Convo AI: Natural language processing (UCS664)

Project on :

Punjabi language Text Summarization

Submitted By:

Pragya Gupta	102103407 3CO15
Yash Dogra	102166002 3CS11

B.E. Third Year – COE

Submitted To:

Dr. Jasmeet Singh

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology

Patiala – 147001

Dataset: XLSum (Punjabi Language)

We have used **Extractive Summarization** -It does not generate new text but rather extracts and compiles existing content.

- The csebuetnlp/xlsum dataset is a comprehensive dataset containing multilingual news articles and their summaries. The dataset includes a wide range of languages. For this project, the XLSum dataset for Punjabi is used from the dataset library is used. There is a 80%-10%-10% split but in Punjabi language we have a **total of 10267 sample counts of which 8215 are for training , 1026 for test and 1026 for validation**.

  ➢
```
from datasets import load_dataset
config_name='punjabi'
dataset = load_dataset("csebuetnlp/xlsum",config_name)
```

Data Fields

- 'id': A string representing the article ID.
- 'url': A string representing the article URL.
- 'title': A string containing the article title.
- 'summary': A string containing the article summary.
- 'text' : A string containing the article text.

id string	url string	title string
international-45496712	https://www.bbc.com/punjabi/international-45496712	ਦੇਖਿਆ ਹੈ ਕਦੇ ਸਮੁੰਦਰ ਦਾ ਇਹ ਰੂਪ - ਤਸਵੀਰਾਂ
india-51503342	https://www.bbc.com/punjabi/india-51503342	ਵੈਲੇਨਟਾਈਨ ਵੀਕ 'ਚ ਇਕੱਲਤਾ ਦਾ ਤੋੜ ਕੀ ਹੈ
international-49854064	https://www.bbc.com/punjabi/international-49854064	ਮੋਦੀ ਪਾਕਿਸਤਾਨ ਅਤੇ ਕਸ਼ਮੀਰ ਦਾ ਨਾਂ ਲਏ ਬਿਨਾਂ ਕਿਹੜੇ ਮੁੱਦਿਆਂ 'ਤੇ ਬੋਲੇ
india-51255271	https://www.bbc.com/punjabi/india-51255271	ਭਾਰਤ ਦੇ 54 ਫ਼ੌਜੀਆਂ ਦੇ 'ਲਾਪਤਾ' ਹੋਣ ਦਾ ਰਹੱਸ

