```
!pip install datasets transformers[sentencepiece] # transformers
!apt install git-lfs # Git Large File Storage
!git lfs install
!git config --global user.email "aakash280500@gmail.com"
!git config --global user.name "Aakash Mahesha"
!pip install -e .
import pandas as pd
import tensorflow as tf
from transformers import TFGPT2LMHeadModel # GPT2 training model
import re
import string
from transformers import GPT2Tokenizer # GPT2 Tokenizer
import numpy as np
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: datasets in /usr/local/lib/python3.9/dist-packages (2.11.0)
Requirement already satisfied: transformers[sentencepiece] in /usr/local/lib/python3.9/dist-packages (4.28.1)
Requirement already satisfied: fsspec[http]>=2021.11.1 in /usr/local/lib/python3.9/dist-packages (from datasets) (2023.4.0)
Requirement already satisfied: multiprocess in /usr/local/lib/python3.9/dist-packages (from datasets) (0.70.14)
Requirement already satisfied: xxhash in /usr/local/lib/python3.9/dist-packages (from datasets) (3.2.0)
Requirement already satisfied: pandas in /usr/local/lib/python3.9/dist-packages (from datasets) (1.5.3)
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.9/dist-packages (from datasets) (2.27.1)
Requirement already satisfied: huggingface-hub<1.0.0,>=0.11.0 in /usr/local/lib/python3.9/dist-packages (from datasets) (0.13.4)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.9/dist-packages (from datasets) (1.22.4)
Requirement already satisfied: dill<0.3.7,>=0.3.0 in /usr/local/lib/python3.9/dist-packages (from datasets) (0.3.6)
Requirement already satisfied: packaging in /usr/local/lib/python3.9/dist-packages (from datasets) (23.0)
Requirement already satisfied: responses<0.19 in /usr/local/lib/python3.9/dist-packages (from datasets) (0.18.0)
Requirement already satisfied: tqdm>=4.62.1 in /usr/local/lib/python3.9/dist-packages (from datasets) (4.65.0)
Requirement already satisfied: pyarrow>=8.0.0 in /usr/local/lib/python3.9/dist-packages (from datasets) (9.0.0)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.9/dist-packages (from datasets) (3.8.4)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.9/dist-packages (from datasets) (6.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.9/dist-packages (from transformers[sentencepiece]) (3.11.0)
Requirement already satisfied: tokenizers!=0.11.3,<0.14,>=0.11.1 in /usr/local/lib/python3.9/dist-packages (from transformers[sentencepi
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.9/dist-packages (from transformers[sentencepiece]) (2022.10.3
Requirement already satisfied: sentencepiece!=0.1.92,>=0.1.91 in /usr/local/lib/python3.9/dist-packages (from transformers[sentencepiece
Requirement already satisfied: protobuf<=3.20.2 in /usr/local/lib/python3.9/dist-packages (from transformers[sentencepiece]) (3.20.2)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.9/dist-packages (from aiohttp->datasets) (6.0.4)
Requirement already satisfied: charset-normalizer<4.0,>=2.0 in /usr/local/lib/python3.9/dist-packages (from aiohttp->datasets) (2.0.12)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.9/dist-packages (from aiohttp->datasets) (1.3.1)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.9/dist-packages (from aiohttp->datasets) (1.3.3)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.9/dist-packages (from aiohttp->datasets) (1.8.2)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in /usr/local/lib/python3.9/dist-packages (from aiohttp->datasets) (4.0.2)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.9/dist-packages (from aiohttp->datasets) (22.2.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.9/dist-packages (from huggingface-hub<1.0.0,>=0.11.0
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.9/dist-packages (from requests>=2.19.0->datasets) (1.26.1
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-packages (from requests>=2.19.0->datasets) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-packages (from requests>=2.19.0->datasets) (2022.12.7
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.9/dist-packages (from pandas->datasets) (2022.7.1)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.9/dist-packages (from pandas->datasets) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.9/dist-packages (from python-dateutil>=2.8.1->pandas->datasets) (1.16.
