IJECBE (2026), **4**, **1**, 202–223
Received (4 August 2025) / Revised (6 June 2026)
Accepted (8 June 2026) / Published (30 March 2026)
https://doi.org/10.62146/ijecbe.v4i1.187
https://ijecbe.ui.ac.id
ISSN 3026-5258

International Journal of Electrical, Computer and Biomedical Engineering

RESEARCH ARTICLE

Development of Prompting Techniques to Trigger Proactive Dialogues in Large Language Models

Muhammad Firdaus Syawaludin Lubis* and Cecilia Inez Reva Manurung

Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, Depok, Indonesia
*Corresponding author. Email: m.firdaus04@ui.ac.id

Abstract

The rise of Artificial Intelligence (AI) challenges the integrity of traditional educational assessments, positioning the oral examination as a robust method for verifying authentic understanding. However, its large-scale implementation is hindered by significant logistical challenges, including time, resources, and evaluation consistency. This research addresses a foundational limitation for using Large Language Models (LLMs) in automated oral exams: their inherent passivity and failure to handle ambiguity, which are critical for effective assessment. To address this critical gap, we established an experimental framework and created two specialized datasets to systematically evaluate prompting techniques designed to elicit proactivity. Using the GPT-4o model, our comparative analysis reveals that the optimal strategy is highly task-dependent: for Clarification Need Prediction (CNP), a Proactive Chain-of-Thought (PCoT) zero-shot approach achieved a 0.99 F1-Score (95% CI [0.99, 1.00]); for Clarification Question Generation (CQG), the PCoT few-shot variant was most effective; and in target-guided scenarios, a simpler proactive prompt was superior on lexicaloverlap metrics while being statistically indistinguishable from PCoT in human judgement. We further report bootstrap confidence intervals and paired significance tests to substantiate these comparisons. As all experiments use a single model in a single domain, our claims are scoped to GPT-4o on computer-engineering content; these findings nonetheless provide foundational insights for developing automated oral examiners capable of nuanced, human-like dialogue, thereby addressing the scalability issues of traditional assessment methods.

Keywords: Proactive Chain-of-Thought, LLM, Prompting, Artificial Intelligence, Proactive Dialogue

1. Introduction

Assessing how well a student understands the material is a core part of teaching. Essays, case studies, applied projects, and structured examinations are all used for this purpose. Ideally such assessment is not only summative, assigning a final grade, but also formative: it gives students feedback that helps them identify their weaknesses and deepen their understanding [1].

The rise of Artificial Intelligence (AI) puts this integrity at risk, because AI tools can be misused for plagiarism [2], [3]. A written answer that looks convincing no longer guarantees authentic understanding [3], which pushes assessment toward more dynamic methods that probe how a student actually reasons. The oral examination is one such method. Because it is interactive, the examiner can ask unexpected followup questions, verify a student's understanding on the spot, and leave little room for AI-assisted cheating.

Oral examinations are effective but hard to scale. They are time-consuming, demand considerable faculty effort, and can be graded inconsistently. To address these scalability and standardization issues, we explore automating the oral examination, with Large Language Models (LLMs) as the core technology for such a system.

LLMs such as the Generative Pre-trained Transformer (GPT) [4] are a natural fit for these simulators, since they understand context and generate human-like responses [3], [5]. Their weakness is passivity. Unlike a human examiner, an LLM rarely takes the initiative: it seldom asks for clarification when a statement is ambiguous, or steers the conversation toward a specific learning objective. This matters, because an oral assessment is not a one-way interrogation; its validity depends on the examiner actively probing and steering the discussion to gauge a student's true competence.

We assume that an oral dialogue can be transcribed into text for an LLM to process. On that basis, we design and evaluate prompting techniques [6], [7] that push an LLM to initiate proactive dialogue. We focus on two forms of proactivity: Clarification Dialogue, to resolve ambiguity, and Target-Guided Dialogue, to keep the conversation on track. A full system would also need speech-to-text and text-to-speech components; here we deliberately limit the scope to generating the proactive textual response.

The approach rests on prompt engineering [8], the practice of crafting input prompts that steer an LLM's behavior. The prompting technique largely determines whether the desired proactive response appears. These techniques range from plain standard prompts and explicit instruction-based prompts to demonstrationbased (few-shot) prompting, where the model learns from examples [9], [10]. To systematically determine which of these strategies is most effective for eliciting clarification and target-guided dialogues, this study adapts the Proactive Chain-of-Thought (PCoT) prompting strategy, originally proposed by Deng et al. [11] for opendomain proactive dialogue, to the automated oral-examination setting, and develops an experimental framework to compare its performance against established methods. We emphasise that PCoT itself is not introduced here; our novelty lies instead in (i) operationalising and stress-testing PCoT for educational clarification and target-guided dialogue, (ii) the two new domain-specific datasets that make this controlled comparison possible, and (iii) the finding that the best prompting strategy is strongly task-dependent, including

the counter-intuitive failure of few-shot PCoT for clarification-need prediction.

This study yields three key contributions to the development of AI-driven educational tools:

1. The design of a reproducible experimental framework to compare three prompting techniques (Standard, Proactive/instruction-based, and PCoT), each under zero-shot and fewshot in-context learning, on the GPT-4o model.
2. The creation of two specialized datasets in the domain of computer engineering and programming algorithms for evaluating proactive dialogue: one for clarification scenarios, mirroring the characteristics of the Abg-CoQA dataset [12], and another for target-guided scenarios, based on the OTTers dataset [13].
3. A comparative analysis—supported by bootstrap confidence intervals and paired significance tests—that identifies the most effective prompting strategy for each specific proactive task, providing insights into how to build more adaptive and effective AI-driven educational tools.

2. Literature Study

This research investigates the use of prompting techniques to trigger proactive dialogue in LLMs. To set the background, this section reviews the two topics our work builds on: prompt engineering and proactive dialogue.

2.1 Prompt Engineering

Prompt engineering is a method of programming LLMs through a series of instructions or directives called prompts [4], [8], [7]. To achieve optimal output, an effective prompt is constructed from several key components. These components, illustrated in Fig. 1, serve as the foundation for directing the model's behavior.

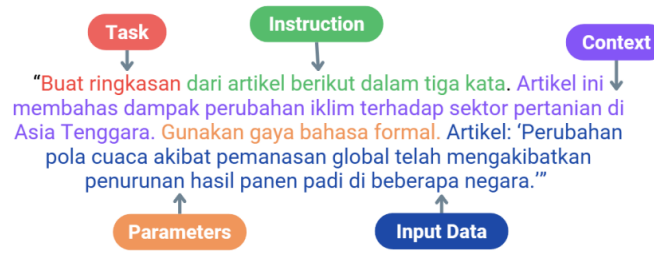

Figure 1. Illustration of the components that form a prompt

The primary approach in prompt engineering is in-context learning, which can be distinguished into two main types: *zero-shot prompting* and *few-shot prompting*. The Zero-shot prompting relies on instructions without examples, primarily using the Task and Instruction components. On the other hand, Few-shot prompting enriches the prompt with several demonstrations, thus heavily relying on the Output Example component to guide the model in understanding the desired response pattern [9].