summary string	text string
ਬ੍ਰਿਟੇਨ ਵਿਚ ਇਸ ਸਾਲ ਸੇਲਰਜ਼ ਲਈ ਜਾਗਰੂਕਤਾ ਹਫ਼ਤਾ ਮਨਾਇਆ ਗਿਆ ਸੀ। ਇਸ ਦੌਰਾਨ ਬ੍ਰਿਟੇਨ ਦੀ ਸੰਸਥਾ 'ਸ਼ਿਪਰੇਕਡ ਮੈਰੀਨ ਸੁਸਾਇਟੀ' ਨੇ ਕੁਝ ਫੋਟੋਗ੍ਰਾਫ਼ਰਾਂ ਨੂੰ ਇਹ...	ਕ੍ਰਿਸ ਹੇਰਿੰਗ ਨੇ ਇਹ ਤਸਵੀਰ ਨੈਰਫਾਕ ਵਿਚ ਖਿੱਚੀ ਅਤੇ ਇਸ ਨੂੰ ਨਾਮ ਦਿੱਤਾ 'ਫਾਇਟਿੰਗ ਟੂ ਦਾ ਐਂਡ' ਮਤਲਬ ਆਖ਼ਰ ਤੱਕ ਸੰਘਰਸ਼ ਓਐੱਨ ਹਮਫ਼ੇਜ਼ ਨੇ ਡਰਹਮ ਦੇ...
ਡਾਂਸ ਫਲੋਰ ਆਪਣੇ ਆਪ ਵਿੱਚ ਇੱਕ ਕਾਇਨਾਤ ਹੈ। ਇੱਥੇ ਗੁਮਨਾਮ ਹੋਣਾ ਉਨ੍ਹਾਂ ਅੰਦਰ ਇੱਕ ਭਰੋਸਾ ਪੈਦਾ ਕਰਦਾ ਹੈ। ਇਸ ਫਲੋਰ 'ਤੇ ਬਹੁਤ ਸਾਰੇ ਲੋਕ ਮਿਲਦੇ ਹਨ, ਜਿਨ੍ਹਾਂ ਦੇ...	ਫਿਰ ਭਾਵੇਂ ਉਹ ਸਾਲਸਾ ਹੋਵੇ, ਕਿਜ਼ੁੰਬਾ ਹੋਵੇ ਜਾਂ ਫਿਰ ਬਸ਼ਾਤਾ, ਕਿਸੇ ਅਜਨਬੀ ਨਾਲ ਨੱਚ ਕੇ ਤੁਸੀਂ ਘਰ ਆ ਜਾਂਦੇ ਹੋ। ਤੁਹਾਡਾ ਸਮਾਂ ਗੁਜ਼ਰ ਜਾਂਦਾ ਹੈ। ਜਿਸ ਸ਼ਹਿਰ ਵਿੱਚ ਉਹ...
ਭਾਰਤ ਦੇ ਪ੍ਰਧਾਨ ਮੰਤਰੀ ਨਰਿੰਦਰ ਮੋਦੀ ਨੇ ਸੰਯੁਕਤ ਰਾਸ਼ਟਰ ਦੀ ਜਨਰਲ ਅਸੈਂਬਲੀ ਨੂੰ ਸੰਬੋਧਿਤ ਕਰਦੇ ਹੋਏ ਕਿਹਾ ਕਿ ਉਹ ਸੰਯੁਕਤ ਰਾਸ਼ਟਰ ਦੀ ਜਨਰਲ ਅਸੈਂਬਲੀ ਦੇ ਲੋਕਾਂ ਵਿੱ...	ਮੋਦੀ ਨੇ ਸਭ ਤੋਂ ਪਹਿਲਾਂ ਆਪਣੇ ਭਾਸ਼ਣ ਵਿੱਚ ਮਹਾਤਮਾ ਗਾਂਧੀ ਨੂੰ ਯਾਦ ਕੀਤਾ। ਉਨ੍ਹਾਂ ਨੇ ਕਿਹਾ ਕਿ ਇਹ ਸਾਲ ਇਸ ਲਈ ਵੀ ਕਾਫ਼ੀ ਅਹਿਮ ਹੈ ਕਿਉਂਕਿ ਭਾਰਤ ਇਸ ਸਾਲ...
ਉਨ੍ਹਾਂ ਨੂੰ 'ਲਾਪਤਾ 54' ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਹ ਉਹ ਭਾਰਤੀ ਫ਼ੌਜੀ ਹਨ ਜੋ ਭਾਰਤ ਅਤੇ ਪਾਕਿਸਤਾਨ ਵਿਚਾਲੇ ਹੋਈਆਂ ਲੜਾਈਆਂ ਦੀ ਨਫ਼ਰਤ ਵਿੱਚ ਭੁਲਾ ਦਿੱਤੇ ਗਏ ਹਨ। ਮੰਨਿਆ...	ਭਾਰਤ-ਪਾਕਿਸਤਾਨ ਲੜਾਈ ਦੇ ਦੌਰਾਨ ਪਾਕਿਸਤਾਨ ਫ਼ੌਜ ਦੁਆਰਾ ਕਬਜ਼ੇ ਵਿੱਚ ਲਏ ਗਏ ਭਾਰਤੀ ਫ਼ੌਜੀ ਭਾਰਤ ਅਤੇ ਪਾਕਿਸਤਾਨ ਨੇ ਕਸ਼ਮੀਰ ਦੇ ਵਿਵਾਦਿਤ ਖੇਤਰ 'ਤੇ ਕਬਜ਼ਾ ਕਰਨ...
"ਇੱਕ ਕੋਰੋਨਾ ਸੀ 2020 ਵਾਲਾ..... ਤੇ ਇੱਕ ਕੋਰੋਨਾ ਹੈ 2021 ਵਾਲਾ।	ਦੋਵਾਂ ਵਿੱਚ ਕਈ ਬੁਨਿਆਦੀ ਫ਼ਰਕ ਹਨ। ਦੂਜੀ ਲਹਿਰ ਦੌਰਾਨ ਕੋਰੋਨਾ ਪਹਿਲਾਂ ਦੇ ਮੁਕਾਬਲੇ ਜ਼ਿਆਦਾ ਫ਼ੈਲ ਰਿਹਾ ਹੈ, ਪਰ ਘੱਟ ਘਾਤਕ ਹੈ। ਬੱਚਿਆਂ ਤੇ ਨੌਜਵਾਨਾਂ ਨੂੰ ਆਪਣੀ ਗ੍ਰਿਫ਼ਤ...

