17-630 Prompt Engineering

Assignment: Question Answering

This assignment introduces students to simple prompting with a question-answering example. Question answering aims to extract entities from one or more texts. To illustrate question answering, students will craft a text from a random Wikipedia article and then write five questions that can be answered by their own reading of the text, in addition to five questions that cannot be answered from the text. Finally, they will write a prompt to a language model to instruct the model to answer each of the 10 questions.

Optional Readings

- Rajpurkar et al., “Know What You Don’t Know: Unanswerable Questions for SQuAD,” ACL pp. 784-789, 2018.

Learning Objectives

- Setup and configure a prompting environment with a large language model.
- Design a basic prompt using answerable and unanswerable questions.

Assignment Deliverables

This is an individual assignment. To complete this assignment, the student should perform the following steps: (1) visit Wikipedia.com and select “Random article” from the navigation menu. (2) Choose a paragraph 700-750 characters in length, and (3) Write five *answerable questions* that can be answered by the text, and five plausible but unanswerable questions that cannot be answered by the text. Include answers to each of your questions. (4) For each question/answer pair, prompt the LLM by providing a single instruction that asks the LLM to answer the question using the provided Wikipedia paragraph. (5) Evaluate the LLM’s responses by comparing the responses with your expected answers using Exact Match (EM) and F1 scores. (6) Re-run your experiments using the following models: gpt-4o, gpt-4o-mini, and gpt-3.5-turbo

Submit a single ZIP archive with the following files and exact filenames:

- notebook.ipynb – your Jupyter Notebook containing your code used to conduct your experiment. Please document major sections of your notebook.
- squad_questions.json – your paragraph and questions that you developed in the SQuAD v2 format
- responses.json – the list of responses from your LM to your prompt.
- questions.txt – Answers to the assignment questions provided in the package.

Further Guidance

- **Important**! When getting started, use the cheapest model to test your experimental setup. Only test a small number of samples to evaluate your code in an end-to-end experiment before running your experiment on the entire dataset.

- If you want to test an out-of-distribution article, instead of choosing a Random Article, choose an article that describes an event that occurred after the model's training cut-off date. Verify the article did not exist in the pre-training by checking the revision history of the article.
- The assignment package includes a notebook.ipynb file to get you started, as well as the datasets for SQuAD v2. Divide the work into three parts: (a) creating the question data structure, (b) prompting the LLM, and (c) evaluating the responses using Exact Match and F1-Score (see next bullet). Create an end-to-end solution before manually tuning the prompt.
- Exact Match (EM) = 1, if every word in the predicted answer matches every word in the expected answer, else EM = 0. To calculate the F1 score, students must count every word in the predicted answer that matches a word in the expected answer as a true positive (TP). Any word in the expected answer that is missing from the predicted answer is counted as a false negative (FN), and every word in the predicted answer that does not appear in the expected answer is a false positive (FP). The F1 score = 2TP / (2TP + FP + FN). Students can run totals of TP, FP and FN over all responses in their experiment.
- If the LLM is instruction-tuned, students may want to instruct the model to only provide the answer and not add additional commentary or elaboration (e.g., "Don't comment or elaborate.") Students could also experiment with JSON output to assist in parsing the result.
- Each question may have only one answer from the paragraph, or it may have more than one answer. The answers should be populated from exact substrings contained within the paragraph, and your data structure should include for each answer the starting character position of the answer in the text. If the question is unanswerable, the answer list in the JSON file should be empty.
- The data structure in squad_questions.json file is the SQuAD v2 format and is only used to replicate experiments by ensuring all data is readable from the same format. Avoid prompting the LLM to answer the question using this data structure, particularly, the starting indices of the answers in the text. Students can easily calculate this number, whereas LLMs are prone to hallucinate the number.

Example:

```
Transistor

The essential usefulness of a transistor comes from its ability to use
a small signal applied between one pair of its terminals to control a
much larger signal at another pair of terminals. This property is
called gain. It can produce a stronger output signal, a voltage or
current, which is proportional to a weaker input signal; that is, it
can act as an amplifier. Alternatively, the transistor can be used to
turn current on or off in a circuit as an electrically controlled
switch, where the amount of current is determined by other circuit
elements.

(A) Why is a transistor so useful?
(A) What is gain?
```

```
(A) What is an additional use of the transistor?
(A) What determines the amount of current in an electrically controlled switch?
(U) What controls how strong the output signal is?
(U) Where are transistors located?
(U) How large is a transistor in comparison to a terminal?
(U) What does a signal become when the transistor does not work properly?
(U) Where would an electrically controlled switch be located?
```

JSON Format for squad_questions.json:

```
{
  "context": "The essential usefulness...", # the paragraph
  "qas": [
    {
      "question": "Why is a transistor...". # first question
      "answers": [
        {"text": "its ability to use..."}, ... # answer list
      ]
      "is_impossible": false # true, if unanswerable
    }, ...
     {

      "question": "What controls how strong...".
      "answers": [],
      ]
      "is_impossible": true
    }, ...
  ]
}
```

JSON Format for responses.json:

The responses file should contain a JSON array of 10 elements, where each element is the i^{th} response from the LM to the i^{th} question in the squad_questions.json file. These responses are the answers provided by your LM to your prompt.

Evaluation Criteria

- Completeness of the paragraph and ten questions and answers.
- Thoughtfulness of the answers to the questions in questions.txt, including examples from your work to illustrate your answers.