Furthermore, various prompting techniques exist. *Standard Prompting* is the most basic, typically containing only the Task component. *Instruction Prompting* is more

structured, adding detailed Instruction and Constraints [14]. *Demo Prompting* explicitly uses the Output Example component to mimic a specific format or style [10]. Finally, *Proactive Chain-of-Thought (PCoT)*—introduced by Deng et al. [11] as an extension of Chain-of-Thought prompting [15]— instructs the model to articulate its internal reasoning before committing to a proactive action (e.g., deciding whether to ask a clarification question or how to bridge to a target topic), often leveraging role-based prompting and structured output templates to guide its thought process before reaching a final answer. Throughout this paper we use the abbreviation "PCoT" consistently; it refers to the same technique that some prior work denotes "ProCoT".

2.2 Proactive Dialogue

Proactive dialogue in a conversation system is defined as the ability of the system to take initiative and direct the conversation, rather than just passively responding to user input, in order to achieve predetermined communicative goals [16]. This capability is crucial as AI evolves from simple chatbots into more autonomous intelligent assistants. In an educational context, proactivity is essential for making interactions between an examiner and a student feel more realistic and effective; the system should not merely await answers but actively guide the exploration of understanding. However, even powerful LLMs like ChatGPT still exhibit limitations in proactivity, often remaining passive when faced with ambiguous queries or failing to handle problematic requests [17].

Previous research has explored developing proactive dialogue systems through various means. Early work often focused on conventional rule-based or statistical models to direct conversations [18], with later studies exploring neural networks and complex dialogue management modules to enhance proactivity [19]. Foundational research includes Kumar and Gupta's comprehensive review of clarification question generation methods [20], the fundamental framework for mixedinitiative dialogue by Allen, Ferguson, and Stent [21], and surveys on empowering LLMs with advanced conversational capabilities by Zhu et al. [22]. While these studies have laid a crucial groundwork, they often require extensive fine-tuning or significant architectural modifications.

More recently, a small but growing body of work has begun to apply LLMs directly to automated oral and spoken assessment, which is the application area closest to ours. Nitze [23] prototypes an LLM-based simulator of oral examinations that generates questions and personalised feedback, and Church et al. [24] propose a viva-voce simulator that interrogates a student's written submission to probe authentic understanding. In specialised domains, Silvestri et al. [25] build and clinically validate a GPT-4-powered chatbot for surgical oral-board scenarios—explicitly reporting hallucination and harm risks—while Chiu et al. [26] introduce a viva benchmark that requires multi-turn, information-seeking reasoning. Beyond text, Ma et al. [27] grade second-language oral proficiency directly from audio using a speech LLM, and, at the answer level, Kortemeyer [28] benchmarks GPT-4 for automated short-answer grading. Compared with these systems, which largely treat the LLM as a question generator, an examinee, or a grader, our work isolates and systematically compares *the prompting strategies* that trigger two specific proactive behaviours—clarification and

target-guided dialogue—and quantifies, with statistical testing, which strategy is best for each. To our knowledge, no prior study combines structured clarification and target-guided proactive-dialogue prompting for educational oral examination, which is the gap this paper addresses.

al examination, which is the gap this paper addresses. In contrast to the fine-tuning–based dialogue systems above, this study takes a different approach. It specifically investigates how prompting techniques can be leveraged to trigger the initiation of proactive dialogue in a state-of-the-art LLM without any architectural modifications. The research focuses on the model's ability to take the first proactive step in response to an initial input. The primary focus is on two types of proactive dialogue critical for effective communication:

- ***Clarification Dialogue:*** The system's ability to recognize ambiguity or incompleteness in a user's input and proactively request additional information to resolve it. This involves two key sub-tasks: predicting when clarification is needed (Clarification Need Prediction) and generating an appropriate question (Clarification Question Generation).
- ***Target-guided Dialogue:*** The system's ability to strategically steer the conversation toward a predetermined topic or goal, a formulation introduced for open-domain conversation by Tang et al. [29]. This is essential in scenarios like an oral exam, where the system must ensure that specific learning objectives are covered.

3. Methodology

This section outlines the methodology established to prepare for the experimental evaluation. The process involved four key preparatory stages: first, the design of the system architecture; second, the generation of specialized datasets; third, the implementation of the proposed prompting strategies; and finally, the definition of appropriate metrics to quantitatively assess performance. Each of these stages is described in detail below.

3.1 System Architecture

The system architecture was implemented as a structured, script-based methodology in Python, developed within Visual Studio Code. This approach is designed to be reproducible and efficient, handling the systematic testing of various prompting techniques on an LLM. The project utilizes key libraries including OpenAI for API communication [30], NLTK [31] and Scikit-learn [32] for evaluation metrics, and Chardet for character encoding detection [33]. The structure is organized into two primary folders corresponding to the dialogue scenarios under study: clarification and target-guided dialogue datasets. The entire workflow, illustrated in Fig. 2, is composed of four stages that correspond to the four labelled groups in the figure: *Data Preparation, Formatting, Model Interaction*, and *Performance Evaluation*.

The process begins with the Data Preparation stage, which gathers the two inputs required by the pipeline: a source dataset file (containing conversation histories and contexts) and a **prompt.txt** template file holding the prompt variants for every technique. In the subsequent Formatting stage, managed by the **process.py** script,

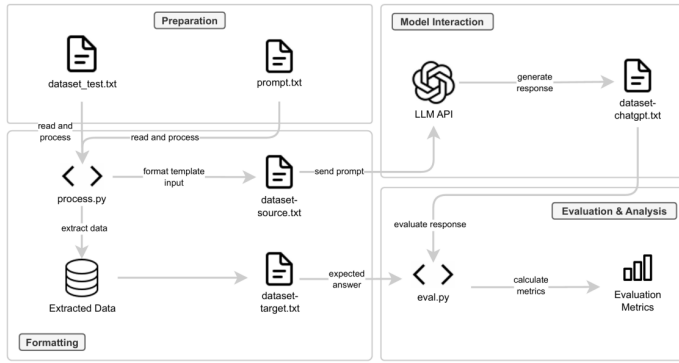

Figure 2. System architecture illustrating the workflow from data preparation and model interaction to final evaluation

the system extracts the data from each test case and dynamically integrates it with the corresponding prompt template to construct the final, fully-formatted prompts. The output is twofold: a **dataset-source.txt** file containing the prompts ready for the model, and a **dataset-target.txt** file holding the ground-truth answers for later evaluation.

Next, in the Model Interaction stage, the **test_chatgpt.py** script handles all communication with the OpenAI API. It systematically iterates through each prompt in **dataset-source.txt**, sending them to the exclusive model used for all experiments, GPT-4o. The model's responses are then collected and stored sequentially in a **dataset-chatgpt.txt** file.

Finally, the Performance Evaluation stage employs the **eval.py** script. This script performs a comparative analysis by measuring the model-generated responses in dataset-chatgpt.txt against the ground-truth answers from **dataset-target.txt**. It then calculates a suite of quantitative performance metrics to determine the effectiveness of each prompting strategy for the specific proactive tasks.

