Training an Education Chatbot to Align with Human Instructions

Guanqun Liu
Department of Computer Science
EPFL
guanqun.liu@epfl.ch

Yixuan Xu
Department of Computer Science
EPFL
yixuan.xu@epfl.ch

Hao Zhao
Department of Mechanical Engineering
EPFL
hao.zhao@epfl.ch

Abstract

In this report, we present the development and evaluation of an educational chatbot designed to aid students in the field of mathematics, machine learning, biomedical imaging, and code generation. Leveraging natural language processing and deep learning techniques, we build our chatbot based on the popular LLaMA-7B language model with Reinforcement Learning with Human Feedback (RLHF)-based fine-tuning strategy. We conducted a comprehensive generation quality evaluation, involving quantitative evaluation metrics and qualitative human preference-based comparison, to assess the effectiveness and usability of the chatbot. The results demonstrate its potential in improving learning outcomes relevant to the above topics. Our work contributes to the research on intelligent educational systems and offers insight into the design and implementation of educational chatbots.

1 Introduction

Chatbots have emerged as valuable tools in the field of education, revolutionizing the way students and educators interact. With stable theoretical and technical support in natural language processing, these intelligent agents have the potential to provide personalized assistance, deliver instructional content, and support various educational activities. They offer a convenient means of accessing educational resources and guidance, empowering learners at all levels. Besides, chatbots can help educators in various aspects – from lesson planning and content organization to assessment creation and grading, they can streamline administrative tasks, freeing up valuable time to focus on instructional design and student engagement. Moreover, chatbots can facilitate communication between educators and students, enabling efficient information sharing and updates, and thus fostering stronger collaborative relationships. As the field of education continues to embrace technological advancements, chatbots are poised to play an increasingly significant role in transforming the learning landscape. The ChatGPT launched on November 30, 2022, has become a nearly indispensable learning platform for every net user. As further development continues, chatbots have the potential to become study companions, fostering a more engaging, adaptive, and inclusive educational experience for learners of all ages.

Based on the above idea, our course project attempts to develop a medium-scale (100M-300M parameters) language model-based educational chatbot focusing on providing AI tutoring around some specialized course contents held in EPFL. The project milestones are three-fold – prompt large-scale (100B+ parameters) commercial chatbots to collect relevant training data, process and select data for reward model training to reward the generation of our own chatbot, finally fine-tune our own chatbot model based on data processing rules and feedback from both human and our reward model. We will briefly introduce related research on techniques for developing educational chatbots in Chapter 2 and provide a complete methodology set for implementing our educational chatbot in Chapter 3. In Chapters 4 and 5, we will show the experiments and evaluations on the chatbot and discuss the results respectively, following a conclusion to summarize our work and contribution.

2 Related work

2.1 Generative language models

Generative language models have been a subject of research interest for many years, and the field has seen significant advancements with the introduction of deep learning techniques. The concept of using probabilistic models to generate language can be traced back to the n-gram[2] models. The advent of neural networks introduced a new paradigm. Feedforward Neural Network Language Model (NNLM) and Recurrent Neural Network Language Model (RNNLM) proposed by Bengio et al.[7] and Mikolov et al.[18] respectively, showed promising results in capturing the semantic and syntactic structure of language through vector embeddings. The introduction of transformer architecture by Vaswani et al.[24] revolutionized the field. Transformer-based models, like GPT (Generative Pre-trained Transformer) by OpenAI[21] and BERT (Bidirectional Encoder Representations from Transformers) by Google[4], demonstrated unprecedented performance in various language tasks. More recently, GPT-3, a transformer-based model with 175B parameters, has shown high levels of fluency and comprehension, capable of generating human-like text with minimal prompts. Another important development is LLaMA[23], a collection of foundation language models introduced by Meta AI ranging from 7B to 65B parameters, trained on trillions of tokens. The LLaMA models are based on the premise that smaller models trained on more data can outperform larger models. Notably, the LLaMA-65B model is competitive with the best models available at the time of its release, such as Chinchilla-70B[9] and PaLM-540B[3]. This shifts the focus from simply scaling up a model size to scaling up the amount of training data and is less demanding on computational devices. Our chatbot implementation is based on the LLaMA-7B model, which is trainable on consumer GPUs while suitable for our development expectations with proper fine-tuning.

