MNLP Project Proposal

1 Overview

In our course project, our objective is to develop an advanced chatbot by leveraging open-sourced large language models like LLaMA [17], specifically tailored to answering and explaining questions from academic courses. Drawing inspiration from the success of InstructGPT [15], we intend to combine supervised training and reinforcement learning with human feedback to enhance the chatbot's performance.

2 Data preprocessing

For our project, we will utilize interactions provided by other students from the course project, public multiple-choice datasets, and question-answering datasets.

2.1 Data collection for supervised training

To effectively train a language model in a specific area, it is crucial to gather an ample amount of training data.

Interactions collection Specifically, we employ different prompting templates to interact with ChatGPT based on the question type. Our prompting methods encompass: 1) asking for explanations for the academic items in the question stems and/or options, 2) analysing step by step, 3) providing the final answer to the question and giving the explanations.

Human evaluation scores A dataset of comparisons between model outputs is collected by all the project students. The collected dataset will consist of approximately 20'000 interactions, on about 6'000 different questions (on average each question will receive two or three interactions).

2.2 Data collection for the reward model

The reward model assigns a score to the answer generated by the chatbot, with a higher value when the quality of the generated output is good. Different types of data will be collected to train this model:

Gold demonstration collection A gold answer is considered the optimal answer to a given question. The reward model will be trained to recognise the gold answer and give them the highest scores. If gold answers are not provided or are too short, they will be collected online (Wikipedia or specialised literature) for a small number of questions (100).

Other data sources For this project we consider a starting point based on a pre-trained language model that has already been trained on language modelling tasks [11]. Nevertheless, in order to train a comprehensive course assistant, we recognize the importance of augmenting the training data beyond the provided course team data. There are two potential sources we can explore: open-world language training data sources such as arXiv articles and Wikipedia articles[1] and seeking relevant public datasets that align with our task. We aim to enhance further the model's understanding and knowledge within the specific academic domain by leveraging these public datasets:

Human ChatGPT Comparison Corpus (HC3)[2] The HC3 dataset compares sentences from ChatGPT with human responses to evaluate whether ChatGPT can keep honest and harmless interactions. The English version contains 24'322 questions, 58'546 human answers and 26'903 ChatGPT answers [7].

Massive Multitask Learning Understanding (MMLU)[3] The MMLU dataset covers 57 academic subjects across STEM, the humanities, the social sciences, and more. It contains 15'908 questions [9].

AI2 Reasoning Challenge (ARC)[4] The dataset contains 7'787 questions natural, grade-school science questions [4].

3 Reward model

3.1 Model design

The reward model is used to automatically predict the quality of answers generated by the chatbot during an interaction with the user. A pre-trained language

[1] https://arxiv.org/ and https://www.wikipedia.org/ respectively
[2] https://github.com/Hello-SimpleAI/chatgpt-comparison-detection
[3] https://github.com/hendrycks/test
[4] https://allenai.org/data/arc

model will be used as a text encoder, and a regression layer will be fine-tuned to predict the reward.

Several models based on the transformer architecture fine-tuned on academic or scientific datasets exist and will be compared:

BioMed-RoBERTa [8] A language model based on the RoBERTa-base [13] architecture, trained across four domains (biomedical and computer science publications, news, and reviews).

SciBERT [2] A pre-trained language model based on BERT [5], trained using a large multi-domain corpus of scientific publications and has been shown to outperform BERT on a variety of scientific NLP tasks.

3.2 Training strategies

First, the reward model will be adapted to our domain with a supervised-learning fine-tuning (SFT) strategy on a language modelling task. For this task, datasets such as HC3, MMLU or ARC can be used.

The second fine-tuning strategy is to teach the model to recognise the best answer among two or more answers, so that the gold answer Y_+ becomes optimal among all possible answers provided [12]. For a given demonstration Y_-, the reward model predicts the reward score $R(Y_-)$ and a ranking loss function is used: $L_{RM} = -\log \sigma(R(Y_+) - R(Y_-))$.

