NLP LAB 1

Ankit Rathwa - 202001190

code:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
import tarfile
import os

# Path to the .tgz file in Google Drive
path_to_tgz = '/content/drive/MyDrive/nlp/cnn_stories.tgz'

# Path to save the extracted files
extracted_folder_path = '/content/Extracted/'

# Create the extracted folder if it doesn't exist
os.makedirs(extracted_folder_path, exist_ok=True)

# Extract the .tgz file
with tarfile.open(path_to_tgz, 'r:gz') as tar:
    tar.extractall(extracted_folder_path)
```

```
mport tarfile
import os

# Path to the .tgz file in Google Drive
path_to_tgz = '/content/drive/MyDrive/nlp/dailymail_stories.tgz'

# Path to save the extracted files
extracted_folder_path = '/content/Extracted/'

# Create the extracted folder if it doesn't exist
```

```
os.makedirs(extracted_folder_path, exist_ok=True)

# Extract the .tgz file
with tarfile.open(path_to_tgz, 'r:gz') as tar:
    tar.extractall(extracted_folder_path)
```

```
# Path to the extracted folder
extracted_folder_path = '/content/Extracted/cnn/stories/'

# List files in the extracted folder
extracted_files = os.listdir(extracted_folder_path)

# Print the list of extracted files
print("Extracted Files:")
for file_name in extracted_files:
    print(file_name)
```

```
import os

# Path to the extracted folder
extracted_folder_path = '/content/Extracted/dailymail/stories/'

# List files in the extracted folder
extracted_files = os.listdir(extracted_folder_path)

# Print the list of extracted files
print("Extracted Files:")
for file_name in extracted_files:
    print(file_name)
```

```
import os
import spacy
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from multiprocessing import Pool
# Load spaCy's English model
```

```
nlp = spacy.load("en core web sm")
def extract noun phrases(text):
   doc = nlp(text)
   noun phrases = [chunk.text for chunk in doc.noun chunks]
   return ' '.join(noun phrases)
def process document(file path):
   with open(file path, 'r') as file:
       text = file.read()
       noun phrases text = extract noun phrases(text)
       return noun phrases text
def compute cosine similarity(noun phrases list):
   tfidf vectorizer = TfidfVectorizer()
   tfidf matrix = tfidf vectorizer.fit transform(noun phrases list)
   cosine similarities = cosine similarity(tfidf matrix)
   return cosine similarities
documents dir1 = '/content/Extracted/cnn/stories'
file names1 = [os.path.join(documents dir1, file name) for file name in
os.listdir(documents dir1) if file name.endswith('.story')][:1000]
with Pool() as pool:
   noun phrases list1 = pool.map(process document, file names1)
orint("Extracted Noun Phrases:")
```

```
for noun_phrases_text in noun_phrases_list1:
    print(noun_phrases_text)
```

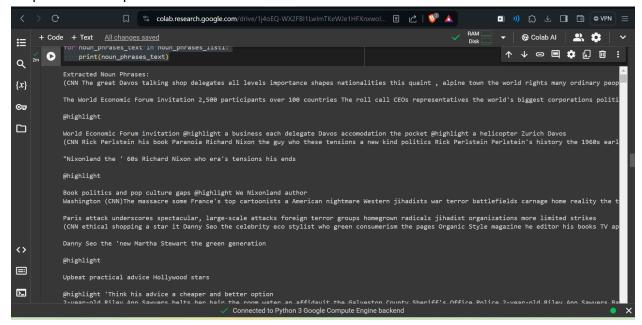
```
# Compute cosine similarities between all pairs of noun phrases
cosine_similarities1 = compute_cosine_similarity(noun_phrases_list1)

