

Final Project Report

Dataset:

Our project's data comes from the DialogSum corpus in the Hugging Face datasets repository. A large-scale dialogue summarizing resource, this dataset was developed by Yulong Chen and colleagues and includes 13,460 talks with matching hand annotated summaries and themes. DialogSum offers a comprehensive and varied collection of task-oriented scenarios that address a wide range of everyday themes, including work, education, medicine, shopping, leisure, and travel. Language specialists analyze each interaction to make sure the summaries maintain significant named entities, are succinct, convey the most relevant information, and are presented in formal language from the viewpoint of an observer. With an extra 100 holdout conversations for topic creation, the dataset is divided into training (12,460 dialogues), validation (500 dialogues), and test sets (1,500 dialogues). A couple of the data snapshots are shown below.

```
data = "knkarthick/dialogsum"
dt = load_dataset(data)
```

Downloading readme: 100% 4.65k/4.65k [00:00<00:00, 374kB/s]

Downloading data: 100% 11.3M/11.3M [00:00<00:00, 44.7MB/s]

Downloading data: 100% 442k/442k [00:00<00:00, 1.57MB/s]

Downloading data: 100% 1.35M/1.35M [00:00<00:00, 3.41MB/s]

Generating train split: 12460/0 [00:00<00:00, 58972.96 examples/s]

/opt/conda/lib/python3.10/site-packages/datasets/download/streaming_download_manager.py:784: DeprecationWarning: keyword in pd.read_csv is deprecated and will be removed in a future version.
return pd.read_csv(xopen(filepath_or_buffer, "rb"), download_config=download_config), **kw

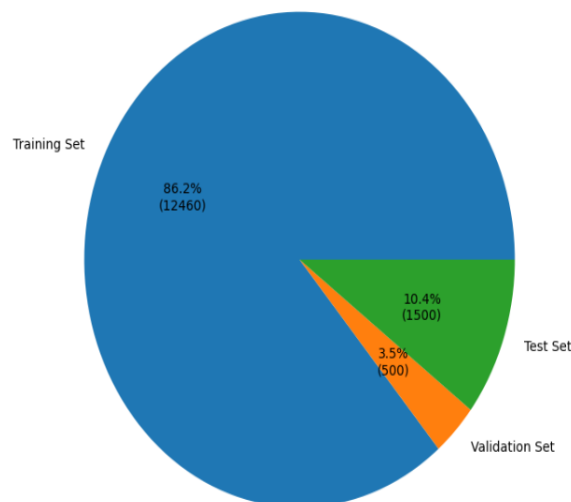
Generating validation split: 500/0 [00:00<00:00, 16830.40 examples/s]

/opt/conda/lib/python3.10/site-packages/datasets/download/streaming_download_manager.py:784: DeprecationWarning: keyword in pd.read_csv is deprecated and will be removed in a future version.
return pd.read_csv(xopen(filepath_or_buffer, "rb"), download_config=download_config), **kw

Generating test split: 1500/0 [00:00<00:00, 48295.14 examples/s]

```
DatasetDict({
  train: Dataset({
    features: ['id', 'dialogue', 'summary', 'topic'],
    num_rows: 12460
  })
  validation: Dataset({
    features: ['id', 'dialogue', 'summary', 'topic'],
    num_rows: 500
  })
  test: Dataset({
    features: ['id', 'dialogue', 'summary', 'topic'],
    num_rows: 1500
  })
})
```

Data Splits Size



```
train_df.head()
```

	id	dialogue	summary	topic
0	train_0	#Person1#: Hi, Mr. Smith. I'm Doctor Hawkins. ...	Mr. Smith's getting a check-up, and Doctor Haw...	get a check-up
1	train_1	#Person1#: Hello Mrs. Parker, how have you bee...	Mrs Parker takes Ricky for his vaccines. Dr. P...	vaccines
2	train_2	#Person1#: Excuse me, did you see a set of key...	#Person1#'s looking for a set of keys and asks...	find keys
3	train_3	#Person1#: Why didn't you tell me you had a gi...	#Person1#'s angry because #Person2# didn't tel...	have a girlfriend
4	train_4	#Person1#: Watsup, ladies! Y'll looking'fine t...	Malik invites Nikki to dance. Nikki agrees if ...	dance

Base-Model:

