Content

1. Introduction

1.1. Author's Details

Name : Nisarg Patel

PRN : 21070126060

Batch : AIML - A3

Git-Repo : GitRepo

1.2. Downloading and Importing the required Libraries

In [ ]:
```
# This Python 3 environment comes with many helpful analytics libraries i
# It is defined by the kaggle/python Docker image: https://github.com/kag
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) wil

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) th
# You can also write temporary files to /kaggle/temp/, but they won't be
```

In [1]:
```
!pip install transformers -q
!pip install keras_nlp -q
!pip install datasets -q
!pip install huggingface-hub -q
!pip install rouge-score -q
```

```
WARNING: Running pip as the 'root' user can result in broken permissions a
nd conflicting behaviour with the system package manager. It is recommende
d to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 23.0.1 -> 23.2.1
[notice] To update, run: pip install --upgrade pip
WARNING: Running pip as the 'root' user can result in broken permissions a
nd conflicting behaviour with the system package manager. It is recommende
d to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 23.0.1 -> 23.2.1
[notice] To update, run: pip install --upgrade pip
WARNING: Running pip as the 'root' user can result in broken permissions a
nd conflicting behaviour with the system package manager. It is recommende
d to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 23.0.1 -> 23.2.1
[notice] To update, run: pip install --upgrade pip
WARNING: Running pip as the 'root' user can result in broken permissions a
nd conflicting behaviour with the system package manager. It is recommende
d to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 23.0.1 -> 23.2.1
[notice] To update, run: pip install --upgrade pip
WARNING: Running pip as the 'root' user can result in broken permissions a
nd conflicting behaviour with the system package manager. It is recommende
d to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 23.0.1 -> 23.2.1
[notice] To update, run: pip install --upgrade pip
```

