NAME : RETHINAGIRI G

ROLL NO : 225229130

COURSE TITLE : NATURAL LANGUAGE PROCESSING LAB

LAB_04 Computing Cocument Similarity using Doc2Vec Model

EXERCISE - I

1. Import dependencies

In [1]:

```
import gensim
```

In [2]:

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

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[2]:

```
True
```

In [3]:

```
from gensim.models.doc2vec import Doc2Vec,TaggedDocument
from nltk.tokenize import word_tokenize
from sklearn import utils
```

2.Create Dataset

In [4]:

```
data=[" I love machine learning. Its awesome.",
     " I love coding in python",
     "I love building in chatbots",
     "they chat amazingly well"]
```

3. Create Tagged Document

In [5]:

```

tagged_data=[TaggedDocument(words=word_tokenize(d.lower()),
                        tags=[str(i)]) for i,d in enumerate(data)]
```

4. Train Model

In [6]:

```
vec_size=20
alpha=0.025
```

In [7]:

```
#create model

model=Doc2Vec(vector_size=vec_size,
             alpha=alpha,
             min_count=1,
              dm=1)
```

In [8]:

```
# Build vocabulary

model.build_vocab(tagged_data)
```

In [9]:

```
# Shuffle data

tagged_data=utils.shuffle(tagged_data)
```

In [10]:

```
# train Doc2Vec model

model.train(tagged_data,total_examples=model.corpus_count,epochs=30)
model.save("d2v.model")
print(" Model Saved")
```

```
 Model Saved
```

5. Find Similar documents for the given document

In [11]:

```
from gensim.models.doc2vec import Doc2Vec as D2V

model=D2V.load("d2v.model")
```

In [12]:

```

# To find the vector of a document which is not in training data

test_data=word_tokenize("I love chatbots".lower())

v1=model.infer_vector(test_data)
print(" V1_infer",v1)
```

```
 V1_infer [-0.01744856  0.0139768   0.00102463 -0.01409235  0.01133437  0.01650609
 -0.01653387 -0.01284875 -0.00099804 -0.018149   -0.01496079  0.0212263
  0.02131033  0.01589404  0.00186414 -0.02312037  0.00964129  0.01834673
 -0.00726666  0.00302069]
```

In [13]:

```
# To find most similar doc using tags

similar_doc=model.docvecs.most_similar('1')
print(similar_doc)
```

```
[('2', 0.32375025749206543), ('0', 0.2837848961353302), ('3', 0.21745406091213226)]

C:\Users\user\AppData\Local\Temp\ipykernel_13512\2042040100.py:3: DeprecationWarning: Call to deprecated `docvecs` (The `do
cvecs` property has been renamed `dv`.).
  similar_doc=model.docvecs.most_similar('1')
```

In [14]:

```
# To find vector of doc i traing data using tags or in other words,
 # printing the vector of documents at index 1 in training data

print(model.docvecs['1'])
```

```
[-0.0192466   0.01301249 -0.02877155  0.01312144  0.02962105 -0.04105275
 -0.04217485 -0.0503067   0.02452813 -0.04621011  0.02930218  0.03441546
 -0.03326039 -0.02350739 -0.00663665  0.00828503 -0.00737227 -0.04289516
 -0.01865279  0.00888925]

C:\Users\user\AppData\Local\Temp\ipykernel_13512\2056527876.py:4: DeprecationWarning: Call to deprecated `docvecs` (The `do
cvecs` property has been renamed `dv`.).
  print(model.docvecs['1'])
```

EXERCISE -II

Q1. Train the following documents using Doc2Vec model

In [15]:

```
docs=["the house had a tiny little mouse",
     "the  cat saw the mouse",
     "the mouse ran away from the house",
     "the cat finally ate the mouse",
     "the end of the mouse story"]
```

In [16]:

```
tagged_data=[TaggedDocument(words=word_tokenize(d.lower()),
                        tags=[str(i)]) for i,d in enumerate(docs)]
```

In [17]:

```
vec_size=20
alpha=0.025
```

In [18]:

```
#Create model

Doc2Vec(vector_size=vec_size,
             alpha=alpha,
             min_count=1,
              dm=1)
```

In [19]:

```
# Build Vocabulary

model.build_vocab(tagged_data)
```

In [20]:

```
tagged_data=utils.shuffle(tagged_data)
```

In [21]:

```
# train Doc2Vec Model

model.train(tagged_data,total_examples=model.corpus_count,epochs=30)
model.save("d2v2.model")
print("Model Saved")
```

```
Model Saved
```

Q2. Find the most simiar Two documents for the query document "cat stayed in the house".

In [22]:

```
from gensim.models.doc2vec import Doc2Vec as D2V

model=D2V.load("d2v2.model")
```

In [23]:

```
test_data2=word_tokenize("cat stayed in the house".lower())

v2=model.infer_vector(test_data2)
print(" V2_infer",v2)
```

```
 V2_infer [ 0.01805817 -0.00423691 -0.0194633  -0.00023666  0.02106751  0.01411113
  0.00166233 -0.01006699 -0.01537784  0.00598021 -0.01467327  0.02139555
 -0.0182403   0.01891387  0.00870299  0.02040237 -0.00996894 -0.02206526
 -0.01960368  0.00322819]
```

In [24]:

```
similar_doc2=model.docvecs.most_similar('2')
print(similar_doc2)
```

```
[('3', 0.34877684712409973), ('1', 0.336925208568573), ('4', 0.23375152051448822), ('0', -0.09642516076564789)]

C:\Users\user\AppData\Local\Temp\ipykernel_13512\291976564.py:1: DeprecationWarning: Call to deprecated `docvecs` (The `doc
vecs` property has been renamed `dv`.).
  similar_doc2=model.docvecs.most_similar('2')
```

In [25]:

```
print(model.docvecs['2'])
```

```
[-0.01075136 -0.03610123  0.02015789 -0.04287454  0.01476685 -0.02390859
  0.00274999 -0.011177    0.02697756 -0.0409523  -0.01088565  0.00035094
 -0.03539685 -0.03443976 -0.0106908   0.04471822 -0.00621852  0.01774278
 -0.02981413  0.04480183]
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_13512\3897255831.py:1: DeprecationWarning: Call to deprecated `docvecs` (The `do
cvecs` property has been renamed `dv`.).
  print(model.docvecs['2'])
```