MIE 451-1513 Decision Support Systems
Lab and Assignment 5: LLM Prompt Engineering for Natural Language Tasks

Due Date: Posted in Syllabus

This lab and assignment involves designing an experiment to evaluate various prompt styles for a pretrained large language model (LLM) on a natural language (NL) task. Specifically, you will need to:

1. Select and sign-up for a NL task.
2. Select an evaluation metric and curate a small dataset for your task.
3. Design several prompt styles for your task.
4. Evaluate LLM performance with your prompts on your data.

Marking scheme and requirements:
Please submit a single PDF report that includes the required tables and written answers for each question. Do not exceed 8 pages for the main report, and use at least 11 pt font for body text and 9 pt font for tables. Short, precisely worded reports are encouraged.

There are **7 points and 5 questions** in this assignment. **There is no code review.**

You must curate your own data for experiments — **sharing data between classmates is not permitted**. Please note the **plagiarism policy** in the syllabus that will be **strictly enforced**.

What/how to submit your work:

Submit your PDF report to Github by the deadline (timestamps after the deadline will be considered late). If you write any code (not required), push it to Github as a Jupyter notebook.

1 Before and in the Introductory lab

1.1 LLM Setup

Scale is critical for LLM performance. The top LLMs are currently OpenAI's GPT 3.5/4 which are closed-source and require over 300 GB of RAM to host. However, the performance of these models by far exceeds that of any model you would be able to host on 12 GB of Colab RAM — see the end of Lab 1 for an example of this performance difference.

For this reason, we will be working with OpenAI models. You have two options to complete the lab and assignment: 1) using the ChatGPT web application, or 2) using the OpenAI API. While the API may require a $5 payment (see Option 2 below), you should be able to sign up for ChatGPT for free.

We will not discriminate between these two options for marking: students who choose to use the free web applications are just as likely to get full marks as students who use the API. You are not graded on the performance of the LLM — rather, you are graded on your experimental design and prompt design. For this reason, paying for access to better LLMs (e.g. GPT 4) will not give you any advantage.

Option 1: ChatGPT Web Application Sign up for a **free account** here.

Option 2: OpenAI API **[Not required]** Sign up for API access here.

- After signing up, you will need to navigate to API Keys in the left ribbon to obtain an API key. **Never share API keys with anyone** — if your payment information is associated with your key, anyone with your key can charge you. If you suspect your API key has been leaked, disable it immediately in your account and create a new one.
- You will receive a free trial with $5 of API credits if you have not had an OpenAI/ChatGPT account for over three months. Otherwise, you will need to purchase $5 of credits for your API key to work.
- Unfortunately, we will not be able to reimburse you for any API costs should you choose this option. However, the expected API cost for completing this assignment with `gpt-3.5-turbo-1106` is well below $5, with the longest prompt in the lab ($\sim$ 1400 words) costing roughly $0.002.

1.2 In the Lab

The lab demonstrates several prompting styles including:

- Zero-shot (ZS)
- Instruction engineering
- Few-shot (FS)
- Negation handling
- Retrieval-augmented generation (RAG)
- Chain-of-thought (COT)

It also provides links to more references on prompt engineering.

2 Main Assignment

The overall objectives of the assignment are to (1) choose a challenging NL task for evaluation of large language models, (2) justify and implement an appropriate evaluation metric for your chosen NL task, (3) curate labeled data relevant to your task, (4) evaluate performance differences with different styles of prompt engineering, and (5) document and report your results.

NL Task Sign-Up

Eight broad NL tasks for decision support are described in Section 3. You will design a small experiment to test various prompt styles on one of these tasks.

While your experiment should address the broad task description, you are allowed to define a specific subtask you are interested in exploring. Several example subtasks are given for each task — you can choose to focus on one of these example subtasks or define your own. You are allowed to reuse code from past assignments but you must curate new data for your task.

Sign-up for a task using your full name here.

The maximum number of students per task is **15**. The sign-up is first-come, first-served — if you delete other students' names, you will receive a zero. (We maintain a history of this form.)

Q1 (1.5 Points). Task Definition and Motivation

Describe the following three components (a,b,c) with no more than 5 sentences each.

(a) Task Description: Given the broad description for your task in Section 3, define a specific subtask you have chosen to study (referred to as "your task" for the rest of Section 2). Clearly identify the inputs and outputs from the prompting portion of your task.

(b) Evaluation Metric: Select, describe, and justify the evaluation metric you will use to measure performance on your task. Human judgement is an acceptable metric (e.g., to determine whether text became more formal or not), but you must use **your judgement and that of at least one other judge**. Note that each person would need to manually judge each test example against each prompt style ($\sim$ 30 judgments).

Inter-rater or LLM Reliability Assessment:

- **If you choose to use human judgment:** You must define a **clear instruction rubric** for the evaluation of your metric (this rubric does not count as part of the answer sentence limit). Further, it is critical to determine whether human judgment is a reliable metric across judges, hence you must also report the **contingency table and Kappa score** between yourself and another judge on a **minimum of 10 examples** when only provided with your rubric as a basis for judgment.
- **If you decide to automate evaluation with the GPT API or another LLM:** You must report the **evaluation prompt** you used (this does not count as part of the answer sentence limit) as well as the **contingency table and Kappa score** assessment of agreement between your judgments and the LLM judgments on a **minimum of 10 examples**. You must make judgments according to the instruction prompt provided to the LLM.

