Data Description and Problem Formulation:

The Alpaca GPT4 dataset encompasses a diverse range of tasks typically undertaken by GPT4 family models. It is well-established that GPT4 family models rely extensively on distinct prompts and instructions to execute various operations, including translation, summarization, entity extraction, question answering, and more. Consequently, the Alpaca GPT4 dataset comprises a multitude of instructions.

Data features: ['instruction', 'input', 'output', 'text']. I have used only 'instruction', 'input', 'output' in this task.

Objective:

Our primary objective in this task is to train the T5 model for versatile text generation. Given the wide array of instructions within the Alpaca GPT4 dataset, our goal is to equip the T5 model with the capability to handle text generation tasks of varying nature. This involves harnessing the rich diversity of instructions present in the dataset to create a highly adaptable and versatile text generation model.

Data Preparation:

In this phase, we will meticulously prepare the data to create a standardized structure for each row. Each row of our dataset will consist of two distinct strings: the input string and the target string. It is imperative to maintain a consistent format for the input string across all rows. The primary task of the model will be to generate an appropriate response based on the provided input string.

Task Overview:

The model's performance will be assessed by evaluating its ability to generate responses that closely match the target string. To achieve this, we will fine-tune the model's parameters to minimize the disparity between the predicted string and the target string. This iterative process ensures that the model becomes adept at generating accurate and contextually relevant responses based on the given input, thus enhancing its overall performance.

I**nput String** is the combination of Instruction and Input columns from Alpaca dataset. Note that if the input column is blank then Input String will only contain Instruction

Target String is the output column of the Alpaca dataset.
Below is an example of how input_text and target_text look like-

'input_text': 'INSTRUCTION #:# Brainstorm 5 adjectives to describe the following photograph INPUT #:# Photo of a colorful flower field'

'target_text': '1. Vibrant\n2. Lush\n3. Picturesque\n4. Serene\n5. Majestic'

Metric for model evaluation: Evaluating the performance of models like T5 or GPT is not very straight forward, we need to check generated results from various perspectives such as-

- Time taken by the model to produce the result.
- Generated results should be ethically accepted.
- How relevant the generated text is to the input.

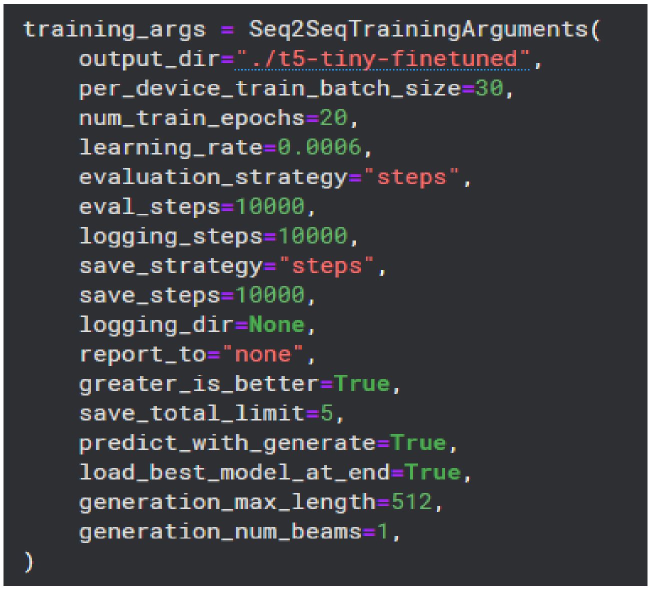
For simplicity, I have focused on the 3rd point.
To check the model relevance, we can use multiple metrics such as BLEU, ROUGH, BERT Score , Semantic similarity between generated and target strings and Human Evaluation. In this particular project, I have used ROUGH, BERT Score and Human Evaluation..

ROUGH Metric: The ROUGH metric is commonly used to evaluate text generation models for tasks like summarization and translation. It measures how well the generated text matches the reference text by comparing n-grams. ROUGH includes ROUGH-Recall (measuring the recall of generated n-grams), ROUGH-Precision (evaluating precision in n-gram generation), and ROUGH-FMeasure (a balanced measure of both). Higher ROUGH scores indicate better model performance. Additionally, ROUGH can be computed for different n-gram sizes, such as unigrams (ROUGH-1) and bigrams (ROUGH-2), offering a versatile evaluation of model-generated text.