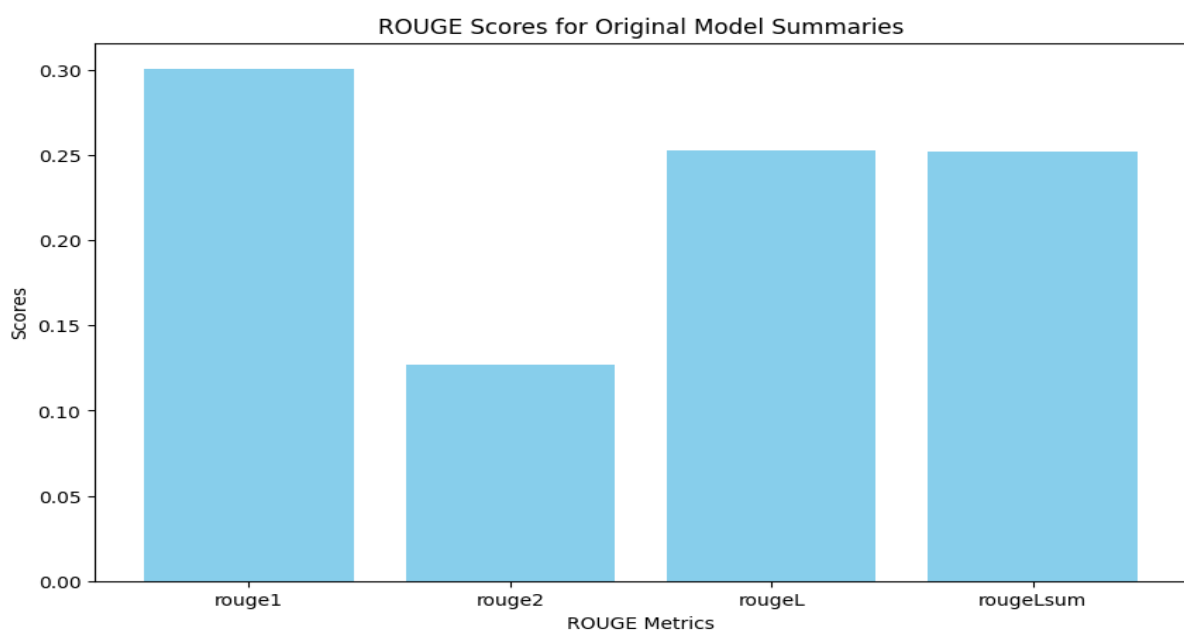
Google Research has created an enhanced language model called FLAN-T5, which stands for "Fine-tuned Language Net - T5". It expands upon the T5 (Text-To-Text Transfer Transformer) architecture, which enables the use of the same model, loss function, and hyperparameters across a variety of natural language processing (NLP) applications by converting them into a common text-to-text format. By adding instruction-based fine-tuning, FLAN-T5 improves the T5 model's generalization and comprehension of task instructions, allowing it to perform better on a variety of NLP tasks. Because of this, FLAN-T5 excels at jobs like chat summarization that need for sophisticated comprehension and generation. Our goal is to optimize conversation summarization performance by utilizing LoRA and RLHF approaches to further refine the tremendous capabilities of the underlying model, FLAN-T5.

LORA Summarization Optimization:

We concentrated on getting the dataset ready and preprocessing it throughout the first part of our endeavor. To make the data easier to understand for our particular goal of conversation summaries, we began by removing the DialogSum dataset's pointless field "topic". After that, we experimented with several prompting strategies, such as learning ways that use one-shot, few-shot, and zero-shot procedures. These methods were used to identify the best prompts for producing summaries of superior quality. In order to determine which random prompts produce the best summaries for a specific conversation, we also evaluated a variety of them.

Ten conversations and the related human-written summaries were chosen at random in order to assess the performance of the basic model. After that, we used the base FLAN-T5 model to create summaries and ROUGE metrics to evaluate the performance. Following were the evaluation results: {'rouge1': 0.300760283032675, 'rouge2': 0.1270529264337314, 'rougeL': 0.2529406199223184, 'rougeLsum': 0.25207893887684474}. With further training methods and fine-tuning, these results offer a baseline for future advancements.

	human summaries	model summary
91	Mr. Polly wants to get a break from work and h...	Person1#: I'm sorry, Mr. Polly. I'm sorry, but...
75	#Person1# and #Person2# are talking about some...	The Ram Bar and Grill is having a special on p...
1264	Ballam is hired by Apple Corporation but he th...	The position of computer engineer is offered t...
330	#Person1# has decided to go on a diet in the n...	#Person1: I'm going on a diet. I'm going to cu...
1349	#Person1# asks about the fundraiser of #Person...	#Person1#: How did your company's fundraiser g...
774	Mary got married. #Person2# is surprised when ...	Mary got married yesterday.
1038	Nathan is going to Chicago for his practice in...	Nathan is going to Chicago for his practice.
1285	#Person1# and #Person2# both get laid off and ...	Person1 is looking for a job.
1165	#Person1# is afraid that eyelash curler may hu...	#Person1: I'm a wimp.
335	Karen Huang wants to register for a class. #Pe...	#Person2#: Hello. Can I help you? #Person1#: Y...