# Output cosine similarities
print("\nCosine Similarities:")
for i in range(len(cosine_similarities1)):
    for j in range(i+1, len(cosine_similarities1[i])):
        similarity = cosine_similarities1[i, j]
        print(f"Cosine similarity between noun phrases {i} and {j}:
{similarity}")
```

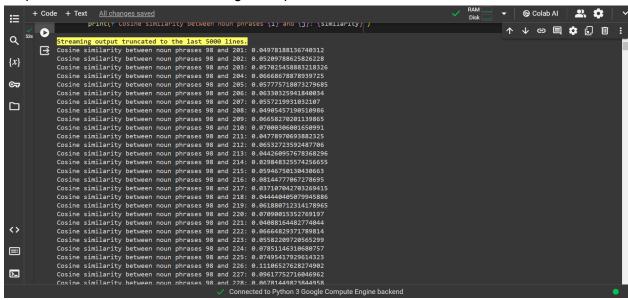
```
import os
import spacy
from sklearn.feature extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine similarity
from multiprocessing import Pool
nlp = spacy.load("en core web sm")
def extract noun phrases(text):
   doc = nlp(text)
   noun phrases = [chunk.text for chunk in doc.noun chunks]
   return ' '.join(noun phrases)
def process document(file path):
   with open(file path, 'r') as file:
        text = file.read()
       noun phrases text = extract noun phrases(text)
        return noun phrases text
def compute cosine similarity(noun phrases list):
```

```
tfidf vectorizer = TfidfVectorizer()
   tfidf matrix = tfidf vectorizer.fit transform(noun phrases list)
    cosine similarities = cosine similarity(tfidf matrix)
   return cosine similarities
documents dir2 = '/content/Extracted/dailymail/stories'
file names2 = [os.path.join(documents dir2, file name) for file name in
os.listdir(documents dir2) if file name.endswith('.story')][:1000]
with Pool() as pool:
   noun phrases list2 = pool.map(process document, file names2)
print("Extracted Noun Phrases:")
for noun phrases text in noun phrases list2:
   print(noun phrases text)
 Compute cosine similarities between all pairs of noun phrases
cosine similarities2 = compute cosine similarity(noun phrases list2)
print("\nCosine Similarities:")
for i in range(len(cosine similarities2)):
   for j in range(i+1, len(cosine similarities2[i])):
        similarity = cosine similarities2[i, j]
       print(f"Cosine similarity between noun phrases {i} and {j}:
{similarity}")
```

Outputs for noun-phraases in cnn stories :



Outputs for cosine similarities among noun-phraases in cnn stories :



Outputs for noun-phraases in dailymail stories :

```
Extracted Noun Phrases:
Mark Duell Matt Chorley Mailonline Political Editor :
08:22 EST 20 March 2013
| 13:36 EST 20 March 2013
Millions Britain's hard-pressed taxpavers themselves the coalition its flagship tax cut Chancellor George Osborne the amount money that workers the
Rising trend This personal allowance graph figures the Institute Fiscal Studies website a person it It part a bid the burden squeezed family budget:
decision VAT 17.5 per cent 20 per cent the coalition the Tories Lib Dems the personal allowance this a \,
longer term policy objective the income tax threshold April the average worker The Government the policy other tax cuts the Tory policy Inheritance
personal allowance inflation the Exchequer £500million free market the Adam
.
Smith Institute Mr Osborne the allowance which full-time minimum wage workers any
£12,875 personal allowance the take-home pay all those up to 1,297,000 people the tax system The institute's policy director Sam Bowman It a national scandal we the people the
standard living Payments April the average taxpayer £705 less tax the 2010 general election Meanwhile hopes the Chancellor the strain savers recent cash four years ultra-low interest rates living costs wages Almost a dozen tax loopholes the government those who payments the exchequer the Chance
 website Moneyfacts the impact inflation savings average interest tax 20 per cent the spending power Former Government policy adviser Ros
Altmann who director-general Saga the rules tax-free Isas people all their allowance cash Isas they the current system only half the £11,280 allowance remainder stocks shares Dr Altmann Older people who their savings the stock
market younger generations who a house deposit It the
dramatic drop savings interest rates the Government savers them cash other assets Mr Osborne that corporation tax 20 per cent He the headline rate 28 to 24 per cent it cent 21 per cent April he corporation tax 20 per cent
```

Outputs for cosine similarities among noun-phraases in dailmail stories :

