1. **Introduction**

Brief overview of the tasks (question answering, translation, summarization) and the model architecture used.

The document presents an analysis of three tasks - question answering, translation, and summarization - using the GPT-2 model with a technique called prompt tuning. Prompt tuning involves freezing the weights of the GPT-2 model and adding a soft prompt embedding layer as the initial layer with dimensions [prompt\_tune, 768]. Subsequently, weight updates are performed only for this prompt embedding layer.

Input + soft prompt. **6, 1018**

For data preprocessing, techniques like lemmatization, stopword removal, and truncating are applied to filter the datasets for all three tasks. The approach is inspired by public repositories and aims to enhance the performance of the model across the tasks.

**Data Description:**

For summarization, the CNN/Daily Mail dataset was utilized, with a soft prompt initialized.

For question answering, the SQuAD dataset was employed, and the soft prompt was initialized.

For machine translation from English to German, the Europarl dataset was used, with the soft prompt initialized.

The tokens selected for each task were:

Summarization: [SUMMARIZE] (Token size: 3)

1021,768 = 1024 \* 768

Question Answering: "Answer the question using given context" (Token size: 6), 1018, 768 = 1024 \* 768

Machine Translation: "translate english to german" (Token size: 6)

1018,768 = 1024 \* 768

**2. Methodology**Explanation of the training setup including:

* Hyperparameters chosen (loss function, number of epochs, learning rate, optimizer, etc.)

Tasks of question answering, machine translation, and summarization employed identical hyperparameter settings due to time constraints, without engaging in hyperparameter optimization.

**CrossEntropyLoss** was chosen as the loss function across all tasks due to its suitability for classification problems inherent in these tasks.

The **Adam optimizer** was utilized alongside CrossEntropyLoss.

7b, 14 B para extra (SGD)

2 more parameter,

A **batch size** of 8 was taken. (Later removed because of memory issue)

The training was limited to **2 epochs** for each task due to computational constraints, with each epoch taking approximately 6-7 hours to complete in Google Colab.

**Learning rate** was set to 2e-3.

A **gradient clip** value of 1.0 was applied.

**3. Performance Evaluation**

**Task 1: Translation**

**Train Data Size: 10000**

**Test Data Size: 2000**

**Model performance metrics (Bert Score, BLEU Score, ROUGE Score) BLURT**

Average Training BLEU Score: 1.1846005662636967e-157

Average Training BERTScore: 0.946 (To context )

Average Training ROUGE-1 Score: 0.966

Average Training ROUGE-2 Score: 0.965

Average Training ROUGE-L Score: 0.966

Average Validation BLEU Score: 5.760720367222536e-160

Average Validation BERTScore: 0.946

Average Validation ROUGE-1 Score: 0.972

Average Validation ROUGE-2 Score: 0.972

Average Validation ROUGE-L Score: 0.972

Train\_loss : 1.084 val\_loss : 0.444

From the results above, it's evident that our model has been trained effectively for translation, as it can readily generalize on the test dataset. This is further supported by its strong performance on both BERT and ROUGE scores. BERTScore, a metric typically utilized for retaining contextual information, also demonstrates the model's proficiency.

**Task 2: Summarization**

**Train Data Size: 2871**

**Test Data Size: 133**

Model performance metrics (Bert Score, BLEU Score, ROUGE Score)

Average Training BLEU Score: 4.444515154690828e-05

Average Training BERTScore: 0.940

Average Training ROUGE-1 Score: 0.931

Average Training ROUGE-2 Score: 0.930

Average Training ROUGE-L Score: 0.931

Average Validation BLEU Score: 3.6364572685947105e-234

Average Validation BERTScore: 0.956

Average Validation ROUGE-1 Score: 0.966

Average Validation ROUGE-2 Score: 0.966

Average Validation ROUGE-L Score: 0.966

Train\_loss : 0.338 val\_loss : 0.416

From the results above, it's evident that our model has been trained effectively for translation, as it can readily generalize on the test dataset. This is further supported by its strong performance on both BERT and ROUGE scores. BERTScore, a metric typically utilized for retaining contextual information, also demonstrates the model's proficiency.

**Task 3: Question Answering**

**Train Data Size: 3000**

**Test Data Size: 1000**

Model performance metrics (Bert Score, BLEU Score, ROUGE Score)

Average Training BLEU Score: 5.704725912199388e-235

Average Training BERTScore: 0.973

Average Training ROUGE-1 Score: 0.968

Average Training ROUGE-2 Score: 0.968

Average Training ROUGE-L Score: 0.968

Average Validation BLEU Score: 3.076248974135019e-234

Average Validation BERTScore: 0.92

Average Validation ROUGE-1 Score: 0.9

Average Validation ROUGE-2 Score: 0.91

Average Validation ROUGE-L Score: 0.90

Train\_loss : 0.101 val\_loss : 0.033

From the results above, it's evident that our model has been trained effectively for translation, as it can readily generalize on the test dataset. This is further supported by its strong performance on both BERT and ROUGE scores. BERTScore, a metric typically utilized for retaining contextual information, also demonstrates the model's proficiency.