CS-552: Modern NLP
Project Description

Antoine Bosselut

EPFL

Context

- ChatGPT released in late 2022
- ChatGPT ingredients are things we already cover in class:
  - Transformers
  - Pretrained Language models
  - Prompting, in-context learning, chain-of-thought reasoning
  - Model interpretability & explainability
  - Text generation: training with RL, evaluation

Context

- Wouldn’t it be fun if students could work with ChatGPT-like models for their class project?
- What would be an interesting setting to design these systems for?
- How could we fit the skills we want you to gain (fine-tuning, prompting, scale, evaluation, data, ethics) in this framework?

Build your own education assistant!

- **Course Project**: Build your own educational chatbot for EPFL courses
  - https://docs.google.com/document/d/1SP8SCHPOZZGEhs2ay-38FjedRE1bS9Q99VJb28eHoYk
- 100 courses on campus shared assessment materials from their courses
- TA staff for CS-552 prepared content from these courses into data that you can use for your project
- You'll use data from some of these courses as a starting to point to train education assistants!

Project: Three Stages

- **Collect preference data**
  - Aligning a LM requires collecting of preference data for the tasks you want your chatbot to perform
  - In the first part of the project, you'll distill these data from 15B+ parameter scale LLMs
- **Train a generator model**
  - Using DPO, you'll train your model to generate answers closer to the preferred demonstrations (compared to the rejected demonstrations)
- **Improve your model**
  - You will finish training your generator model, adapting it for MCQA
  - You will improve it in one or two different ways: RAG and/or Quantization

Part 1: Planning, Lit Review, Data Collection

- Collect preference data (i.e., ranked pairs of demonstrations) for your chat assistant by distilling them from ChatGPT
  - Given ~100 questions from a course at EPFL
  - For each question, prompt ChatGPT to provide two potential responses, where one is better than the other
  - Compare and rank each pair of answers
- Literature Review
  - Read and review a paper related to the project
  - **Each** team member should submit a review of a different paper
- Project proposal for completing the other parts of the project over the rest of the semester
  - Details in the project description

Part 2: Train a generator model

- Using DPO, train a generator model that generates text aligned to the preferences captured by your data
- Organize a dataset for training this generator model
  - Use the preferences you collected in Part 1
  - Use other sources of data
- Reflect on your progress, preliminary results, and next steps

Part 3: Improve your model

- Finish training your generator model (using the data you collected in the first two parts of the project).
- Specialize your model to answer MCQ
  - You must train your model to output a single letter – either ‘A’, ‘B’, ‘C’ or ‘D’ – as its output
- Improve this model in one or two ways (groups of 3 and 4, respectively):
  - RAG (Retrieval Augmented Generation)
  - Quantization

Advising Policy

- Each team will be assigned a mentor (AE or TA) to advise them over the course of your project
- When you have questions, you should first reach out to your mentor before contacting other members of the course staff
- When you request it, your mentor should make themselves available to you for discussion about the project during normal course hours
  - Wednesdays, 13h15-14h; Thursdays, 14h15-16h
- If you (or they) are not available at that time, it is also possible to set an alternate time for an in-person or remote meeting.

Compute

- Each team will have access to the SCITAS cluster
  - Can be used to train your model
  - Information session **May 2nd**
  - See the project description for documentation
- Each student will also have access to $50 Google Cloud Credits
  - Instructions for redeeming and using them on the project description
- You are free to find other sources of compute, but these are the ones we provide as part of the course

Grading

- Milestone 1 (15%)
  - Collect first dataset of preference data
  - Conduct literature review
  - Proposal for remainder of project
- Milestone 2 (15%)
  - Collect additional data from any source
  - Trained generator model
  - Progress report
- Final models, code, data, report (30%)

Grading

- For fairness, we will use parameter tranches to compare model performances, so as to not penalize you over compute limitations.
  - 0-150M
  - 151M-999M
  - 1B-2.9B
  - 3B+
- On the other hand, we will reward ambition (beyond model scale), so don’t be afraid to explore bigger model sizes and leverage the compute we make available to you!

Collaboration Policy

- Project teams are composed of 3 or 4 members, who should contribute roughly equally.
- Grading breakdown:
  - The data collection and literature review of the first project milestone will be individually graded.
  - The project proposal, second milestone, and the final report will be graded for all team members.
  - With your final report, you should submit a Contributions statement (See project description).
- You may 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 must clearly specify your team's contributions and how they differ from the pre-existing codebases in your reports. AI-based tools are allowed but must be properly cited (see project description doc).

Next few days: Project Sign-ups

- **Today - Monday:**
  - Look over the Project Description
  - Accept the assignment on GitHub, using your team name
- **Monday:**
  - Project packages have been sent by email
  - API keys activated to provide access to ChatGPT