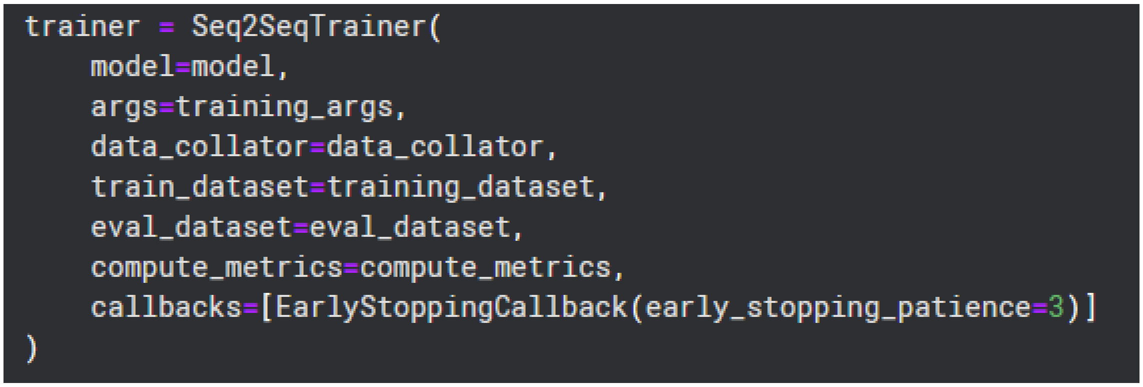
BERT Score Metric: BERTScore is a valuable metric for assessing text generation quality. While the ROUGH metric emphasizes syntactical similarity, BERTScore prioritizes semantic similarity between two sentences. This distinction allows BERTScore to provide a more comprehensive evaluation of the generated text's quality by considering not only syntax but also semantic meaning.

Train, validation and test splits: There are a total of 52002 rows in the data. Of which, 95% data (49401 rows) have been selected randomly for the train set, 3% data (1560 rows) for the validation set and 2% data (1041 rows) for the test set.

Model Parameters: Below are snapshots of the model parameters. The default Adam optimizer is utilized without explicit definition. The model undergoes 20 epochs, and fine-tuning can extend if loss reduction continues beyond this. After every 10,000 batch (each batch is of 30 data points) processes, ROUGH-FMeasure and BETScore validation scores are assessed. Early stopping is implemented, with "restore best model" set to True in case of declining model performance during training.

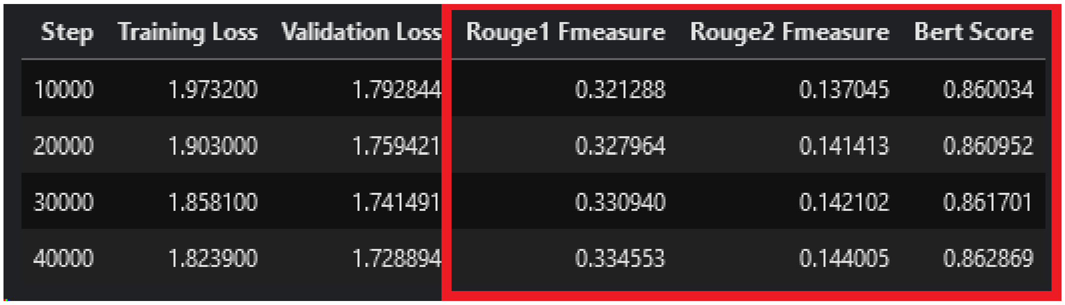


```
training_args = Seq2SeqTrainingArguments(
    output_dir="./t5-tiny-finetuned",
    per_device_train_batch_size=30,
    num_train_epochs=20,
    learning_rate=0.0006,
    evaluation_strategy="steps",
    eval_steps=10000,
    logging_steps=10000,
    save_strategy="steps",
    save_steps=10000,
    logging_dir=None,
    report_to="none",
    greater_is_better=True,
    save_total_limit=5,
    predict_with_generate=True,
    load_best_model_at_end=True,
    generation_max_length=512,
    generation_num_beams=1,
)
```



```
trainer = Seq2SeqTrainer(
    model=model,
    args=training_args,
    data_collator=data_collator,
    train_dataset=training_dataset,
    eval_dataset=eval_dataset,
    compute_metrics=compute_metrics,
    callbacks=[EarlyStoppingCallback(early_stopping_patience=3)]
)
```