In [2]:

```
# Importing the needed libraries
import nltk
import numpy as np
import tensorflow as tf
from tensorflow import keras
```

```
D0927 13:09:51.240667260      15 config.cc:119]                         gRP
C EXPERIMENT tcp_frame_size_tuning               OFF (default:OFF)
D0927 13:09:51.240691774      15 config.cc:119]                         gRP
C EXPERIMENT tcp_rcv_lowat                       OFF (default:OFF)
D0927 13:09:51.240695346      15 config.cc:119]                         gRP
C EXPERIMENT peer_state_based_framing            OFF (default:OFF)
D0927 13:09:51.240698246      15 config.cc:119]                         gRP
C EXPERIMENT flow_control_fixes                  ON  (default:ON)
D0927 13:09:51.240700761      15 config.cc:119]                         gRP
C EXPERIMENT memory_pressure_controller          OFF (default:OFF)
D0927 13:09:51.240703686      15 config.cc:119]                         gRP
C EXPERIMENT unconstrained_max_quota_buffer_size OFF (default:OFF)
D0927 13:09:51.240706300      15 config.cc:119]                         gRP
C EXPERIMENT new_hpack_huffman_decoder           ON  (default:ON)
D0927 13:09:51.240709794      15 config.cc:119]                         gRP
C EXPERIMENT event_engine_client                 OFF (default:OFF)
D0927 13:09:51.240712399      15 config.cc:119]                         gRP
C EXPERIMENT monitoring_experiment               ON  (default:ON)
D0927 13:09:51.240714981      15 config.cc:119]                         gRP
C EXPERIMENT promise_based_client_call           OFF (default:OFF)
D0927 13:09:51.240717523      15 config.cc:119]                         gRP
C EXPERIMENT free_large_allocator                OFF (default:OFF)
D0927 13:09:51.240720131      15 config.cc:119]                         gRP
C EXPERIMENT promise_based_server_call           OFF (default:OFF)
D0927 13:09:51.240722819      15 config.cc:119]                         gRP
C EXPERIMENT transport_supplies_client_latency   OFF (default:OFF)
D0927 13:09:51.240725359      15 config.cc:119]                         gRP
C EXPERIMENT event_engine_listener               OFF (default:OFF)
I0927 13:09:51.240926792      15 ev_epoll1_linux.cc:122]                grp
c epoll fd: 62
D0927 13:09:51.246311038      15 ev_posix.cc:144]                       Usi
ng polling engine: epoll1
D0927 13:09:51.246341979      15 dns_resolver_ares.cc:822]              Usi
ng ares dns resolver
D0927 13:09:51.246741129      15 lb_policy_registry.cc:46]              reg
istering LB policy factory for "priority_experimental"
D0927 13:09:51.246753005      15 lb_policy_registry.cc:46]              reg
istering LB policy factory for "outlier_detection_experimental"
D0927 13:09:51.246757156      15 lb_policy_registry.cc:46]              reg
istering LB policy factory for "weighted_target_experimental"
D0927 13:09:51.246760808      15 lb_policy_registry.cc:46]              reg
istering LB policy factory for "pick_first"
D0927 13:09:51.246764443      15 lb_policy_registry.cc:46]              reg
istering LB policy factory for "round_robin"
D0927 13:09:51.246768013      15 lb_policy_registry.cc:46]              reg
istering LB policy factory for "weighted_round_robin_experimental"
D0927 13:09:51.246776060      15 lb_policy_registry.cc:46]              reg
istering LB policy factory for "ring_hash_experimental"
D0927 13:09:51.246795623      15 lb_policy_registry.cc:46]              reg
istering LB policy factory for "grpclb"
D0927 13:09:51.246836567      15 lb_policy_registry.cc:46]              reg
istering LB policy factory for "rls_experimental"
D0927 13:09:51.246853547      15 lb_policy_registry.cc:46]              reg
istering LB policy factory for "xds_cluster_manager_experimental"
D0927 13:09:51.246858420      15 lb_policy_registry.cc:46]              reg
istering LB policy factory for "xds_cluster_impl_experimental"
D0927 13:09:51.246862987      15 lb_policy_registry.cc:46]              reg
istering LB policy factory for "cds_experimental"
D0927 13:09:51.246875434      15 lb_policy_registry.cc:46]              reg
istering LB policy factory for "xds_cluster_resolver_experimental"
```

```
D0927 13:09:51.246879674      15 lb_policy_registry.cc:46]             reg
istering LB policy factory for "xds_override_host_experimental"
D0927 13:09:51.246883730      15 lb_policy_registry.cc:46]             reg
istering LB policy factory for "xds_wrr_locality_experimental"
D0927 13:09:51.246888797      15 certificate_provider_registry.cc:35]  reg
istering certificate provider factory for "file_watcher"
I0927 13:09:51.249397944      15 socket_utils_common_posix.cc:408]     Dis
abling AF_INET6 sockets because ::1 is not available.
I0927 13:09:51.265229770     397 socket_utils_common_posix.cc:337]     TCP
_USER_TIMEOUT is available. TCP_USER_TIMEOUT will be used thereafter
E0927 13:09:51.272197672     397 oauth2_credentials.cc:236]            oau
th_fetch: UNKNOWN:C-ares status is not ARES_SUCCESS qtype=A name=metadata.
google.internal. is_balancer=0: Domain name not found {created_time:"2023-
09-27T13:09:51.272182351+00:00", grpc_status:2}
```