For this task, we will first use a large-scale reading comprehensions dataset such as the HC3 dataset. For a subset of questions, a ChatGPT interaction will be collected. The reward model will then be trained to recognise the gold answer among the two provided demonstrations.

4 Supervised training

4.1 Model design

We find several open-source language models to base our chatbot on, such as LLaMA [17] and Vicuna [3].

LLaMA LLaMA is a collection of open-source language models with 7B, 13B, 33B, and 65B parameters. LLaMA-13B has a superior performance compared to GPT-3 (175B) on multiple benchmarks.

Vicuna Vicuna-13B is an open-source chatbot that fine-tuned LLaMA on user-shared conversations from ShareGPT. Preliminary evaluation using GPT-4 as a judge shows Vicuna-13B achieves more than 90% quality of OpenAI ChatGPT and Google Bard.

4.2 Training strategies

There are two potential strategies to train the chatbot:

Fine-tuning the whole model If we have enough data and computation resources, we can build upon the open-sourced LLaMA language model.This approach enables us to leverage techniques such as Fully Sharded Data Parallel and mixed precision training, enhancing the training process's efficiency and performancein order to maximize the capabilities of the LLaMA7B model for our specific task.

Training an adapter In line with the innovation introduced by the LLaMA adapter [18], we can also explore the concept of a lightweight adaption module to facilitate efficient fine-tuning of the LLaMA model for instruction following tasks, while keeping computation costs low.

5 Reinforcement learning with human feedback

5.1 Model design

The final model will be obtained by fine-tuning the language model again using reinforcement learning with human feedback (RLHF).

5.2 Training strategies

The training loop is composed of three steps, similar to most training strategies in machine learning [1]: sample from the policy and generate responses; pass the responses to the reward model to get a rating; and run a PPO [16] step. There is a PyTorch PPO implementation by Spinning Up [14]. There is also a HuggingFace TRL (Transformer Reinforcement Learning) implementation [10]. The latter seems more practical to use, so it will be the first option. Datasets such as HC3 and HH-RLHF can be used for this task.

6 Evaluation

We plan to compare our final model with several baselines on both quantitative and qualitative evaluations.

6.1 Metric-based evaluation

To do the quantitative evaluation, we will compute the BLEU (BiLingual Evaluation Understudy), ROUGE (Recall-Oriented Understudy for Gisting Evaluation) and BERTScore on variants of our model (including the pretrained baseline) and lightweight chatbots.

6.2 Human evaluation

Finally, to get a more in-depth analysis of the performance of our final model, we will do some qualitative human evaluation following the method presented in [6]. For this, we will do binary comparisons of our model outputs with the chosen baseline and ChatGPT. For each output, the evaluators will have to choose the best answer without knowing the origin of the output.

References

[1] E. Beeching, K. Rasul, Y. Belkada, L. Tunstall, , L. von Werra, N. Rajani, and N. Lambert. Hugging face – the ai community building the future., 2023.

[2] I. Beltagy, K. Lo, and A. Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[3] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[4] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[6] X. Geng, A. Gudibande, H. Liu, E. Wallace, P. Abbeel, S. Levine, and D. Song. Koala: A dialogue model for academic research. Blog post, April 2023.

[7] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding, J. Yue, and Y. Wu. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*, 2023.

[8] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL*, 2020.

[9] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

[10] HuggingFace. Trl training customization.

[11] P. Lewis, M. Ott, J. Du, and V. Stoyanov. Pretrained language models for biomedical and clinical tasks: understanding and extending the state-of-the-art. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 146–157, 2020.

[12] Z. Li, X. Jiang, L. Shang, and H. Li. Paraphrase generation with deep reinforcement learning. *arXiv preprint arXiv:1711.00279*, 2017.

[13] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[14] OpenAi. Proximal policy optimization - spinning up documentation.

[15] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017.

[17] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[18] R. Zhang, J. Han, A. Zhou, X. Hu, S. Yan, P. Lu, H. Li, P. Gao, and Y. Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023.