3.2 Dataset Development

To evaluate the proactive capabilities of the LLM, two specialized datasets were designed and generated, consisting of *course-related questions* tailored to the domain of computer engineering and programming algorithms. This approach ensures that the evaluation scenarios are relevant and realistically complex. Both datasets were initially created using a GPT model to achieve diversity and scale, followed by a meticulous manual curation process to guarantee the accuracy and quality of each test case. *Construction procedure.* The generation was steered by a structured prompt that fixed the output schema (the JSON fields described below) and explicitly required the items to span the six cognitive levels of Bloom's taxonomy [34] (remembering, understanding, applying, analysing, evaluating, and creating) so that the questions and answers vary systematically in difficulty rather than being near-duplicates. *Curation and quality control.* Every auto-generated item was then manually reviewed by the first author— a computer-engineering graduate familiar with the subject matter—against

four criteria: (i) factual correctness with respect to the programming-algorithms material, (ii) conformance to the required schema, (iii) internal consistency between the answer, the ambiguity label, and the reference clarification/bridging response, and (iv) appropriateness of the intended Bloom level. Items that were factually wrong, malformed, incoherent, or off-topic were either corrected or discarded. *Language*. Because the target deployment is an Indonesian-language oral examination, the datasets and all prompts were authored and the experiments were executed in Bahasa Indonesia; the examples shown throughout this paper are faithful English translations provided for readability. We acknowledge as a limitation that curation was performed by a single annotator, so no inter-annotator agreement could be computed; this is discussed in Section 6.

The first, the *Clarification Dialogue Dataset*, was developed to assess the model's ability to handle ambiguity, drawing its characteristics from the Abg-CoQA dataset [12]. This dataset consists of 500 unique items, where each entry contains a conversation history, a simulated student answer designed to be either clear or ambiguous, and an explicit label ('ambiguous' or 'non ambiguous') that serves as the ground truth. Of the 500 items, 340 (68%) are labelled 'ambiguous' and 160 (32%) 'non ambiguous'; this class distribution is summarised in Table 1. The full set of 500 items is used for the CNP task, while the 340 ambiguous items (each carrying a reference clarification question) constitute the test set for the CQG task. This label is critical for the *Clarification Need Prediction (CNP)* task. For entries marked as 'ambiguous', an ideal clarification question is also provided, serving as the ground truth for the *Clarification Question Generation (CQG)* task. To ensure a comprehensive evaluation, each data entry is structured with all necessary components: the '**story**' field provides the broader academic context, the '**history_turns**' array establishes the dialogue history, and the '**target_turn**' object contains the critical examiner's question and the simulated student's answer.

The second, the *Target-Guided Dialogue Dataset*, was created to evaluate the model's proficiency in steering a conversation toward a specific goal, inspired by the OTTers dataset [13]. This dataset comprises 258 items. Each sample includes a dialogue sequence representing a source topic (the student's response) and a desired target topic. To test the model's robustness, the dataset was intentionally designed so that within a group of scenarios, the initial question and the ideal target response remain consistent while the simulated student's answer is varied. For deeper analysis, the dataset also specifies key concepts for each turn: **s_c** (Source Concepts) from the initial question, **b_c** (Bridge Concepts) from the student's answer which the LLM must bridge from, and **t_c** (Target Concepts) representing the target topic that the LLM's generated response should achieve.

To make the structure concrete despite the dataset not being publicly released, Fig. 3a shows the schema of a representative Clarification entry and Fig. 4 that of a *Target-Guided* entry (both English-translated). For the clarification data, the **ambiguity** label is the CNP ground truth and **clarification_turn** the CQG reference; for the target-guided data, **dialog[2]** is the reference bridging response and **s_c/b_c/t_c** are the source/bridge/target concept lists.

Table 1. Dataset Statistics and Task Splits

Dataset	Items	Class balance	Used by
Clarification (Abg-CoQA-style)	500	340 amb. / 160 non-amb.	CNP (500)
ambiguous subset	340	_	CQG (340)
Target-Gided (OTTers-style)	258	_	Response gen. (258)

```
{
  "id": "programming_algorithms_2",
  "story": "Programming algorithms are step-by-step procedures ...",
  "history_turns": [ {"turn_id": 1, "question": "What is an algorithm?",
                      "answer": "A way to solve problems step by step."} ],
  "target_turn": {"turn_id": 2,
     "question": "Name an algorithm and explain how it works.",
     "answer": "Bubble Sort compares adjacent elements and swaps them ..."},
  "ambiguity": "ambiguous",
  "clarification_turn": {"question": "What is the time complexity of Bubble
  ↪  Sort?"}
}
```

(a) Clarification Dialogue entry.

```
{
  "sid": 301,
  "dialog": ["Sorting helps organize data efficiently.",
             "Merge sort uses divide and conquer to sort data.",
             "This leads to faster sorting on large datasets."],
  "s_c": ["sorting","organize","data","efficiently"],
  "b_c": ["merge","sort","divide","conquer"],
  "t_c": ["faster","sorting","large","datasets"]
}
```

(b) Target-Guided Dialogue entry. `dialog[0]` is the examiner context, `dialog[1]` the student answer to bridge from, and `dialog[2]` the reference target response.

Figure 3. Schema of a representative Clarification Dialogue entry (English translation)

Train/test separation and leakage control. The datasets are used purely for inference-time evaluation; no model weights are trained on them. For the few-shot conditions, a single fixed demonstration (a hand-crafted DFS example) is prepended to every prompt; this exemplar is authored separately and is *not* drawn from the test items, so the reported scores do not reflect train/test leakage. Importantly, for CNP the ground-truth **ambiguity** label is never included in the prompt—the model must infer it—so the near-perfect CNP result cannot be attributed to label leakage. We return to the interpretation of this result in Section 5.4.

Data availability. The two datasets are not publicly released at this time, because they were generated and curated under an institutional API account and because premature public posting risks contaminating future LLM training/benchmark corpora for this task. The complete schema and representative entries are provided above for transparency, and the datasets are available from the authors on reasonable request for non-commercial research use.

3.3 Prompting Design

The design of the prompts is a critical component of this methodology. The following subsections first detail the base templates created for each dialogue task, and then explain how these templates are used to construct the final prompting schemes for the experiments.

3.3.1 Clarification Dialogue Template

For the clarification dialogue scenario, the following prompt template is used:

```
Based on the provided document and answer, think about whether the answer is
↪ ambiguous or not. If it is ambiguous, ask a clarification question; the
↪ response must start with "Therefore, the answer is ambiguous. The
↪ clarification question is". If it is not ambiguous, answer the question.
```

Figure 4. Template of Prompting in Clarification Dialogue

The prompt template in Fig. 5 is an implementation of the Proactive Chain-of-Thought (PCoT) strategy with a zero-shot approach. PCoT uses an instruction ("**Based on the provided document and answer, think about whether...**") to think step-by-step and be proactive in clarification ("**If it is ambiguous, ask a clarification question...**"). This prompt will receive the context from the previous dialogue and the input data in the form of the student's response to be evaluated. This is a zero-shot approach because the prompt does not include an Output Example (input-output pair) to help the model understand the desired response pattern.