1.3. Loading the Dataset

In [3]:
```
# Loading the Xsum Dataset
from datasets import load_dataset
dataset = load_dataset("xsum", split="train")
print(dataset)
```

```
/usr/local/lib/python3.8/site-packages/tqdm/auto.py:21: TqdmWarning: IProg
ress not found. Please update jupyter and ipywidgets. See https://ipywidge
ts.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
Downloading builder script: 100%|██████████| 5.76k/5.76k [00:00<00:00, 10.
5MB/s]
Downloading readme: 100%|██████████| 6.24k/6.24k [00:00<00:00, 20.0MB/s]
Downloading data files:   0%|          | 0/2 [00:00<?, ?it/s]
Downloading data:   0%|          | 0.00/255M [00:00<?, ?B/s]
Downloading data:   1%|          | 1.57M/255M [00:00<00:16, 15.7MB/s]
Downloading data:   2%|          | 6.34M/255M [00:00<00:07, 34.5MB/s]
Downloading data:   4%|▏         | 11.3M/255M [00:00<00:05, 41.3MB/s]
Downloading data:   6%|▏         | 16.2M/255M [00:00<00:05, 44.3MB/s]
Downloading data:   8%|▏         | 21.2M/255M [00:00<00:05, 46.3MB/s]
Downloading data:  10%|▏         | 26.1M/255M [00:00<00:04, 47.4MB/s]
Downloading data:  12%|▏         | 31.1M/255M [00:00<00:04, 48.2MB/s]
Downloading data:  14%|▏         | 36.0M/255M [00:00<00:04, 48.6MB/s]
Downloading data:  16%|▏         | 41.0M/255M [00:00<00:04, 48.9MB/s]
Downloading data:  18%|▏         | 46.0M/255M [00:01<00:04, 49.1MB/s]
Downloading data:  20%|▏         | 51.0M/255M [00:01<00:04, 49.4MB/s]
Downloading data:  22%|▏         | 55.9M/255M [00:01<00:04, 49.4MB/s]
Downloading data:  24%|▏         | 60.9M/255M [00:01<00:03, 49.6MB/s]
Downloading data:  26%|▏         | 65.9M/255M [00:01<00:03, 49.7MB/s]
Downloading data:  28%|▏         | 70.9M/255M [00:01<00:03, 49.8MB/s]
Downloading data:  30%|▏         | 75.9M/255M [00:01<00:03, 49.9MB/s]
Downloading data:  32%|▏         | 81.0M/255M [00:01<00:03, 50.1MB/s]
Downloading data:  34%|▏         | 86.0M/255M [00:01<00:03, 50.2MB/s]
Downloading data:  36%|▏         | 91.1M/255M [00:01<00:03, 50.3MB/s]
Downloading data:  38%|▏         | 96.1M/255M [00:02<00:03, 50.4MB/s]
Downloading data:  40%|▏         | 101M/255M [00:02<00:03, 50.5MB/s]
Downloading data:  42%|▏         | 106M/255M [00:02<00:02, 50.5MB/s]
Downloading data:  44%|▏         | 111M/255M [00:02<00:02, 50.5MB/s]
Downloading data:  46%|▏         | 116M/255M [00:02<00:02, 50.3MB/s]
Downloading data:  48%|▏         | 121M/255M [00:02<00:02, 50.3MB/s]
Downloading data:  50%|▏         | 126M/255M [00:02<00:02, 50.1MB/s]
Downloading data:  52%|▏         | 131M/255M [00:02<00:02, 50.0MB/s]
Downloading data:  54%|▏         | 136M/255M [00:02<00:02, 49.9MB/s]
Downloading data:  56%|▏         | 141M/255M [00:02<00:02, 49.4MB/s]
Downloading data:  57%|▏         | 146M/255M [00:03<00:02, 49.1MB/s]
Downloading data:  59%|▏         | 151M/255M [00:03<00:02, 49.3MB/s]
Downloading data:  61%|▏         | 156M/255M [00:03<00:02, 49.1MB/s]
Downloading data:  63%|▏         | 161M/255M [00:03<00:01, 48.8MB/s]
Downloading data:  65%|▏         | 166M/255M [00:03<00:01, 49.0MB/s]
Downloading data:  67%|▏         | 171M/255M [00:03<00:01, 49.2MB/s]
Downloading data:  69%|▏         | 176M/255M [00:03<00:01, 49.2MB/s]
Downloading data:  71%|▏         | 181M/255M [00:03<00:01, 49.4MB/s]
Downloading data:  73%|▏         | 186M/255M [00:03<00:01, 48.8MB/s]
Downloading data:  75%|▏         | 191M/255M [00:03<00:01, 48.6MB/s]
Downloading data:  77%|▏         | 196M/255M [00:04<00:01, 48.3MB/s]
Downloading data:  79%|▏         | 201M/255M [00:04<00:01, 48.3MB/s]
Downloading data:  81%|▏         | 205M/255M [00:04<00:01, 48.6MB/s]
Downloading data:  83%|▏         | 210M/255M [00:04<00:00, 48.7MB/s]
Downloading data:  85%|▏         | 215M/255M [00:04<00:00, 47.8MB/s]
Downloading data:  86%|▏         | 220M/255M [00:04<00:00, 47.3MB/s]
Downloading data:  88%|▏         | 225M/255M [00:04<00:00, 47.5MB/s]
Downloading data:  90%|▏         | 230M/255M [00:04<00:00, 46.9MB/s]
Downloading data:  92%|▏         | 234M/255M [00:04<00:00, 45.3MB/s]
Downloading data:  94%|▏         | 239M/255M [00:04<00:00, 44.5MB/s]
Downloading data:  96%|▏         | 243M/255M [00:05<00:00, 43.8MB/s]
Downloading data:  97%|▏         | 248M/255M [00:05<00:00, 43.4MB/s]
```

```
Downloading data: 100%|██████████| 255M/255M [00:05<00:00, 47.6MB/s]
Downloading data files:  50%|█████     | 1/2 [00:06<00:06,  6.14s/it]
Downloading data: 2.72MB [00:00, 36.5MB/s]
Downloading data files: 100%|██████████| 2/2 [00:06<00:00,  3.36s/it]
Generating train split: 100%|██████████| 204045/204045 [00:57<00:00, 3519.
53 examples/s]
Generating validation split: 100%|██████████| 11332/11332 [00:31<00:00, 35
6.35 examples/s]
Generating test split: 100%|██████████| 11334/11334 [00:31<00:00, 355.24 e
xamples/s]
Dataset({
    features: ['document', 'summary', 'id'],
    num_rows: 204045
})
```

