17-630 Prompt Engineering

Assignment: In-Context Learning

This assignment introduces students to In-Context Learning (ICL) by tuning the model at inference time using demonstrations. Unlike traditional fine-tuning, which can require 10^4 training samples, ICL can be used to train a large language model with 10^1 training samples. Still, evidence shows that the number of demonstrations affects accuracy (Brown et al., 2020), as well as which demonstrations are selected and the order in which they are presented (Lu et al., 2022; Zhao et al., 2023).

In this assignment, students explore ways to categorize customer reviews to improve product comparisons. Customer reviews reflect the unique perspectives and needs of their authors. While language models can effectively generate a small number of labels per review, the number of unique labels is linear with the number of reviews. This scaling factor prohibits product comparisons when the number of reviews approaches the tens of thousands.

This assignment explores the efficacy of prompting a language model to categorize reviews under a smaller number of general categories that are weighted by whether the review author is writing positively or negatively with regard to the category.

Recommended Readings

- Brown et al., “Language Models are Few-Shot Learners,” Advances in Neural Information Processing Systems 33 (NeurIPS 2020)
- Zhao et al., “Calibrate Before Use: Improving Few-Shot Performance of Language Models” (ACL 2023)
- Lu et al. “Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity,” (ACL 2022)

Learning Objectives

- Tune prompts using demonstration number, demonstration selection and order.
- Design and evaluate basic prompt tuning experiments.

Assignment Deliverables

This is an individual assignment. In this assignment, students will develop one or more prompts to classify Amazon Reviews using a product-specific list of categories in the form of a hashtag (e.g., `#DurabilityAndLongevity`) . Training demonstrations for each category have been provided for students to use in tuning their prompt(s). As discussed in class, the selection and order of demonstrations influences evaluation metrics.

Develop a Jupyter Notebook that includes a section for conducting the experiment and a section for evaluating the experimental results. Students are encouraged to use the notebook provided in the assignment package.

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.
- results.json – the results of your experiment formatted according to the example format below under further guidance.
- questions.txt – Answers to the questions provided in the package.

Further Guidance

- The assignment package includes three datasets: dataset-dev.json, which is used for testing the prompt and experimental design; dataset-train.json, which is used for sampling demonstrations for in-context learning; and dataset-test.json, which is used for the final evaluation.
- There are at least two ways to present the task: (1) prompt the model to generate all relevant tags; or (2) prompt the model to decide if a specific tag applies to a review. While the first approach may be more challenging, depending on the model, the second approach requires more compute than the first.
- When generating categories from a list of allowable categories, be sure to check for hallucinations, especially if the input context is large.
- Try a few prompts and inspect the responses before scaling your data to the entire dataset. Write and test your evaluation code using only a few examples to limit the cost of debugging. Compute the token length of your prompts to estimate the total cost of your experiments before you running the experiments.
- The results.json file content should contain the same records included in the dataset-test.json file. For each record, add a new key “predicted” and include the predicted hashes from your prompt as the key-value.

Evaluation Criteria

- Correctness of the implementation of the experiment and JSON file.
- Thoughtfulness of the answers to the questions in questions.txt, including examples from your work to illustrate your answers.