GENERATING LESS TOXIC CONTENT USING REINFORCEMENT LEARNING ON LLMs

PROJECT DESCRIPTION

Large language models (LLMs) have grown significantly in capability, enabling them to produce complex and nuanced text. However, this power also raises concerns, as these models can inadvertently generate harmful content, including hate speech and misinformation. Addressing this issue is crucial to ensure that the use of LLMs aligns with ethical standards and promotes positive societal impacts.

Our approach to mitigate this risk is using reinforcement learning (RL), a method within machine learning. In RL, a model is trained to minimize toxic output by implementing a system of rewards and penalties: it receives incentives for producing safe, non-toxic content and consequences for generating unacceptable content.

ENVIRONMENT

The "environment" refers to the operational setup, including the data, models, and evaluation metrics used for the RL-based fine-tuning process, rather than a specific software environment designed for RL experiments. The RL techniques are applied to the domain of NLP and content generation with the T5 model, focusing on reducing hate speech in generated outputs. For implementing such an RL setup, we are using a combination of machine learning frameworks (PyTorch) and libraries specialized for working with large language models (Hugging Face's Transformers). These tools provide the necessary infrastructure to simulate the RL environment for training and evaluating the model based on interactions with the data and the reward, even though it does not specify a particular RL simulation environment by name.

KEY COMPONENTS

1. *Hugging face libraries:* Hugging Face is a machine learning (ML) and data science platform and community that helps users build, deploy and train machine learning models. We have many hugging face components used in our project, transformers, dataset on which the model is trained, PEFT adapter and so on.
2. *Transformer:* Text-To-Text Transfer Transformer (T5) is a pre-trained encoder-decoder model handling all NLP tasks as a unified text-to-text-format where the input and output are always text strings. T5-Small is the checkpoint with 60 million parameters.

3. *Dataset:* DialogSum is a large-scale dialogue summarization dataset, consisting of 13,460 dialogues (+1000 tests) split into train, test and validation. Fields of the dataset:

   dialogue: text of dialogue.
   summary: human written summary of the dialogue.
   topic: human written topic/one liner of the dialogue.
   id: unique file id of an example.

4. *PEFT (Parameter Efficient Fine-Tuning):* Parameter Efficient Fine-Tuning (PEFT) presents a viable approach by diminishing the number of fine-tuning parameters and memory consumption, all while maintaining performance levels comparable to those achieved through full fine-tuning.
5. *QLoRA (Quantized Low-Rank Adaptation):* QLoRA, integral to PEFT, functions by integrating fresh, trainable parameters into the model without augmenting its overall parameter count. This strategy, akin to the adapter approach, maintains the model's size while leveraging parameter-efficient fine-tuning. QLoRA utilizes quantization methods to decrease memory usage while preserving or potentially improving model performance. It introduces innovations like 4-bit Normal Float, Double Quantization, and Paged Optimizers to achieve superior computational efficiency with minimal storage demands.
6. *Reward model (DaNLP/da-electra-hatespeech-detection):* The hate speech model, is a Danish language model fine-tuned on Facebook data. It accurately predicts the offensiveness of text, categorizes hate speech into specific types such as personal attacks and language usage, and provides valuable insights for content moderation and fostering a safer online environment. It can maximum take 512 tokens as input at a time. The sentences are automatically truncated if longer.
7. *PPO (Proximal Policy Optimization):* Once the reward model is established, the language model undergoes additional training, which prioritizes maximizing the rewards (score) predicted by the reward model. This phase employs techniques from reinforcement learning, such as Proximal Policy Optimization (PPO), to fine-tune the language model's parameters, aiming to generate outputs more likely to garner high rewards. Through this iterative approach, the language model progressively learns to generate responses that align better with human preferences.

METHODOLOGY

- *LLM Fine-tuning*
    - Pre-trained model - 'google-t5/t5-small' : Begin by choosing a base model for fine-tuning. In this case, the 'google-t5/t5-small' model is selected. This

model is part of the T5 (Text-to-Text Transfer Transformer) series developed by Google, known for its versatility in handling a variety of text-based tasks. The 'small' variant is chosen for its balance between performance and resource efficiency, also for the less number of parameters, making it suitable for experimental setups and initial prototypes.