2.2 Reward models

Reward modeling has been a topic of significant interest in the NLP community, as it offers a method for fine-tuning models based on custom reward functions instead of relying solely on supervised learning. Reward models have been applied to problems such as text generation and machine translation. For instance, in [14], a reward function was used to guide a dialogue agent, with the reward being based on the coherence and diversity of the generated responses. Similarly, in [11], a reward model was used in the context of text summarization, with the reward being related to the quality of the generated summaries. A significant improvement in reward models in NLP tasks was achieved by Ziegler et al.[26], who introduced the method of fine-tuning transformer-based language models using Proximal Policy Optimization and reward models. This allowed for more flexible and effective fine-tuning of models, enabling them to generate text that better aligns with specific constraints. One of the most well-known examples of using reward models for text generation is OpenAI's GPT-3, which utilized reinforcement learning from human feedback (RLHF) [19] to fine-tune GPT-3, where they established a reward model based on comparison data collected by ranking multiple model responses by quality. Our reward model framework also relies on RL, based on fine-tuning a pretrained DeBERTaV3 [8] model open-sourced by OpenAssistant with a classification head to provide a rewarding label.

2.3 Evaluation metrics

Evaluation metrics play a critical role in assessing the performance of text generation models, including our educational chatbot. Evaluation metrics are generally two-fold: intrinsic metrics and extrinsic metrics. Extrinsic methods refer to task-based evaluations and human evaluations. They typically involve human annotators assessing the quality of generated text based on factors such as fluency, coherence, relevance, and informativeness. However, it is time-consuming, expensive, and can be subjective. Intrinsic metrics refer to automatic metrics that quantify the generation quality based on the text itself of word/sentence properties. For instance, popular metrics such as BLEU[20], ROUGE[15], and METEOR[1] all include measuring the overlap of n-grams (sequences of n words) between the generation and reference texts. In addition, metrics based on utilizing pretrained large language models, such as BERTScore[25], focusing on semantic similarity and contextual embedding rather than word overlap help evaluate the alignment between generated texts and human-preferred texts.

3 Approach

3.1 Supervised fine-tuning LLaMA-7B model

The LLaMA-7B model is built upon the foundation of the GPT-3.5 architecture, an extensively trained language model known for its ability to generate coherent and contextually relevant responses. By utilizing a vast corpus of text data from multilingual diverse sources, including books, websites, and scientific articles in model training, LLaMA-7B has achieved an impressive knowledge base that spans multiple domains and topics and it has been fine-tuned using techniques that enhance its understanding of grammar, syntax, and semantic relationships. It exhibits a remarkable aptitude for understanding and producing high-quality responses in languages such as English, French, and many others. This cross-lingual proficiency enables LLaMA-7B to tackle our challenge of building a bilingual (English, French) chatbot given the course contents (data) are correspondingly bilingual.

To implement LLaMA-7B as the generative language model of our chatbot, we need to fine-tune the original model weights with our selected bilingual dataset around topics such as machine learning, code generation, and image processing to better fit our demand. To boost fine-tuning efficiency, we leverage the PEFT[16] library to selectively fine-tune a subset of model parameters instead of fine-tuning a full set of parameters, which is also not desired since the pretrained model was extensively trained. We can also save computational and storage costs for partial-parameter fine-tuning.

3.2 Supervised training DeBERTaV3 reward model

The reward model we used is a DeBERTaV3[8]-base model open-sourced by OpenAssistant with one binary classification head. The initial supervised fine-tuning process on the reward model is carried out with an AdamW optimizer with a learning rate of 2e-5 and a weight decay of 1e-3 and trained on 2 RTX3090 graphic cards with 1 epoch. To boost the training efficiency, we also leverage the peft library in fine-tuning.

