Assignment 4

Megha 2021337

Google drive link : [Fine Tunned GPT2 Model]

Text Preprocessing

```
import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
```

Removing html tags

```
def html_remove(text):
   word=BeautifulSoup(text,"html.parser")
   return word.get_text()
```

Removing accented chars

```
def accented(text):
   text = unicodedata.normalize('NFKD', text).encode('ascii',
'ignore').decode('utf-8', 'ignore')
   return text
```

Preprocessing

```
def preprocess(text):
   text = html_remove(text)
   text = accented(text)
   text = re.sub(r'[^a-zA-Z0-9\s]', '', text)
   text = text.lower()
   tokens = word_tokenize(text)
   stop_words = set(stopwords.words('english'))
   tokens = [word for word in tokens if word not in stop_words]
   tokens = [word for word in tokens if word not in string.punctuation]
   lemmatizer = WordNetLemmatizer()
   lemmatized_tokens = [lemmatizer.lemmatize(word) for word in tokens]

   return ' '.join(lemmatized_tokens)

df['Summary'] = df['Summary'].apply(preprocess)
df['Text'] = df['Text'].apply(preprocess)
```

```
df.to_csv('Preprocessed_Review1.csv', index=False)
```

GPT2 Model Fine Tunning

Importing important libraries

```
import pandas as pd
import re
from transformers import GPT2LMHeadModel, GPT2Tokenizer
from transformers import TextDataset, DataCollatorForLanguageModeling
from transformers import Trainer, TrainingArguments
```

Creating Custom dataset :

```
class Modified_Set(Dataset):
   def __init__(self, texts, summaries, tokenizer, max_length=512):
       self.texts = texts
       self.summaries = summaries
       self.tokenizer = tokenizer
       self.max_length = max_length

   def __len__(self):
       return len(self.texts)

   def __getitem__(self, idx):
       text = self.texts[idx]
       summary = self.summaries[idx]
       combined_text = (text, '[SEP]', summary)
       encoding = self.tokenizer(combined_text,
max_length=self.max_length, padding="max_length", truncation=True,
return_tensors="pt")
       input_ids = encoding.input_ids.squeeze()
       attention_mask = encoding.attention_mask.squeeze()
       labels = input_ids.clone()

       return {
           'input_ids': input_ids,
           'attention_mask': attention_mask,
           'labels': labels
       }
```

Saved (Text,Summary) as tuples.

After training the model, we tested it on our test dataset.The ROUGE score is :

{'rouge-1': {'r': 0.1171142857142857, 'p': 0.15500000000000003, 'f': 0.12349206228711104}, 'rouge-2': {'r': 0.013, 'p': 0.024333333333333332, 'f': 0.01569523795928073}, 'rouge-l': {'r': 0.1171142857142857, 'p': 0.15500000000000003, 'f': 0.12349206228711104}}

Parameters :

```
batch_size = 2
model = GPT2LMHeadModel.from_pretrained('gpt2')
epochs = 3
model.train()
```

Model is trained on 1000 dataset.Then then tested on 250 test datasets.