RLHF on Sentence Generating

DDA4210 Final Report

119010233 Ma Yufei
120090339 Sui Haoran
120090784 Yin Qihua
120040065 Yao Xingtong

1 Introduction

Reinforcement learning (RL) has made significant strides in various domains, including natural language processing (NLP). In this report, we explore the integration of RL with human feedback for improving sentence generation. We discuss the research's significance and novelty.

1.1 Significance

Incorporating human feedback in RL for sentence generation provides several benefits: it captures language nuances, addresses biases and fairness issues, enables personalized interactions, and fosters more accurate and coherent sentence generation. Moreover, it promotes the development of interpretable and controllable AI systems.

1.2 Novelty

The combination of RL and human feedback is a relatively new approach in sentence generation. Our research introduces a novel framework that merges both methods, leveraging human feedback during training and traditional supervised learning for fine-tuning. We examine the role of various feedback modalities, such as rankings, comparisons, and rewards, to optimize the learning process. Our findings contribute to the growing literature on RL and NLP, shedding light on effectively incorporating human feedback for enhanced sentence generation.

2 Data

2.1 Data Collection

Downloaded from Kaggle, the dataset contains more than 2 million comments of 28 movies in Douban Movie website,which is a widely used Chinese movie review website. The features include:

- **ID:** the ID of the comment (start from 0)
- **Movie_Name_EN:** the English name of the movie
- **Movie_Name_CN:** the Chinese name of the movie
- **Crawl_Date:** the date that the data are crawled
- **Number:** the number of the comment
- **Username:** the username of the account
- **Date:** the date that the comment posted
- **Star:** the star that users give to the movie (from 1 to 5, 5 grades)
- **Like:** the count of "like" on the comment

2.2 Data Preprocessing

First, to make sure the data we input to the model is valid and meaningful, we first removed all the emojis, websites and redundant punctuation using regular expression.

Second, the dataset needs to be ordered from positive emotion to negative emotion according to the comments to do the sentiment analysis. The *Star* is selected to measure the emotions conveyed by the comments. We consider the higher *Star* given by the user, the more positive emotion the user convey. Thus, we selected the *Comment* feature and rank them according to the *Star* feature in a descending order.

Third, for training and validation, the dataset is split into 8:2 as training set and validation set separately.

The tokenization is completed by the *AutoTokenizer* function from *Transformer*.

3 Method

3.1 Language Model

3.1.1 introduction

Language models play a vital role in natural language processing (NLP) tasks. Among various language models, GPT-2 is one of the most prominent ones, which has achieved state-of-the-art performance on many NLP benchmarks. In this part, we will discuss how we utilized GPT-2 to generate sentences in a Chinese conversational setting and its role in our RLHF project.

3.1.2 GPT-2 for Sentence Generation

GPT-2 is an unsupervised language model that is based on the transformer architecture. The model consists of multiple layers of self-attention and feed forward neural networks. More specifically, GPT-2 is a generative model, which means that it can generate new text that follows the underlying distribution of the training data.

To use GPT-2 for sentence generation, we first fine-tuned it on our dataset, a preprocessed large Chinese conversation corpus. We trained the model to predict the next word given the previous context, while also conditioning it on a conversational start token for each sentence. This way, the generated sentences would follow a conversational style and be more relevant to our RLHF project.

After fine-tuning, we can generate new sentences by sampling from the output distribution of the model. Given a prompt or context, we can feed it into the model and let it generate the next word, and repeat this process until we reach the desired length or terminate the sentence. The sampling technique we used is called top-p sampling, which selects the most probable words that cumulatively exceed a certain probability threshold. This way, we can control the diversity and randomness of the generated sentences. Moreover, we can also perform beam search, which explores multiple hypotheses and selects the most likely one by maximizing the joint probability of the whole sentence.

3.1.3 Applications in RLHF

The human feedback system of our project starts with the GPT-2-generated sentences, which are used as the language input. The purpose of the system is to obtain high-quality feedback from human annotators on the positivity of the emotion expressed in each sentence. To facilitate this process, we created an interface that presents the generated sentences to a human player, who must rate its emotion on a scale from -5 to 5. Using this feedback, we can train and optimize our core component, the Reward Model, and generate a numerical reward signal that guides the GPT-2 model's training, which, in turn, determines the output of the language model.

3.2 Reward Model

3.2.1 Sorting Sequence

We refer to chatGPT's reward model training method. A RM is trained on human-annotated sorted sequences. The purpose is not to let humans directly feedback the true score of each sentence, but let people sort the sentences according to their degree of quality specified by humans. Because it is difficult to unify the standard to directly score the generated text. For the same generated answer, if some annotators score 5 points and some annotators score 3 points, it will be difficult for the model to know whether the sentence is good or not when learning. Since it is difficult to unify the absolute scores, it is much easier to convert this task into a relative ranking task. So we expect to train a Reward model through ranking sequence: when the sentence is more positive, the higher the Reward the model gives.