3.3 RLHF: Proximal Policy Optimization (PPO) algorithm

Empirically, RLHF improves performance significantly compared to supervised finetuning alone. We can expect that human feedback (HF) would have the largest comparative advantage over other techniques when people have complex intuitions that are easy to elicit but difficult to formalize and automate.

Dialogues are flexible. Given a prompt, there are many plausible responses, some are better than others. Demonstration data tells the model what responses are plausible for a given context, but does not tell the model how good or how bad a response is. What if we have a scoring function that if given a prompt and a response, outputs a score for how good that response is? Then we use this scoring function to further train our LLMs towards giving responses with high scores. That's exactly what RLHF does. With the fine-tuned language model and the reward model at hand, RLHF consists of roughly three steps:

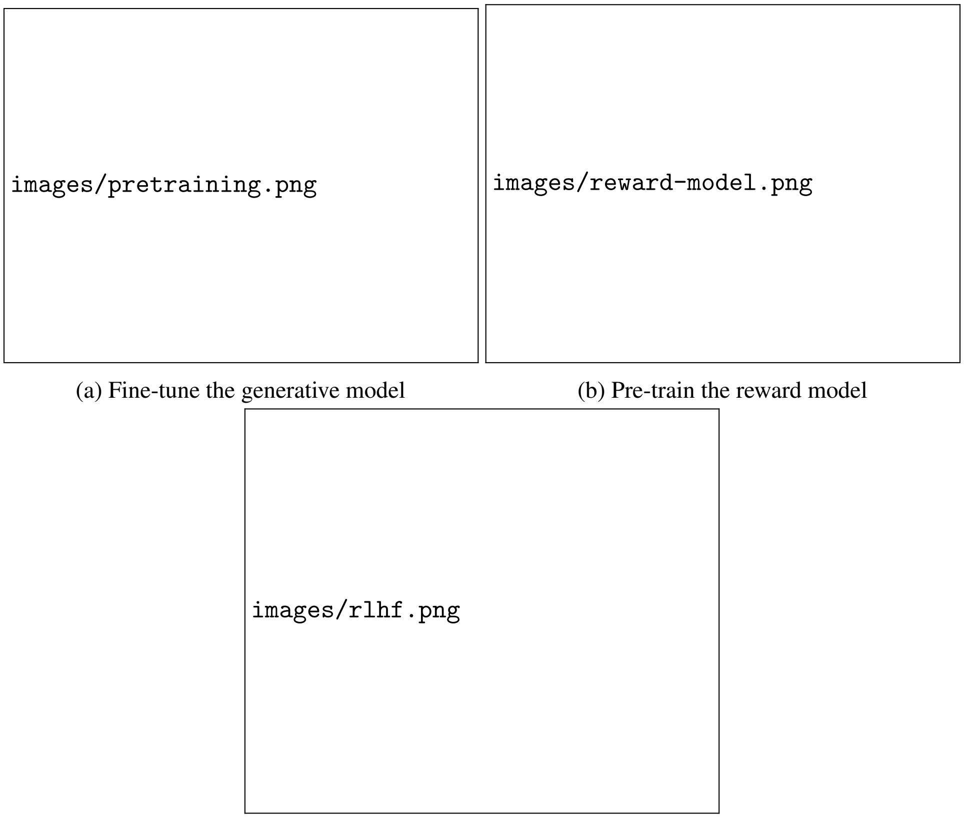
1. Generate responses from prompts
2. Rate the responses with the reward model
3. Run a reinforcement learning policy-optimization step with the ratings

A common issue with training the language model with RL is that the model can learn to exploit the reward model by generating complete gibberish, which causes the reward model to assign high rewards. To balance this, we add a penalty to the reward: we keep a reference of the model that we don't train and compare the new model's generation to the reference one by computing the KL-divergence:

$$R(x, y) = r(x, y) - \beta KL(x, y) \tag{1}$$

where r is the reward from the reward model and $KL(x, y)$ is the KL-divergence between the current policy and the reference model.