Transformer Model: T5
(Text-To-Text Transfer Transformer)

In this project, we utilized 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.

Model Architecture and Features:

- Model Name: T5 (Text-To-Text Transfer Transformer)
- Architecture: Encoder-Decoder architecture, suitable for sequence-to-sequence tasks
- Preprocessing: The input texts are prefixed with "summarize: " before tokenization to indicate the task.
- Optimizer: AdamWeightDecay with a learning rate of 2e-5 and weight decay of 0.01.
- Data Collation: Using DataCollatorForSeq2Seq to dynamically pad inputs and outputs during batching.
- Batch Size: 8 for both training and evaluation.
- Evaluation Metric: **rouge score-** To use it during training, we will create a function compute_metrics that passes the predictions and labels as a parameter. The length of prediction is added under key ‘gen_len’

CODE :

Implementation Details:

Loading Libraries:

```
!pip install transformers
!pip install datasets
!pip install evaluate
!pip install rouge_score
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-packages (4.41.1)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers) (3.14.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.23.1)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (1.25.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (24.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (2024.5.15)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers) (2.31.0)
Requirement already satisfied: tokenizers<0.20,>=0.19 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.19.1)
Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.4.3)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.66.4)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.23.0->transformers) (2023.6.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.23.0->transformers) (4.11.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2024.2.2)
Collecting datasets
  Downloading datasets-2.19.1-py3-none-any.whl (542 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 542.0/542.0 kB 7.7 MB/s eta 0:00:00
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from datasets) (3.14.0)
```

Loading the Dataset:

```
+ Code + Text                                                    Connect T4   Gemini
Successfully installed datasets-2.19.1 dill-0.3.8 multiprocess-0.70.16 xxhash-3.4.1
```

```
from datasets import load_dataset
config_name='punjabi'

dataset = load_dataset("csebuetnlp/xlsum",config_name)
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:89: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab a
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
Downloading data: 100% 35.1M/35.1M [00:00<00:00, 91.4MB/s]
Downloading data: 100% 2.93M/2.93M [00:00<00:00, 22.1MB/s]
Downloading data: 100% 2.96M/2.96M [00:00<00:00, 19.6MB/s]
Generating train split: 100% 8215/8215 [00:01<00:00, 7313.74 examples/s]
Generating test split: 100% 1026/1026 [00:00<00:00, 12787.85 examples/s]
Generating validation split: 100% 1026/1026 [00:00<00:00, 9173.35 examples/s]
```

```
[ ] dataset
```

```
DatasetDict({
    train: Dataset({
        features: ['id', 'url', 'title', 'summary', 'text'],
        num_rows: 8215
    })
    test: Dataset({
        features: ['id', 'url', 'title', 'summary', 'text'],
        num_rows: 1026
    })
    validation: Dataset({
        features: ['id', 'url', 'title', 'summary', 'text'],
        num_rows: 1026
    })
})
```

```
dataset['train'][0]
```

```
{'id': 'international-45496712',
 'url': 'https://www.bbc.com/punjabi/international-45496712',
 'title': 'ਦੇਖਿਆ ਹੈ ਕਦੇ ਸਮੁੰਦਰ ਦਾ ਇਹ ਰੂਪ - ਤਸਵੀਰਾਂ',
 'summary': "ਬ੍ਰਿਟੇਨ ਵਿਚ ਇਸ ਸਾਲ ਸੇਲਰਜ਼ ਲਈ ਜਾਗਰੂਕਤਾ ਹਫ਼ਤਾ ਮਨਾਇਆ ਗਿਆ ਸੀ। ਇਸ ਦੌਰਾਨ ਬ੍ਰਿਟੇਨ ਦੀ ਸੰਸਥਾ 'ਸ਼ਿਪਰੇਕਡ ਮੈਰੀਨ ਸੁਸਾਇਟੀ' ਨੇ ਕੁਝ ਫੋਟੋਗ੍ਰਾਫ਼ਰਾਂ ਨੂੰ ਇਹ ਤਸਵੀਰਾਂ ਖਿੱਚਣ ਦੀ ਚੁਣੌਤੀ ਦਿੱਤੀ ਸੀ। ਉਸ ਮੁਕਾਬਲੇ ਵਿੱਚੋਂ ਹੀ ਇਹ ਕੁਝ ਤਸਵੀਰਾਂ ਚੁਣੀਆਂ ਗਈਆਂ ਹਨ।",
 'text': "ਕ੍ਰਿਸ ਹੇਰਿੰਗ ਨੇ ਇਹ ਤਸਵੀਰ ਨੌਰਫਾਕ ਵਿਚ ਖਿੱਚੀ ਅਤੇ ਇਸ ਨੂੰ ਨਾਮ ਦਿੱਤਾ 'ਫਾਇਟਿੰਗ ਟੂ ਦਾ ਐਂਡ' ਮਤਲਬ ਆਖ਼ਰ ਤੱਕ ਸੰਘਰਸ਼ ਓਐੱਨ ਹਮਫੇਜ਼ ਨੇ ਡਰਹਮ ਦੇ ਲਾਇਟ ਹਾਊਸ ਦੇ ਨੇੜੇ ਉੱਠਦੀਆਂ ਲਹਿਰਾਂ ਦੇ ਜੋਸ਼ ਨੂੰ
```