Training of PEFT with LORA:

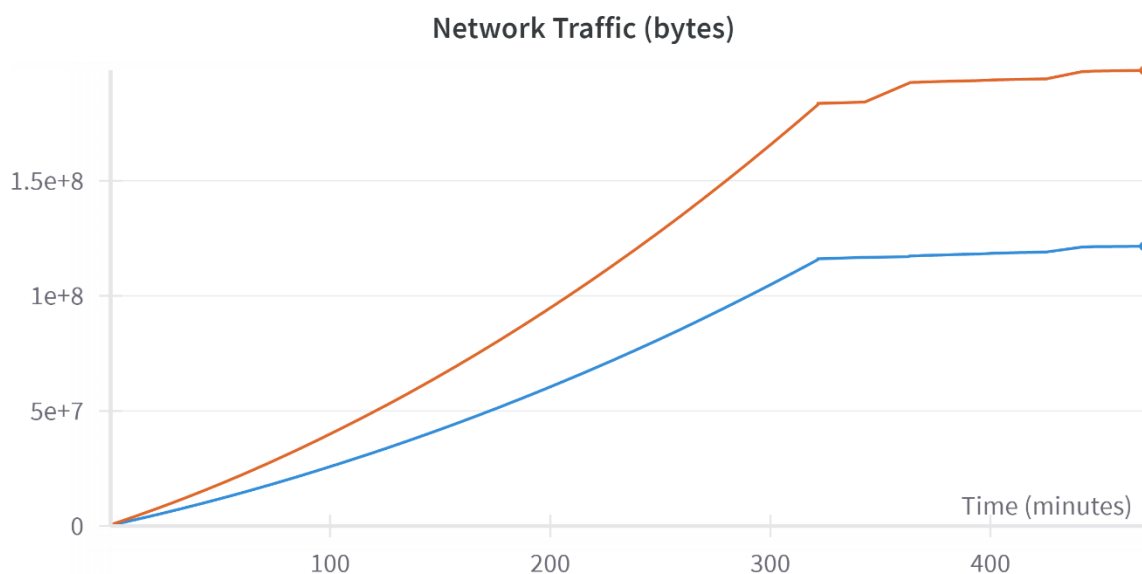
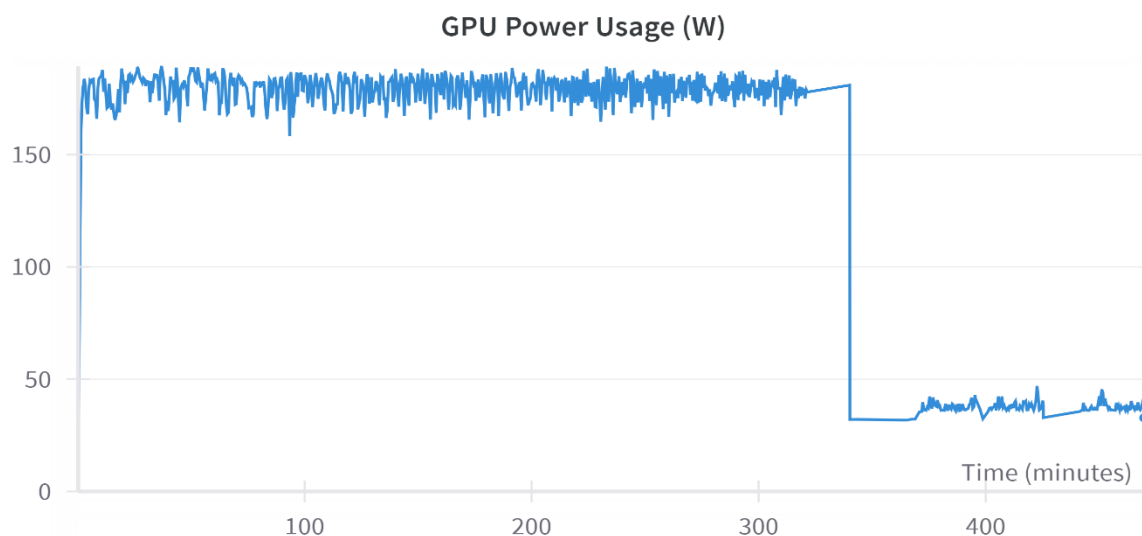
The FLAN-T5 model's advanced capabilities were carefully considered in the training process design, which employed the PEFT (Parameter-Efficient Fine-Tuning) approach along with LoRA (Low-Rank Adaptation). Initially, we configured the training environment on PyTorch GPU 100, which offered the processing capacity required to effectively manage the demanding training assignments.