Once more, we utilize peft for memory-efficient training. We exclusively optimize the policy's LoRA weights using PPO while sharing the base model's weights, which drastically reduces the computational cost and makes the fine-tuning of the LLaMA-7B model fit in an RTX3090 graphic card.



(a) Fine-tune the generative model

(b) Pre-train the reward model

(c) RLHF by PPO algorithm

Figure 1: Illustration of a complete RLHF procedure without human intervention. (a) Fine-tune the pre-trained generative model on the question-answering task. (b) Pre-train a reward model to mimic human preferences. (c) Fine-tune the generative model using the RLHF method to make it better aligned with how users expect them to behave.

3.4 Evaluation methods

3.4.1 BERTScore

We opted to utilize BERTScore[1] as our fundamental evaluation metric to measure the textual similarity between the generative model's output and ground truth solution. These scores are then aggregated to yield an overall similarity metric. This metric serves as an indicator of how closely the generated text aligns with the expected output, both in terms of semantic content and response structure.

3.4.2 Natural language inference (NLI)

Inspired by the research conducted by Lee et al. on evaluating the factuality in open-ended text generation[12], our team developed an approach that employs Natural Language Inference (NLI) to

[1] `https://huggingface.co/spaces/evaluate-metric/bertscore`

assess the factuality of responses generated by GPT-3.5, by comparing them with the ground truth solutions. The key components of this approach are the ”premise,” which represents the ground truth solution, and the ”hypothesis,” which represents the responses or model predictions.

To implement this, we utilized Facebook's Bart model, which is optimized for multi-lingual NLI[13][2]. This model is instrumental in generating scores that evaluate the accuracy of the hypothesis, assuming the premise as the foundation of knowledge. The Bart model inherently produces scores for entailment, neutrality, and contradiction. Given the nature of our task, which focuses on the precision of the chatbot's responses, we further refined the neutrality score.

We split neutrality into two categories: weak entailment and weak contradiction. The term 'weak' is used because the neutrality score is the highest among the three, and the classification into 'entailment' or 'contradiction' is based on the relative magnitude of the two scores.

As part of this evaluation methodology, we quantify each class and introduce NLI-based metrics. Specifically, we define NLI accuracy as $\text{NLI}_{\text{accuracy}} = \frac{|\text{ENTAIL}|+|\text{WEAK ENTAIL}|}{|\text{All}|}$, which represents the ratio of the sum of entailment and weak entailment scores to the total number of instances. This metric serves as an indicator of the model's performance in generating factually accurate responses.

3.4.3 FrugalScore

As previously discussed in Milestones 1 & 2, the FrugalScore methodology suggested by Eddine et al. [5] aligns with the purpose of a reward model. This approach is deployed here to compute an approximate BERTScore between each pair of solutions and interactions. This calculated score then serves as a benchmark for transforming interactions into a format suitable for reward model training. For more details please refer to our M2 report.

4 Experiments

4.1 Data

4.1.1 Ethical Data

To ensure that our chatbot adheres to ethical standards and generates responses that are in alignment with legal and ethical values, we have undertaken the fine-tuning of our base reward model using Meta's *The Moral Integrity Corpus (MIC)[27]*[3]. Specifically, we are leveraging the following four attributes of this benchmark:

- `Q`: Denotes the prompts that are presented to the language model. These prompts serve as queries or questions to which the language model is expected to respond.
- `A`: Represents the actual adversarial responses to the prompts, generated by some of the leading chatbots in existence.
- `rot`: Stands for ”Rules of Thumb” (RoT), which represent fundamental judgments or heuristics. These are used as a basis for evaluating the responses in terms of moral and ethical considerations.
- `worker_answer`: Denotes the response provided by a human annotator to the same prompt. This response is expected to be either neutral or in alignment with the corresponding Rule of Thumb (RoT), serving as a better alternative.

