Ling 282/482 hw8

Due 11pm on November 25, 2024

In this assignment, you will

- Review the key training innovations incorporated in modern LLMs
- Discuss societal and scientific consequences to modern LLM usage
- Experiment with an LLM of your choice

Submission Instructions

This assignment contains only a written portion (no programming). Your answers must be submitted in a *.txt or *.pdf file to Blackboard.

1 LLM Training and Background [28 pts]

Q1: Ingredients for LLMs [8 pts] What are the three main "ingredients" that make up the difference between traditional language models like GPT-3, and modern "LLMs" like ChatGPT? Explain each in your own words (about 2 sentences each).

Q2: Precursors to LLMs [4 pts] What is one of the techniques described as a "precursor" to LLMs in Lecture 17? In your own words, describe this technique and how it relates to later processes for training full-blown LLMs (3-4 sentences).

Q3: Data requirements [4 pts] There are two types of data required for training modern LLMs that go beyond raw, unlabeled text that might be scraped from the internet. In your own words, describe one of the new types of data, as well how this data might be curated by someone building an LLM (3-4 sentences).

Q4: Reinforcement Learning [8 pts] Reinforcement Learning has become a key part of training modern LLMs:

- What is RLHF? What does it stand for, what does the process involve, and why is it critical for training LLMs? [3-4 sentences; 4 pts]
- What is DPO (in general terms), and how does it change the way RLHF is conducted? Why has it become widespread? [3-4 sentences; 4 pts]

Q5: Proprietary modeling [4 pts] What are a few ways in which the papers released along with LLMs trained by large corporations have diverged from normal scientific documents? What are a few advantages and disadvantages to keeping details of model training and architecture proprietary? [3-4 sentences; 4 pts]

2 Experimenting with an LLM [38 pts]

For this section, choose a publically-available LLM with which you can interface/experiment. You can also use a non-public model if you happen to already be subscribed to one. If you are unsure if a model qualifies, you can look to slide 30 of lecture 17, but most publically-available "chat-bots" are likely to be based on LLMs as we describe them in this class. If you are unsure what to choose, ChatGPT might be a good default choice: `https://chatgpt.com`

Q1: Your model choice [16 pts]

- What model did you pick? Try to be as specific as possible, since terms like "ChatGPT" can actually refer to a family of related models. [2 pts]
- Try to find an academic-style description ("paper") for your model choice. Organizations that train LLMs will often release a "white-paper" in the format of a traditional academic paper, which will often be hosted on a pre-print server such as `arxiv.org`. Googling `your_model_name paper` might be a good way to find this. Give the url for the paper you find here. [2 pts]
- Take a brief glance at the paper, especially the abstract (don't read the whole thing). What is claimed to be a main innovation of the model in question? What sort of training and architecture details can you find? What training and architecture details seem to be left out or hidden? [8 pts]
- Try to find whether the LLM you choose has features that go beyond what we discussed in class. E.g. some LLMs are multimodal (can interact with images or video), have access to the internet during generation (Retrieval-Augmented Generation), or have access to a coding language compiler during generation. Some of these might be more evident from the product website rather than the academic paper. Report your findings here, and keep them in mind for the next section. [4 pts]

Q2: Prompting the LLM [8 pts] Familiarize yourself with the "chat-bot" interface to your model if you are not already familiar. Most of us are familiar with the impressive things these models can do, but try to come up with a prompt/request/task that **challenges** the model's abilities. I.e. try to come up with something the model can't do well or fails to do entirely.

- If you succeed at coming up with something the model fails at, describe it: what prompt did you give, and how did the model respond? How does the model's behavior differ from the desired behavior? If you could not come up with something that the model couldn't do, describe what you tried. [4 pts]
- Why do you think the model either failed or succeeded at your challenging prompts? Are there aspects of LLM training that you think either failed or succeeded at capturing the behavior you were trying to elicit? [4 pts]

Q3: Prompt templates/engineering [8 pts] Try to come up with an every-day task that you might be able to solve with a **prompt template**. This is a type of prompt which shares a common format, but has "slots" to be filled with interchangeable information depending on the exact task you want the model to perform (see this guide for more information: `https://www.codecademy.com/article/getting-started-with-lang-chain-prompt-templates`). **Note:** you do **not** have to code-up an actual automated prompting system. Simply come up with the template.

- Report schematic version of your prompt template here (without specific information filled in) [4 pts]
- Try feeding this template to your chosen LLM with a few different versions (filling the template slots in). Does the model behave like you expect it to? [4 pts]

Q4: Reflection [6 pts] These questions are simply intended to have you reflect on your (non-)usage of LLMs in every-day life. There is no correct answer.

- Do you use LLMs already in your daily life? If not, do you think you will after experimenting with them here? Explain why or why not. [2 pts]
- What are some reasons you could imagine for a typical person to decide for or against incorporating LLMs into their daily workflow? Do you think they make work/school overall easier? [2 pts]
- It is generally agreed upon that AI and LLMs are driving a large growth in energy consumption (see for example this article). Do you think pursuing this technology and making it widely available is still worth it? Why or why not? [2 pts]