Natural Language Processing Lab

Name : Ezhilarasan C

Roll No:225229151

Lab 4: Document Similarity using Doc2vec

1.Import dependencies

In [1]:
```
import gensim
```

In [4]:
```
from gensim.models.doc2vec import Doc2Vec, TaggedDocument
from nltk.tokenize import word_tokenize
from sklearn import utils
```

2. Create dataset

In [5]:
```
data=["I love machine learning. Its awesome.",
"I love coding in python",
"I love building chatbots",
"they chat amagingly well"]
```

3.Create TaggedDocument

In [6]:
```
tagged_data=[TaggedDocument(words=word_tokenize(d.lower()),tags=[str(i)]) for i,d in enumerate(data)]
```

4.Train Model

In [7]:
```
#model parameters
vec_size=20
alpha=0.025

#create model
model=Doc2Vec(vector_size=vec_size,
              alpha=alpha,
              min_alpha=0.00025,
              min_count=1,
              dm=1)

#build vocabulary
model.build_vocab(tagged_data)

#shuffle data
tagged_data=utils.shuffle(tagged_data)

#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 [8]:
```
from gensim.models.doc2vec import Doc2Vec

model=Doc2Vec.load("d2v.model")

#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)

#to find most similar doc using tags
similar_doc=model.dv.most_similar('1')
print(similar_doc)

#to find vector of doc in training data using tags or
#In other words,printing the vector of document at index 1 in training data

print(model.dv["1"])
```

```
v1_infer [-0.00628174  0.01117978  0.0078928   0.02358329  0.0240783   0.0241926
  0.01790185  0.01491485 -0.01218563  0.00336643 -0.00813564  0.00505789
  0.00430344  0.00052596  0.00542781 -0.01847247 -0.01744725 -0.02444072
 -0.02125897  0.02318843]
[('2', 0.3250403106212616), ('0', 0.2968985438346863), ('3', 0.22365665435791016)]
[-0.01990577  0.01304763 -0.02958454  0.01350018  0.03025302 -0.0416232
 -0.0429488  -0.05067174  0.0239734  -0.04663915  0.02936365  0.03494433
 -0.0342547  -0.02422205 -0.00711292  0.00778666 -0.00734602 -0.04273351
 -0.01970573  0.00895928]
```

Exercise-2

Question-1. Train the following documents using Doc2Vec model

In [9]:
```
docs = ["the house had a tiny little mouse",
        "the mouse ran away from the house",
        "the cat finally ate the mouse",
        "the end of the mouse story"]
```

In [10]:
```
tagged_docs=[TaggedDocument(words=word_tokenize(d.lower()),tags=[str(i)]) for i,d  in enumerate(data)]
```

In [11]:
```
#model parameters
vec_size=20
alpha=0.025

#create model
model=Doc2Vec(vector_size=vec_size,
              alpha=alpha,
              min_alpha=0.00025,
              min_count=1,
              dm=1)

#build vocabulary
model.build_vocab(tagged_docs)

#shuffle data
tagged_docs=utils.shuffle(tagged_docs)

#train Doc2Vec model
model.train(tagged_docs,
            total_examples=model.corpus_count,
            epochs=30)

model.save("d2v.model")
print("Model Saved")
```

```
Model Saved
```

In [12]:

```
from gensim.models.doc2vec import Doc2Vec

model=Doc2Vec.load("d2v.model")

#to find the vector of a document which is not in training data
test_data=word_tokenize("cat stayed in the house".lower())
v1=model.infer_vector(test_data)
print("v1_infer",v1)

#to find most similar doc using tags
similar_doc=model.dv.most_similar('2')
print(similar_doc)

#to find vector of doc in training data using tags
print(model.dv["2"])
```

```
v1_infer [ 0.01325687 -0.01836687  0.00975062  0.02452783  0.02382731 -0.01649335
 -0.00994847  0.01145699  0.01769195 -0.01408347  0.01660025 -0.01438435
 -0.01571622 -0.01222706 -0.00617121  0.01248548 -0.0068709  -0.00465107
  0.01529026 -0.00246679]
[('3', 0.3407185971736908), ('1', 0.33049604296684265), ('0', -0.1114959642291069)]
[-0.0107779  -0.03629022  0.02062683 -0.0430157   0.01423227 -0.02366377
  0.00278364 -0.0108782   0.02691032 -0.04073556 -0.01055875 -0.00018295
 -0.03411321 -0.03360157 -0.01021269  0.04415133 -0.00633329  0.01769288
 -0.02959119  0.04442726]
```