3.3.2 Target-guided Dialogue Template

For the target-guided dialogue scenario, the following prompt template is used:

```
Based on the answer history and the target answer, consider the relationship
↪ between the current answer and the target answer, then predict a suitable
↪ next topic that can smoothly bridge the current answer to approach the
↪ target answer. Then, based on the predicted next topic, generate an
↪ appropriate response. Please reply by completing the output template "The
↪ current answer is []. To bridge the current answer with the target answer,
↪ the next topic is []. Based on the predicted next topic, the response is"
```

Figure 5. Template of Prompting in Target-guided Dialogue

The prompt template in Fig. 6 is also an implementation of the PCoT strategy with a zeroshot approach. The PCoT aspect is evident from the detailed instruction to think step-by-step ("**consider the relationship between... then predict the next topic... Then based on the predicted next topic...**") as well as the proactive goal of bridging and directing the dialogue. This prompt will utilize the context from the previous dialogue history and the input data in the form of the current answer and the target answer.

3.3.3 Prompting Schemes

The final input prompts were generated using prompting schemes, which are defined as the dynamic combination of a base template (see Sections 3.3.1 and 3.3.2), the data

of a specific *test case*, a *prompting technique*, and a *learning approach*. This customization was applied distinctly for the Clarification and Target-Guided Dialogue scenarios. The number of test cases is fixed by the datasets described above: 500 for CNP, 340 (the ambiguous subset) for CQG, and 258 for target-guided response generation.

This study evaluates three prompting techniques: **Standard** (a plain task instruction), **Proactive** (an instruction-based prompt that explicitly directs the model to act proactively, e.g., to begin its reply with a fixed signal phrase), and **PCoT** (which additionally asks the model to reason step-by-step before acting). Each technique is crossed with two learning approaches— zero-shot and few-shot—giving the strategy labels used in Section 5 (e.g., "PCoT, 1-shot"). The zero-shot variant gives the model a direct instruction with no preceding example; the few-shot variant augments the same base prompt with a single, complete demonstration of the task to provide a contextual anchor. We deliberately treat the demonstration (sometimes called "demo" prompting) as the *mechanism that realises the few-shot approach* rather than as a separate technique; this is why the experimental tables report Standard/Proactive/PCoT each at 0-shot and 1-shot rather than a separate "Demo" row. Not every technique is meaningful for every sub-task: because Standard prompting does not emit a clarification-need signal, CNP compares only Proactive and PCoT, whereas CQG (which also includes a Standard baseline) and target-guided response generation are reported for the applicable techniques in Section 5.

3.4 Evaluation Metrics

The clarification and target-guided tasks serve fundamentally different functions, and the clarification task is itself split into predicting the *need* for clarification and generating the clarifying *question*. This functional divergence makes it essential to evaluate each sub-task with metrics suited to its nature. All of the automatic measures below are established metrics from the machine-translation and summarisation literature and are not contributions of this paper; we report them with the standard citations and the exact implementations noted so that the numbers can be reproduced.

Clarification Need Prediction. Because CNP is a binary decision, whether a given answer requires clarification or not, we treat it as a classification problem and report Precision, Recall, and their harmonic mean, the F1-score. Precision is the fraction of the answers the model flagged as needing clarification that genuinely did, so a high value reflects a low false-positive rate and a model that is reliable when it chooses to act. Recall is the fraction of the answers that truly needed clarification that the model actually caught, so a high value reflects a low false-negative rate and a model that rarely misses an opportunity to clarify. The F1-score balances the two into a single figure that penalises both error types, and we take it as our primary summary of CNP performance.

Clarification Question Generation. Once the need for clarification is established, the task becomes generating a well-formed question, which we score against the groundtruth reference using BLEU [35]. BLEU measures the n-gram overlap between the generated text and the reference, capturing lexical similarity and fluency, with higher scores indicating a closer match to the target question. We report sentence-level BLEU-1 through BLEU-4 computed with the NLTK implementation [31]

without smoothing, averaged over the test items.

Target-Guided Dialogue. Judging a bridging response requires capturing several aspects of similarity at once, so we combine three complementary metrics. BLEU [35] again measures n-gram precision and lexical overlap with the reference, here computed at the corpus level with sacreBLEU [36]. ROUGE-L [37] rewards words shared in the same order by measuring the Longest Common Subsequence between the generated and reference texts; we use the F-measure variant ($\beta = 1.2$) from the MS-COCO caption-evaluation toolkit. METEOR [38] aligns the two texts word by word, crediting not only exact matches but also stemmed forms and synonyms, and so gives a more semantically aware picture than BLEU or ROUGE alone; we use the NLTK implementation [31].

Human evaluation. Because these overlap-based metrics are known to under-credit responses that are semantically valid but lexically divergent, we complement them with a human-judgement score: a simple author-defined 1–10 Likert scale (1 = very poor, 10 = excellent) capturing the overall quality of a generated response, that is, its relevance, correctness, and appropriateness as an examiner turn. This is not a standardised instrument; the detailed rating protocol (two evaluators, blinding, and coverage) is described in Section 4.2, and the small-panel limitation is acknowledged in Section 6.

4. Experiments

This section details the experimental procedure designed to evaluate the performance of the various prompting strategies. The experiments were conducted across two distinct proactive dialogue scenarios: clarification and target-guidance, with the specific prompting schemes applied as described in the previous sections. The following subsections describe the execution flow for each dialogue scenario, including examples of the prompts used.

4.1 Experimental Setup and Reproducibility

All responses were generated with OpenAI's gpt-4o model through the Chat Completions API. To maximise determinism and reproducibility we set temperature = 0, max tokens = 512, and n = 1 (a single completion per prompt); other decoding parameters were left at their API defaults. Each prompt was issued once. The pipeline is implemented in Python; evaluation uses scikit-learn [32] for precision/recall/F1, NLTK [31] for sentence-BLEU and METEOR, sacreBLEU [36] for corpus-BLEU, and an MS-COCO-style ROUGE-L. The exact prompt templates for every technique (Standard, Proactive, PCoT) and learning approach (Fig. 10), together with the single few-shot demonstration (Fig. 11), are reproduced in the Appendix. Because the experiments were run in Bahasa Indonesia, the Appendix also provides the original Indonesian templates alongside the English translations used in the main text.

4.2 Human Evaluation Protocol

The human-judgement scores reported in Tables 4 and 6 were produced under the following protocol. Two evaluators rated the generated responses on the 1–10 scale defined above: one is a researcher (the first author, a computer-engineering graduate

familiar with the programmingalgorithms material) and the other a non-researcher, included to reduce single-perspective bias. For each test item the evaluators were shown the reference (target) answer together with the competing systems' generated responses, but not the identity of the prompting strategy that produced each response (the responses were extracted programmatically by their trigger phrases and presented without method labels), providing a degree of blinding against method bias. The clarification responses for the six strategies and the target-guided responses for the four strategies were each scored across the evaluation set, and the per-item scores were averaged over the two evaluators before computing the per-method means reported here. Given the small twoperson panel, we did not compute a formal inter-rater agreement coefficient; we note this as a limitation (Section 6) and, to support the comparisons statistically, we report 95% confidence intervals and paired significance tests in Section 5.

4.3 Clarification Dialogue Scenario

The clarification experiment was conducted in two distinct phases: Clarification Need Prediction (CNP) and Clarification Question Generation (CQG).

For the CNP phase, the model's task was to infer the need for clarification without being given the ground-truth ambiguity label. The evaluation was automated by a script that analyzed the response's prefix. If the generated text began with a key phrase like "**The clarifying question is...**", it was labeled as ambiguous (1); otherwise, it was labeled as not ambiguous (0). An example of the input and expected outputs for this task is shown in Fig. 6a.