Reading package lists... Done
Building dependency tree
Reading state information... Done
git-lfs is already the newest version (2.9.2-1).
0 upgraded, 0 newly installed, 0 to remove and 24 not upgraded.
Error: Failed to call git rev-parse --git-dir: exit status 128
Git LFS initialized.
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Obtaining file:///content
ERROR: file:///content does not appear to be a Python project: neither 'setup.py' nor 'pyproject.toml' found.
```

```
# loading the dataset, which was extracted from HuggingFace API
fan_story_df = pd.read_csv('/content/dataset.csv')
fan_story_df
```

	Unnamed: 0	story	title	category
0	0	just a question\n\nWhat I thought when I first...	just a question by multifics123	Zom-B
1	1	pain\n\nIt may be sad but it can be better\n\n...	pain by Isaiah Thomas	Zenda
		Zombie Mayhem\n\nZombie	Zombie Mayhem by	Zoombie

```
fan_story_df.drop(columns=['Unnamed: 0','title','category'],inplace=True)
fan_story_df
```

	story
0	just a question\n\nWhat I thought when I first...
1	pain\n\nIt may be sad but it can be better\n\n...
2	Zombie Mayhem\n\nZombie Mayhem\n\nIn 2050, the...
3	Zero Wing 2: AYBASBTU Game Script\n\nZERO WING...
4	Not prosaic\n\nDisclaimer, I do not claim to o...
...	...
938	More\n\n**I've watched this movie 3 times this...
939	Change Comes From Within\n\n**As usual, I own ...
940	Changes\n\nYou know that feeling where people ...
941	Undying Hero\n\nZero percent.\n\nVictory canno...
942	Unlosing Haunted Mansion\n\n**Disclaimer**: I ...

943 rows × 1 columns

Pre-processing

```
# handling contractions
contractions_dict = { "ain't": "are not","'s":" is","aren't": "are not",
                     "can't": "cannot","can't've": "cannot have",
                     "'cause": "because","could've": "could have","couldn't": "could not",
                     "couldn't've": "could not have", "didn't": "did not","doesn't": "does not",
                     "don't": "do not","hadn't": "had not","hadn't've": "had not have",
                     "hasn't": "has not","haven't": "have not","he'd": "he would",
                     "he'd've": "he would have","he'll": "he will", "he'll've": "he will have",
                     "how'd": "how did","how'd'y": "how do you","how'll": "how will",
                     "I'd": "I would", "I'd've": "I would have","I'll": "I will",
                     "I'll've": "I will have","I'm": "I am","I've": "I have", "isn't": "is not",
                     "it'd": "it would","it'd've": "it would have","it'll": "it will",
                     "it'll've": "it will have", "let's": "let us","ma'am": "madam",
                     "mayn't": "may not","might've": "might have","mightn't": "might not",
                     "mightn't've": "might not have","must've": "must have","mustn't": "must not",
                     "mustn't've": "must not have", "needn't": "need not",
                     "needn't've": "need not have","o'clock": "of the clock","oughtn't": "ought not",
                     "oughtn't've": "ought not have","shan't": "shall not","sha'n't": "shall not",
                     "shan't've": "shall not have","she'd": "she would","she'd've": "she would have",
                     "she'll": "she will", "she'll've": "she will have","should've": "should have",
                     "shouldn't": "should not", "shouldn't've": "should not have","so've": "so have",
                     "that'd": "that would","that'd've": "that would have", "there'd": "there would",
                     "there'd've": "there would have", "they'd": "they would",
                     "they'd've": "they would have","they'll": "they will",
                     "they'll've": "they will have", "they're": "they are","they've": "they have",
                     "to've": "to have","wasn't": "was not","we'd": "we would",
                     "we'd've": "we would have","we'll": "we will","we'll've": "we will have",
                     "we're": "we are","we've": "we have", "weren't": "were not","what'll": "what will",
