Homework 7

11-967

Due date: 03/16/2025 11:59 PM EST

In this homework, you will improve your understanding of how different models can behave very differently given the same prompt. You will also implement a method to mitigate label bias.

*Note: there are **two** Gradescope submission pages, one for your code and one for your `report.pdf`. Please submit your code to **Assignment** as instructed, and your `report.pdf` to **Assignment (PDF)**.*

Problem 1: Comparison of Model Behaviors

In class, we've learned how the data that models are trained on can have significant impact on their capabilities. This includes the choices in pre-training data as well as any post-training that was performed. In this question, your goal is to identify tasks for which one or another model is best suited. In the real world, it is typical to initially choose a model based on qualitative exploration and intuition before moving on to expensive, quantitative evaluations to assess which model is best for your task.

We have provided you with some example code in `src/olmo/inference.py` for doing inference with the OLMo family of models on your machine. You may either modify this starter code or write your own code to answer this question. ***Note: this question asks you to experiment with several different models, some of which might require you to get approvals on HuggingFace. If you run out of disk space you may need to delete model files before moving onto the next question.***

[Question 1.1] (***Written, 7 points***) You would like to use an LLM to explain difficult topics in physics at an elementary level. For example, here are a few prompts you might try (one per line).

- `Explain quantum computing to me like I'm five years old.`
- `Tell me how particle accelerators work in the most basic language possible.`
- `Pretend I'm really dumb and explain why gravity exists.`

Your goal is to decide on a good instruction-tuned model to use for this task. Try the above prompts (or others of your choice) with three different open-source LLMs that

- have undergone instruction tuning.
- are in the range of 7-9 billion parameters
  (e.g. `OLMo-7B-Instruct`, `gemma-2-9b-it`, `Meta-Llama-3.1-8B-Instruct`, etc.).

DELIVERABLES FOR Q1.1

In your report, answer the following:

A. Write down the three models you are comparing.

B. Which of the 3 models seemed most effective at the task of generating elementary explanations of physics concepts?

C. Write down the prompts that most informed your decision of which model was best.

D. In a paragraph, explain your answer to B. What properties of the generations led to your decision? To what extent did the phrasing of the prompt impact which model seemed to do the best?

[Question 1.2] (***Written, 5 points***) Suppose you would like to use an LLM to generate a recipe given the name of a dish. Sometimes it's not always clear whether a pre-trained model, or one post-trained is best for a given task. Experiment with the following models, designing a verbalizer for each that can accomplish the recipe generation task:

- `OLMo-7B`: Language pre-model trained on Dolma, a largely web-derived dataset.
- `OLMo-7B-SFT`: OLMo-7B that has been post-trained with supervised finetuning on Tulu, a dataset designed to teach instruction following.

DELIVERABLES FOR Q1.2

In your report, answer the following:

A. Write down the verbalizers that worked the best for each model. Note: verbalizer is not the exact prompt. It's the prompt format.

B. Which of the two models seemed most effective at the task of generating recipes?

C. Compare and contrast the generations from the two models. Is this a task that can be effectively performed using a pre-trained model?

[Question 1.3] (***Written, 5 points***) There is one other version of OLMo:

- `OLMo-7B-Instruct`: OLMo-7B-SFT that has been further post-trained with DPO on UltraFeedback.

Since this version has been tuned on human-preferences, it will avoid generating answers that human annotators might perceive as harmful. For example, when asked for instruction on how to build a bomb, `OLMo-7B-Instruct` might respond "Sorry, I can't help with that." However, as we discussed in lecture, through clever prompting, it is possible to "jailbreak" an aligned language model, getting it to generate text that violates principles of harmlessness.

DELIVERABLES FOR Q1.3

In your report, answer the following:

A. Identify a question or task (other than bomb-making) for which `OLMo-7B-Instruct` generates a response that includes a refusal. Write down your prompt and the model's response in your report.

B. Experiment with modifying your prompt such that the model successfully performs the task. Write down at least 3 of the prompts you tried and the model's responses to them in your report. In a paragraph, discuss what strategies worked and which ones didn't.

C. In a few sentences, explain why building aligned language models which cannot be jail-broken is difficult.

Problem 2: Calibration and Bias Mitigation

As we learned in class, few-shot learning is sometime unstable. The model's performance may vary significantly based on prompt formats and also the order of examples. This instability arises from the bias of language models towards predicting certain answers, e.g., those that are placed near the end of the prompt or that are common in the pretraining data. Zhao et al.(2021) propose a method to "calibrate" a language model to reduce these biases.

[Question 2.1] (***Written, 3 points***) When applying language models to binary classification, the model's predicted probabilities may be miscalibrated, meaning they don't reflect true class frequencies. Consider a scenario where the true labels are evenly balanced (50% positive, 50% negative), but the model is biased towards predicting positive - it frequently outputs $p(Positive) > 0.5$ even for negative cases, resulting in a skewed distribution of predicted probabilities, as shown in Figure 1.

DELIVERABLES FOR Q2.1

Suppose you have computed the distribution of predicted $p_\theta(Positive)$ values across all instances in the test set, and find it to look like Figure 1 above. You also know that the true labels are balanced. How can you determine a better decision boundary to correct for this bias than $p_\theta(Positive) > 0.5$, without access to any true labels or development set? Answer in two sentences.

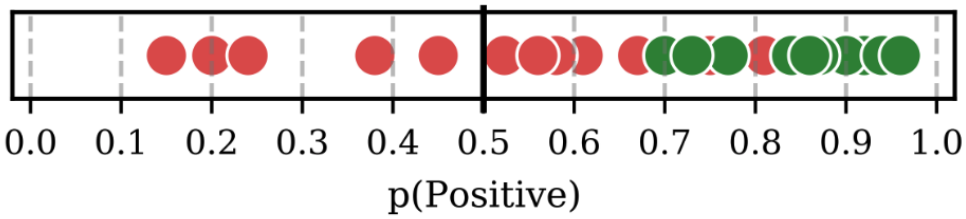

Figure 1: A miscalibrated classifier that is biased to the positive class, taken from Zhao et al.(2021). Negative groundtruth examples are marked with ●, and positive groundtruth examples are marked with ●.

[Question 2.2] (***Coding, 10 points***) The paper Zhao et al.(2021) proposed a contextual calibration method to correct biases in language model outputs. This approach estimates a model's inherent biases using *content-free* inputs like **"N/A"**, then adjusts the output probabilities to ensure uniform class scores for these inputs. You will implement and evaluate this method by using GPT-3 to perform binary sentiment classification on the SST-2 dataset through the OpenAI API, allowing you to observe how calibration can improve classification accuracy in practice.

DELIVERABLES FOR Q2.2

Read the paper and complete the code in `src/bias/utils.py`. Please see the README for additional instructions. Submit your code on Gradescope.

[Question 2.3] (***Written, 6 points***) Now you have completed the code of mitigation classification bias in few-shot learning. You are free to play around with it by modifying the following hyperparameters:

- `seed`: random seed used to randomly choose examples from a dataset
- `num_shots`: the number of examples provided in few-shot learning
- `content_free_input`: the content-free input used in the algorithm

DELIVERABLES FOR Q2.3

Report the results of **two** different hyperparameter configurations. For each, describe what hyperparameter values were modified, and the resulting classification accuracy scores before and after your calibration. Your report should answer the following questions:

A. Describe your experimental procedure in a few sentences.

B. In one paragraph, describe your experimental results. Include the following accuracy **before and after** calibration and discuss whether the bias calibration worked as expected.