For the CQG phase, the prompt explicitly informed the model that the student's answer was ambiguous. The model was then evaluated solely on its ability to generate a high-quality clarification question against the ground-truth reference, as shown in the example in Fig. 6b.

```
Story: "Depth-First Search (DFS) is a graph traversal algorithm..."
Question (System): "What is DFS?"
Answer (Student): "DFS is a sorting algorithm that arranges data in ascending
↪  order."

Prompt (pcot): Based on the given document and answer, think about whether the
↪  answer is ambiguous or not. If it is ambiguous, ask a clarification
↪  question; the response must start with "Therefore, the answer is ambiguous.
↪  The clarification question is". If it is not ambiguous, answer the question.

Expected LLM Response (Ambiguous): "Therefore, the answer is ambiguous. The
↪  clarification question is: Can you provide more detail that DFS is a sorting
↪  method...?"
```

(a) CNP testing scenario.

```
Story: "Depth-First Search (DFS) is a graph traversal algorithm..."
Question (System): "What is DFS?"
Answer (Student): "DFS is a sorting algorithm that arranges data in ascending
↪  order."

Prompt (zs_pcot_cqg): Based on the document and conversation history, the answer
↪  is ambiguous. The response must start with the reason why the answer is
↪  ambiguous, followed by "Therefore, the answer is ambiguous. The
↪  clarification question is".

Reference clarification question (ground truth): "Are you sure DFS is used to
↪  sort data?"
Expected LLM Response: "The student confuses DFS with a sorting method, which
↪  contradicts the document. Therefore, the answer is ambiguous. The
↪  clarification question is: Are you sure DFS is used to sort data?"
```

(**b**) CQG testing scenario. Unlike CNP, the model is told the answer is ambiguous and is scored only on the generated clarification question against the ground-truth reference.

Figure 6. Testing scenarios for the two clarification sub-tasks. Panel (a) shows the CNP task, where the model must infer whether clarification is needed; panel (b) shows the CQG task, where the model is told the answer is ambiguous and must generate the clarification question.

4.4 Target-Guided Dialogue Scenario

In this scenario, the model's objective was to strategically guide a conversation toward a predefined target. It was provided with the conversation history and a target topic. The prompts instructed the model to analyze the relationship between the current and target answers and then generate a response that would smoothly bridge the conversation, as illustrated in Fig. 7. The model's success was measured by its ability to produce an effective and coherent transitional response.

```
Dialog 1 (Question): "What is a recursive function?"
Dialog 2 (Answer): "It's a function that calls itself."
Dialog 3 (Target Topic): "It helps solve smaller problems"

Prompt (pcot): Based on the answer history and the target answer, consider the
↪  relationship between the current answer and the target answer, then predict
↪  a suitable next topic... Please reply by completing the output template "The
↪  current answer is []. To bridge..., the next topic is []. Based on..., the
↪  response is".

Expected LLM Response (brackets filled by the model):
"The current answer is [a recursive function calls itself]. To bridge the
current answer with the target answer, the next topic is [how recursion
breaks a problem into smaller sub-problems]. Based on the predicted next
topic, the response is: This self-calling behaviour lets the function solve
a problem by breaking it down into smaller, more manageable parts."
```

Figure 7. Example of a Target-Guided Dialogue Testing Scenario. The bracketed slots "[]" in the prompt are output placeholders that the model fills in: the first restates the current answer, the second names the predicted bridging topic, and the third is the final bridging response that is scored.

5. Results and Discussions

The experiments yielded distinct results for each proactive dialogue scenario, revealing that the optimal prompting strategy is highly dependent on the specific task. To substantiate the comparisons rather than relying on point estimates alone, we accompany the metric tables with uncertainty and significance analysis: 95% confidence intervals (CIs) obtained by nonparametric bootstrap (5,000 resamples) for the CNP F1-score and for the human-judgement means, McNemar's exact test for paired CNP decisions, and the two-sided Wilcoxon signedrank test for paired human-score comparisons. Reported p-values are for the specific contrasts named in the text.

5.1 Clarification Need Prediction (CNP) Performance

Table 2 compares the prompting strategies on the Clarification Need Prediction (CNP) task.

Table 2. CNP EVALUATION RESULTS

Prompting Strategy		CNP Performance		
Method	Shot	Precission	Recall	F1-Score
Proactive	0	1.00	0.30	0.47
Proactive	1	1.00	0.85	0.92
PCoT	0	0.99	0.99	0.99
PCoT	1	1.00	0.17	0.29

For the task of identifying whether a student's answer was ambiguous, the PCoT Zero-Shot method performed exceptionally well, achieving a nearly perfect F1 Score of 0.99, with balanced Precision (0.99) and Recall (0.99). This indicates that instructing the model to follow a logical chain of thought is highly effective for ambiguity detection. Unexpectedly, the PCoT Few-Shot method performed the worst, with an F1 Score of only 0.29. This drastic drop in performance, primarily due to a collapse in Recall to 0.17, suggests that providing a single example caused the model to overfit. The model likely latched onto the specific pattern in the demonstration and failed to generalize to other forms of ambiguity.

These point estimates are statistically robust. A non-parametric bootstrap (5,000 resamples over the 500 items) places the PCoT zero-shot F1 at 0.99 with a 95% CI of [0.99, 1.00], while the PCoT few-shot collapse is tightly estimated at F1 [0.24, 0.35] and the Proactive zero-shot value at [0.41, 0.53], confirming that these large gaps are not sampling artefacts. A paired McNemar exact test on per-item decisions confirms that PCoT zero-shot is significantly more accurate than its closest competitor, Proactive few-shot ($p < 10^{-7}$), and than Proactive zeroshot ($p < 10^{-6}$). We stress that this near-perfect score is not a label-leakage artefact: the ground-truth ambiguity label is never shown to the model during CNP, and the very fact that a single in-context example can drive performance from 0.99 down to 0.29 demonstrates that the task is non-trivial and sensitive to prompt design. Nonetheless, two factors likely inflate the absolute score relative to naturalistic classroom dialogue, and we state them plainly: (a) the test items are synthetic and share a templated structure, and (b) the CNP decision is recovered by a rule-based match on the response prefix (the trigger

phrase "the clarification question is"), which rewards the very output format that PCoT is instructed to produce. We therefore read 0.99 as an upper bound under controlled conditions rather than an expected field accuracy; validating CNP on human-generated answers with a learned or human-judged decision rule is left to future work (Section 6).

5.2 Clarification Question Generation (CQG) Performance

For the generative task of Clarification Question Generation (CQG), the results presented in Table 3 reveal a stark performance gap, with the PCoT Few-Shot method emerging as the superior strategy by a significant margin. It not only achieved the highest scores across all BLEU metrics (e.g., a BLEU-1 of 41.8) but also produced the most efficient outputs, with the lowest average response length of just 12.2 tokens.