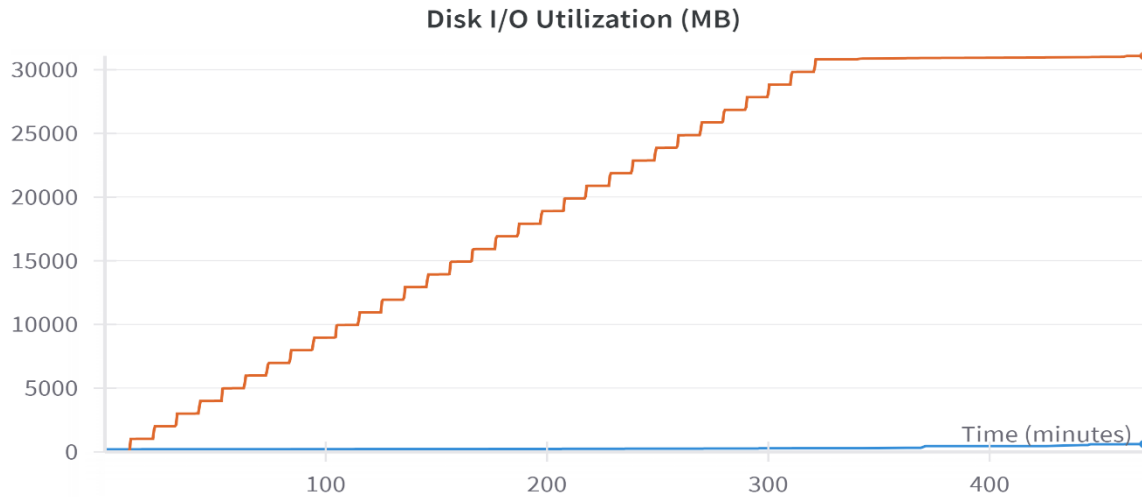
We utilized the following exact LoRA parameters to configure the model: $r=16$, $\text{lora_alpha}=16$, targeting the modules "q" and "v", with a bias-free dropout rate of 0.05. This setup reduced the total number of trainable parameters to 1,769,472, or 0.7096414524241463% of the total parameters, in an effort to optimize the fine-tuning process by concentrating on the most influential parameters.

To set up the training regimen, we used the TrainingArguments from the transformers library. The learning rate of $1e-3$, an automatic batch size finder, an output directory for storing model checkpoints, and a total of 10 training epochs were the critical parameters. Every 500 steps,

logging was set up to record progress, and data were sent to "wandb" for thorough tracking and training metrics visualization.

The Trainer class was used to train the model, which expedited the training loop and guaranteed effective use of the GPU resources. The training configuration made sure the model saw a wide range of conversations from the DialogSum dataset, which helped it pick up on the subtleties and complex patterns needed to produce conversation summaries of the highest caliber. The `save_pretrained` method was used to save the model and tokenizer after training was finished, making it simple to load and evaluate them later. The extensive training procedure laid the groundwork for notable advancements in the model's summarizing powers.





We tracked the model's performance over time and ensured effective resource use by keeping an eye on a number of critical metrics while our PEFT model was being trained using LoRA. A steady increase in network traffic, measured in bytes, is seen in the first chart, which shows a considerable amount of data being sent between the compute and storage units. The disk I/O use in megabytes is shown in the second chart. This step-wise rise pattern reflects the periodic checkpointing and storing of model states. The GPU power usage is shown in watts in the final chart. It shows that the training period was characterized by intense computation, with a high and stable power consumption level maintained until a rapid decline that most likely signified the end of the training session. Together, these measurements show how effective the training system is at utilizing the available hardware and achieving maximum performance, despite the high resource demands.



The training loss during the course of the training procedure is depicted in the chart. The training loss is much higher at start, but it rapidly decreases in the first few hundred steps and stabilizes at a much lower value for the duration of the training session. This quick decline shows that the model learned and converged effectively early in the training phase. Once the

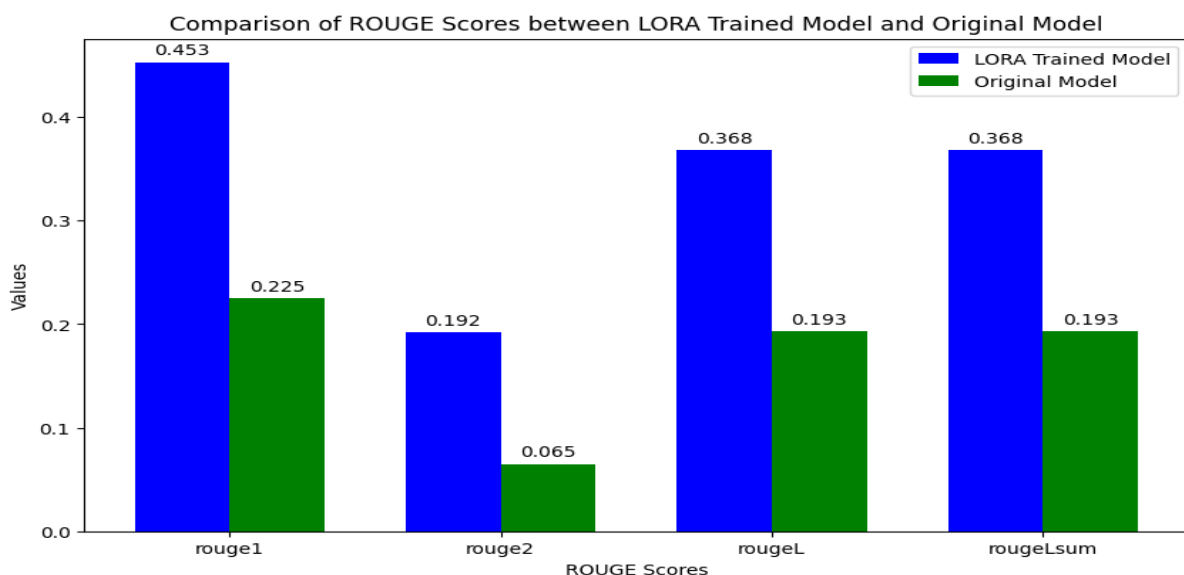
model has successfully learned the underlying patterns in the data, it is likely to sustain minimal error as training advances, as indicated by the continuously low training loss beyond the initial phase.