                     "what'll've": "what will have","what're": "what are", "what've": "what have",
                     "when've": "when have","where'd": "where did", "where've": "where have",
                     "who'll": "who will","who'll've": "who will have","who've": "who have",
                     "why've": "why have","will've": "will have","won't": "will not",
                     "won't've": "will not have", "would've": "would have","wouldn't": "would not",
                     "wouldn't've": "would not have","y'all": "you all", "y'all'd": "you all would",
                     "y'all'd've": "you all would have","y'all're": "you all are",
                     "y'all've": "you all have", "you'd": "you would","you'd've": "you would have",
                     "you'll": "you will","you'll've": "you will have", "you're": "you are",
```

```
                    "you've": "you have"}

# Regular expression for finding contractions
contractions_re=re.compile('(%s)' % '|'.join(contractions_dict.keys()))

# Function for expanding contractions
def expand_contractions(text,contractions_dict=contractions_dict):
  def replace(match):
    return contractions_dict[match.group(0)]
  return contractions_re.sub(replace, text)
```

```
fan_story_df['story_cleaned'] = fan_story_df['story']
fan_story_df['story_cleaned'] = fan_story_df['story_cleaned'].apply(lambda a: " ".join(a.split('\n\n'))) # handling "\n\n" characteres ir
fan_story_df['story_cleaned'] = fan_story_df['story_cleaned'].apply(lambda a: a.replace("\nend file\n","")) # replacing "\nend file\n" st
fan_story_df['story_cleaned'] = fan_story_df['story_cleaned'].apply(lambda a: expand_contractions(a)) # expanding contractions
fan_story_df['story_cleaned'] = fan_story_df['story_cleaned'].apply(lambda a: re.sub('\*+','',a)) # replacing * in the text
fan_story_df['story_cleaned'] = fan_story_df['story_cleaned'].apply(lambda a: re.sub(' +',' ',a)) # replacing multiple spaces with single
fan_story_df['story_cleaned'] = fan_story_df['story_cleaned'].apply(lambda a: re.sub('\n+','\n',a)) # replacing multiple new line charact
```

```
fan_story_df['story_cleaned'][10]
```

```
'Masks Title: My Mask Rating: T Summary: Connor and this thoughts about how he will never fit in with the rest of the group. One-shot D
isclaimer: I do not own anyone or anything from Zoom! AN: This is my first ever Zoom story and I hope that it comes out alright. Basica
lly it is a short one-shot about Connor. I always liked Connors character and it left me wondering how he really felt when he got back.
Nobody could ever be that happy after everything he went through. This is my try and writing Connor. I hope I do him justice. Have you
ever put on a front for those around you? Have you ever had to pretend to be something you are not? I have had to do that every single
day since my return from the dimensional time rift I had been trapped in for some thirty years. I have to act like I am the same teenag
```

```
tokenizer = GPT2Tokenizer.from_pretrained('gpt2') # loading the GPT2 English tokenizer from HuggingFace API

fan_story_df['tokenized_data'] = fan_story_df['story_cleaned']

#tokenizing the textual data using the tokenizer
fan_story_df['tokenized_data'] = fan_story_df['tokenized_data'].apply(lambda a:tokenizer.encode(a,max_length=1024,truncation=True))

fan_story_df
```

```
Downloading (…)olve/main/vocab.json: 100%  1.04M/1.04M [00:00<00:00, 26.0MB/s]
Downloading (…)olve/main/merges.txt: 100%  456k/456k [00:00<00:00, 14.5MB/s]
Downloading (…)lve/main/config.json: 100%  665/665 [00:00<00:00, 49.6kB/s]
```

	story	story_cleaned	tokenized_data
0	just a question\n\nWhat I thought when I first...	just a question What I thought when I first re...	[3137, 257, 1808, 1867, 314, 1807, 618, 314, 7...
1	pain\n\nIt may be sad but it can be better\n\n...	pain It may be sad but it can be better The ye...	[35436, 632, 743, 307, 6507, 475, 340, 460, 30...
2	Zombie Mayhem\n\nZombie Mayhem\n\nIn 2050, the...	Zombie Mayhem Zombie Mayhem In 2050, the milit...	[57, 9081, 35450, 14609, 35450, 554, 32215, 11...