Table 3. CQG EVALUATION RESULTS

Prompting Strategy		CQG Performance				
Method	**Shot**	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**Avg. Length**
Standard	0	15.4	7.7	4.1	2.4	62.8
Standard	1	28.7	16.0	9.3	5.7	35.6
Proactive	0	24.3	11.3	5.4	3.1	33.6
Proactive	1	32.6	19.0	11.4	7.2	30.0
PCoT	0	27.8	12.1	4.7	2.6	18.8
PCoT	1	**41.8**	**26.9**	**17.7**	**12.7**	**12.2**

This success comes from a two-fold mechanism in its design. First, the PCoT prompt compelled the model to perform an internal reasoning step to identify the source of the ambiguity in the student's answer. Second, the single demonstration (few-shot) provided a clear structural template, teaching the model to output its reasoning and then use a distinct key phrase, "The clarification question is:", before the question itself. This structure was critical for the evaluation script, which was programmed to isolate and assess only the text following this trigger phrase, ensuring a clean and accurate evaluation of the question's quality.

In contrast, the Standard zero-shot method failed because, lacking specific structural guidance, the model defaulted to a general explanatory mode. Instead of asking a targeted question, it generated long, descriptive paragraphs attempting to correct the student's misconception. Since this entire irrelevant text block was compared against a short, concise ground-truth question, the n-gram overlap was minimal, leading to extremely low BLEU scores and a high average response length (62.8).

Table 4. Human Evaluation of Clarification Response Quality

Method	Shot	Avg. Human Score	95% CI
Standard	0	5.54	[5.44, 5.64]
Standard	1	6.56	[6.46, 6.66]
Proactive	0	7.36	[7.27, 7.46]
Proactive	1	8.40	[8.30, 8.49]
PCoT	0	8.45	[8.35, 8.55]
PCoT	1	**8.83**	**[8.73, 8.93]**

Scores averaged over n = 500 items; CIs from a 5,000-sample bootstrap.

These automated findings were strongly corroborated by the human evaluation results presented in Table 4. The concise, well-structured questions from the PCoT methods were perceived as being of the highest quality, receiving the top scores (8.45 for Zero-Shot and 8.83 for Few-Shot). A two-sided Wilcoxon signed-rank test confirms that PCoT few-shot is rated significantly higher than every other strategy, including PCoT zero-shot ($p < 10^{-7}$) and Proactive few-shot ($p < 10^{-9}$); the difference between PCoT zero-shot and Proactive few-shot is not significant ($p = 0.47$), indicating these two are of comparable perceived quality. This confirms that the qualities favored by the BLEU metric, brevity and lexical relevance, also align with what human evaluators consider to be an effective clarification.

5.3 Target-Guided Dialogue Performance

For guiding a dialogue towards a target topic, the Proactive method outperformed PCoT on automatic metrics, as detailed in Table 5. Specifically, the Proactive Few-Shot approach achieved the highest METEOR (0.57) and ROUGE-L (0.38) scores, indicating it was most effective at capturing the lexical and structural similarity of the reference response. The PCoT methods scored lower on these metrics primarily because their prompts induced verbose, explicit reasoning that was not present in the concise ground-truth responses. This added text, which is visible in PCoT's higher average response length, penalized the scores that are based on direct token overlap.

Table 5. Response Generation Evaluation Results

Prompting Strategy		Response Generation Performance			
Method	Shot	BLEU	METEOR	R-L	Avg. Length
Proactive	0	**17.0**	0.47	0.30	33.9
Proactive	1	10.8	0.57	**0.38**	27.6
PCoT	0	**7.0**	**0.39**	0.26	33.6
PCoT	1	4.0	0.45	0.23	**54.5**

However, this highlights a potential limitation of the automatic metrics alone. When turning to the human evaluation results, summarized in Table 6, a more nuanced picture emerges. Here, the PCoT Few-Shot strategy was rated almost as highly (9.00) as the top-performing Proactive Few-Shot method (9.03). Crucially, this near-equality is statistically confirmed: a two-sided Wilcoxon signed-rank test

finds no significant difference between Proactive few-shot and PCoT few-shot ($p = 0.56$), even though the automatic-metric gap between them is large. This is direct quantitative evidence that the automatic metrics penalised the verbosity of the PCoT responses without a corresponding loss in human-perceived quality: evaluators found the explicit reasoning to be valuable, coherent, and effective, perceiving its quality to be on par with the more concise, direct approach.

Table 6. Human Evaluation of Target-Guided Response Quality

Method	Shot	Avg. Human Score	95% CI
Proactive	0	**7.43**	[7.33, 7.53]
Proactive	1	9.03	[8.95, 9.10]
PCoT	0	**8.49**	**[8.39, 8.59]**
PCoT	1	9.00	[8.93, 9.07]

Scores averaged over n = 500 items; CIs from a 5,000-sample bootstrap. Proactive 1-shot vs. PCoT 1-shot: Wilcoxon $p = 0.56$ (n.s.).

5.4 Discussions and Interpretation of Results

While the experimental results affirm that advanced prompting techniques such as PCoT are generally superior to standard, direct methods, they also validate the central hypothesis of this study: the optimal implementation of these techniques is highly task-dependent. The findings demonstrate that a one-size-fits-all approach is insufficient. Instead, achieving the desired model behavior for different proactive scenarios requires a nuanced and tailored application of prompting, particularly regarding the use of zero-shot versus few-shot learning.

The performance in the Clarification Dialogue Scenario provides the clearest evidence for this. For the classification-like task of Clarification Need Prediction (CNP), the PCoT Zero-Shot method was superior. This demonstrates that for tasks requiring logical inference, providing a structured reasoning framework is more effective than showing a single example, which proved to cause overfitting and a drastic loss of generalization. Conversely, for the structured generative task of Clarification Question Generation (CQG), the PCoT Few-Shot method excelled. Here, the demonstration was critical not for teaching the model what to do, but how to format the output cleanly, separating its internal reasoning from the final, concise question.

A qualitative analysis of the generated responses provides deeper insight into these quantitative results. In the clarification scenario, when faced with an incorrect student answer ("DFS is a sorting algorithm"), both few-shot (fs) and PCoT zero-shot (zs-pcot) methods successfully identified the ambiguity and produced clarification questions. However, their quality varied; the response from zs-pcot tended to be more analytical ("Provide more information or context..."), whereas the response from fs pcot cqg was more direct ("Are you sure DFS is used to sort data?"). It is this variation in relevance and structure that directly influenced the BLEU scores.

Similarly, in the Target-Guided Dialogue Scenario, the effectiveness of the PCoT method can be seen in its ability to integrate target concepts into its response. For instance, when given the target of discussing how a recursive function "helps solve

a smaller problem," the model generated the response: "...This allows the function to solve a problem by breaking it down into smaller, more manageable parts." This success in embedding key concepts from the target (e.g., "solve a problem," "smaller") is what theoretically leads to high scores on semantic similarity metrics like METEOR and ROUGE-L, even if the overall verbosity was penalized.

The simpler "Proactive" prompts ultimately outperformed the PCoT strategy in the targetguided scenario due to the nature of the evaluation metrics. The verbose, step-by-step reasoning inherent to PCoT's output was penalized by metrics like BLEU and ROUGE-L, which compare the generated text against a concise ground-truth reference. This highlights a crucial interaction between prompt design and evaluation methodology: a prompt that encourages explicit reasoning may be highly effective in practice but score poorly on standard similarity metrics.

