Homework 8: Fine-tuning LLMs

Due December 6th

In this homework, you will revisit one of the tasks that we looked at last week. This time, instead of relying entirely on a pre-trained language model, we will **fine-tune** a language model. Because we now need to run the models in training mode, however, we will have to work with smaller models.

This homework should be done in a Google Colab notebook with a GPU session. To make sure that your session is using GPU, navigate to Edit > Notebook Settings, and select GPU from the Hardware Accelerator drop-down menu.

If you run out of Colab credits, let me know. I have a small budget that I can use for this.

You should submit your Python files and your write-up (as a PDF) on Gradescope. Download Colab notebooks in Jupyter notebook format (.ipynb) to submit them.

1 Commonsense reasoning

Fine-tune DistilBERT on the COPA dataset from last week. Use the training set for training and the validation set for validation and evaluation. Use the pre-trained DistilBertForMultipleChoice model available through the Hugging Face Transformers library as your base model.

There is a well-written Hugging Face tutorial that covers how to fine-tune a BERT model on a sentence classification task. Here is the link to the tutorial: https://huggingface.co/course/chapter3.

However, you do not need to follow along with the instructions about loading a dataset, because we have our dataset downloaded.

1.1 Loading data

I have given you some data loading code, but you will need to upload the dataset files to Colab. In main, you should use the provided **load_data** function to load the test and train files (train.jsonl and val.jsonl).

1.2 Formatting data

You will need to fill in the **tokenize_example** function to make **load_data** work. Follow the DistilBert documentation to get this working.

1.3 Training loop

Using the tutorial, set up a training loop for your model. It should run for num_epochs, looping completely through the batches in train_dataloader in each epoch.

1.4 Evaluation

I have given you a function called **evaluate_sample**. This function takes a random sample of the dataset that is passed in, evaluates the model on the sampled examples, and prints out the examples, their correct labels, and their predicted labels. This should be useful in confirming that your model is correctly configured.

Write a function called **evaluate** that evaluates all batches in a dataset that is passed in, printing out accuracy. Call this function at the end of each training epoch in **main**.

1.5 Analysis questions

1. How well does the LLM perform on the task?
2. Did you observe signs of overfitting to the training data? If so, about how many epochs did this take?
3. Do you notice any trends in the model performance?

2 AITA Task

Am I The Asshole is a popular advice subreddit where posters seek an opinion about whether they have done wrong or been wronged in a given situation. In this part of the homework, you will build a system that takes a post and classifies it as either “YTA" (You’re the Asshole) or “NTA" (Not the Asshole).

I have given you both a training set and test set. If you pursue fewshot learning, you can use examples from the training set as part of your prompt.

You may either fine-tune an existing model or pursue a fewshot learning approach. **You may not use OpenAI’s ChatGPT or GPT-4 models.** You can use any other model, whether for finetuning or fewshot prompting.

2.1 Analysis questions

1. What approach did you take? How did you decide what to use?
2. How well does your system do?
3. What challenges did you run into?

3 Final project preparation

3.1 Project update

- What progress have you made on your final project?
- Have your plans changed or evolved since last week?

3.2 Literature review

Please identify three academic research papers that provide relevant background for your proposed project. Read them and provide a paragraph summary of each.

Two good places to look for NLP papers are Semantic Scholar and the ACL Anthology.