3	Zero Wing 2: AYBASBTU Game Script\n\nZERO WING...	Zero Wing 2: AYBASBTU Game Script ZERO WING 2 ...	[28667, 13405, 362, 25, 317, 56, 33, 1921, 193...
4	Not prosaic\n\nDisclaimer, I do not claim to o...	Not prosaic Disclaimer, I do not claim to own ...	[3673, 10360, 18452, 3167, 17111, 11, 314, 466...
...	...	...	...
938	More\n\n**I've watched this movie 3 times this...	More I have watched this movie 3 times this we...	[5167, 314, 423, 7342, 428, 3807, 513, 1661, 4...
939	Change Comes From Within\n\n**As usual, I own ...	Change Comes From Within As usual, I own nothi...	[19400, 34606, 3574, 12511, 1081, 6678, 11, 31...

```
def generate_sequences(encoded_data, seq_length):
    '''
    Generates sequqneces of context and target encodings.

    Arguments:
        encoded_data(list) : of encoded data of the texts
        seq_length(list) : the length of the sequences to be created

    Returns:
        input_sequneces, target_sequences (tuple) : tuples of input and target
```

```
                                                encoding sequences
    '''
    seq_length = seq_length


    sequences = [encoded_data[i:i+seq_length] for i in
                 range(0, len(encoded_data), seq_length)]
    sequences = [s for s in sequences if len(s) == seq_length]
    input_sequences = [s[:-1] for s in sequences]
    target_sequences = [s[1:] for s in sequences]

    return input_sequences, target_sequences
```

```
def create_train_val_datasets(dataframe, sequence_length, batch_size):
    """
    Generates training and validation datasets.

    Arguments:
        dataframe (DataFrame): fan_fiction_story dataframe with encodings
        sequence_length(int): the length of the sequences to be created
        batch_size(int): the size of the batches

    Returns:
        train_dataset, validation_dataset(tuple) : tuple of train and validation datasets
    """
    train_input_sequences = []
    train_target_sequences = []

    val_input_sequences = []
    val_target_sequences = []

    # Training size is set to 90% of the data. Validation size is set to 10% of the data.
    train_size = int(fan_story_df.shape[0]*0.9)

    # encoding text data for train dataset
    for encoded_data in fan_story_df['tokenized_data'][:train_size]:
        inpt_seq, trgt_seq = generate_sequences(encoded_data, sequence_length,
                                                batch_size)
        train_input_sequences += inpt_seq
        train_target_sequences += trgt_seq

    # encoding text data for validation dataset
    for encoded_data in fan_story_df['tokenized_data'][train_size: ]:
        inpt_seq, trgt_seq = generate_sequences(encoded_data, sequence_length,
                                                batch_size)
        val_input_sequences += inpt_seq
        val_target_sequences += trgt_seq

    # creating Tensorflow Datasets of Train Data
    train_input_dataset = tf.data.Dataset.from_tensor_slices(train_input_sequences)
    train_target_dataset = tf.data.Dataset.from_tensor_slices(train_target_sequences)
    train_dataset = tf.data.Dataset.zip((train_input_dataset, train_target_dataset)).batch(batch_size)

    # creating Tensorflow Datasets of Validation Data
    val_input_dataset = tf.data.Dataset.from_tensor_slices(val_input_sequences)
    val_target_dataset = tf.data.Dataset.from_tensor_slices(val_target_sequences)
    val_dataset = tf.data.Dataset.zip((val_input_dataset, val_target_dataset)).batch(batch_size)

    return train_dataset, val_dataset
```

```
def train_gpt(train_data, learning_rate, epochs):
    """
    Initializes a GPT2 model by loading it from HuggingFace API. The model
    is trained on custom dataset. The trained model and its respective training
    history are returned back.