Using T5-small Transformer model :

\+ Code + Text Connect T4 ✦ Gemini

```
[ ] from transformers import AutoTokenizer

    checkpoint = "t5-small"
    tokenizer = AutoTokenizer.from_pretrained(checkpoint)
```

```
tokenizer_config.json: 100% 2.32k/2.32k [00:00<00:00, 166kB/s]
spiece.model: 100% 792k/792k [00:00<00:00, 3.08MB/s]
tokenizer.json: 100% 1.39M/1.39M [00:00<00:00, 4.49MB/s]
```

```
prefix = "summarize: "

def preprocess_fn(examples):
    inputs = [prefix + doc for doc in examples["text"]]
    model_inputs = tokenizer(inputs, max_length=1024, truncation=True)
    text_target=[doc for doc in examples['summary']]
    with tokenizer.as_target_tokenizer():
      labels = tokenizer(text_target, max_length=128, truncation=True)

    model_inputs["labels"] = labels["input_ids"]
    return model_inputs
```

\+ Code + Text Connect T4 ✦ Gemini

```
[ ] tokenized_dataset = dataset.map(preprocess_fn, batched=True)
```

```
Map: 100% 8215/8215 [00:44<00:00, 295.93 examples/s]
/usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils_base.py:3946: UserWarning: `as_target_tokenizer` is deprecated and will be removed in v5
  warnings.warn(
Map: 100% 1026/1026 [00:03<00:00, 274.42 examples/s]
Map: 100% 1026/1026 [00:02<00:00, 409.78 examples/s]
```

```
[ ] tokenized_dataset['train'][0]
```

```
ਦੇ ਚਾਲਕ ਦਲ ਦੇ ਦੋ ਮੈਂਬਰਾਂ ਨੂੰ ਕੈਮਰੇ ਵਿਚ ਕੈਦ ਕੀਤਾ। ਇਹ ਤਸਵੀਰ ਇਆਨ ਰੀਡ ਦੇ ਕੈਮਰੇ ਰਾਹੀ ਸਾਹਮਣੇ ਆਈ। ਇਹ ਬੇੜੀ 1980 ਵਿਆਂ ਤੋਂ ਸਕਾਲ੍ਹਵਾ ਦੇ ਤਟ ਉੱਤੇ ਖੜ੍ਹੀ ਹੈ। ਤੁਹਾਨੂੰ ਇਹ ਵੀਡੀਓ ਵੀ ਪਸੰਦ ਆ ਸਕਦੇ ਹਨ-
(ਬੀਬੀਸੀ ਪੰਜਾਬੀ ਨਾਲ FACEBOOK, INSTAGRAM, TWITTERਅਤੇ YouTube 'ਤੇ ਜੁੜੋ।)",
 'input_ids': [21603,
  10,
  3,
  2,
  3,
  2,
  3,
  2,
  3,
  2,
  3,
```

