ASSIGNMENT 2: PROMPT ENGINEERING & AGENTS

Minghao Wu, Shunian Chen, Benyou Wang*
The teaching team of Natural Language Processing
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
minghaowu@link.cuhk.edu.cn, wangbenyou@cuhk.edu.cn
Fall 2026 (AY 2026–27, Term 1)

1 INTRODUCTION

Prompt engineering involves developing and optimizing prompts to use language models (LMs) effectively across a wide variety of applications and research topics. These skills help us understand the capabilities and limitations of large language models (LLMs). Researchers use prompt engineering to improve LLM performance on tasks such as question answering and arithmetic reasoning. Developers use it to design prompting techniques that connect LLMs with users and other tools.

Prompt engineering also supports the development of AI agents: systems that use an LLM to make decisions and take actions toward a goal. Carefully designed prompts can guide agents through multi-step tasks, help them evaluate intermediate results, and support interactions with digital tools. In this assignment, you will explore both prompting techniques and agent-based approaches.

2 PRELIMINARY

We have prepared a Google Colab notebook to help you get started:

https://colab.research.google.com/drive/1n_2uDVFr_sYMAlhw0mjHMWCiC1NT8l8L?usp=sharing

The notebook covers:

- How to call the DeepSeek API.
- Useful prompt engineering techniques.
- How to get started with Task 1 and Task 2.

We strongly recommend reviewing the notebook before starting your experiments.

3 TASKS

Complete **one and only one** of the four tasks below. Carefully refine your prompts to improve task performance. You must implement and compare **at least three prompting techniques** introduced in the tutorial or in the Prompt Engineering Guide (https://www.promptingguide.ai/techniques).

In addition, include a comparative analysis of **non-agent and agent-based approaches**. Evaluate how an LLM agent performs relative to conventional prompting using appropriate measures such as accuracy, completion time, reasoning quality, and the ability to handle task complexity. Describe the agent's workflow and how it uses intermediate observations or feedback to decide its next action. Use the same evaluation data for comparisons and report the model and experimental settings.

*Benyou Wang is the instructor.

3.1 TASK 1: LLMS AS A KNOWLEDGEABLE DOCTOR

The pharmacist licensure exam assesses the knowledge and skills needed for professional pharmacy practice. LLMs such as DeepSeek can assist with exam preparation by explaining concepts and answering questions. However, poorly designed prompts may lead to inaccurate or incomplete answers. In this task, investigate how prompting and agent-based approaches affect performance on pharmacist examination questions.

Input: Multiple-choice questions.
Output: The selected answer for each question.
Example: See Figure 1.
Criterion: Answer accuracy.

Code and data:

- 100 questions sampled from the 2021 National Pharmacist Professional Qualification Examination.
- Task 1 code and data in the course repository.

An example prompt for multiple-choice questions

Prompt:
你是一个药剂师考试能手，每次都考100分，这道题对你来说不在话下，深呼吸，并一步一步思考，并给出正确的答案。回答格式为答案是A、B、C或者D。
【最佳选择题】根据健康中国战略，推进健康中国建设主要遵循的原则不包括哪个选项?
A: “健康优先”
B: “改革创新”
C: “科学发展”
D: “公开透明”

Expected output:
答案是D。根据健康中国战略，推进健康中国建设主要遵循的原则不包括“公开透明”。

Figure 1: An example prompt and answer for Task 1.

3.2 TASK 2: LLMS FOR AI FEEDBACK

Language models can be trained using preference feedback from human annotators (Ouyang et al., 2022) or AI systems (Bai et al., 2022). Such feedback can be represented as a triplet consisting of a question, a *chosen answer*, and a *rejected answer*, and can be used to train a reward model.

Use DeepSeek to provide preference feedback by choosing the better of two candidate answers. Preferences can be biased by answer order, so **shuffle the order of the answers** and map the resulting choices back to the original answer identities when evaluating performance.

Input: A question Q and two answers A_1 and A_2.
Output: The chosen answer and the rejected answer.
Example: See Figure 2.
Criterion: Agreement with the ground-truth preference labels.

