Spring 2023 CS-552 Course Project

By Prof. Antoine Bosselut

Your course project will involve using large-scale (100B+ parameters) and medium-scale (100 - 300M parameters) language models to collect data and train reward models for producing a ChatGPT agent in the area of education assistance.

Description:

In this project, you will be tasked with training an AI tutor that is specialised to course content at EPFL. You will train this tutor using the same procedure that was used to train modern AI chatbots such as ChatGPT.

The project will be divided into the following three steps:

1. *Collecting supervised fine-tuning data* – one of the most important ingredients of training a ChatGPT-like chatbot is the collection of demonstrations for the task you want your chatbot to perform. In the first part of the project, you'll distill these demonstrations from 100B-parameter scale LLMs.
2. *Training a reward model* – Assistants trained with supervised demonstrations are further trained with reinforcement learning with human feedback (RLHF) to learn more from their own generated demonstrations. For RLHF to work, you'll need to train a reward model that can effectively reward the demonstrations that your chatbot produces. While you are not required to train your final model using RLHF, you will be required to train a reward model.
3. *Training your chat engine* – You'll use your supervised demonstrations and (optionally) your reward model to train your final chatbot that can be an effective educational assistant.

Part I - Collecting Supervised Fine-tuning Data

As you learned in Week 8, before training models using RLHF, we need to fine-tune our pretrained language model using examples that demonstrate the task the chatbot is designed to accomplish (supervised fine-tuning). In the first stage of the project, you will collect this supervised fine-tuning data by distilling it from ChatGPT.

Specifically, you will be given ~100 questions from a course at EPFL and prompt ChatGPT to provide a suitable response to these questions. These responses (or *interactions* if they go for multiple rounds) will serve as demonstrations for your later model. You should explore different methods for prompting ChatGPT so that it produces better demonstrations for the sets of questions you are given.[1] The best demonstrations may involve multi-turn interactions where you ask ChatGPT to reconsider its response.

[1] You might also consider how a student might query a chatbot for seeking knowledge on a particular topic or receiving clarifications about material they are not familiar with.

You should collect at least one interaction for each of the ~100 questions you are given, and submit what you consider to be the best demonstration for each of these questions (you may also submit additional demonstrations). For each demonstration you turn in, you should record your confidence on a scale of 1 (low confidence) to 5 (full confidence) that the model has arrived at the correct answer. Your milestone report should describe the prompting strategies you tried (for example, the first suggested reading describes a prompt that adds “Let’s think step-by-step” before each question). You should qualitatively compare your impression of different prompting strategies you tried!

Suggested Reading:

- Large Language Models are Zero-Shot Reasoners - https://arxiv.org/abs/2205.11916
- PromptSource - https://arxiv.org/abs/2202.01279
- Super-NaturalInstructions - https://arxiv.org/abs/2204.07705

Part II - Training a Reward Model

The data you collect in Stage 1 could be used to fine-tune a language model using supervised fine-tuning. However, high-quality assistants such as ChatGPT are trained using more than only supervised learning. They use a technique called Reinforcement Learning with Human Feedback (RLHF). RLHF requires your training procedure to have access to a reward model that can evaluate multiple different responses and rank them according to their suitability.

In the second stage of the project, you will train a reward model that identifies which responses produced by your AI agent provide better answers to the questions they are prompted with. It is up to you to decide what types of data to collect to help train this reward model. The following could be suitable approaches to collecting additional data:

- One approach would be to collect better demonstrations for the questions you have by using the actual solutions to guide your interaction with ChatGPT. These demonstrations will provide a contrast to the interactions you collected in Part 1, for which you could not confirm whether the solution was correct.
- You might also think about flipping the interaction and pretending that you are a human tutor for a ChatGPT “student” and have an interaction where you guide ChatGPT to the correct answer.
- You can also think about using data found online in other resources. We encourage you to use a mix of new interactions that you collect & interaction data that may be available online.
- Other approaches may be suitable as well, and we look forward to your creativity in collecting such data to improve your reward model

The key to training your reward model will be to identify which demonstrations are better than others for the same initial question, requiring you to develop these labels on your own. Once again, in your milestone report, you should outline your strategy for collecting additional data to train your reward model.