In [4]:
```
# Printing the first element of the dataset
print(dataset[0])
```

```
{'document': 'The full cost of damage in Newton Stewart, one of the areas
worst affected, is still being assessed.\nRepair work is ongoing in Hawick
and many roads in Peeblesshire remain badly affected by standing water.\nT
rains on the west coast mainline face disruption due to damage at the Lami
ngton Viaduct.\nMany businesses and householders were affected by flooding
in Newton Stewart after the River Cree overflowed into the town.\nFirst Mi
nister Nicola Sturgeon visited the area to inspect the damage.\nThe waters
breached a retaining wall, flooding many commercial properties on Victoria
Street - the main shopping thoroughfare.\nJeanette Tate, who owns the Cinn
amon Cafe which was badly affected, said she could not fault the multi-age
ncy response once the flood hit.\nHowever, she said more preventative work
could have been carried out to ensure the retaining wall did not fail.\n"I
t is difficult but I do think there is so much publicity for Dumfries and
the Nith - and I totally appreciate that - but it is almost like we\'re ne
glected or forgotten," she said.\n"That may not be true but it is perhaps
my perspective over the last few days.\n"Why were you not ready to help us
a bit more when the warning and the alarm alerts had gone out?"\nMeanwhil
e, a flood alert remains in place across the Borders because of the consta
nt rain.\nPeebles was badly hit by problems, sparking calls to introduce m
ore defences in the area.\nScottish Borders Council has put a list on its
website of the roads worst affected and drivers have been urged not to ign
ore closure signs.\nThe Labour Party\'s deputy Scottish leader Alex Rowley
was in Hawick on Monday to see the situation first hand.\nHe said it was i
mportant to get the flood protection plan right but backed calls to speed
up the process.\n"I was quite taken aback by the amount of damage that has
been done," he said.\n"Obviously it is heart-breaking for people who have
been forced out of their homes and the impact on businesses."\nHe said it
was important that "immediate steps" were taken to protect the areas most
vulnerable and a clear timetable put in place for flood prevention plan
s.\nHave you been affected by flooding in Dumfries and Galloway or the Bor
ders? Tell us about your experience of the situation and how it was handle
d. Email us on selkirk.news@bbc.co.uk or dumfries@bbc.co.uk.', 'summary':
'Clean-up operations are continuing across the Scottish Borders and Dumfri
es and Galloway after flooding caused by Storm Frank.', 'id': '35232142'}
```

1.4. Splitting the Dataset into Train and Test

In [5]:
```
# Splitting the dataset into train and test
datasets = dataset.train_test_split(train_size=0.1,test_size=0.1)
```