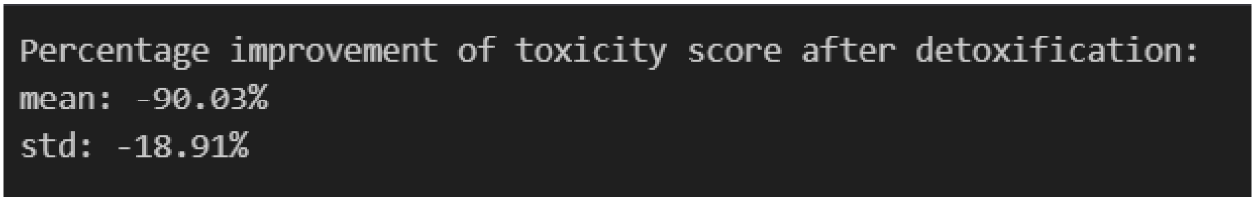
- Dataset - 'knkarthick/dialogsum' : Collect the necessary data for training. The dataset 'knkarthick/dialogsum' is specifically chosen, which is likely a collection of dialogues designed for summarization tasks. This dataset will provide the conversational content needed to fine-tune the model on dialogue understanding and summarization.
- Preprocess Dataset :
    - Install required libraries : Ensure that all necessary Python libraries and dependencies are installed to handle dataset loading, preprocessing, and model training. This might include libraries like transformers, datasets, and others specific to handling large language models.
    - Loading dataset : Load the 'dialogsum' dataset into the working environment, preparing it for preprocessing and fine-tuning.
    - Create Bitsandbytes configuration : Optimize model training by configuring bitsandbytes, a library that enhances the performance of transformer models through efficient memory usage and faster processing.
    - Loading the Pre-Trained model : Load the initially selected 't5-small' model. This step is critical to begin the fine-tuning process, utilizing the pre-trained weights as the starting point.
    - Tokenization : Convert the raw text data from the dataset into a format that the model can understand, typically involving converting text into tokens or small chunks of text represented by numerical IDs.
    - Test the Model with Zero Shot Inferencing : Evaluate the fine-tuned model's ability to perform tasks it hasn't been explicitly trained on, using zero-shot learning techniques. This involves providing the model with a task description and observing how well it can generate appropriate responses based on its general understanding and adaptations from the fine-tuning process.
- Fine-tuning - Parameter Efficient Fine-Tuning (PEFT), achieved PEFT using QLoRA (Low-Rank Adaptation)
    - Preparing the model for QLoRA: This step involves configuring the language model to utilize QLoRA (Quantized Low-Rank

Adaptation), a method that modifies only a small part of the model’s parameters, specifically targeting those parameters that can provide significant benefits in learning new tasks while minimizing computational overhead.

        - Setup PEFT for Fine-Tuning: Establish the environment and settings for Parameter Efficient Fine-Tuning (PEFT), which ensures that the model's adaptation to new data (like summarization) uses fewer resources and is more scalable.
        - Train PEFT Adapter: Train the adapter modules added to the base model that enable efficient learning of new tasks without extensive re-training of the entire network.
        - Prepare a PPO model & a Reference model: Set up a Proximal Policy Optimization (PPO) model, a type of reinforcement learning algorithm, alongside a reference model for comparative analysis and validation during the training process.
- *Reward Model*
    - ‘DaNLP/da-electra-hatespeech-detection’ : Utilize this specific model to detect hate speech within the text. This model categorizes text into offensive or non-offensive classes, aiding in the identification of undesirable content.
    - Evaluate toxicity: Metric used is 'evaluate-measurement/toxicity'. Employ a predefined metric for assessing the level of toxicity in the content generated by the model, ensuring the output meets acceptable standards of discourse.
- *Fine-Tuning to Detoxify the Summaries*
    - PPO - RL Algorithm
        - Implement the Proximal Policy Optimization (PPO) algorithm, a reinforcement learning approach, to further fine-tune the summaries generated by the model. This involves adjusting the model’s responses based on a reward system that discourages the generation of toxic content.
        - Initialize PPOTrainer : Start the PPOTrainer with specific configurations to optimize the learning process towards generating non-toxic, high-quality text summaries. We load the ppo_model and the tokenizer. We will also load a frozen version of the model ref_model. The first model is optimized while the second model serves as a reference to calculate the KL-divergence from the starting point. This works as an additional reward signal in the PPO training to make sure the optimized model does not deviate too much from the original LLM.

- Fine Tune the Model
    - Get the query responses from the policy LLM (PEFT model): Process of extracting responses from the language model that has been adapted using the Parameter Efficient Fine-Tuning (PEFT) approach. This involves running specific queries through the model and observing the responses to evaluate how well the model is performing in real-time scenarios.
    - Get sentiments for query/responses from 'da-electra-hatespeech-detection' model: Response from the LLM is passed through the 'da-electra-hatespeech-detection' model to assess whether the content is offensive or not. This step is crucial for determining the initial quality and safety of the responses.
    - Optimize policy with PPO(query, response, reward): PPO to fine-tune the model's responses based on a reward system. The reward is calculated based on the sentiment analysis from the hate speech detection model, thus promoting responses that are deemed non-offensive and penalizing inappropriate outputs.
- Evaluate the Model
    - After detoxification, we load the PPO/PEFT model back & use the test dataset split to evaluate the toxicity score of the RL-fine-tuned model: In the evaluation phase where the model, now refined with reinforcement learning strategies to reduce toxicity, is tested again. This test uses a separate portion of the dataset to ensure that the evaluation is unbiased and represents the model's general performance.
    - Compare the toxicity scores of the reference model (before detoxification) and fine-tuned model (after detoxification): A comparative analysis of toxicity scores between the original model and the one that has undergone detoxification. This comparison will highlight the effectiveness of the fine-tuning process in reducing undesirable content in the model's outputs.
    - Compare the results of the PPO model with the Reference model: Finally, comparison of the overall performance, including aspects like response quality and adherence to content guidelines, between the original reference model and the newly optimized PPO model. This will help quantify the improvements made through the fine-tuning processes.