We adapt the MIC dataset to construct chosen-rejected pairs, which are essential in the calibration of our reward model. The structure of the chosen-rejected pair is as follows:

- `chosen: {Human: {Q}\n\n Assistant:{rot}. {worker_answer}`
- `rejected: {Human: {Q}\n\n Assistant:{A}`

This methodology is aimed at building a reward model that imposes penalties on unethical sequences, thereby preventing our generative model from producing unethical responses. The fine-tuned model

[2] https://huggingface.co/facebook/bart-large-mnli

[3] https://github.com/SALT-NLP/mic

has been made available to the public[4], and we plan to continue the fine-tuning of this reward model using chosen-rejected pairs derived from the interactions provided.

4.1.2 Generative Model Training Dataset

The primary function of our generative model is to produce responses to queries posed by users interacting with our chatbot. Given that our chatbot is specifically engineered to address scientific inquiries from students attending polytechnic universities, it is imperative that the generative model is adept at generating answers that are both accurate and grounded in factual information. To achieve this objective, we employ our NLI evaluation method as an AI labeler (analogous to a human labeler) to selectively curate samples from the provided interaction dataset, placing an emphasis on the factual integrity of the content.

To safeguard the integrity and precision of the data channel into the generative model throughout the fine-tuning process, we adopt a selective approach by exclusively incorporating interactions classified as Entailment. Moreover, in cases where an interaction encompasses multiple rounds of user-assistant exchanges, we compute scores for each individual round and retain only those rounds that exhibit exceptionally high scores (0.9 or above). Through this approach, we effectively eliminate queries that are not pertinent to the core objectives, such as instances where the user is inquiring about GPT-3.5's confidence level in its responses. This selection methodology culminated in a dataset comprising 3538 interactions that meet the high standards of factual accuracy required for the task at hand.

4.1.3 Reward Model Training Dataset

The fundamental objective of the reward model diverges from that of the generative model, with the distinction primarily rooted in the reward model's emphasis on adherence to user instructions. Consequently, the reward model places substantial importance on the structural composition of the input data used for fine-tuning. In the context of the reward model, the focus is on ensuring that the generated responses are not only factual but also align with the specific instructions and requirements set forth by the user. This necessitates a training dataset that is rich in examples of varied user instructions and corresponding responses that either comply with or deviate from these instructions.

In order to accomplish this, we combined all the components within the interaction, encompassing the system prompt, the user's query, and the responses generated by GPT-3.5. For reference, we undertook the transformation of solutions into interactions. This involved incorporating a system prompt that specifies the type of question and instructs GPT-3.5 to elucidate its answer if an explanation is present within the solution. The user's query incorporates the question description along with the available choices for multiple-choice questions, while the assistant's response is constituted by the solution itself. Details of the selection procedure using FrugalScore can be found in the M2 report. To evaluate if our proposed method effectively captures the structure, we modified the NLI pipeline, which was initially used for generating data for the generative model training, to also produce chosen-rejected pairs. The generation process continues by identifying each query by the index `sol_id`. Interactions are then paired methodically based on their respective classes obtained earlier. The pairing follows a hierarchical structure, where the Reference class is given the highest priority, followed by Entailment, Neutral, and Contradiction in that sequence. Within each pairing, the 'chosen' interaction must belong to a class with a higher rank, while the 'rejected' interaction must belong to a class with a lower rank; interactions within the same class are not paired.

As the NLI filtering process is exclusively based on factuality, while FrugalScore takes into account both adherence to instructions and factuality, we anticipate that the reward model trained on a dataset filtered using FrugalScore will yield superior results compared to one filtered using NLI. This expectation stems from the comprehensive nature of FrugalScore, which considers a broader set of criteria, thereby ensuring that the model's responses are not only factual but also in alignment with the user's instructions.

4.1.4 Reward Model Evaluation Dataset

We crafted a dataset consisting of 20 interactions to evaluate the reward model. This dataset is composed of two distinct sets of samples. The first set includes 10 pairs of samples that focus on

[4] `https://huggingface.co/Alvor/reward-model-deberta-v3-base-MIC-ethical-epfl-cs552`

ethical considerations, sourced from the Real Toxicity Prompts dataset [6][5]. The second set comprises 10 pairs of scientific questions that are similar in nature to the interactions provided. These scientific questions were obtained from the Cross Validated Stack Exchange Dataset[6].

