Carnegie Mellon University

17-630: Prompt Engineering

Class Time: Tuesday and Thursday, 2:00-3:20
Location: 3SC 265
Semester: Spring 2025, 12 units

Instructor

Prof. Travis Breaux
Office Hours: By Appointment
Email: tdbreaux@andrew.cmu.edu

Teaching Assistant(s)

Venu Arvind Arangarajan
Office Hours: TBD
Email: varangar@andrew.cmu.edu

Neel Bhandari Bhandari
Office Hours: TBD
Email: neelbhan@andrew.cmu.edu

Harivallabha Rangarajan
Office Hours: TBD
Email: hrangara@andrew.cmu.edu

Course Introduction. Advances in large language models have created opportunities to adapt natural language processing tasks to new domains without the computational labor of fine-tuning models on tens of thousands of training examples. Instead, users can write prompts with instructions, demonstrations and other context to facilitate in-context learning using a model with frozen parameters.

In this course, prompt engineering is recast from only "how to write prompts" to how to construct reliable prompt-based systems by choosing techniques from a emerging scientific foundation. Students begin learning a brief history of large language models (LLM), as well as contemporary approaches to LLM design and development, such as instruction-tuning, alignment, and calibration. Under alignment, students will study sources and manifestations of hallucinations, bias, and sycophancy, in addition to topics within super-alignment, such as deceptive, immoral and unethical LLM behavior.

In addition, students learn contemporary prompt engineering strategies and techniques, such as chain-of-thought, retrieval augmented generation (RAG), tool usage, various ways to self-prompt for response verification and consistency. Within the repertoire of multi-stage prompting strategies, students will also study agentic frameworks, covering the use of personas, memory models, and the benefits of agent-based debate and collaboration.

The course covers standard prompt engineering benchmarks and metrics to evaluate the efficacy of prompt designs and explores emerging practices for metric design and development.

Finally, students will practice using cloud-based language models to complete coursework. Various options exist, including OpenAI's GPT, Google's Gemini, and Anthropic's Claude.

Learning Objectives. After completing this course, you will be able to:

- Intuit design of prompt instructions and templates to shape answers
- Decompose complex inference tasks into multi-stage prompts
- Design experiments to evaluate prompt designs and optimize prompt performance

Assessments. Students learn more by applying and explaining ideas to others, thus, the course requires the following activities:

- **Homework assignments,** or individual work to help students focus on important points in the readings and to exercise particular skills
- **Project,** or group work to allow students to construct larger, more complex systems by combining multiple prompting strategies and techniques
- **Quizzes** to check your learning and reinforce key concepts
- **Final Exam,** to demonstrate your cumulative knowledge on practical examples. The final exam will be a take-home, open book, open notes exam that is due one week after the final day of class.
- **Class participation,** to enrich the discussion with your insight, relevant experience, critical questions, and analysis of the material. The quality of contribution is more important than the quantity. Students may participate using synchronous and asynchronous modalities.

Assessment	**Final Grade %**
Individual Assignments	30%
Project	30%
Quizzes	15%
Final exam	15%
Class participation	10%

Homework Assignments. The course includes three homework assignments as follows.

1. An in-context learning assignment to illustrate efficacy of instruction authoring, template usage and demonstration practices on a text classification task.

2. A tool usage assignment to call functions that contrasts single-stage prompting with multi-stage task decomposition.
3. A retrieval augmented generation assignment to evaluate content-ingest strategies and retrievers using question-answering and text summarization.

In addition to the above homework assignments, students will complete three in-class recitation projects.

Course Project. At mid-semester, students propose an independent or group project for the remainder of the course. Students are expected to provide biweekly check-ins to report progress and troubleshoot issues as their projects progress.

Required Readings. The concepts and ideas taught in this class are largely drawn from emerging research, some of which are only available in pre-publication format. Students will be directed to read papers curated from over 150 foundational papers published in NeurIPS, ICLR, ICML, and ACL, among others.

Lectures will cover the readings in-depth with examples drawn from papers and from other works cited by the readings. Reading ahead allows students to engage in richer discussions during the course and thus this is *highly encouraged* to get the most out of class time.

Resources. Students will be provided $50 in credit at OpenAI for using the latest GPT and GPT Turbo models to complete coursework.

Doctoral Students. PhD students who wish to apply this course toward their PhD requirements should consult with their program director and/or advisor. They may wish to enroll in 17-730, wherein students are expected to (1) attempt the "reach goals" included in assignments and (2) to choose a course project relevant to their research interests to increase their depth of engagement with course material.

Late Coursework Policy. Students are expected to turn-in assignments before the due date. To help students balance their busy schedules and multiple due dates across courses, this course provides each student a total of five late days that can be spent at any time during the course and at the discretion of the student. Students must notify both the professor and teaching assistants before the due date of their intent to use one or more of their late days.