In [6]:
```
# Checking the length of the train and test split
print(len(datasets['train']))
print(len(datasets['test']))
```

```
20404
20405
```

1.5. Describing the parameters for the model

In [7]:
```
# Define maximum input length for tokenization
MAX_INPUT_LENGTH = 1024

# Define minimum and maximum target lengths for summarization
MIN_TARGET_LENGTH = 5
MAX_TARGET_LENGTH = 128

# Define batch size for training
BATCH_SIZE = 8

# Define learning rate for optimizer
LEARNING_RATE = 2e-5

# Define maximum number of epochs for training
MAX_EPOCHS = 1

# Define the pre-trained T5 model checkpoint to be used
MODEL_CHECKPOINT = "t5-small" # Name of Model
```

2. Tokenization

2.1. Importing the Tokenizer

In [8]:
```
# Importing the tokenizer
from transformers import AutoTokenizer
tokenizer = AutoTokenizer.from_pretrained(MODEL_CHECKPOINT)
```

```
Downloading (…)okenizer_config.json: 100%|██████████| 2.32k/2.32k [00:00<0
0:00, 1.38MB/s]
Downloading (…)ve/main/spiece.model: 100%|██████████| 792k/792k [00:00<00:
00, 12.0MB/s]
Downloading (…)/main/tokenizer.json: 100%|██████████| 1.39M/1.39M [00:00<0
0:00, 25.6MB/s]
```

2.2. Adding the summarization prefix

In [9]:
```
# Check if the specified model checkpoint is either "t5-small" or "t5-bas
if MODEL_CHECKPOINT in ["t5-small", "t5-base"]:
    # If yes, set the prefix for summarization task
    prefix = "summarize: "
else:
    # If not, set an empty prefix
    prefix = ""
```

3. Preprocessing

3.1. Defining the Preprocessing Function

In [10]:
```
#Preprocessing
def preprocess_function(examples):
    inputs = [prefix + doc for doc in examples["document"]]
    model_inputs = tokenizer(inputs, max_length=MAX_INPUT_LENGTH,truncati
    # Setup the tokenizer for targets
    with tokenizer.as_target_tokenizer():
        labels = tokenizer(
            examples["summary"], max_length=MAX_TARGET_LENGTH, truncation
    model_inputs["labels"] = labels["input_ids"]
    return model_inputs
```

3.2. Mapping the Preprocessing Function to the Train and Test Datasets

In [11]:
```
# Mapping the preprocessing function to the train and test datasets
tokenized_datasets = datasets.map(preprocess_function, batched=True)
```

```
Map:   0%|          | 0/20404 [00:00<?, ? examples/s]/usr/local/lib/python
3.8/site-packages/transformers/tokenization_utils_base.py:3660: UserWarnin
g: `as_target_tokenizer` is deprecated and will be removed in v5 of Transf
ormers. You can tokenize your labels by using the argument `text_target` o
f the regular `__call__` method (either in the same call as your input tex
ts if you use the same keyword arguments, or in a separate call.
  warnings.warn(
Map: 100%|██████████| 20404/20404 [00:12<00:00, 1614.67 examples/s]
Map: 100%|██████████| 20405/20405 [00:12<00:00, 1615.06 examples/s]
```

4. Training

4.1. Improting the Tensorflow Model

In [12]:
```
# Using tensorflow AutoModelForSeq2SeqLM for summarization
from transformers import TFAutoModelForSeq2SeqLM, DataCollatorForSeq2Seq
model = TFAutoModelForSeq2SeqLM.from_pretrained(MODEL_CHECKPOINT)
```

```
Downloading (…)lve/main/config.json: 100%|██████████| 1.21k/1.21k [00:00<0
0:00, 166kB/s]
Downloading model.safetensors: 100%|██████████| 242M/242M [00:05<00:00, 4
4.2MB/s]
All PyTorch model weights were used when initializing TFT5ForConditionalGe
neration.

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 traine
d on, you can already use TFT5ForConditionalGeneration for predictions wit
hout further training.
```