(c) Motivation: Describe why the task you selected is important as one component of a system used for Decision Support, including at least two real-world examples.

Q2 (1.5 Points). Data Curation

Curate 10 test examples for your task that should include (a) an input and (b) an example of the desired output (from the prompting portion of your task). Sometimes the desired output is

a ground truth label or answer when there is one correct response (e.g., classification or factoid question answering), while in open-ended tasks there may be one or more examples of a "good" output (e.g., text summarization). Discuss any questions with the TA during lab or on Piazza. You can make up your own data and/or obtain data online. **Remember to cite any sources.**

(a) Example listing: Include 5 test examples in a table (with clearly labeled input and target output columns) in the main report, and the remaining 5 examples in a table in an appendix (the appendix does not count towards the page limit).

(b) Process Description: Briefly describe your data curation process.

Q3 (1.5 Points). Prompt Styles

Design 3 prompt styles for your task (you are welcome to design and experiment with more prompt styles, but 3 are sufficient to achieve full credit). These styles should include:

1. A basic zero-shot prompt as a baseline. You can keep this prompt minimal (e.g., no instruction engineering) to see if prompt engineering techniques give improvements over this minimal zero-shot baseline.
2. A few-shot prompt.

For the remaining prompt style (minimum of 1 additional prompt in addition to the 2 above), explore the other techniques demonstrated in the lab.

(a) Prompt Documentation and Description: In a cleanly formatted answer (e.g., in a table or clearly labeled sections), show each of your prompt templates in full and briefly describe each technique. If a prompting style involves multiple steps (e.g. chain-of-thought), include a template for each step. Since your input data will change for each test example, clearly label where the input text should go in the prompt.

Q4 (1.5 Points). Results and Discussion

Use your evaluation metric to evaluate each of your prompt styles on each test example. This would amount to 30 evaluations for 3 prompt styles.

(a) Aggregated results In a table, report results aggregated across all 10 test examples for each prompt style. Briefly discuss any differences in the performance of the prompt styles.

(b) Examples of Success and Failure In another table, for one of the prompt styles, show three test examples for which the LLM did well and three examples for which the LLM did poorly. Hypothesize why the LLM performed well for the successful examples and poorly for the unsuccessful examples.

If you do not have 3 examples of failure, add three new test examples that force the LLM to fail. Some ideas to make the LLM fail are: corrupting syntax and grammar, adding irrelevant input information, and adding out-of-distribution (OOD) data (e.g., made-up product names).

If you do not have 3 examples of success, your task/data is too difficult or your prompts/data are low-quality. Go back to Q1-Q3 until you have at least 3 successful examples.

Q5 (1 Points). Limitations

(a) Deployment Issues: Think of **two deployment issues** that could occur when using an LLM for your task. Discuss what could go wrong and how these issues might be handled in practice. You can cite blogs or papers if they help you answer this question. Use at most 5 sentences for each issue.

3 NL Task Descriptions

1. **Recommendation:** Recommend relevant items for a user. Example subtasks:
   - Recommend items given their NL descriptions and a user's item interaction history, such as:
     - The user's past item ratings.
     - A sequence of items recently liked by the user.
   - Recommend items given their reviews and a NL query from a user.
   - Summarize a user's interests given a set or sequence of recently liked books or movies.
2. **Summarization:** Compressing information into a shorter form while preserving important content. Hint: two common summary evaluation metrics are ROUGE and BERTScore (but you do not have to use these metrics). Example subtasks:
   - Generate paragraph and/or bullet point summaries of a document or a set of documents.
   - Extract the key topics in a long document as a list or hierarchy.
   - Given a debate containing multiple perspectives (e.g., Fox News vs. CNN articles on the same topic), extract key issues and perspectives and/or generate opinion summaries.
   - Summarize reviews (e.g., hotels on Tripadvisor, products on Amazon, restaurants on Yelp). While you get to define the task you want to perform, here are some possibilities:
     - The summary can be be over a single item or a summary overview of multiple items.
     - A summary of single or multiple items can be aspect-oriented (e.g., location, price, staff helpfulness).
     - Contrastive summaries can focus on the key differences between items.
     - The summary can be general or targeted w.r.t. an NL query (e.g., "what do people like or dislike about an item"; "what are popular menu items", etc.).
     - The summary can be extractive or abstractive (you can Google for these terms). The advantage of an extractive summarization is that you can cite your sources.
3. **Information extraction:** Generate a JSON file which structures information from an interaction (e.g., an email chain) and/or set of documents. The JSON format is essentially a multi-level dictionary such as:

```
{
    "name": "Emily Yang",
    "interests": ["biking", "cooking", "data science"],
    "address": {
        "street": "1 Yonge Street",
        "apartement": "not provided"
```

```
        "city": "Toronto",
        "province": "ON",
        "postal code": "1A1 A1A"
    }
    ...
}
```

Example subtasks:

- Given an email chain, extract meeting information including meeting date, time, participants, location, and agenda.
- Given a recommendation dialogue (see this paper for examples), extract user preferences (both soft constraints and hard constraints), previously recommended items, and user feedback on recommended items (e.g., "accept"/"reject").
- Extract information in the domains of: financial documents, resumes and job postings, scientific papers, legal documents, medical records.
- Extract entities (e.g., people, places, institutions) and relationships (e.g., "visited", "spoke to") from news articles and represent them as triples, such as (person: "Biden", action: "spoke to", person: "Trudeau").

