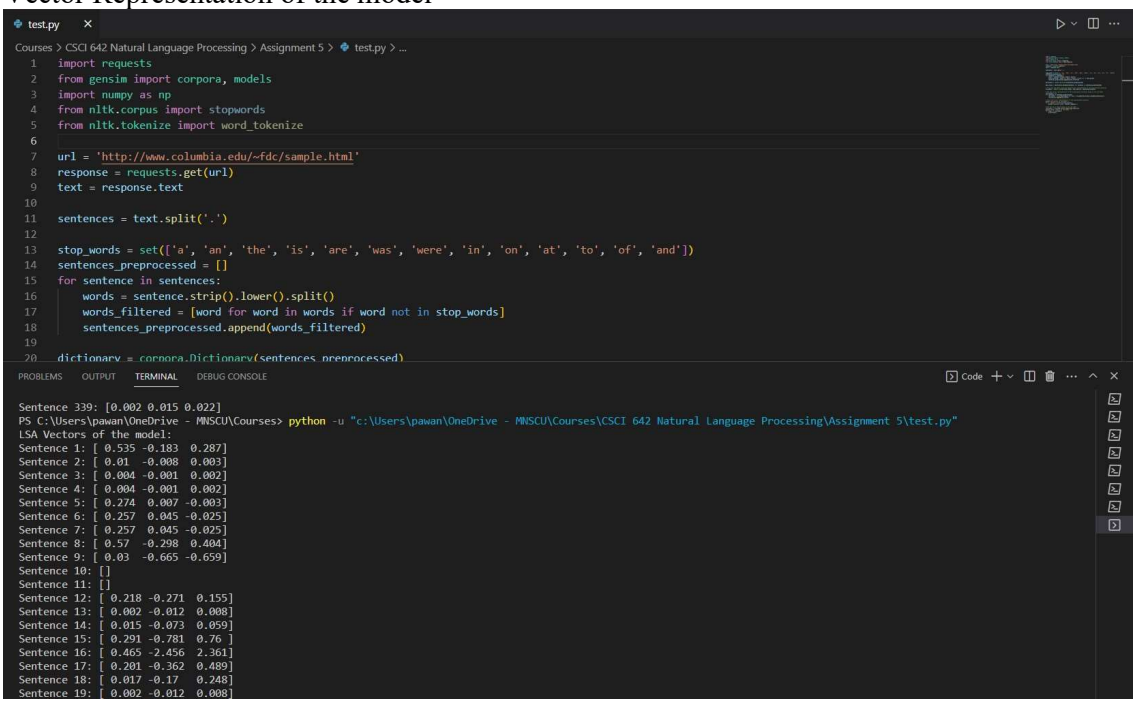
Assignment #5

CSCI 642 Natural Language Processing

Yesho Reddipalli
15896337

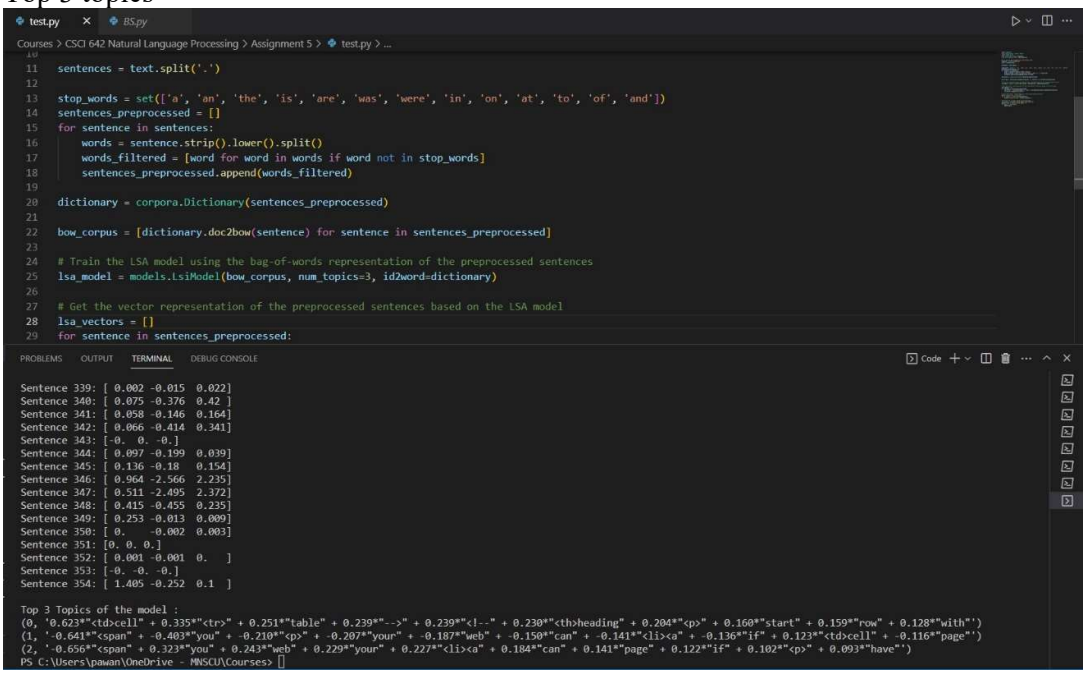
Vector Representation of the model



```
import requests
from gensim import corpora, models
import numpy as np
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

url = 'http://www.columbia.edu/~fdc/sample.html'
response = requests.get(url)
text = response.text

sentences = text.split('.')

stop_words = set(['a', 'an', 'the', 'is', 'are', 'was', 'were', 'in', 'on', 'at', 'to', 'of', 'and'])
sentences_preprocessed = []
for sentence in sentences:
    words = sentence.strip().lower().split()
    words_filtered = [word for word in words if word not in stop_words]
    sentences_preprocessed.append(words_filtered)

dictionary = corpora.Dictionary(sentences_preprocessed)
```

```
