Exercise-1 1 Import dependencies Rimport gensim from gensin. anolek-docevac import Docevec, Tagged Document from Alt. totoria import word-topenia. from stleam import calils. data = [" I love machine learning. Its aubsone." "I love cooling in python", " I love building chartbots" . " they chart amagingly well" import in 1tk. NI bk. down load ( pun kt') tagged - Dorta = [Tagged Document ( woods tokenize) Logs = [str(?)])...for i'd in enunerate (data) Vec-Size = 20 altha = 0.005 model = Docevec (vector - size = vec - size, alpha soplaa minialpha = 0.000 25/ min\_count=1 , dm = 1)

model. build rocab (fagged data)

# Natural Language Processing Lab Lab4. Computing Document Similarity using Doc2Vec Model

#### **EXERCISE-1**

# 1. Import dependencies

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

#### 2. Create dataset

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

# 3. Create TaggedDocument

# 4. Train Model

```
# 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")
```

### 5. Find Similar documents for the given document

DERIOFDATA SOUNCE (BISHOP (FEER COULEGE) TRICKY

```
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
```

total = exemples = model. Coopers - Count, cpochs = 30) madel. sale ("der. Madel") point (1 platel Saved!) from gention. models decerved import mudel = Docovec. local ("He v. moole") test-data = word-tokonizo(") love chaket" lower()) 1, = madel. Paper- vocto (lest. da) por ("V, - infer", V) Sinular doc = model. Houses nost similar (111) print ( similar doc) print (does madel. doglecs [11]) does = [" the hous had pliny. little mouse! 14 the cat saw the mouse, "The mouse 1 cap . caway from the house" "The cart finally are the mouse", "The and of the mouse istory"

```
similar_doc = model.docvecs.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.docvecs['1'])
EXERCISE-2
Question1. Train the following documents using Doc2Vec model
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"
 Question2. Find the most similar TWO documents for the query document "cat stayed in the
 house".
 tagged - dada = [ Tagged Document (words - Word- Fokenize
   tags = [str (P)]) for P, d Pr entire (d. Lower ())),
     vec size = 20
       , alpha 20.025
      model = Doc 2vec(vector-size= vec-size, alpha=alpha,
                                        MPn-alpha 50.000 05)
min-count=1)
    model. build-vocab (dagged. dala)
  itagged data shtils shuffle (tagged data)
```

•

-3

-3

madel. train (tagged. datagradal. examples much. 1. compus

. count, create 20)

Model. ave. (Mov. murdel")

point ("Model Laved")

from . gensim. mudels. ducevee import Mocover.

model = .D.ocver. load ("dov. mudel")

- test - data = word. dokonfro ("cal. staged in the

house". lower()

V1 = model. figer. vo eter (test. data)

Print ("V1 - inter", V1)

Strilar - doc = model. docvers-most. similar (\*2")

prat (samilar-doc)