    Parameters:
        train_data : training data
        learning_rate(float) : learning rate of the optimizer
        epochs: number of epochs to train the model
    """
    # GPT2 model
    gpt_model = TFGPT2LMHeadModel.from_pretrained("gpt2")

    # Adam Optimizer
```

```
    optimizer = tf.keras.optimizers.Adam(
        learning_rate = learning_rate,
        weight_decay = 0.002,
    )

    # Loss function
    loss_metric = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)

    gpt_model.compile(optimizer, loss = [loss_metric,*[None] * gpt_model.config.n_layer], metrics=['accuracy'])

    # callbacks
    early_stopping = tf.keras.callbacks.EarlyStopping(monitor = 'train_loss', patience = 2, restore_best_weights=True)

    # training the model
    history = gpt_model.fit(train_data, epochs = epochs, callbacks = [early_stopping])

    return gpt_model, history
```

```
import itertools
learning_rate = [1e-5, 5e-5]
batch_size =  [8]
sequence_length = [64, 128, 256]
num_epochs= [10, 15]

all_hyper_parameters_lists = []
all_hyper_parameters_lists.append(learning_rate)
all_hyper_parameters_lists.append(batch_size)
all_hyper_parameters_lists.append(sequence_length)
all_hyper_parameters_lists.append(num_epochs)
all_hyper_parameters_lists
hyper_parameter_combinations = [p for p in itertools.product(*all_hyper_parameters_lists)]

# all combinations of hyper parameters
hyper_parameter_combinations
```

```
[(1e-05, 8, 64, 10),
 (1e-05, 8, 64, 15),
 (1e-05, 8, 128, 10),
 (1e-05, 8, 128, 15),
 (1e-05, 8, 256, 10),
 (1e-05, 8, 256, 15),
 (5e-05, 8, 64, 10),
 (5e-05, 8, 64, 15),
 (5e-05, 8, 128, 10),
 (5e-05, 8, 128, 15),
 (5e-05, 8, 256, 10),
 (5e-05, 8, 256, 15)]
```

```
all_models = []
all_histories = []
all_validation_loss = []
for combination in hyper_parameter_combinations:
    # training a gpt2 model for each combination of hyper parameters
    lr, batch_size, sequence_length, num_epochs = combination

    # creating training and validation datasets for the respective sequence_length and batch size
    train_dataset,val_dataset = create_train_val_datasets(fan_story_df, sequence_length, batch_size)

    # training the model with respective learning rate and epochs
    trained_model, model_history = train_gpt(train_dataset, lr, num_epochs)

    # finding the validation loss of the model by evaluating it on validation dataa
    validation_loss = trained_model.evaluate(val_dataset, batch_size=batch_size,verbose =1 )

    all_models.append(trained_model)
    all_histories.append(model_history)
    all_validation_loss.append(validation_loss)

print(all_models)
print(all_histories)
print(all_validation_loss)
```

```
333/333 [==============================] - 29s 86ms/step - loss: 3.2987 - accuracy: 0.3550
Epoch 6/10
333/333 [==============================] - ETA: 0s - loss: 3.2643 - accuracy: 0.3586WARNING:tensorflow:Early stopping conditioned on m
333/333 [==============================] - 29s 86ms/step - loss: 3.2643 - accuracy: 0.3586
Epoch 7/10
333/333 [==============================] - ETA: 0s - loss: 3.2333 - accuracy: 0.3624WARNING:tensorflow:Early stopping conditioned on m
333/333 [==============================] - 29s 86ms/step - loss: 3.2333 - accuracy: 0.3624
Epoch 8/10
333/333 [==============================] - ETA: 0s - loss: 3.2057 - accuracy: 0.3659WARNING:tensorflow:Early stopping conditioned on m
333/333 [==============================] - 29s 86ms/step - loss: 3.2057 - accuracy: 0.3659
Epoch 9/10
333/333 [==============================] - ETA: 0s - loss: 3.1789 - accuracy: 0.3678WARNING:tensorflow:Early stopping conditioned on m
333/333 [==============================] - 29s 86ms/step - loss: 3.1789 - accuracy: 0.3678
Epoch 10/10
333/333 [==============================] - ETA: 0s - loss: 3.1537 - accuracy: 0.3712WARNING:tensorflow:Early stopping conditioned on m
333/333 [==============================] - 29s 86ms/step - loss: 3.1537 - accuracy: 0.3712
35/35 [==============================] - 4s 40ms/step - loss: 3.3972 - accuracy: 0.3584
All model checkpoint layers were used when initializing TFGPT2LMHeadModel.