4. **Query reformulation/augmentation:** Reformulate (rephrase) or augment (expand) a query to improve information retrieval performance (improving recall is a good focus for this task, but other metrics will work as well). You can reuse your IR system and corpus from Assignment 1; however, even if your end task and evaluation is a standard IR metric on the existing queries, you should still curate examples of good query reformulations. Example subtasks:

   - Without using additional information, rephrase a query (e.g., add synonyms, modify phrasing) to improve recall-oriented metrics.
   - Infer an underlying information need that is not directly expressed in the query (see lab 1 for examples of information need) and augment the query to improve recall.
   - Use pseudo-relevance feedback methods (Google for this term) to perform query expansion or reformulation.

5. **Retrieval-augmented question answering (QA):** Answer a question by performing a search and using the results to generate an answer. You can reuse your IR system from Assignment 1 or use Internet search (manually storing and inserting the top-k results for a query into your prompt template). Example subtasks:

   - Given a question that requires knowledge which was not in the training data of an LLM, e.g., "What was the interest rate decision announced by the Federal Reserve in November 2023?", use the results of a search to generate answers.
   - Post-hoc source attribution: given a retrieval-augmented LLM generated answer, attribute parts of the answer to spans in the retrieved documents which support it (see this paper for examples).

6. **Retrieval-augmented summarization:** Given an information need (e.g., a query), perform a search and generate a summary of the results. Check the summarization task above for further hints on task possibilities and evaluation metrics. Example subtasks:

   - Generate paragraph or bullet point summaries of the top retrieved document or documents.
   - Attribute parts of a generated summary to passages in the retrieved documents (see this paper for examples).
   - Decompose a complex query into multiple queries and aggregate the results into a single summary.
   - Determine whether the top-retrieved documents can provide the answer to a query or whether the system should respond "I don't know".

7. **Natural language inference (NLI) [Bowman et al.(2015)]:** Given two statements, a premise and a hypothesis, infer whether the releationship between the statements is entailement, contradiction, or neutral, as defined by:

   - Entailment: If the premise is true, the hypothesis is also true, e.g.:
     - Premise: "A soccer game with multiple children playing."
     - Hypothesis: "Some kids are playing a sport."
   - Contradiction: If the premise is true, the hypothesis is false, e.g.:
     - Premise: "A man inspects the uniform of a figure in some East Asian country."
     - Hypothesis: "The man is sleeping."
   - Neutral: The truth of the premise does not imply the truth of the hypothesis, e.g.:
     - Premise: "An older and younger woman are smiling."
     - Hypothesis: "Two women are smiling and laughing at the cats playing on the floor."

   Example subtasks:

   - Given a specific item description (premise) and a generic item feature (hypothesis), such as "2020 Pinot Noir Sandbanks Estates" and "red wine", respectively, infer whether the specific item description entails, contradicts, or is neutral to the generic item feature. In the example above, the relationship should be entailment because Pinot Noir is a type of red wine.
   - Perform NLI that requires knowledge of a specific domain such as geography or politics. For example "The President of the US attended the 47th G7 summit." should entail "Joe Biden was in Cornwall, UK in June 2021".
   - Determine whether an item description in a recommendation system entails an NL query, e.g., if I query for a "a great place to have a meal and watch FIFA with friends" and the item options are (a) "a sports bar with TVs and a variety of food options" and (b) "a television store with TVs tuned to every channel" then (a) entails the query while (b) does not.
   - Perform NLI that requires negated, temporal, and/or commonsense reasoning — see Table 2 in the RecipeMPR dataset [Zhang et al.(2023)] for examples of these reasoning types.

8. **Writing assistance** Modify or improve text written by a user in terms of aspects such as style, syntax, or clarity. Example subtasks:

    - Identify unclear portions of text that could be rewritten and propose clarifications.
    - Change text style:
        - Formal vs. informal
        - For a 5-year-old vs. for a college graduate
        - Detoxify text
        - Make text florid (i.e., elaborate and ornate), e.g., for marketing text.
    - Identify syntax errors and propose corrections.

References

[Bowman et al.(2015)] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326* (2015).

[Zhang et al.(2023)] Haochen Zhang, Anton Korikov, Parsa Farinneya, Mohammad Mahdi Abdollah Pour, Manasa Bharadwaj, Ali Pesaranghader, Xi Yu Huang, Yi Xin Lok, Zhaoqi Wang, Nathan Jones, et al. 2023. Recipe-MPR: A Test Collection for Evaluating Multi-aspect Preference-based Natural Language Retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2744–2753.