4.2. Using the DataCollatorForSeq2Seq for collating the data

In [13]:
```
# Using the DataCollatorForSeq2Seq for collating the data
from transformers import DataCollatorForSeq2Seq
data_collator = DataCollatorForSeq2Seq(tokenizer, model=model,return_tens
```

4.3. Creating the Training, Test and Generation Datasets

In [14]:
```
# Create a training dataset from the tokenized training data
train_dataset = tokenized_datasets["train"].to_tf_dataset(
    batch_size=BATCH_SIZE,                  # Set the batch size for train
    columns=["input_ids", "attention_mask", "labels"],  # Specify columns
    shuffle=True,                           # Shuffle the data for randomn
    collate_fn=data_collator                # Use the specified data colla
)

# Create a test dataset from the tokenized test data
test_dataset = tokenized_datasets["test"].to_tf_dataset(
    batch_size=BATCH_SIZE,                  # Set the batch size for test
    columns=["input_ids", "attention_mask", "labels"],  # Specify columns
    shuffle=False,                          # Do not shuffle the test data
    collate_fn=data_collator                # Use the specified data colla
)

# Create a generation dataset by selecting a subset of data from the toke
generation_dataset = (
    tokenized_datasets["test"]              # Access the tokenized test da
    .shuffle()                              # Shuffle the data for randomn
    .select(list(range(200)))               # Select a specific range of s
    .to_tf_dataset(                         # Convert the selected data to
        batch_size=BATCH_SIZE,              # Set the batch size
        columns=["input_ids", "attention_mask", "labels"],  # Specify col
        shuffle=False,                      # Do not shuffle the generatio
        collate_fn=data_collator            # Use the specified data colla
    )
)
```

```
You're using a T5TokenizerFast tokenizer. Please note that with a fast tok
enizer, using the `__call__` method is faster than using a method to encod
e the text followed by a call to the `pad` method to get a padded encodin
g.
```

5. Fine-Tuning

5.1. Tuning optimizer and compiling the model

In [15]:
```
# Tuning the optimizer and compiling the model
optimizer = keras.optimizers.Adam(learning_rate=LEARNING_RATE)
model.compile(optimizer=optimizer)
```

5.2. Model Summary

In [16]:
```
model.summary()
```

```
Model: "tft5_for_conditional_generation"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 shared (Embedding)          multiple                  16449536

 encoder (TFT5MainLayer)     multiple                  35330816

 decoder (TFT5MainLayer)     multiple                  41625344

=================================================================
Total params: 60,506,624
Trainable params: 60,506,624
Non-trainable params: 0
_________________________________________________________________
```

6. Evaluation

6.1. Importing the RougeL Metric

In [17]:
```
# Import the RougeL metric from keras_nlp.metrics module
import keras_nlp
rouge_l = keras_nlp.metrics.RougeL()

# Define a function to compute the RougeL metric
def metric_fn(eval_predictions):
    # Unpack the predictions and labels
    predictions, labels = eval_predictions

    # Decode the predicted tokens to text
    decoded_predictions = tokenizer.batch_decode(predictions, skip_specia
```

```
    # Replace masked label tokens
    for label in labels:
        label[label < 0] = tokenizer.pad_token_id

    # Decode the label tokens to text
    decoded_labels = tokenizer.batch_decode(labels, skip_special_tokens=T

    # Compute the RougeL score
    result = rouge_l(decoded_labels, decoded_predictions)

    # We will print only the F1 score, you can use other aggregation metr
    result = {"RougeL": result["f1_score"]}

    return result
```

```
Using TensorFlow backend
```