All the layers of TFGPT2LMHeadModel were initialized from the model checkpoint at gpt2.
If your task is similar to the task the model of the checkpoint was trained on, you can already use TFGPT2LMHeadModel for predictions
Epoch 1/15
333/333 [==============================] - ETA: 0s - loss: 3.5894 - accuracy: 0.3272WARNING:tensorflow:Early stopping conditioned on m
333/333 [==============================] - 68s 87ms/step - loss: 3.5894 - accuracy: 0.3272
Epoch 2/15
333/333 [==============================] - ETA: 0s - loss: 3.4506 - accuracy: 0.3396WARNING:tensorflow:Early stopping conditioned on m
333/333 [==============================] - 29s 87ms/step - loss: 3.4506 - accuracy: 0.3396
Epoch 3/15
333/333 [==============================] - ETA: 0s - loss: 3.3842 - accuracy: 0.3460WARNING:tensorflow:Early stopping conditioned on m
333/333 [==============================] - 29s 87ms/step - loss: 3.3842 - accuracy: 0.3460
Epoch 4/15
333/333 [==============================] - ETA: 0s - loss: 3.3370 - accuracy: 0.3506WARNING:tensorflow:Early stopping conditioned on m
333/333 [==============================] - 29s 87ms/step - loss: 3.3370 - accuracy: 0.3506
Epoch 5/15
333/333 [==============================] - ETA: 0s - loss: 3.2988 - accuracy: 0.3552WARNING:tensorflow:Early stopping conditioned on m
333/333 [==============================] - 29s 87ms/step - loss: 3.2988 - accuracy: 0.3552
Epoch 6/15
333/333 [==============================] - ETA: 0s - loss: 3.2649 - accuracy: 0.3589WARNING:tensorflow:Early stopping conditioned on m
333/333 [==============================] - 29s 86ms/step - loss: 3.2649 - accuracy: 0.3589
Epoch 7/15
333/333 [==============================] - ETA: 0s - loss: 3.2346 - accuracy: 0.3617WARNING:tensorflow:Early stopping conditioned on m
333/333 [==============================] - 29s 86ms/step - loss: 3.2346 - accuracy: 0.3617
Epoch 8/15
333/333 [==============================] - ETA: 0s - loss: 3.2067 - accuracy: 0.3648WARNING:tensorflow:Early stopping conditioned on m
333/333 [==============================] - 29s 87ms/step - loss: 3.2067 - accuracy: 0.3648
Epoch 9/15
333/333 [==============================] - ETA: 0s - loss: 3.1793 - accuracy: 0.3686WARNING:tensorflow:Early stopping conditioned on m
333/333 [==============================] - 29s 86ms/step - loss: 3.1793 - accuracy: 0.3686
Epoch 10/15
333/333 [==============================] - ETA: 0s - loss: 3.1537 - accuracy: 0.3711WARNING:tensorflow:Early stopping conditioned on m