Comparison of results:

The bar chart compares the ROUGE scores of the LORA trained model and the original model across four different metrics: ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-Lsum. The LORA trained model, represented in blue, significantly outperforms the original model, represented in green, across all metrics. Notably, this substantial improvement is achieved by training only 0.7% of the model's parameters.

- **ROUGE-1:** The LORA trained model achieves a score of 0.453, which is more than double the score of 0.225 obtained by the original model, resulting in a 101.6% improvement.
- **ROUGE-2:** The LORA trained model scores 0.192, whereas the original model scores only 0.065, reflecting a 195.7% improvement.
- **ROUGE-L:** The LORA trained model achieves a score of 0.368, significantly higher than the 0.193 scored by the original model, indicating a 90.7% improvement.
- **ROUGE-Lsum:** The LORA trained model also scores 0.368 in ROUGE-Lsum, compared to the original model's 0.193, showing a 90.7% improvement.

These results highlight the significant enhancement in summarization performance, demonstrating that the LORA trained model provides more accurate and coherent summaries. Achieving such remarkable improvements by training only 0.7% of the model's parameters underscores the efficiency and effectiveness of the PEFT approach with LoRA.



When compared to the original model, the LORA-trained model shows notable gains in its capacity to summarize discussions. Prior to LORA training, the model produced imprecise answers that were unable to fully capture the essential context and specifics of the discussions. For example, the original model's response, while describing a discourse about placing an order for meals, only mentioned a portion of the order and omitted important features like drinks and dessert. On the other hand, the model's summaries following LORA training are significantly

more thorough and consistent with the initial summaries. It faithfully catches important details from the chats, like a trip to Hebei and the intense sandstorms that are plaguing the area. The updated model generates summaries that accurately reflect the talks' real substance in addition to being more thorough. This improvement in performance, which was attained with just 0.7% of the model's parameters trained, demonstrates how well the LORA approach works to improve language model capabilities for improved summarization. This improvement in performance, which was attained with just 0.7% of the model's parameters trained, demonstrates how well the LORA approach works to improve language model capabilities for improved summarization.

Before LORA:

```
Prompt: summarize
#Person1#: May I take your order?
#Person2#: We haven't decided yet. Could you give us a little longer?
#Person1#: Yes, take your time, please.
#Person2#: Can we get something to drink? We want two bottles of beer.
#Person1#: Fine.
#Person2#: Could you tell us your specials today?
#Person1#: The special today is steak.
#Person2#: We'll take this steak dinner.
#Person1#: What would you like to go with your steak?
#Person2#: Peas and carrots.
#Person1#: I see. What would you like for dessert?
#Person2#: Icecream, please.
Response: #Person1#: We're going to take the steak dinner.

original summary #Person1# serves #Person2# to order two bottles of beer, a steak dinner, and ice cream.
```

After LORA:

```
Prompt: summarize
#Person1#: Where are you going for your trip?
#Person2#: I think Hebei is a good place.
#Person1#: But I heard the north of China are experiencing severe sandstorms!
#Person2#: Really?
#Person1#: Yes, it's said that Hebes was experiencing six degree strong winds.
#Person2#: How do these storms affect the people who live in these areas?
#Person1#: The report said the number of people with respiratory tract infections tended to rise after sandstorms. The sand gets into people's noses and throats and creates irritation.
#Person2#: It sounds that sandstorms are trouble for everybody!
#Person1#: You are quite right.
Response: #Person2#'s going to Hebei for a trip. #Person1# tells #Person2# the north of China is experiencing severe sandstorms. #Person2# thinks sandstorms are trouble for everybody.

original summary #Person2# plans to have a trip in Hebei but #Person1# says there are sandstorms in there.
```

RLHF Training Using PEFT (LoRA):

The required models and tokenizers were built up before the training process started. The basic model for the PEFT (Low-Rank Adaptation) training was a FLAN-T5 model. The appropriate tokenizer was loaded into this basic model. Furthermore, a pre-trained model that was especially optimized for dialogue summary was included. To improve training efficiency, this model was relocated to a suitable device, a GPU if one was available.