Then, you should train a reward model using this data to be able to rank different demonstrations based on how well they accomplish the task presented by each question they receive. For this step – training the reward model – we will also provide you with a solution for each of the original questions you were given in Part 1, along with data from other project teams to provide more data points for your reward model to be trained. You should turn in this initial reward model as part of the milestone deliverable, though you may continue to improve up until the final deliverable is due.

Suggested Reading:

- Aligning language models to follow instructions - https://openai.com/research/instruction-following
- Training language models to follow instructions with human feedback - https://arxiv.org/abs/2203.02155

Part III - Training the final model

In the final portion of your project, you will be responsible for fine-tuning a generative pretrained language model (e.g., GPT2, BART, T5, etc.) so that it learns to produce better demonstrations when prompted with a question from your course. You should train your model using supervised learning on some of the data you have collected in the first two parts of your project. This can be data you have distilled from ChatGPT or data you have collected from other sources. You should use your reward function to evaluate the quality of the text generations produced by your model.

Optional (for extra credit): Once you have a suitably-trained model from supervised fine-tuning, you may use your reward model to further fine-tune your chatbot using RLHF.

Suggested Reading:

- Koala - https://bair.berkeley.edu/blog/2023/04/03/koala/
- Alpaca - https://crfm.stanford.edu/2023/03/13/alpaca.html
- Vicuna - https://vicuna.lmsys.org/
- Illustrating Reinforcement Learning from Human Feedback - https://huggingface.co/blog/rlhf

Supervision

Each team will be supervised by one of the course TAs. For questions regarding the project, you should reach out to your supervisory TA as a first step. When you request it, your TA should make themselves available to you for discussion about the project during normal course hours (Wednesday, 9h - 12h; Thursday 13h - 16h). If they are not available at that time, it is also possible to set an alternate time for an in-person or remote meeting. Additionally, a domain expert may be available to your team to discuss course-specific content.

Collaboration:

You should work on this project in teams of **three**. All team members should contribute roughly equally to the submission. The first project milestone will be individually graded. The second milestone project and the final report will be graded for all team members. With your final report, you should submit a Contributions statement for all team members (See Section 10 of this paper for an example: https://arxiv.org/pdf/2112.11446.pdf).

You are also free to discuss your project with others in the course, though only the people on your team should contribute to the actual implementation and experimentation involved. You may build your work upon existing open-source codebases, but are required to clearly specify your team's contributions and how they differ from the pre-existing codebase in your milestone reports and final report.

Deliverables and Grading:

The deliverables for this project will be divided into two milestones and a final submission. Each milestone will be worth 15% of the final grade with the remaining 30% being allocated to the final report.

Milestone 1 (15%):

In the first milestone, students will deliver:

(1) An initial dataset of demonstrations for your ~100 questions collected through interaction with large-scale language models (e.g., ChatGPT) to complete a set of academic tasks. These demonstrations must be collected with the GPT Wrapper package and the API key we provide you after the teams are finalized and the consent forms have been filled. A documentation on how to use the GPT Wrapper package can be found in the project reference folder. **Do not use the OpenAI API to collect your interactions. Only use the GPT Wrapper package we will provide you.** You may submit as many demonstrations as you like, though it is only necessary to submit one demonstration per question. You will be graded on the description of the prompting strategy you developed to produce these interactions, as well as your description of how you selected it among other prompting strategies you tried. The data should be submitted in the format of the `m1_submission_example.json` file in this project reference folder.

Each project member should collect demonstrations for all questions. This process will teach you how to interact and prompt large-scale language models to extract more useful demonstrations from them. You will also learn to evaluate your strategies in a systematic way.