333/333 [==============================] - 29s 87ms/step - loss: 3.1537 - accuracy: 0.3711
Epoch 11/15
333/333 [==============================] - ETA: 0s - loss: 3.1289 - accuracy: 0.3741WARNING:tensorflow:Early stopping conditioned on m
```

```
# finding the perplexity of all the models
all_perplexity = list(map(lambda a : tf.math.exp(a[0]), all_validation_loss))
```

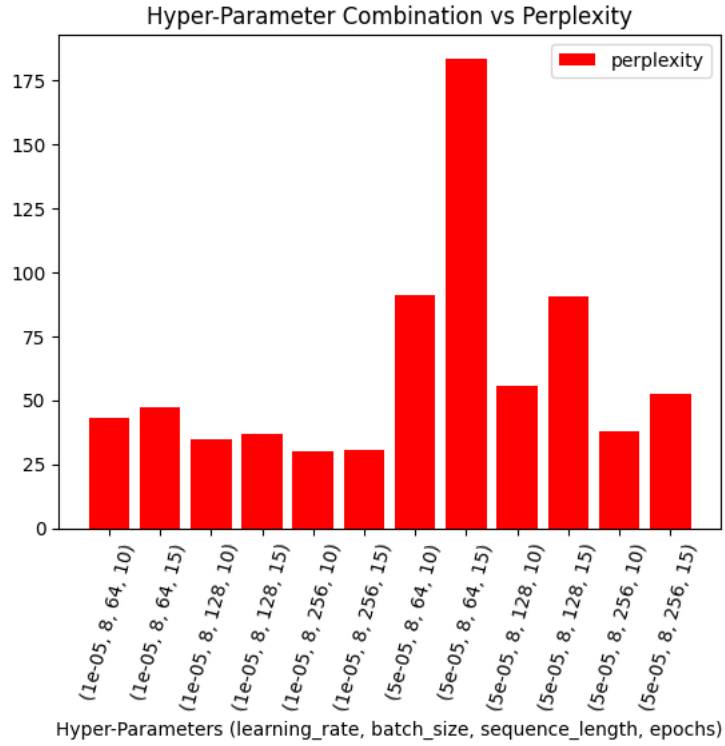
```
# the lowest perplexity
min_perplexity = min(all_perplexity)
```

```
loss_dataframes = pd.DataFrame([ a.numpy() for a in all_perplexity],columns = ['perplexity'])
loss_dataframes['hyper_parameter_combinations'] = hyper_parameter_combinations
loss_dataframes.to_csv('combination_to_perplexity.csv')
```

```
loss_dataframes
```

	perplexity	hyper_parameter_combinations
0	43.148060	(1e-05, 8, 64, 10)
1	46.983692	(1e-05, 8, 64, 15)
2	34.738026	(1e-05, 8, 128, 10)
3	36.543987	(1e-05, 8, 128, 15)
4	29.879160	(1e-05, 8, 256, 10)

```
# plotting the graph of the hyper parameter combination to the perplexity of the models
from matplotlib import pyplot as plt
plt.bar([str(a) for a in loss_dataframes['hyper_parameter_combinations']], loss_dataframes['perplexity'], color = 'red', label='perplexi
plt.xticks(rotation = 75)
plt.xlabel('Hyper-Parameters (learning_rate, batch_size, sequence_length, epochs)')
plt.title('Hyper-Parameter Combination vs Perplexity')
plt.legend()
plt.show()
```




```
print(min_perplexity)
```

```
tf.Tensor(29.87916, shape=(), dtype=float32)
```

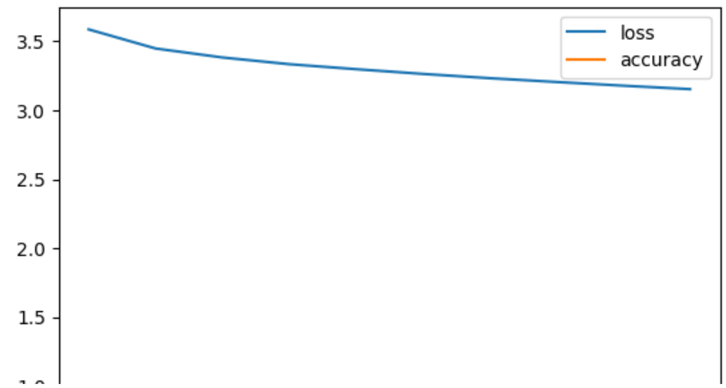
```
best_combination = hyper_parameter_combinations[all_perplexity.index(min_perplexity)]
best_combination
```

```
(1e-05, 8, 256, 10)
```

```
# storing the training history of the model which gave the lowest perplexity
best_model_training_history = all_histories[all_perplexity.index(min_perplexity)]
history_df = best_model_training_history.history
history_df = pd.DataFrame(history_df)
history_df.to_csv('best_model_training_history.csv')
```

```
history_df.plot()
```

```
<Axes: >
```




```
# selecting the model which gave the lowest perplexity
best_model = all_models[all_perplexity.index(min_perplexity)]
```

```
best_model.save('best_gpt2_model')
```

```
WARNING:absl:Found untraced functions such as wte_layer_call_fn, wte_layer_call_and_return_conditional_losses, dropout_333_layer_call_fr
```

```
def story_generation(prompt, temp):
    """
    Generates text using the trained GPT2 model.