In conclusion, the findings demonstrate that enhancing LLM proactivity requires a deliberate choice of strategy aligned with the specific task. For classification and structured generation, a Chain-of-Thought approach is highly effective, but the decision to use a zero-shot or few-shot implementation is critical. For more creative or open-ended generative tasks, simpler, more direct prompts may yield better performance when measured with traditional metrics. This framework provides a robust validation of how to strategically engineer prompts to build more effective and reliable AI-driven educational tools.

Comparison with related LLM oral-examination work. Our results are consistent with, and extend, the emerging literature on LLM-based assessment. Like the oral-examination and viva simulators of Nitze [23] and Church et al. [24], we find that a single state-of-the-art LLM can produce examiner-style turns without any fine-tuning. However, those systems treat the LLM mainly as a question generator or interrogator and do not isolate which prompting strategy elicits the desired proactive behaviour; our task-dependent comparison fills that gap. The reliability concerns documented by Silvestri et al. [25] for a GPT-4 surgical oral-board chatbot—hallucination and the risk of misleading examinees—reinforce why explicit, inspectable reasoning (as in PCoT) is attractive for high-stakes assessment, and why our human-judged comparison matters alongside lexical metrics. Whereas Chiu et al. [26] cast the LLM as the examinee in a viva benchmark and Ma et al. [27] grade spoken proficiency directly from audio, our work targets the complementary role of the LLM as a proactive examiner operating on transcribed text, and at a finer granularity (clarification-need prediction, clarification-question generation, and target bridging) than answer-level grading approaches such as Kortemeyer [28]. The principal merit of our approach is therefore methodological: a controlled, statistically tested protocol for selecting prompting strategies per proactive sub-task, rather than a single end-to-end demonstration.

6. Conclusion

This research demonstrates that eliciting specific proactive dialogues from LLMs requires tailored prompting strategies, as no single approach proved universally effective. To systematically investigate this, we successfully delivered on three primary contributions. First, we designed and implemented a **reproducible experimental**

framework to systematically compare various prompting techniques under both zero-shot and few-shot conditions. Second, we created **two novel, specialized datasets** for clarification and target-guided scenarios, which enabled our controlled experiments.

Finally, the **comparative analysis** conducted within this framework yielded several key findings. For the ambiguity detection task of Clarification Need Prediction (CNP), the PCoT zero-shot approach was the top performer, achieving a near-perfect F1-Score of 0.99. Surprisingly, its few-shot variant failed drastically (F1-Score 0.29), indicating that a single example caused significant overfitting. Conversely, for the generative task of asking the question (CQG), the PCoT few-shot approach was most effective, attaining the highest BLEU scores by leveraging the provided example. In the Target-Guided Dialogue scenario, a simpler Proactive prompt consistently performed strongly; the zero-shot variant excelled in lexical flexibility (BLEU), while the few-shot variant achieved the highest scores in semantic similarity (METEOR and ROUGE-L). These findings underscore the critical importance of prompt design and provide actionable insights for building more adaptive and effective AI-driven educational tools.

Despite these findings, it is important to acknowledge the limitations of this study, and we deliberately scope our claims accordingly. (*i*) Single model. All results were obtained with a single LLM, GPT-4o; we therefore frame our conclusions as holding for GPT-4o and do not claim they transfer to other model families without further testing. (*ii*) Single domain and language. The datasets cover only computer-engineering/programming-algorithms content and were authored and evaluated in Bahasa Indonesia, which may limit generalisability to other subjects and languages. (*iii*) Synthetic, templated data. Both datasets were LLM-generated and manually curated rather than collected from real examinations; their templated structure means the absolute scores—particularly the near-perfect CNP F1—are likely optimistic relative to naturalistic dialogue. (*iv*) Small evaluation panel. The human-judgement scores come from only two evaluators (one researcher and one non-researcher); with such a small panel no formal inter-rater agreement coefficient was computed, so the reported confidence intervals and significance tests quantify sampling uncertainty but not rater subjectivity. (*v*) Single deterministic run. Each prompt was issued once at **temperature** = 0; we did not study sensitivity to decoding randomness or prompt paraphrase. (*vi*) *Text-only*. The work is confined to the textual response-generation core and does not include speech components or real students. Future work should therefore validate these prompting strategies across multiple LLMs, subjects, and languages; replace synthetic items with naturalistic and human-annotated data scored by multiple raters with reported agreement; probe robustness across random seeds and prompt variants; and integrate the techniques into an end-to-end system with speech-to-text and text-to-speech capabilities, followed by user studies evaluating the effectiveness and naturalness of the automated oral exam in a real-world setting.

Appendix 1. Prompt Templates and Few-Shot Demonstration

For full reproducibility, this appendix lists the complete instruction component of every prompting technique used, for both scenarios, together with the single demonstration that realises the few-shot conditions. The experiments were run in Bahasa Indonesia; we give faithful English translations here and one verbatim Indonesian example for reference. Each technique is run zeroshot (instruction only) and few-shot (instruction prepended with the demonstration). Per-item data (**story/history/answer** for clarification; **dialog/target** for target-guided) is inserted programmatically before the instruction.

```
== CLARIFICATION ==
[Proactive | CNP] Given the document and the answer, the response should
  start with either "The answer is" or "The clarifying question is".
[PCoT | CNP] Given the document and the answer, think whether the answer is
  ambiguous or not. If it is ambiguous, ask a clarifying question; the
  response should start with "Therefore, the answer is ambiguous. The
  clarifying question is". If it is not ambiguous, answer the question.
[Proactive | CQG] Given the document and the conversation history, the answer
  is ambiguous. Please ask a clarifying question starting with "The
  clarifying question".
[PCoT | CQG] Given the document and the conversation history, the answer is
  ambiguous. The response should start with the reason why the answer is
  ambiguous, then follow by "Therefore, the answer is ambiguous. The
  clarifying question is".

== TARGET-GUIDED ==
[Standard] Given the answer history and the target answer, generate a response.
[Proactive] Given the answer history and the target answer, predict a suitable
  next topic that can bridge the current answer toward the target answer
  smoothly. Then generate an appropriate response. Reply by completing:
  "The next topic is []. The response is".
[PCoT] Given the answer history and the target answer, consider the relationship
  between the current and target answer, then predict a suitable bridging
  topic, then generate a response. Reply by completing: "The current answer is
  []. To bridge the current answer with the target answer, the next topic is
  []. Based on the predicted next topic, the response is".
```

(a) Instruction component of every prompting technique (English translation), issued verbatim under the zero-shot conditions.

```
Few-shot demonstration (clarification, PCoT), verbatim Bahasa Indonesia:
Dokumen: "Depth-First Search (DFS) adalah algoritma penelusuran graf ...".
Pertanyaan: "Apa itu DFS?" Jawaban: "DFS adalah algoritma pengurutan yang
menyusun data dalam urutan menaik." Silakan hasilkan respons: DFS bukanlah
algoritma pengurutan. Jawaban Anda salah. Pertanyaan klarifikasinya adalah:
Apakah Anda yakin DFS digunakan untuk mengurutkan data?

English translation: Document: "Depth-First Search (DFS) is a graph traversal
algorithm ...". Question: "What is DFS?" Answer: "DFS is a sorting algorithm
that arranges data in ascending order." Response: DFS is not a sorting
algorithm; your answer is incorrect. The clarifying question is: Are you sure
DFS is used to sort data?
```

(b) The single demonstration used in all few-shot (1-shot) conditions, prepended to the instruction above, shown in the original Indonesian and in English.