7. Training the Model

7.1. Training the Model

In [ ]:
```
from transformers.keras_callbacks import KerasMetricCallback
metric_callback = KerasMetricCallback(metric_fn,eval_dataset=generation_d
callbacks = [metric_callback]
# For now we will use our test set as our validation_data
model.fit(train_dataset, validation_data=test_dataset, epochs=MAX_EPOCHS,
```

```
https://symbolize.stripped_domain/r/?
trace=7af9b354c4f9,7af9b3486f8f,7af842acb0e8,7af842ac8692,7af842ac8b96,7af843
7af844256728,70f32832c9000a2f02b2bd353a9dfb35c743ca88:7af842017000-
7af842264750,8f79f803f683427be94b1cfeea32716e6ef365e4:7af829915000-
7af84186a830 *** SIGTERM received by PID 15 (TID 15) on cpu 44 from PID 1; stack
trace: *** PC: @ 0x7af9b354c4f9 (unknown) syscall @ 0x7af828c3605a 1152
(unknown) @ 0x7af9b3486f90 1872873984 (unknown) @ 0x7af842acb0e9 176
nsync::nsync_sem_wait_with_cancel_() @ 0x7af842ac8693 144
nsync::nsync_cv_wait_with_deadline_generic() @ 0x7af842ac8b97 32
nsync::nsync_cv_wait_with_deadline() @ 0x7af84387d09b 464
tensorflow::ProcessFunctionLibraryRuntime::RunSync() @ 0x7af8333f5a7d 368
tensorflow::KernelAndDeviceFunc::Run() @ 0x7af82d829615 480
tensorflow::EagerKernelExecute() @ 0x7af82d82a1eb 528
tensorflow::ExecuteNode::Run() @ 0x7af8333ed5fc 96
tensorflow::EagerExecutor::SyncExecute() @ 0x7af82d823976 1152 tensorflow::
(anonymous namespace)::EagerLocalExecute() @ 0x7af82d8265d4 1136
tensorflow::EagerExecute() @ 0x7af82d80b149 208
tensorflow::EagerOperation::Execute() @ 0x7af8333fe244 208
tensorflow::CustomDeviceOpHandler::Execute() @ 0x7af8393b835b 80
TFE_Execute @ 0x7af84214ae25 1440 TFE_Py_ExecuteCancelable() @
0x7af71777ca23 208 tensorflow::TFE_Py_ExecuteCancelable_wrapper() @
```

0x7af71777cd14 208 pybind11::cpp_function::initialize<>()::{lambda()#3}::_FUN() @ 0x7af717780616 768 pybind11::cpp_function::dispatcher() @ 0x7af9b377fb3e (unknown) cfunction_call_varargs @ 0x100000001 (unknown) (unknown) https://symbolize.stripped_domain/r/?trace=7af9b354c4f9,7af828c36059,7af9b3486f8f,7af842acb0e8,7af842ac8692,7af842 7af844256728,70f32832c9000a2f02b2bd353a9dfb35c743ca88:7af842017000-7af842264750,8f79f803f683427be94b1cfeea32716e6ef365e4:7af829915000-7af84186a830,1278088d049ad36cb636fbbc76303cb3:7af81d600000-7af828e4d7c0 E0927 15:10:54.481297 15 coredump_hook.cc:360] RAW: Remote crash gathering disabled for SIGTERM.

Sir i have tried using TPU's GPU's and normal CPU's for the computation purpose but since the dataset is too large everytime it was either thorwing oom error or it used to get disconnected. Despite of trying for 2 days i was unable to train the model however upon reaching the college i will use the GPU lab and get this model trained over there meanwhile i will be trying to train the model on google colab and kaggle as well if it get's trained i will update the file in the github where this is uploaded. Thanks sir for understanding.

8. Testing the Model

8.1. Testing the Model

In [ ]:
```
# Import the summarization pipeline from transformers library
from transformers import pipeline

# Create a summarization pipeline with the specified model and tokenizer
summarizer = pipeline("summarization", model=model, tokenizer=tokenizer,

# Print a separator line
print("-" * 1000)

# Print the original article content
print("________Original Article__________")
print(datasets['test'][0]['document'])

# Print a separator line
print("-" * 1000)

# Print the original summary
print("________Original Summary__________")
print(datasets['test'][0]['summary'])

# Print a separator line
print("-" * 1000)

# Generate and print the predicted summary
summarizer(datasets["test"][0]["document"], min_length=MIN_TARGET_LENGTH,
```

9. Saving the Model

In [ ]:
```
# Saving the model
model.save_pretrained("T5CustomSummarizer",from_pt=True)
tokenizer.save_pretrained("T5CustomSumTokenizer",from_pt=True)
```

In [ ]:
```

```