A toxicity evaluation mechanism was established to guarantee that the resulting summaries were not only accurate but also devoid of any harmful substance. To do this, a pre-trained RoBERTa model that has been optimized for hate speech detection had to be loaded. The resulting summary and the original discussion were concatenated, and this text was then ran through the toxicity detection algorithm to perform the toxicity evaluation. For additional research, the text's likelihood of being labeled as hate speech was noted.

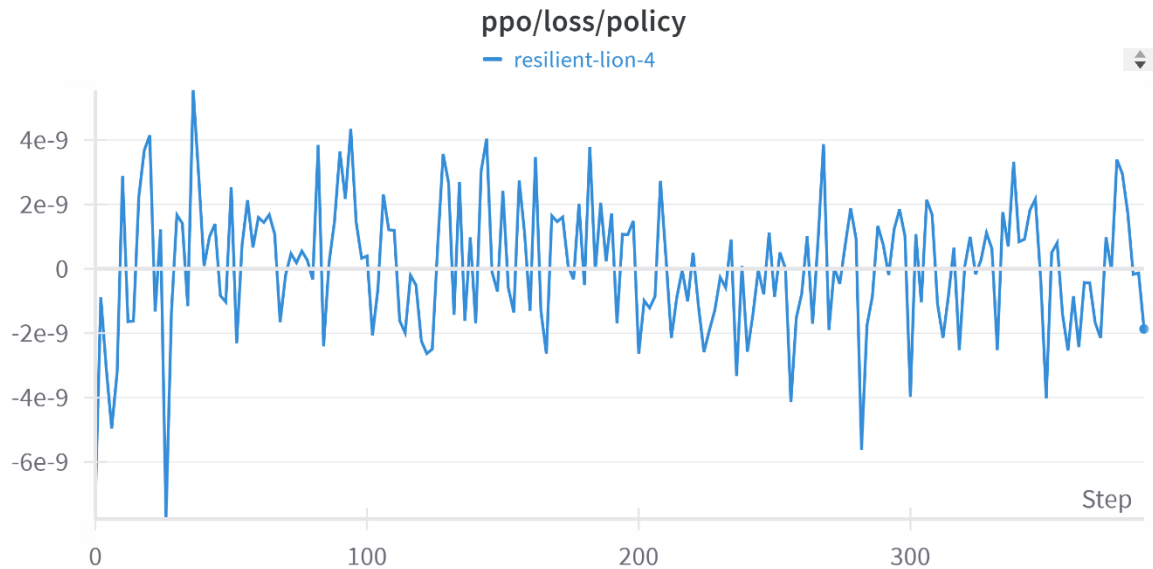
An initial performance assessment of the model was carried out prior to training. This involved utilizing the pre-trained algorithm to summarize conversations from the test dataset and assessing the toxicity of these summaries. The procedure gave toxicity scores a baseline, which was necessary to compare the outcomes after training. The first evaluation's findings showed that the initial model had a tendency to produce harmful content, if any, with a mean toxicity score of roughly 0.028 and a standard deviation of roughly 0.082.

The PEFT technique, specifically with LoRA (Low-Rank Adaptation), was the main focus of the instruction. By training a small subset of parameters, the LoRA technique was able to preserve or even improve the model's performance while drastically lowering the processing needs. For PPO (Proximal Policy Optimization) training, which is appropriate for reinforcement learning with human feedback, the model was modified. In this step, reference models were also set up, and training settings such a batch size of 64, a learning rate of 3e-5, and 10 PPO epochs were configured.