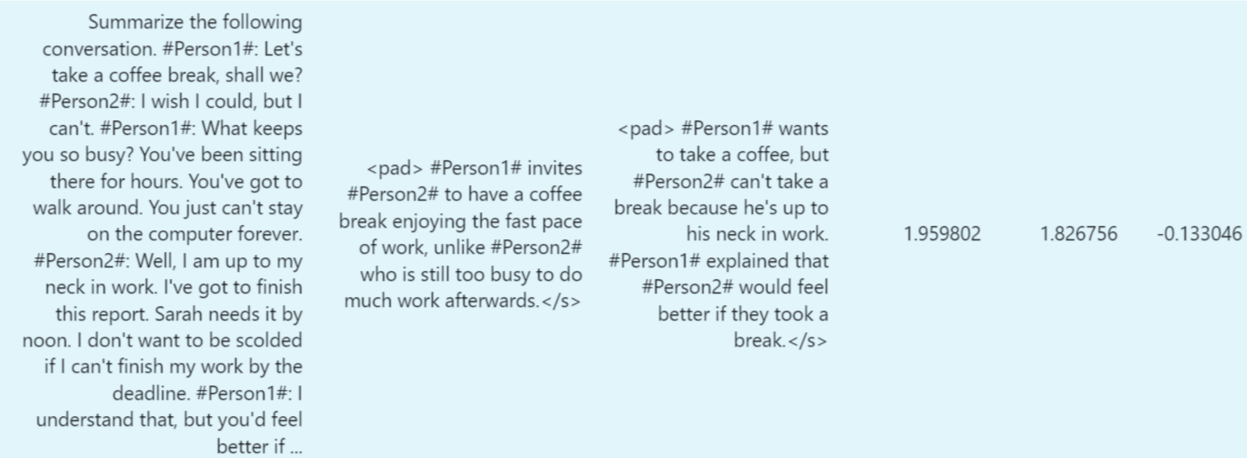
RESULTS



```
Percentage improvement of toxicity score after detoxification:
mean: -90.03%
std: -18.91%
```

The negative values for the percentage improvement in toxicity scores indicate that there has been a reduction in toxicity levels, which is indeed a positive outcome. Here's a breakdown of what these numbers mean and why they are presented as negative:

Mean Toxicity Score Improvement (-90.03%): This value represents the change in the average toxicity score from the reference model to the RL-fine-tuned model after detoxification. A -90.03% change means that the average toxicity score of the RL-fine-tuned model is 90.03% lower than that of the reference model. The negative sign indicates a reduction, which in this context, is beneficial as it implies less toxicity.
Standard Deviation of Toxicity Score Improvement (-18.91%): This value shows the change in the variability of toxicity scores. A -18.91% change indicates that there is 18.91% less variation in the toxicity scores in the RL-fine-tuned model compared to the reference model. Again, the negative sign denotes a decrease in variability, suggesting that the RL-fine-tuned model's scores are not only lower on average but also more consistent (less spread out), which is generally desirable.
In summary, the negative percentages are good in this scenario because they signify a decrease in both the average toxicity and its variability, thus showing that the detoxification process has effectively made the RL-fine-tuned model less toxic. This is a desired outcome in efforts to create more ethical and less harmful AI systems.



Summarize the following conversation. #Person1#: Let's take a coffee break, shall we? #Person2#: I wish I could, but I can't. #Person1#: What keeps you so busy? You've been sitting there for hours. You've got to walk around. You just can't stay on the computer forever. #Person2#: Well, I am up to my neck in work. I've got to finish this report. Sarah needs it by noon. I don't want to be scolded if I can't finish my work by the deadline. #Person1#: I understand that, but you'd feel better if ...	<pad> #Person1# invites #Person2# to have a coffee break enjoying the fast pace of work, unlike #Person2# who is still too busy to do much work afterwards.</s>	<pad> #Person1# wants to take a coffee, but #Person2# can't take a break because he's up to his neck in work. #Person1# explained that #Person2# would feel better if they took a break.</s>	1.959802	1.826756	-0.133046

Query: This column contains the input or prompt given to the fine-tuned model.
Response Before Detoxification: This column shows the model’s response to the input query before any detoxification process was applied.
Response After Detoxification: This column displays the model's response after undergoing detoxification, which is intended to make the response less toxic or more appropriate.
Reward Before Detoxification: This column represents the toxicity score of the response before detoxification.
Reward After Detoxification: Similar to the previous column but evaluates the response after the detoxification process.
Reward Difference: This column shows the difference in the reward scores before and after detoxification, indicating the effect of the detoxification process. A positive value would suggest an improvement in the reward score, while a negative value indicates a decrease.

REFERENCES

https://medium.com/@hamzafergougui/llms-for-good-reinforcement-learning-to-reduce-hate-speech-57883b68250a
https://huggingface.co/datasets/knkarthick/dialogsum
https://huggingface.co/docs/transformers/en/index