CS-552: Modern NLP
Project Description

Antoine Bosselut

Context

- Ingredients of ChatGPT, Claude, Gemini are topics we cover in class:
  - Transformers
  - Pretrained Language models
  - Prompting, in-context learning, chain-of-thought reasoning
  - Text generation: training with RL, evaluation
  - Retrieval-/Tool-augmented Generation

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?

Course Project

- **Course Project**: Build your own education assistant for EPFL courses
    - https://docs.google.com/document/d/1_pEhrREF1IhR0xWFsJ3yvrh-wNJFtKz5HJfEIMupKtE/
- TA staff for CS-552 prepared content from EPFL courses into data that you can use for your project
- You'll use data from these courses as a starting to point to train education assistants!

Project: Three Stages

- **Planning:** Project Proposal & Literature Review

- **Data Collection:**
  - Training chat assistants requires collecting preference data for the task you want the chatbot to perform
  - In the first part of the project, you'll distill these preference pairs from state-of-the-art LLMs.

- **Model Development**
  - Chatbots must demonstrate complex reasoning capabilities to be useful in practice.
  - In the second stage, you will train a generative reasoning model using Direct Preference Optimization (DPO) and Supervised Fine-Tuning (SFT).
  - Reasoning models must be efficient and have up-to-date information, so you will also implement Quantization and Retrieval Augmented Generation (RAG) training pipelines.
  - In the third stage of the project, you'll improve these models as much as you can!

Part 0: Proposal

- Project plan for completing the other parts of the project over the remaining project period
  - Details in the project description

Part 1: Literature Review, Data Collection

- Collect preference data (i.e., demonstration pairs) for your chat assistant by distilling them from ChatGPT
    - Given ~100 questions from a course at EPFL
    - For each question, prompt ChatGPT to provide responses to create a preference pair
    - Explore different prompting methods for extracting suitable demonstrations for preference pairs

- Literature Review
    - Read about prompting methods, RLHF, and other related topics for the project
    - Team members should compile a literature review of the explored papers, synthesising their findings

Part 2: Training Models

- Train a **generative assistant model** using your collected data and other datasets
- Train a **preference model** to score demonstrations produced by your generative system
- Train a **retrieval-augmented assistant model** that has access to external information
- Train a **quantised assistant model** that is more efficient than your original model, while retaining as much of its original performance
- Organise datasets for training these models (using data from part 1, and other sources)
- Evaluate the quality of your models

Grading

- Proposal (1%)
- Milestone 1 (10%)
    - Collect first dataset of preference pairs (5%)
    - Conduct literature review and submit related work report (5%)
- Milestone 2 (5%)
    - Collect additional data from any source
    - Submit v1 of preference model, generative reasoning model, quantised model, RAG model (4%)
    - Submit Progress Report (1%)
- Final models, code, data, report (54%)

Grading: Group vs. Individual

- Project deliverables will have both individual and group components
- **Data:** Every student will collect their own preference data
- **Modeling:** Every student will be **principally** responsible for one of the four models developed by the team.
  - Each student's grade will be more weighted by the development of their **principal** model
  - However, part of each student's grade will be linked to the performance of *all* models from the team, encouraging collaboration and team work

Grading: Group vs. Individual

- Individual (21% of course grade)
  - **Milestone 1**: First dataset of preference pairs (5%)
  - **Milestone 2**: Principal model runs (1%)
  - **Final Deliverable:** Final performance of principal model (15%)
    - Code and data for training model must be submitted

- Group (49% of course grade)
  - **Proposal** (1%)
  - **Milestone 1:** Literature review and related work report (5%)
  - **Milestone 2:** Progress Report (1%) and model runs from other team members (1% each = 3%)
  - **Final Deliverable**
    - Other models, code, data (9%)
    - Final report (30%)

Failure to submit an individual principal model in Milestone 2 and Final Deliverable will result in 0 for progress report and final report, respectively, for the student responsible for that principal model.

Advising Policy

- Each team will be assigned a mentor (course AE) 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
    - Wednesday, 13h15 - 14h; Thursday 13h15 - 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.

Collaboration Policy

- You should work on this project in teams of **four**. All team members should contribute roughly equally to the submission.

- Grading breakdown:
  - The data collection and majority of the model development will be individually graded.
  - The project proposal, literature review, progress report, and the final report will be jointly graded for all team members. Additionally, team members will be partly graded on the performance of the models of their teammates.
  - 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.

Next few days: Project Sign-ups

- **Today - Tuesday next week (15.04.2025):**
  - Look over the Project Description
  - Fill out team registration form
- **Wednesday next week (16.04.2025):**
  - Proposal Released