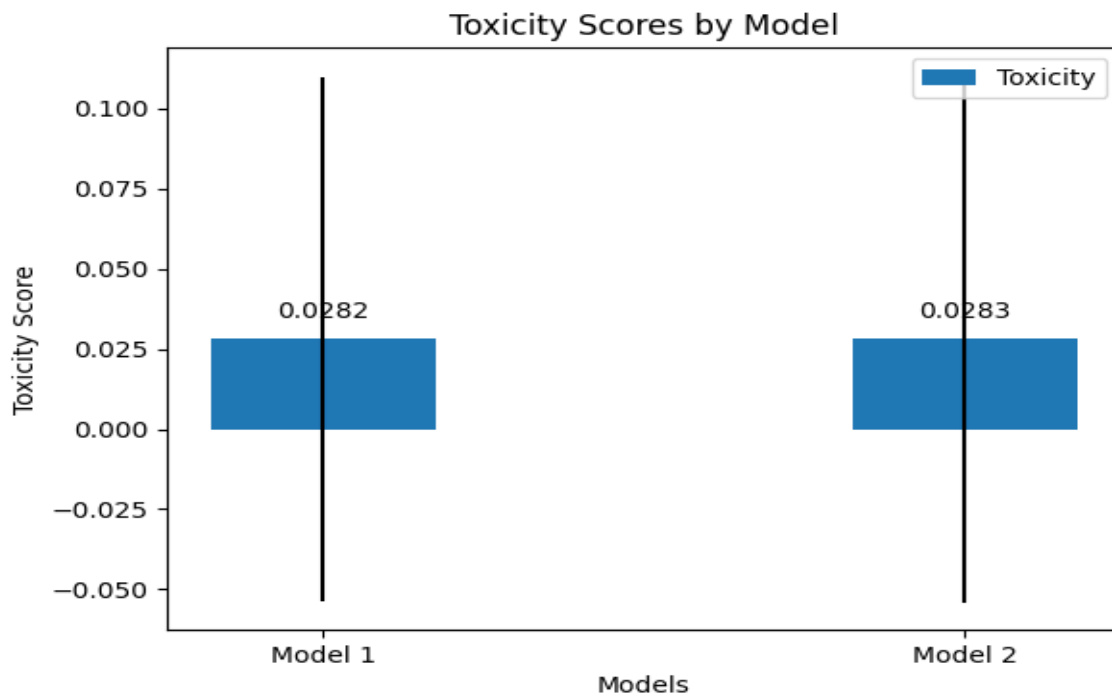
id	query	response	reward
55	Summarize the conversation #Person1#: This small town has many serious problems. #Person2#: What's the cause? #Person1#: The mayor and high officials practice corruption and a large number of guys who are good at nothing occupy many important positions. #Person2#: I see why it lags behind	<pad> #Person1# tells #Person2# about the problems of a small town. </s>	1.485
56	Summarize the conversation #Person1#: Why are you at the police station? What happened to you? #Person2#: Well, I went to a restaurant after work and after I finished dinner, I went to my car. I tried to open the door. But my key broke in the lock. So I called a slot company and they opened the car for	<pad> #Person2# told #Person1# #Person2# went to a restaurant after work. #Person2# called a slot company and they opened the car and took #Person2# home. </s>	2.267
57	Summarize the conversation #Person1#: Can I help you? #Person2#: Yes, I need to buy a computer for this semester. I was told it is cheaper to buy computers here. #Person1#: Well, you heard right. You can get an excellent deal on a new computer here. We have great discounts for students. #Person2#: I don't	<pad> #Person2# wants a PC to buy for this semester but #Person1# suggests a Macintosh and #Person2# wants to buy a computer with a lot of memory. #Person1# introduces the Macintosh to #Person2# and #Person2# decides to buy a computer with Macintosh. #Person1# isn't sure but says	3.855
58	Summarize the conversation #Person1#: So do you think you can get to the office alone? #Person2#: Sure, I'll catch the train. If I can't catch the train, I'll get a taxi. #Person1#: Do you think the taxi driver will understand you? #Person2#: If he doesn't understand me, I'll show him the office address on my	<pad> #Person2# is going to get to the office alone. #Person1# asks #Person2# what to do if #Person2# is late. </s>	1.235
59	Summarize the conversation #Person1#: I like that picture you put up on the wall yesterday. #Person2#: Thank you. It's a photo that I took on my vacation last year. #Person1#: You took it yourself? #Person2#: I've been taking pictures for years. #Person1#: Sounds interesting. Can I see some	<pad> #Person2# took a picture of #Person1# and tells #Person1# #Person2# doesn't want to let people know about #Person2#'s pictures. </s>	3.634

The FLAN-T5 tokenizer was used to tokenize the conversations from the training dataset. To assist the model in producing summaries, a summary cue was placed before each dialogue. Tokenized datasets made training iterations more efficient by preparing them for batch processing. In order to guarantee consistency between batches, the data preparation stage made sure that the input sequences were suitably truncated and padded.

The model was saved when training was finished, and its performance was assessed once more using the test dataset. Repeating the toxicity assessment allowed for the measurement of any gains brought about by the training procedure. With a mean of roughly 0.028 and a standard deviation of roughly 0.082, the final toxicity ratings were compared to the pre-training scores in order to evaluate how well the training reduced the creation of hazardous content.