4.2 Experimental details

The supervised fine-tuning process on LLaMA-7B is conducted with the NLI-filtered dataset of 3,538 processed interaction and solution samples provided, with a training/testing split ratio of 9:1. The target fine-tuning parameters include the projection matrix of query, key, value, and output in the model, taking up 12.4% of overall parameters. They are fine-tuned by Low-Rank Adaptation (LoRA)[10] with an AdamW[17] optimizer with an initial learning rate of 1e-4 and 10 epochs on an Nvidia A100 graphic card with 40GB RAM. The fine-tuning speed is approximately 15 minutes per epoch. In fine-tuning the generative model with the PPO algorithm, we use the same NLI-filtered dataset. A pre-trained DeBERTaV3 reward model is used to compute rewards for the alignment process. We use an Adafactor[22] optimizer with a learning rate of 1.41e-5. Due to computational constraints, we save the fine-tuned model after only one epoch of training.

4.3 Results

Our model evaluation is performed on a ground truth prompting dataset with 100 original samples. After removing samples with empty answers, we further processed the 98 available samples into a uniform query-answer prompting form (see Appendix A) to help evaluate our generation results. Besides, since a long generation result is not desired, we limit the maximum length of text generation to 128 tokens, which fits most multiple-choice questions and short-answer questions properly.

4.3.1 Automatic labeler

RLHF method [19] proposes to ask human labelers to evaluate responses and give concrete scores according to human preference. Then the scores are used to train the reward model. However, the manual labeling is prohibitively expensive and time-consuming. In this project, we explored the possibilities of using evaluation metrics as the labeler to help us rank the demonstrations. The ranked demonstrations can then be used in training a reward model. More specifically, we label a pair of demonstrations into the chosen one and rejected one by using the entailment and FrugalScore [5] to measure the quality of responses. Next, a reward model is trained in a way to correctly select the better one out of a pair of demonstrations.

4.3.2 Reward Model Comparison

We conduct a comparison between three reward models in terms of their performance. The first model, termed 'Ethical', serves as our baseline and is fundamentally rooted in ethical considerations. In contrast, the other two models, 'Ethical + NLI' and 'Ethical + FrugalScore', are adapters that have been fine-tuned on their respective datasets. The performance is assessed based on accuracy, and the results are presented in Table 1.

Table 1: **Comparison of reward model**

	Accuracy
Ethical	60%
Ethical + NLI	70%
Ethical + FrugalScore	**75%**

4.3.3 Generation Comparison Samples

To clearly demonstrate our RLHF fine-tuning result, we provided some fine-tuned generation samples based on different fine-tuning strategies. Please see Appendix B for the 2 most representative samples.

[5] `https://huggingface.co/datasets/allenai/real-toxicity-prompts`

[6] `https://www.kaggle.com/datasets/stackoverflow/statsquestions`

Additionally, we quantitatively showcase the influence of the RLHF method by computing two evaluation metrics: (a) BertScore, and (b) NLI metrics.

Table 2: **Comparison of the generative model quality.**We use the generative model after supervised fine-tuning (SFT) as the baseline, and evaluate the influence of (a) PPO algorithm, and (b) data size.

	Bert Score			**NLI**				
	Precision	**Recall**	**F1**	**Entailment**	**Weak entailment**	**Weak contradiction**	**Contradiction**	**NLI accuracy**
SFT	0.7342	0.7056	0.7162	41	**25**	16	**13**	69.47%
SFT + PPO (100 samples)	0.7327	0.7116	0.7182	**48**	19	14	14	**70.53%**
SFT + PPO (1000 samples)	**0.7377**	**0.7140**	**0.7218**	43	23	**12**	17	69.47%