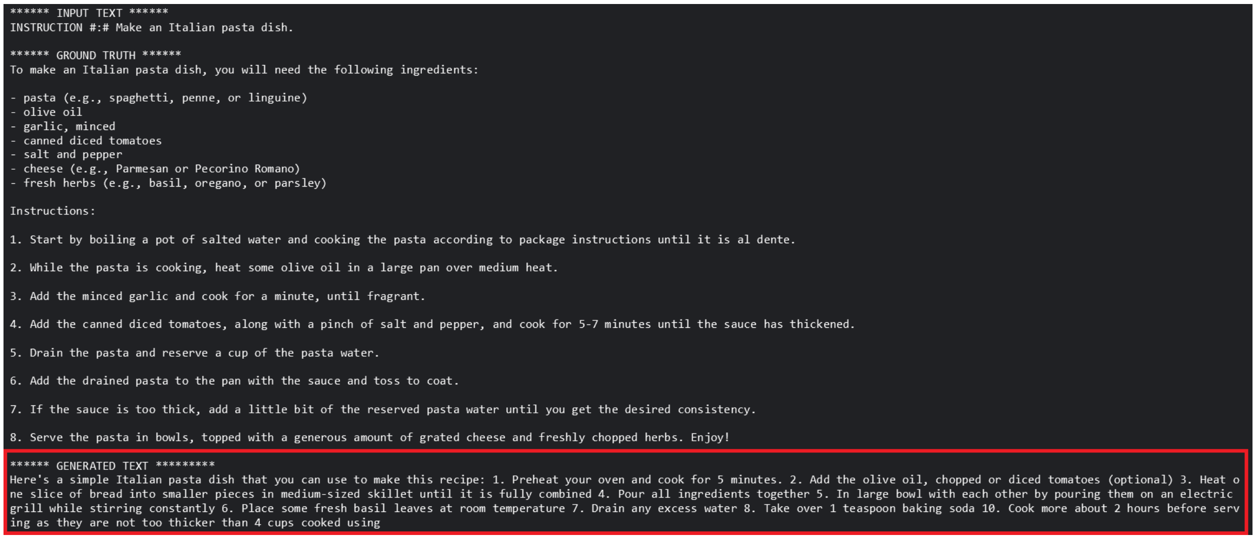
Training Result:



Step	Training Loss	Validation Loss	Rouge1 Fmeasure	Rouge2 Fmeasure	Bert Score
10000	1.973200	1.792844	0.321288	0.137045	0.860034
20000	1.903000	1.759421	0.327964	0.141413	0.860952
30000	1.858100	1.741491	0.330940	0.142102	0.861701
40000	1.823900	1.728894	0.334553	0.144005	0.862869

Results:

Example



```
****** INPUT TEXT ******
INSTRUCTION #:# Make an Italian pasta dish.

****** GROUND TRUTH ******
To make an Italian pasta dish, you will need the following ingredients:

- pasta (e.g., spaghetti, penne, or linguine)
- olive oil
- garlic, minced
- canned diced tomatoes
- salt and pepper
- cheese (e.g., Parmesan or Pecorino Romano)
- fresh herbs (e.g., basil, oregano, or parsley)

Instructions:

1. Start by boiling a pot of salted water and cooking the pasta according to package instructions until it is al dente.

2. While the pasta is cooking, heat some olive oil in a large pan over medium heat.

3. Add the minced garlic and cook for a minute, until fragrant.

4. Add the canned diced tomatoes, along with a pinch of salt and pepper, and cook for 5-7 minutes until the sauce has thickened.

5. Drain the pasta and reserve a cup of the pasta water.

6. Add the drained pasta to the pan with the sauce and toss to coat.

7. If the sauce is too thick, add a little bit of the reserved pasta water until you get the desired consistency.

8. Serve the pasta in bowls, topped with a generous amount of grated cheese and freshly chopped herbs. Enjoy!

****** GENERATED TEXT *********
Here's a simple Italian pasta dish that you can use to make this recipe: 1. Preheat your oven and cook for 5 minutes. 2. Add the olive oil, chopped or diced tomatoes (optional) 3. Heat o
ne slice of bread into smaller pieces in medium-sized skillet until it is fully combined 4. Pour all ingredients together 5. In large bowl with each other by pouring them on an electric
grill while stirring constantly 6. Place some fresh basil leaves at room temperature 7. Drain any excess water 8. Take over 1 teaspoon baking soda 10. Cook more about 2 hours before serv
ing as they are not too thicker than 4 cups cooked using
```