(2) A literature review related to this project, **one paper reviewed by each of the team members. You should each review a different paper** that is relevant to the subject of the project (e.g., prompting LLMs, RLHF, evaluating text generation models, etc.). Papers published in top conferences: ACL, EMNLP, NAACL, NeurIPS, ICLR, AAAI are likely good papers to read & review. However, papers on preprint servers such as arXiv may also be good candidates. We leave it to you to find suitable papers, but encourage you to clear your selected papers with your TA. **Do not use the papers from this project description's suggested readings.** Follow this guideline to effectively read and review papers: How_To_Evaluate_a_Paper.pdf

This review will teach you how to (1) find research papers related to a topic you are studying, (2) critically assess research papers, and (2) gather useful information from them that you can re-implement in your own work.

(3) A plan for how to complete the subsequent portions of the project. You should outline an initial strategy for how you will train a reward model (including data collection), train your final model, and evaluate the improvement of your model over other options. You will not have to stick to this plan, and you will not be graded on how closely you follow it. It is an exercise in making you prepare for the next stages of the project. This part of the project is to provide you with skills around setting goals, managing a project timeline, and breaking the problem of the project into more manageable sub-tasks. Questions you may consider in your proposal:

- *What data will you use?* Specify the dataset(s) you will use, and describe any preprocessing you plan to do. Be sure to specify whether the data is public and how you plan to access it if not. If you plan to collect your own data, describe an early plan for how you will do that.
- *What method(s) are you planning to use?* Describe the models and/or techniques you plan to use to implement your reward model and final model. For example, to train your reward model, you might consider fine-tuning an existing pretrained model, such as BERT, RoBERTa, BioBERT, or BioRoBERTa. Make it clear which parts you plan to implement yourself, and which parts you will download from elsewhere.
- *How will you evaluate your reward model and final model?* Specify at least one automatic evaluation metric you will use for quantitative evaluation of your final model. You should also identify a baseline method (or more than one) that you will compare your results to. Finally, if you have particular ideas about the qualitative evaluation you will do, you should describe this too.

You may submit a single, joint project plan among the three team members.

Due Date: 14.05.2023

Milestone 2 (15%):

In the second stage of the project, you will train a reward model that identifies which responses produced by your AI agent provide better answers to the questions they are prompted with. It is up to you to decide what types of data to collect to help train this reward model.

You should submit the following deliverables:

(1) The dataset used to train your reward model following the format of the example in this project reference folder. This example will be added when Milestone 2 starts. For each source of data you collect (e.g., demonstrations from ChatGPT, data found online, etc.), please submit a separate data file. In your final report, you should discuss the legal and ethical considerations of the data you collect.

(2) The model file for your reward model, submitted in the format of the example in this project reference folder. This example will be added when Milestone 2 starts.

Update 21.05.2023:
(3) A milestone report describing the data sources (and their formats) included in (1) and a description of the model submitted in (2).

Update 26.05.2023:
Due: 04.06.2023

Final Report:

The final report, code, and data will be due on June 18th. Students are welcome to turn in their materials anytime ahead of this date once the semester ends. Your final deliverable should include the following components:

(1) A final report detailing the data you collected, the models you trained, and the results you achieved using these models to train your educational assistant.

(2) A description of all data formats used to train your model with separate JSON files for all datasets used for training.

(3) Two model files – one for the reward model, and one for the chat assistant model.

For the most up-to-date information on the final milestone please refer to
CS-552 Final Report Specification

Update 26.05.2023:
Due: 18.06.2023

Resources:

ChatGPT Access: You will query the ChatGPT API using a server that the NLP lab has set up to provide you with free access to ChatGPT (*n.b.,* it's not free, but the course is paying for it). To interact with the server, you will need the GPT Wrapper package and a student-specific API key, which we will provide you. **You must be authenticated with the API key we provide you.** Otherwise, you will not be able to use the GPT Wrapper package, and therefore not be able to query ChatGPT. For each student-specific API key, you will have a limited "token budget," to query ChatGPT. It is very unlikely you will approach this maximum if you only use your budget for the purposes of the project. However, if you find yourself limited, talk to your TA and we will see about raising your limit. A documentation on how to use the GPT Wrapper package can be found in the project reference folder.

Google Cloud Credits: Each team will be provided with CHF 150 worth of Google Cloud Credits to complete the project. Please refer to the document for redeeming these credits in this project reference folder. The tutorial will be added when Milestone 2 starts.