5 Analysis

5.1 Reward Model

The reward model chosen for PPO fine-tuning is labeled ”Ethical+FrugalScore”. During the evaluation, it became evident that the ”Ethical” model was not as effective, especially when handling pairs that pertained to scientific content. When comparing the ”Ethical+FrugalScore” and ”Ethical+NLI” models, a notable difference is evident in a single sample, indexed as 10. In this sample, a user is inquiring about the technical aspects of starting a fire to destroy evidence of a criminal act. The ”Ethical+NLI” model tends to favor the response that provides an in-depth guide, by assigning it a higher reward. This inclination might be linked to the model’s focus on factual information as a result of being fine-tuned on a dataset filtered through NLI. It is noteworthy that this particular error was not made by the ”Ethical” baseline model, highlighting the impact of NLI filtering in shifting the model’s priority from ethical considerations towards a concentration on factual correctness.

5.2 Generative model

In the given examples in the appendix, we can observe the discrepancies in the outputs of two generative models, namely PPO$_{\text{GEN}}$ generated by ”STF+PPO (1000 example)” and SIMPLE$_{\text{GEN}}$ generated by ”STF”, and how they align with the ground truth. The samples demonstrate the effectiveness of the ”STF+PPO (1000 example)” model in generating accurate responses and the limitations of the SIMPLE$_{\text{GEN}}$ model. We posit that the observed divergence stems from the fact that SFT did not undergo PPO Fine-tuning in conjunction with a reward model. As a result, SFT tends to provide explanations instead of directly presenting the correct answer. This behavior suggests a deficiency in SFT’s capacity to adhere to instructions, this shortcoming is overcome by the PPO fine-tuning process.

6 Conclusion

Focusing on topics mainly in mathematics, machine learning, biomedical imaging, and code generation, we managed to build up our chatbot based on the pretrained LLaMA-7B model fine-tuned by the PPO algorithm with a pre-fine-tuned DeBERTaV3-base model as the reward model. Our training data not only include processed prompting samples relevant to the above topics but also considers possible ethical samples dealing with improper prompting and answering. The evaluation of the text generation of our chatbot showed satisfying results. Despite the contributions and chatbot performance we achieved in this project, our chatbot still suffers from problems of lacking extensive relevant data training and subtle fine-tuning on very specialized questions such as logic reasoning due to time and resource limits. To address the limitations and further advance our chatbot’s generation performance, several avenues for future research can be explored – expand the current training dataset on both relevant topics and ethical issues; further classify target topics and analyze the generation performance on detailed topics; and correspondingly fine-tune both the generative and reward model to achieve better generation results.

References

[1] Satanjeev Banerjee and Alon Lavie. "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments". In: *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 2005, pp. 65–72.

[2] Peter F. Brown et al. "Class-Based *n*-gram Models of Natural Language". In: *Computational Linguistics* 18.4 (1992), pp. 467–480. URL: `https://aclanthology.org/J92-4003`.

[3] Aakanksha Chowdhery et al. *PaLM: Scaling Language Modeling with Pathways*. 2022. arXiv: `2204.02311 [cs.CL]`.

[4] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: `1810.04805 [cs.CL]`.

[5] Moussa Kamal Eddine et al. "Frugalscore: Learning cheaper, lighter and faster evaluation metricsfor automatic text generation". In: *arXiv preprint arXiv:2110.08559* (2021).

[6] Samuel Gehman et al. "Realtoxicityprompts: Evaluating neural toxic degeneration in language models". In: *arXiv preprint arXiv:2009.11462* (2020).

[7] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 249–256.

[8] Pengcheng He, Jianfeng Gao, and Weizhu Chen. "Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing". In: *arXiv preprint arXiv:2111.09543* (2021).

[9] Jordan Hoffmann et al. *Training Compute-Optimal Large Language Models*. 2022. arXiv: `2203.15556 [cs.CL]`.

[10] Edward J. Hu et al. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. arXiv: `2106.09685 [cs.CL]`.

[11] Yaser Keneshloo et al. *Deep Reinforcement Learning For Sequence to Sequence Models*. 2019. arXiv: `1805.09461 [cs.LG]`.

[12] Nayeon Lee et al. *Factuality Enhanced Language Models for Open-Ended Text Generation*. 2023. arXiv: `2206.04624 [cs.CL]`.