Model and optimizer Initialization:

```
[ ] from transformers import DataCollatorForSeq2Seq
    datacollator=DataCollatorForSeq2Seq(tokenizer=tokenizer,model=checkpoint)

[ ] from transformers import create_optimizer, AdamWeightDecay

    optimizer = AdamWeightDecay(learning_rate=2e-5, weight_decay_rate=0.01)

    from transformers import TFAutoModelForSeq2SeqLM

    model = TFAutoModelForSeq2SeqLM.from_pretrained(checkpoint)

    config.json: 100% 1.21k/1.21k [00:00<00:00, 83.8kB/s]
    model.safetensors: 100% 242M/242M [00:00<00:00, 296MB/s]
    All PyTorch model weights were used when initializing TFT5ForConditionalGeneration.

    All the weights of TFT5ForConditionalGeneration were initialized from the PyTorch model.
    If your task is similar to the task the model of the checkpoint was trained on, you can already use TFT5ForConditionalGeneration for predictions without further
```

Breakup of Datasets:

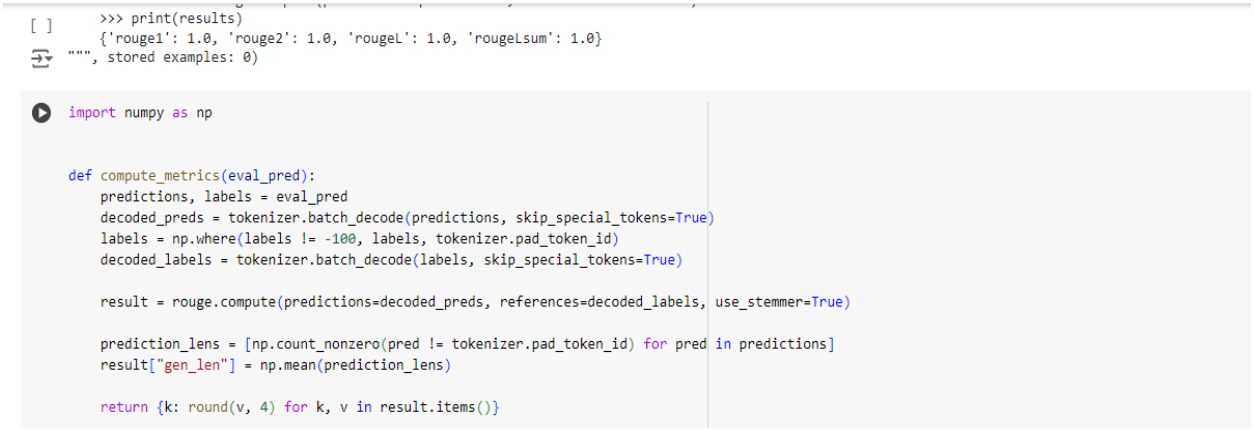
The XLSum dataset, configured for Punjabi, which has training and validation sets to ensure effective training and evaluation of the model.

The validation set is used to tune the model's hyperparameters and evaluate its performance during training. It helps monitor the model's progress and prevents overfitting.

```
train_set = model.prepare_tf_dataset(
    tokenized_dataset["train"],
    shuffle=True,
    batch_size=8,
    collate_fn=datacollator,
)

val_set = model.prepare_tf_dataset(
    tokenized_dataset["validation"],
    shuffle=False,
    batch_size=8,
    collate_fn=datacollator,
)
```