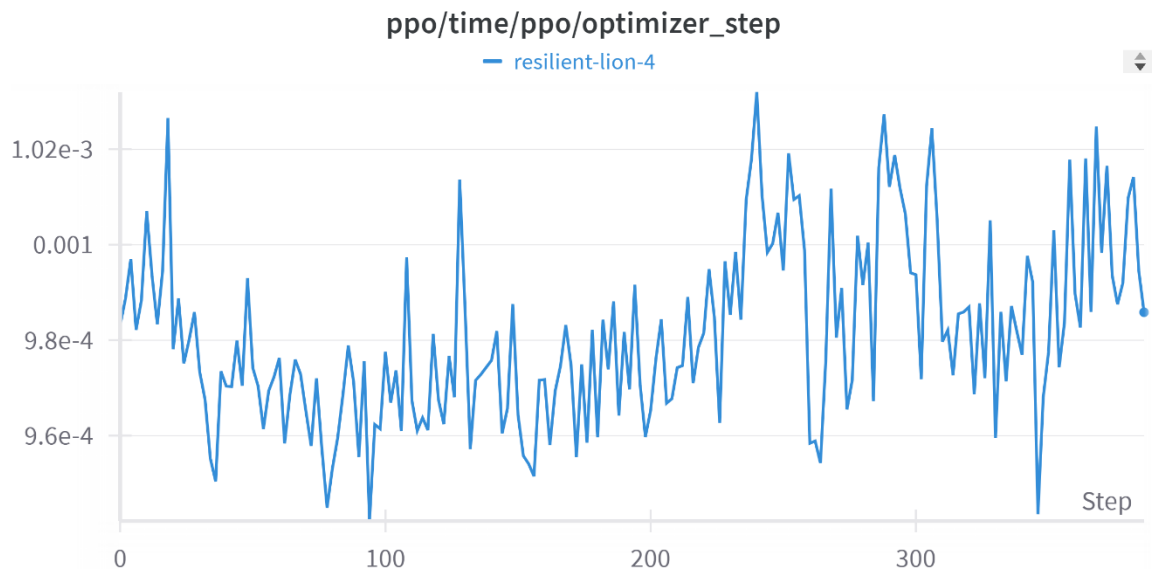


The results of the final toxicity evaluation showed that performance had not changed appreciably even after receiving extensive RLHF training using PEFT (LoRA). With a standard deviation of roughly 0.082 and a mean toxicity score that stayed around 0.028, the model was already performing at its best in terms of toxicity prior to the RLHF training. This result implies that the original model was well-tuned and that additional training did not significantly improve the model's capacity to generate non-toxic content.



The graph of the PPO time optimizer steps shows a training process characterized by active adjustments and stabilization over time. Initial optimizer step values start high around 0.001

and exhibit significant fluctuations, suggesting dynamic adjustments in the early and mid-phases of training (steps 0-200). In the latter phase (steps 200-350), the optimizer steps show a slight upward trend but remain within a relatively narrow band, indicating a well-contained optimization process. Despite these adjustments, the overall model performance, particularly in terms of toxicity, has not significantly changed after reinforcement learning with human feedback (RLHF), suggesting the model's earlier performance was already optimal.



Interphase:

The interface for your project, designed using Flask, serves as a user-friendly web application for text summarization. The interface comprises two main pages: the input page and the result page. The input page, styled with a clean and modern design, features a text area for users to input the text they want to summarize. A "Summarize" button triggers the summarization process. This page is rendered using the `index.html` template, which includes a form that sends the input text to the server for processing when submitted.

On the server side, the application is powered by a FLAN-T5 model fine-tuned for summarization tasks. Upon receiving the input text, the Flask app processes it through a sequence of steps. It first tokenizes the input text using the tokenizer from the pre-trained model, then generates a summary using the model. The summarized text is then decoded and sent back to the user via the `result.html` template, which displays the summary in a well-formatted manner. This two-page structure ensures a smooth user experience, allowing for easy input and clear presentation of the summarized content.

Text Summarization

#Person1#: Excuse me, did you see a set of keys?
#Person2#: What kind of keys?
#Person1#: Five keys and a small foot ornament.
#Person2#: What a shame! I didn't see them.
#Person1#: Well, can you help me look for it? That's my first time here.
#Person2#: Sure. It's my pleasure. I'd like to help you look for the missing keys.
#Person1#: It's very kind of you.
#Person2#: It's not a big deal. Hey, I found them.
#Person1#: Oh, thank God! I don't know how to thank you, guys.
#Person2#: You're welcome.



Summarize

Summary Result

#Person1# asks #Person2# to help look for the missing keys. #Person2# finds them and thanks #Person1#.

Back

References:

1. https://huggingface.co/docs/transformers/en/model_doc/flan-t5
2. <https://medium.com/@eren9677/text-summarization-387836c9e178>
3. <https://huggingface.co/spaces/evaluate-metric/rouge>
4. <https://www.promptingguide.ai/techniques/fewshot>
5. <https://github.com/philschmid/deep-learning-pytorch-huggingface/blob/main/training/peft-flan-t5-int8-summarization.ipynb>
6. <https://www.kaggle.com/code/paultimothymooney/fine-tune-flan-t5-with-peft-lora-deeplearning-ai>
7. "Some of the code is based on the CSE 676 Deep Learning Assignment 0 Bonus submission by My Deep Aman Ganta"
8. <https://pemargr.medium.com/build-a-web-app-using-pythons-flask-for-beginners-f28315256893>