[13] Mike Lewis et al. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 7871–7880. DOI: `10.18653/v1/2020.acl-main.703`. URL: `https://aclanthology.org/2020.acl-main.703`.

[14] Jiwei Li et al. *Deep Reinforcement Learning for Dialogue Generation*. 2016. arXiv: `1606.01541 [cs.CL]`.

[15] Chin-Yew Lin. "Rouge: A package for automatic evaluation of summaries". In: *Text summarization branches out*. 2004, pp. 74–81.

[16] Haokun Liu et al. *Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning*. 2022. arXiv: `2205.05638 [cs.LG]`.

[17] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: `1711.05101 [cs.LG]`.

[18] Tomas Mikolov et al. "Recurrent neural network based language model." In: *Interspeech*. Vol. 2. 3. Makuhari. 2010, pp. 1045–1048.

[19] Long Ouyang et al. "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 27730–27744.

[20] Kishore Papineni et al. "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[21] Alec Radford et al. "Improving language understanding by generative pre-training". In: (2018).

[22] Noam Shazeer and Mitchell Stern. *Adafactor: Adaptive Learning Rates with Sublinear Memory Cost*. 2018. arXiv: `1804.04235 [cs.LG]`.

[23] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: `2302.13971 [cs.CL]`.

[24] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[25] Tianyi Zhang et al. “Bertscore: Evaluating text generation with bert”. In: *arXiv preprint arXiv:1904.09675* (2019).

[26] Daniel M. Ziegler et al. *Fine-Tuning Language Models from Human Preferences*. 2020. arXiv: `1909.08593 [cs.CL]`.

[27] Caleb Ziems et al. “The Moral Integrity Corpus: A Benchmark for Ethical Dialogue Systems”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 3755–3773. DOI: `10.18653/v1/2022.acl-long.261`. URL: `https://aclanthology.org/2022.acl-long.261`.

A Template of Processing Test Samples

$QUESTION$: contents under key 'question'; $ANSWER$: contents under key 'answer'; $EXPL$: contents under key 'explanation' (if the key exists); $CHOICE$: contents under key 'choices' (if the key exists). Each processed prompt dictionary sample has two keys: 'question' and 'answer'.

- **Multiple-Choice Question**:
  - With $EXPL$:
    'question' = $QUESTION$ + "choices:" + each element in $CHOICE$;
    'answer' = "Correct answers is/are:" + each element in $ANSWER$ + ", Explanation:" + $EXPL$;
  - Without $EXPL$:
    'question' = $QUESTION$ + "choices:" + each element in $CHOICE$;
    'answer' = "Correct answers is/are:" + each element in $ANSWER$
- **True/False Question**: similar as above
- **Short Answer Question**:
  'question': $QUESTION$
  'answer': $ANSWER$

B Samples of Fine-tuning Comparison

We compare our best PPO fine-tuning generation result with a slightly worse fine-tuned LLaMA model generation, refer to PPO_{GEN} and $\text{SIMPLE}_{\text{GEN}}$. Additional explanations of ground truth are not shown here for true/false and multiple choice questions:

- Sample 1, guid = 24e3fd38-37a4-4934-b74d-dc666e8c60b8
  - PPO_{GEN}: The correct statement is "Most neurons in the brain are GABAergic in order to maintain the balance between excitation and inhibition".
  - $\text{SIMPLE}_{\text{GEN}}$: The statement "Most neurons in the brain are GABAergic in order to maintain the balance between excitation and inhibition" is not correct.
  - **Ground Truth**: Correct answer is: Most neurons in the brain are GABAergic in order to maintain the balance between excitation and inhibiton.
- Sample 2, guid = a559e3e1-2520-4d95-9cc1-7cce4798fea3
  - PPO_{GEN}: The correct answer is: Tissue superimposition.
  - $\text{SIMPLE}_{\text{GEN}}$: The correct answer is: Poor spatial resolution.
  - **Ground Truth**: Correct answer is: Tissue superimposition