    Arguments:
        prompt(str): the initial seed text for the generation
        temp(float): to determine the randomness of the generation
    Return:
        generated_text(str): story generated by the GPT2 model
    """
    input_ids = tokenizer.encode(prompt, return_tensors='tf')
    output = best_model.generate(input_ids = input_ids, max_length = 100,num_beams = 5, no_repeat_ngram_size=2, do_sample = True, early_
    generated_text = tokenizer.decode(output[0], skip_special_tokens=True)
    return generated_text

```

```
# checking with different temperature parameter value for generation
temperature_values = [ 1.0, 1.5, 2.0, 3.0]
generated_texts = [story_generation('pain\n It may be ', temp) for temp in temperature_values]
print(generated_texts)
```

```
The attention mask and the pad token id were not set. As a consequence, you may observe unexpected behavior. Please pass your input's `a
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
The attention mask and the pad token id were not set. As a consequence, you may observe unexpected behavior. Please pass your input's `a
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
The attention mask and the pad token id were not set. As a consequence, you may observe unexpected behavior. Please pass your input's `a
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
The attention mask and the pad token id were not set. As a consequence, you may observe unexpected behavior. Please pass your input's `a
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
['pain\n It may be \nthe most beautiful day of my life, but I am afraid it is not going to be for you. I do not know what I will do if I
```

```
text_gen_df = pd.DataFrame(temperature_values, columns =['temperature'])
text_gen_df['generated_text'] = generated_texts
text_gen_df.to_csv('generated_text_temperature_comparison.csv')
text_gen_df
```

1 to 4 of 4 entries Filter

index	temperature	generated_text
0	1.0	pain It may be the most beautiful day of my life, but I am afraid it is not going to be for you. I do not know what I will do if I see you again. It is been so long since I last saw you, and I have not seen you in a long time. You are the only one I know that I care about. But I cannot let you go, no matter how hard I try. What I want is you to see me again,
1	1.5	pain It may be icky but ive never felt ick in my life. I wish I could change that. It is the only thing I can do right now. I do not think that I will ever have the chance to make a difference, even if I have to go back to my old school and do something about it. This is what happened to me when I was about to give up hope of a brighter future and live in a world full of peace and love
2	2.0	pain It may be the last day of my life, but then again, I had expected it. I was not here to tell them my real name, not when it would make sense to me; they had decided not to. This time though, all that was important had to be their choice. No one would have

```
# loging in to HuggingFace profile
from huggingface_hub import notebook_login
notebook_login()
```

```
Token is valid.
Your token has been saved in your configured git credential helpers (store).
Your token has been saved to /root/.cache/huggingface/token
Login successful
```

```
# saving the model to the HuggingFace hub
best_model.push_to_hub('aakash-mahesha/fan-story-generation-gpt2-mini')
tokenizer.push_to_hub('aakash-mahesha/fan-story-generation-gpt2-mini')
```

```
Upload 1 LFS files: 100%                                    1/1 [00:47<00:00, 47.63s/it]

tf_model.h5: 100%                                           498M/498M [00:47<00:00, 7.87MB/s]

CommitInfo(commit_url='https://huggingface.co/aakash-mahesha/fan-story-generation-gpt2-
mini/commit/f667fbd07d1ebc1813c705b3f6fc312542d194bf', commit_message='Upload tokenizer', commit_description='',
oid='f667fbd07d1ebc1813c705b3f6fc312542d194bf', pr_url=None, pr_revision=None, pr_num=None)
```

```
1
```