Enhanced B2 Model: Tuned for B2 difficulty level	3.71	4.0	0.593
B2 Distractor Difficulty	**Baseline Model**: Standard settings	2.08	2.17	1.026
	Enhanced B2 Model: Tuned for B2 difficulty level	2.40	2.5	1.189

Table 1: Experiment results comparing baseline and enhanced models tuned for A1 and B2 difficulty levels across various experiments.

	Percentage Preference by GPT-4	
	A1 Level (%)	**B2 Level (%)**
PCGL	42.0	60.0
CDGP	33.0	34.0
Both	25.0	6.0

Table 2: Comparative Quality Evaluation by GPT-4 Across A1 and B2 Difficulty Levels

level. The DG model, baseline model training on 20000 entries and enhanced model fine-tuning on 2000 and 10% baseline model training entries, demonstrated a higher frequency of producing distractors within the targeted difficulty range (1.5 2.5).

4.4.3 Generation Quality

Evaluations using GPT-4 show a clear preference for questions from the PCGL system over the CDGP system, as detailed in Tables 2. At the A1 level, GPT-4 chose PCGL questions 42% of the time compared to CDGPs 33%. This preference increased at the B2 level, with PCGL questions chosen 60% versus CDGP's 34%.

These findings indicate that the PCGB Framework not only more accurately targets difficulty levels but also enhances question quality, consistently outperforming CDGP. The PCGL system's effectiveness in improving educational assessments suggests its potential to transform personalized learning experiences and contribute to more effective educational environments.

5 Conclusion

Our research demonstrates that fine-tuning two pre-trained models and enabling their cooperation can generate a complete cloze task from a single word while also allowing for the adjustment of the task's difficulty level. Although there remains room for improvement in fine-tuning the difficulty adjustments, the quality of the generated tasks already surpasses recent studies on cloze distractors.

6 Limitations

There is still room for improvement in adjusting the difficulty of the questions. Although our experimental results show that, compared to the default model, the difficulty-adjusted model tends to generate sentences and distractors that are closer to the target difficulty, some experimental results were not ideal. In several instances, the default model outperformed the difficulty-adjusted model.

Acknowledgement

This work is supported by NSTC 112-2634-F-005 -002-project Smart Sustainable New Agriculture Research Center (SMARTer), NSTC Taiwan Project under grant 112-2221-E-005 -075 -MY3, and Delta Research Center, Delta Electronics, Inc.

References

Shang-Hsuan Chiang, Ssu-Cheng Wang, and Yao-Chung Fan. 2024. Cdgp: Automatic cloze distractor generation based on pre-trained language model. *arXiv preprint arXiv:2403.10326*.

Siyu Ren and Kenny Q Zhu. 2021. Knowledge-driven distractor generation for cloze-style multiple choice questions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4339–4347.

Atikah Shemshack and Jonathan Michael Spector. 2020. A systematic literature review of personalized learning terms. *Smart Learning Environments*, 7(1):33.

Hui-Juan Wang, Kai-Yu Hsieh, Han-Cheng Yu, Jui-Ching Tsou, Yu An Shih, Chen-Hua Huang, and

Yao-Chung Fan. 2023. Distractor generation based on text2text language models with pseudo kullback-leibler divergence regulation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12477–12491.

Qizhe Xie, Guokun Lai, Zihang Dai, and Eduard Hovy. 2017. Large-scale cloze test dataset created by teachers. *arXiv preprint arXiv:1711.03225*.

Han Cheng Yu, Yu An Shih, Kin Man Law, KaiYu Hsieh, Yu Chen Cheng, Hsin Chih Ho, Zih An Lin, Wen-Chuan Hsu, and Yao-Chung Fan. 2024. Enhancing distractor generation for multiple-choice questions with retrieval augmented pretraining and knowledge graph integration. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11019–11029, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.