Code and data:

- 100 answer pairs sampled from the Anthropic HH-RLHF dataset on Hugging Face.
- Task 2 code and data in the course repository.

An example prompt for AI feedback

Prompt:
Given a question and two answers, select the answer that better addresses the question and briefly explain your choice.
Question: Do you know why turkeys became the official food of Thanksgiving?
Answer 1: To be honest, I don't know anything about that. I know that I'm meant to know a lot about history and current events, but I haven't been programmed with those particular facts, sorry.
Answer 2: I know that the American Indians were the first to celebrate the first Thanksgiving, and the first European settlers came to New England in the early 1600s, and it is likely that they celebrated the first Thanksgiving in the late 1600s. However, it is also likely that some European settlers on the continent celebrated the first Thanksgiving in the mid-1500s. A lot of people think that the main factor in Thanksgiving is that the settlers in the new world were thankful for finding a new land, and that turkey was their traditional food. Another factor that has often been thought to be important is the Pilgrims' relationship with the Native Americans, and the feast may be one way of trying to show them respect. I can't tell you the definitive answer, but maybe this will help you figure it out?

Illustrative preference output:
Chosen: Answer 2. Rejected: Answer 1. Answer 2 attempts to address the question, whereas Answer 1 provides no substantive information.
This example illustrates a preference decision, not verified historical facts. A longer answer can still contain factual errors; discuss such limitations when analyzing AI feedback.

Figure 2: An example preference comparison for Task 2, adapted from the previous assignment.

3.3 Task 3: LLMs as a Data Quality Evaluator

Data quality is central to language model development: "garbage in, garbage out." Use an LLM to evaluate data quality and select better-quality examples for training.

Input: A dataset D.
Output: A cleaner or higher-quality dataset D'.
Criteria: Define and justify your own quality criteria.

Data examples: You may filter one of the following datasets from Peng et al. (2023):

- English Alpaca (GPT-4-generated instruction data).
- Chinese Alpaca (GPT-4-generated instruction data).

3.4 Task 4: Any Other Topic

You may choose another topic that allows you to investigate prompt engineering. The same requirements apply: compare at least three prompting techniques and include a comparison between non-agent and agent-based approaches. Clearly define the task, data, and evaluation criteria.

4 Submission Format

Submit a short report using the Overleaf report template. You may adjust the report format as needed; keep the report concise and informative. Your report should cover:

- Code used to call the DeepSeek API, if applicable.
- Your prompts and the rationale behind their design.
- How the agent is used to solve the selected task.
- Representative cases (e.g., two examples).
- Experimental results and comparisons, including the required prompting and agent comparisons.
- Insights learned (optional).

Submit the report as a **PDF via Blackboard**. Code may be included as supplementary material.

Deadline: 11:59 p.m., November 6, 2026 (China Standard Time, UTC+8).

5 Evaluation Criteria

The assignment is graded as follows:

- **[25 marks] Quality of prompt engineering:** The quality of your prompt designs, their task-specific rationale, and the refinements supported by your experiments.
- **[75 marks] Quality of the report:**
  - **25 marks** for clarity and organization.
  - **25 marks** for experimental results, including completeness, experimental rigor, and technical insight.
  - **25 marks** for experimental analysis, including case studies, sensitivity to individual components, ablation studies, and other relevant discussion.
- **[Up to 5 bonus marks] Significance of an original task:** A well-motivated research or engineering task beyond Tasks 1–3 may receive a bonus. Choosing a new topic alone does not guarantee bonus marks.
- **[Up to 5 bonus marks] Code quality:** Clean, clear, and well-documented code may receive a bonus. Attaching code alone does not guarantee bonus marks.

The final score, including any bonus marks, is **capped at 100**.

Acknowledgment

This is Assignment 2 for CSC6052/CSC5051/MDS5110 in Fall 2026 (AY 2026–27, Term 1). For course information, see https://nlp-course-cuhksz.github.io/.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with GPT-4. *arXiv preprint arXiv:2304.03277*, 2023. URL https://arxiv.org/abs/2304.03277.