Exercise 5 – LLM Prompting and Prompt Engineering

Advanced Prompt Engineering

Deadlines

Deadline for Exercise 5: **Sunday, 1.12.2024, 23:59 (Zurich Time)**.

Deadline for the peer review: **Monday, 9.12.2024, 23:59 (Zurich Time).**

- Note, the peer reviews will be assigned one day after the exercise deadline. You will find instructions for the peer review process at the end of this document.

Learning goals

The exercise consists of two parts. In the first part, we experiment with advanced prompt engineering techniques to assess the symbolic reasoning abilities of a base LLM. In the second part, we consider instruction-tuned LLMs, and evaluate their performance for linguistic annotation task involving structured predictions. The learning goals of this exercise are:

- to understand the difference between base LLMs and instruction-tuned LLMs
- to understand various prompting strategies that can be used to query base LLMs (e.g. zero-shot, few-shot, chain-of-thought prompting, etc.)
- to better understand prompting strategies for querying instruction-tuned models
- to gain experience with finetuning base LLMs for downstream tasks
- to gain experience with handling and validating both structured and non-structured outputs from LLMs
- to evaluate performance of LLMs under different settings

Please keep in mind that you can always consult us and use the exercise forum if you get stuck (note that we have a separate forum for the exercises).

Deliverables

Your solutions for each part should be submitted as self-contained iPython notebooks (ipynb files). The notebooks should contain your well-documented, EXECUTED, and

EXECUTABLE code. That way, your reviewers can view and execute your code without potential dependency issues and/or installing new packages or versions of packages. We encourage you to solve the exercise on Google Colab and download the .ipynb file when everything is completed.

You will also have to write a short lab report at the end of each script. The lab report should be written in a markdown cell and contain a detailed description of the approaches you used to solve this exercise. You can discuss what worked and what didn't work, but in both cases, you please provide reasoning as to why something did or didn't work. Please also include the results. Inside each notebook, we have provided specific questions denoted with "📝❓" that should be addressed in the report.

Please submit a zip folder named **ex05.zip** containing the following working ipynb notebook scripts, named as follows:

- **ex05_part1.ipynb**
- **ex05_part2.ipynb**

Please note:

- Organise your code such that it is executed and executable.
- Executed code means that the cells must have already been run and the output must be visible to anyone checking your notebook without having to run the code again.
- Your assessors should be able to run your code. If it doesn't work, you can't get the maximum score.
- Please make sure that your answers to the specific questions outlined in the notebook and denoted with "📝❓" are clearly marked. If the assessors cannot easily find the answers they need, you can't get the maximum score. We recommend including these clearly in your lab report.
- Also, we highly recommend testing that your notebook runs as expected before submitting it. To do this, hit "Runtime" > "Restart runtime and run all", and make sure that there are no errors! Be sure to submit the version of the file with the executed outputs.

Data

The datasets for this exercise can be downloaded in each of the starter notebooks provided.
In part 1, following Wei et al., 2022, we consider a toy task that is intended to test a model's symbolic reasoning abilities.
In part 2, we consider a classic NLP task of part-of-speech tagging (and tokenization).

Part 1 - LLM Symbolic Reasoning & Advanced Prompt Engineering

To facilitate the start, you can use this iPython notebook in Google Colab which loads the dataset and provides the necessary boilerplate code for your experiments. We recommend making a copy of the notebook to build on. Simply hit the “Copy to Drive” button to do this.

The aim of part 1 is to familiarise yourself with different prompting strategies for querying base LLMs and finetuning them on a downstream task.

Your task is to work through the notebook provided and implement the necessary code to:

1. Familiarise yourself with the Unsloth library for faster inference and finetuning of LLMs.
2. Design a prompt template for zero- and few-shot prompting.
3. Extend the prompt template to support chain-of-thought prompting and supervised finetuning.
4. Implement an inference pipeline to process queries with LLMs at scale.

Part 2 - LLMs as Linguistic Annotators

After experimenting with base LLMs, we now consider instruction-tuned LLMs, which give rise to applications like ChatGPT and Claude. Again, we provide you with this iPython notebook to get you started.

Your task is to work through the notebook provided and implement the necessary code to:

1. Apply an instruction-tuned LLM to the task of POS-tagging (and tokenization).
2. Experiment with manipulating the system prompt used to direct instruction-tuned models.

Submission & Peer Review Guidelines:

As with the first exercise, peer reviews will be carried out on OLAT.

As soon as the deadline for handing in the exercise expires, you will have time to review the submissions of your peers. You need to do **2 reviews** to get the maximum points for this exercise.

Here are some additional rules:

- **All file submissions are anonymous (for peer review purposes): Do not write your name into the Python scripts.**
- **ONLY ONE** member of each team submits on OLAT.

- Please submit a zip folder containing all the deliverables.

Groups & Peer Reviews:

- You should work in the same group (pair) as for exercise 1. Each member should contribute equally!
- Only **one team member submits the solutions.**
- Only the submitting team member will have access to the peer review, however, you should distribute the workload evenly!
- If you do not submit 2 reviews, the maximum number of points you can achieve is 0.75 (out of 1).
- Please use full sentences when giving feedback.
- Provide **constructive** and helpful comments to your peers! If you criticise something, you should also be able to provide actionable and realistic suggestions for how something can be improved.
- Please answer all the review questions of the peer review.