Figure 8. Prompt templates and the few-shot demonstration used in all experiments

References

[1] Paul Black and Dylan Wiliam. *Inside the Black Box: Raising Standards Through Classroom Assessment*. London, UK: King's College London, 1998. URL: https://www.rdc.udel.edu/wp-content/uploads/2016/06/Inside-the-Black-Box-C.A.-1998.pdf.

[2] Debby R. E. Cotton, Peter A. Cotton, and James R. Shipway. "Chatting and Cheating: Ensuring Academic Integrity in the Era of ChatGPT". In: *Innovations in Education and Teaching International* 61.2 (2024), pp. 228–239. DOI: 10.1080/14703297.2023.2190148.

[3] Enkelejda Kasneci et al. "ChatGPT for Good? On Opportunities and Challenges of Large Language Models for Education". In: *Learning and Individual Differences* 103 (2023), p. 102274. DOI: 10.1016/j.lindif.2023.102274.

[4] Luciano Floridi and Massimo Chiriatti. "GPT-3: Its Nature, Scope, Limits, and Consequences". In: *Minds and Machines* 30.4 (2020), pp. 681–694. DOI: 10.1007/s11023-020-09548-1.

[5] Sébastien Bubeck et al. "Sparks of Artificial General Intelligence: Early Experiments with GPT-4". In: *arXiv preprint* (2023). arXiv: 2303.12712 [cs.CL].

[6] ElNiak. *Understanding Prompt Engineering: A Comprehensive 2024 Survey*. Medium. May 2024. URL: https://medium.com/@elniak/understanding-prompt-engineering-a-comprehensive-2024-survey.

[7] Jules White et al. "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT". In: *arXiv preprint* (2023). arXiv: 2302.11382 [cs.SE].

[8] Pengfei Liu et al. "Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing". In: *ACM Computing Surveys* 55.9 (2023). DOI: 10.1145/3560815.

[9] Tom B. Brown et al. "Language Models are Few-Shot Learners". In: *arXiv preprint* (2020). arXiv: 2005.14165 [cs.CL].

[10] L. Feng, M. Hong, and C. J. Zhang. "Auto-Demo Prompting: Leveraging Generated Outputs as Demonstrations for Enhanced Batch Prompting". In: *arXiv preprint* (2024). arXiv: 2410.01724 [cs.CL].

[11] Y. Deng et al. "Prompting and Evaluating Large Language Models for Proactive Dialogues: Clarification, Target-Guided, and Non-Collaboration". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. 2023, pp. 10602–10621. arXiv: 2305.13626.

[12] M. Guo et al. "Abg-CoQA: Clarifying Ambiguity in Conversational Question Answering". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. 2023.

[13] Alessandro Sevegnani, David Vandyke, and David Schlangen. "OTTers: One-turn Topic Transitions for Open-Domain Dialogue". In: *Proceedings of the 21st Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 2020, pp. 153–162. DOI: 10.18653/v1/2020.sigdial-1.19.

[14] Learn Prompting. *Instructions*. https://learnprompting.org/docs/basics/instructions. Accessed: 2025-07-31. 2025.

[15] Jason Wei et al. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models". In: *arXiv preprint* (2022). arXiv: 2201.11903 [cs.CL].

[16] Y. Wang et al. "A Survey on Proactive Dialogue Systems: Problems, Methods, and Prospects". In: *arXiv preprint* (2023). arXiv: 2305.02750 [cs.CL].

[17] Ziwei Ji et al. "Survey of Hallucination in Natural Language Generation". In: *ACM Computing Surveys* 55.12 (2023). DOI: 10.1145/3571730.

[18] David Traum and James Allen. "A Speech Acts Based Approach to Dialogue Processing". In: *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*. 1992, pp. 203–214. DOI: 10.3115/981967.982000.

[19] Milica Gasic and Steve Young. "Gaussian Processes for Sentence-Level and Word-Level Confidence Estimation in Dialogue Systems". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24.2 (2016), pp. 241–253. DOI: 10.1109/TASLP.2015.2503721.

[20] Ankit Kumar and Pawan Gupta. "Survey on Clarification Question Generation". In: *arXiv preprint* (2020). arXiv: 2006.12608 [cs.CL].

[21] James F. Allen, George Ferguson, and Amanda Stent. "Towards a Robust Mixed-Initiative Dialogue System". In: *Proceedings of the ACL-01 Workshop on Conversational Systems*. 2001, pp. 1–8.

[22] J. Zhu et al. "A Survey on Large Language Models in Dialogue Systems". In: *arXiv preprint* (2023). arXiv: 2307.13501 [cs.CL].

[23] A. Nitze. "Future-Proofing Education: A Prototype for Simulating Oral Examinations Using Large Language Models". In: *arXiv preprint* (2024). arXiv: 2401.06160 [cs.AI].

[24] I. M. Church, L. Drake, and M. Harris. "Using LLMs to Support Assessment of Student Work in Higher Education: A Viva Voce Simulator". In: *arXiv preprint* (2025). arXiv: 2511.05530 [cs.CY].

[25] C. Silvestri et al. "Evaluation of a Novel Large Language Model (LLM)-Powered Chatbot for Oral-Boards Scenarios". In: *Global Surgical Education – Journal of the Association for Surgical Education* 3 (2024). DOI: 10.1007/s44186-024-00303-z.

[26] C. Chiu, S. Pitis, and M. van der Schaar. "Simulating Viva Voce Examinations to Evaluate Clinical Reasoning in Large Language Models". In: *arXiv preprint* (2025). arXiv: 2510.10278 [cs.CL].

[27] R. Ma et al. "Assessment of L2 Oral Proficiency Using Speech Large Language Models". In: *arXiv preprint* (2025). arXiv: 2505.21148 [cs.CL].

[28] Gerd Kortemeyer. "Performance of the Pre-Trained Large Language Model GPT-4 on Automated Short Answer Grading". In: *arXiv preprint* (2023). arXiv: 2309.09338 [cs.AI].

[29] Jianheng Tang et al. "Target-Guided Open-Domain Conversation". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2019, pp. 5624–5634. DOI: 10.18653/v1/P19-1568.

[30] OpenAI. *OpenAI API*. https://openai.com. Accessed: 2025-07-31. 2024.

[31] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. Sebastopol, CA: O'Reilly Media, 2009. ISBN: 9780596516499.

[32] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[33] The chardet developers. *chardet: The Universal Character Encoding Detector (Python)*. https://github.com/chardet/chardet. Accessed: 2025-07-31. 2024.

[34] Benjamin S. Bloom, ed. *Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain*. New York: David McKay Company, 1956.

[35] Kishore Papineni et al. "BLEU: A Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2002, pp. 311–318. DOI: 10.3115/1073083.1073135.

[36] Matt Post. "A Call for Clarity in Reporting BLEU Scores". In: *Proceedings of the Third Conference on Machine Translation: Research Papers (WMT)*. 2018, pp. 186–191. DOI: 10.18653/v1/W18-6319.

[37] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*. 2004, pp. 74–81.

[38] Satanjeev Banerjee and Alon Lavie. "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments". In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. 2005, pp. 65–72.