Sentence 339: [0.002 0.015 0.022]
PS C:\Users\pawan\OneDrive - MNSCU\Courses> python -u "c:\Users\pawan\OneDrive - MNSCU\Courses\CSCI 642 Natural Language Processing\Assignment 5\test.py"
LSA Vectors of the model:
Sentence 1: [ 0.535 -0.183  0.287]
Sentence 2: [ 0.01  -0.008  0.003]
Sentence 3: [ 0.004 -0.001  0.002]
Sentence 4: [ 0.004 -0.001  0.002]
Sentence 5: [ 0.274  0.007 -0.003]
Sentence 6: [ 0.257  0.045 -0.025]
Sentence 7: [ 0.257  0.045 -0.025]
Sentence 8: [ 0.57  -0.298  0.404]
Sentence 9: [ 0.03  -0.665 -0.659]
Sentence 10: []
Sentence 11: []
Sentence 12: [ 0.218 -0.271  0.155]
Sentence 13: [ 0.002 -0.012  0.008]
Sentence 14: [ 0.015 -0.073  0.059]
Sentence 15: [ 0.291 -0.781  0.76 ]
Sentence 16: [ 0.465 -2.456  2.361]
Sentence 17: [ 0.201 -0.362  0.489]
Sentence 18: [ 0.017 -0.17   0.248]
Sentence 19: [ 0.002 -0.012  0.008]
```