3.2.2 Rank Loss

Suppose now there is a sorted sequence: $A > B > C > D$. We need to train a model whose scores for four sentences must satisfy the sequence $r(A) > r(B) > r(C) > r(D)$. The rank loss should be:

$$loss = r(A) - r(B) + r(A) - r(C) + r(A) - r(D) + r(B) - r(C) + ... + r(C) - r(D)$$
$$loss = -loss$$

We use the following loss function:

$$loss(\theta) = -\frac{1}{\binom{K}{2}} E_{(x,y_w,y_l)\sim D}[log(\sigma(r_\theta(x,y_w) - r_\theta(x,y_l)))]$$

where $r_\theta(x, y)$ is the scalar output of the reward model for prompt x and completion y with parameter θ, y_w is the preferred completion out of the pair of y_w and y_l, and D is the dataset of human comparison.
The value of loss is equal to the sum of the rewards of all the previous items in the sorted list minus the rewards of the later items. We hope that the model can maximize the difference between the good sentence score and the bad sentence score. Since the gradient descent is the minimization operation, we need to take a negative number for loss to achieve the effect of maximizing the difference. Since the sentences in the data set have been arranged from positive to negative during preprocessing, we only need to traverse and add up the score difference of the front and back items.

4 Results

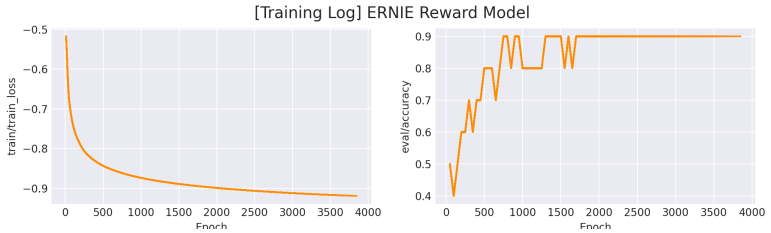

Figure 1: Training Curve of Reward Model for Presentation data set

The above Figure 1 shows the training curve for RM using data set for the presentation, which contains 7.9 MB for train.tsv and 2 MB for dev.tsv. The train loss and the accuracy for this RM both converges after multiple iterations.

Figure 2 shows the the training curve for RM using the movie comments data set, which is introduced in detail in $Data$ section. During more than 3500 iterations the training loss converges, but the accuracy of the RM does not. After reaching peak at around the 500th iteration, the accuracy curve shows a periodic rises and falls and the maximum never exceeds 0.55. We speculate that this is because the RM reachs a local optimum, but the loss function matches the data set used in the presentation better instead of the report data set.

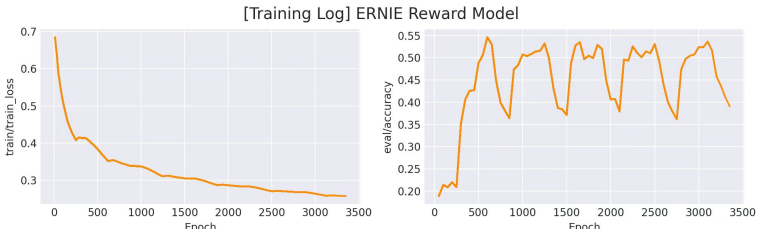

Figure 2: Training Curve of Reward Model for Report data set

Figure 3: Testing Output for the Reward Model

The reward model would be saved as a checkpoint in each 50 iterations. The model with the highest accuracy is set as the model_best. Figure 3 shows a sample that how model_best scores two completely opposite movie comments.

5 Conclusion

Reviewing the whole project, we basically achieved our goal of achieving an reinforcement learning with human feedback. We use our own data to train a reward model and use the reward model to implement RLHF. We met many difficulties on the way to achieve our goal. We tried many models before we found a language model that could be used for reinforcement learning in the specific situation of the group. In order to obtain the data set of the reward model trained to evaluate film reviews, we searched the relevant data and performed a lot of processing on it.From the complexity of the model to the computational power required, RLHF is indeed a very challenging research project.

RLHF is an important research direction in natural language processing, but it has a lot of potential to be explored. Many Chinese language models can be further fine-tuned in this way. More specific data sets can be used to train text generators for specific application scenarios.The experience we have learned from this project will lay a solid foundation for our future research.

6 Reference

https://wandb.ai/ayush-thakur/RLHF/reports/Understanding-Reinforcement-Learning-from-Human-Feedback-RLHF-Part-1–VmlldzoyODk5MTIx
https://huggingface.co/uer/roberta-base-finetuned-jd-binary-chinese
https://huggingface.co/uer/gpt2-chinese-cluecorpussmall
https://github.com/lvwerra/trl
https://arxiv.org/abs/2203.02155
https://www.kaggle.com/datasets/utmhikari/doubanmovieshortcomments