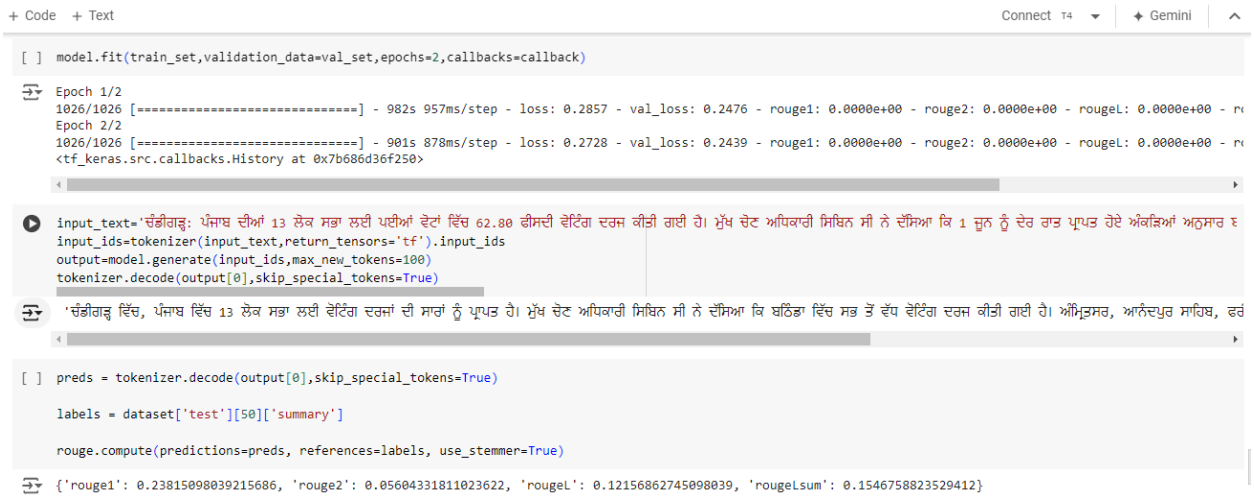
Training:

```
+ Code + Text                                                    Connect T4

[ ] import tensorflow as tf

    model.compile(optimizer=optimizer)

    import evaluate

    rouge = evaluate.load("rouge")
    rouge

EvaluationModule(name: "rouge", module_type: "metric", features: [{'predictions': Value(dtype='string', id='sequence'), 'references':
Sequence(feature=Value(dtype='string', id='sequence'), length=-1, id=None)}, {'predictions': Value(dtype='string', id='sequence'), 'references':
Value(dtype='string', id='sequence')}], usage: """
Calculates average rouge scores for a list of hypotheses and references
Args:
    predictions: list of predictions to score. Each prediction
        should be a string with tokens separated by spaces.
    references: list of reference for each prediction. Each
        reference should be a string with tokens separated by spaces.
    rouge_types: A list of rouge types to calculate.
        Valid names:
        `"rouge{n}"` (e.g. `"rouge1"`, `"rouge2"`) where: {n} is the n-gram based scoring,
        `"rougeL"`: Longest common subsequence based scoring.
        `"rougeLsum"`: rougeLsum splits text using `"
"`.
        See details in https://github.com/huggingface/datasets/issues/617
    use_stemmer: Bool indicating whether Porter stemmer should be used to strip word suffixes.
    use_aggregator: Return aggregates if this is set to True
```



```
[ ]     >>> print(results)
        {'rouge1': 1.0, 'rouge2': 1.0, 'rougeL': 1.0, 'rougeLsum': 1.0}
    """, stored examples: 0)

    import numpy as np


    def compute_metrics(eval_pred):
        predictions, labels = eval_pred
        decoded_preds = tokenizer.batch_decode(predictions, skip_special_tokens=True)
        labels = np.where(labels != -100, labels, tokenizer.pad_token_id)
        decoded_labels = tokenizer.batch_decode(labels, skip_special_tokens=True)

        result = rouge.compute(predictions=decoded_preds, references=decoded_labels, use_stemmer=True)

        prediction_lens = [np.count_nonzero(pred != tokenizer.pad_token_id) for pred in predictions]
        result["gen_len"] = np.mean(prediction_lens)

        return {k: round(v, 4) for k, v in result.items()}
```

Model compilation and Evaluation:



Results:

After training the model for 2 epochs, we evaluate its performance using the rouge metric for the 50th entry of test data. The input_text given is the ‘text’ of dataset[‘test’][50].

The results are as follows:

- {'rouge1': 0.23815098039215686,
- 'rouge2': 0.05604331811023622,
- 'rougeL': 0.12156862745098039,
- 'rougeLsum': 0.1546758823529412}