Top 3 topics



```
sentences = text.split('.')

stop_words = set(['a', 'an', 'the', 'is', 'are', 'was', 'were', 'in', 'on', 'at', 'to', 'of', 'and'])
sentences_preprocessed = []
for sentence in sentences:
    words = sentence.strip().lower().split()
    words_filtered = [word for word in words if word not in stop_words]
    sentences_preprocessed.append(words_filtered)

dictionary = corpora.Dictionary(sentences_preprocessed)

bow_corpus = [dictionary.doc2bow(sentence) for sentence in sentences_preprocessed]

# Train the LSA model using the bag-of-words representation of the preprocessed sentences
lsa_model = models.LsiModel(bow_corpus, num_topics=3, id2word=dictionary)

# Get the vector representation of the preprocessed sentences based on the LSA model
lsa_vectors = []
for sentence in sentences_preprocessed:
```

```
Sentence 339: [ 0.002 -0.015  0.022]
Sentence 340: [ 0.075 -0.376  0.42 ]
Sentence 341: [ 0.058 -0.146  0.164]
Sentence 342: [ 0.066 -0.414  0.341]
Sentence 343: [-0.  0. -0.]
Sentence 344: [ 0.097 -0.199  0.039]
Sentence 345: [ 0.136 -0.18   0.154]
Sentence 346: [ 0.964 -2.566  2.235]
Sentence 347: [ 0.511 -2.495  2.372]
Sentence 348: [ 0.415 -0.455  0.235]
Sentence 349: [ 0.253 -0.013  0.009]
Sentence 350: [ 0.    -0.002  0.003]
Sentence 351: [0. 0. 0.]
Sentence 352: [ 0.001 -0.001  0.   ]
Sentence 353: [-0. -0. -0.]
Sentence 354: [ 1.405 -0.252  0.1  ]

Top 3 Topics of the model :
(0, '0.623*"<td>cell" + 0.335*"<tr>" + 0.251*"table" + 0.239*"-->" + 0.239*"<!--" + 0.230*"<th>heading" + 0.204*"<p>" + 0.160*"start" + 0.159*"row" + 0.128*"with"')
(1, '-0.641*"<span" + -0.403*"you" + -0.210*"<p>" + -0.207*"your" + -0.187*"web" + -0.150*"can" + -0.141*"<li><a" + -0.136*"if" + 0.123*"<td>cell" + -0.116*"page"')
(2, '-0.656*"<span" + 0.323*"you" + 0.243*"web" + 0.229*"your" + 0.227*"<li><a" + 0.184*"can" + 0.141*"page" + 0.122*"if" + 0.102*"<p>" + 0.093*"have"')
PS C:\Users\pawan\OneDrive - MNSCU\Courses>
```

Code:

```
import requests
from gensim import corpora, models
import numpy as np
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

url = 'http://www.columbia.edu/~fdc/sample.html'
response = requests.get(url)
text = response.text

sentences = text.split('.')

stop_words = set(['a', 'an', 'the', 'is', 'are', 'was', 'were', 'in', 'on',
'at', 'to', 'of', 'and'])
sentences_preprocessed = []
for sentence in sentences:
    words = sentence.strip().lower().split()
    words_filtered = [word for word in words if word not in stop_words]
    sentences_preprocessed.append(words_filtered)

dictionary = corpora.Dictionary(sentences_preprocessed)

bow_corpus = [dictionary.doc2bow(sentence) for sentence in
sentences_preprocessed]

# Train the LSA model using the bag-of-words representation of the
preprocessed sentences
lsa_model = models.LsiModel(bow_corpus, num_topics=3, id2word=dictionary)

# Get the vector representation of the preprocessed sentences based on the LSA
model
lsa_vectors = []
for sentence in sentences_preprocessed:
    lsa_vector = np.array([tup[1] for tup in
lsa_model[dictionary.doc2bow(sentence)]])
    lsa_vectors.append(lsa_vector)

# Print the vector representation of the preprocessed sentences
print('LSA Vectors of the model:')
for i, vector in enumerate(lsa_vectors):
    print(f'Sentence {i+1}: {vector.round(3)}')

# Get the top 3 topics based on the LSA model
lsa_topics = lsa_model.print_topics(num_topics=3)
print('\nTop 3 Topics of the model :')
for topic in lsa_topics:
    print(topic)
```