Project Report: GPT-2 Fine-Tuning with WikiText-2

Overview

This project involves fine-tuning the GPT-2 language model using the WikiText-2 dataset. The primary goal is to reduce the perplexity of the model, a measure of how well the model predicts the next word in a sequence. The original perplexity before fine-tuning was 54.729, and after three epochs of fine-tuning, it reduced to 26.979.

Methodology

Data Preparation

The WikiText-2 training dataset was used for fine-tuning the GPT-2 model. The dataset was tokenized using the GPT-2 tokenizer, and sequences of length 128 were used for training. The `TextDataset` class from the `transformers` library was employed for efficient handling of the dataset, and a custom `DataCollatorForLanguageModeling` was used to collate the data.

Model Fine-Tuning

The GPT-2 model was fine-tuned using the `Trainer` class from the `transformers` library. The training process involved updating the model parameters over three epochs, with a batch size of 1. The model was saved at regular intervals during training, allowing for the selection of the best-performing checkpoints.

```
# Training Arguments
training_args = TrainingArguments( output_dir="./model",
                                   overwrite_output_dir=True,
                                   num_train_epochs=3,
                                   per_device_train_batch_size=1,
                                   save_steps=10_000,
)
# Trainer Initialization
trainer = Trainer( model=model,
                   args=training_args,
                   data_collator=data_collator,
```

```
                    train_dataset=train_dataset,
)
# Training Loop
for epoch in range(int(training_args.num_train_epochs)):
          trainer.train()
```

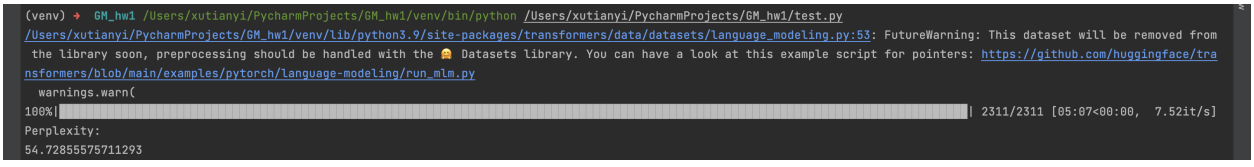
Model Evaluation

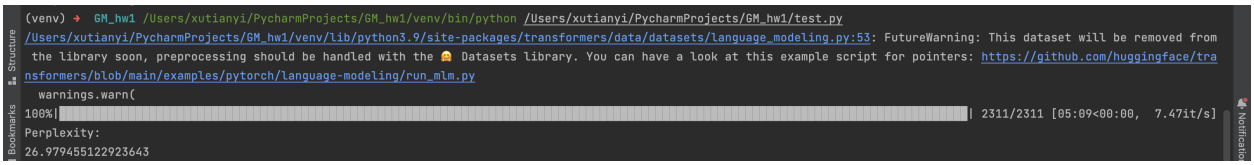
The fine-tuned model was evaluated on the WikiText-2 test dataset to calculate the perplexity. The `Trainer` class was utilized for evaluation, and a custom metric function was defined to compute perplexity from the loss.

```
# Evaluation Arguments
training_args = TrainingArguments( per_device_eval_batch_size=1,
                                   output_dir="./dummy_output_dir",
)

# Metric Calculation
def compute_metrics(p: EvalPrediction):
          perplexity = math.exp(p.loss)
          return {"perplexity": perplexity}

# Trainer Initialization for Evaluation
trainer = Trainer( model=model,
                   args=training_args,
                   data_collator=data_collator,
                   compute_metrics=compute_metrics,
)
# Evaluation
results = trainer.evaluate(eval_dataset=val_dataset)
print("Perplexity: ", math.exp(results["eval_loss"]))
```

Results

The perplexity of the GPT-2 model significantly improved from 54.729 before fine-tuning to 26.979 after three epochs of fine-tuning. This reduction in perplexity suggests that the fine-tuning process enhanced the model's ability to generate more contextually relevant and coherent text.





Conclusion

Fine-tuning the GPT-2 model on the WikiText-2 dataset resulted in a notable improvement in perplexity. The reduced perplexity indicates that the fine-tuned model has a better understanding of the underlying language structure and context. This project demonstrates the effectiveness of transfer learning and fine-tuning for improving the performance of pre-trained language models.