**NATURAL LANGUAGE PROCESSING (SE 313)**

**Practical File**

**(2024- 2025)**

**Submitted By**

**Sanoj (2K21/SE/159)**

**Under the Supervision of**

**Prof. Geetanjali Garg**



**DELHI TECHNOLOGICAL UNIVERSITY**

**(Formerly Delhi College of Engineering)**

**Bawana Road, Delhi-110042**

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Experiment Name** | **Date** | **Teacher Sign** | **Remarks** |
| 1. | Import nltk and download the ‘stopwords’  and ‘punkt’ packages | 13 Jan 2023 |  |  |
| 2. | Import spacy and load the language model. | 13 Jan 2023 |  |  |
| 3. | WAP in python to tokenize a given text. | 20 Jan 2023 |  |  |
| 4. | WAP in python to get the sentences of a  text document. | 3 Feb 2023 |  |  |
| 5. | WAP in python to tokenize text with  stopwords as delimiters. | 3 Feb 2023 |  |  |
| 6. | WAP in python to add custom stop words in  spaCy. | 3 Feb 2023 |  |  |
| 7. | WAP to remove punctuations, perform  stemming, lemmatize given text and extract  usernames from emails | 24 Feb 2023 |  |  |
| 8. | WAP to do spell correction, extract all  nouns, pronouns and verbs in a given text | 7 Mar 2023 |  |  |
| 9. | WAP to find similarity between two words  and classify a text as positive/negative  sentiment | 31 Mar 2023 |  |  |

**EXPERIMENT 1 – NLTK, STOPWORDS AND PUNKT**

**OBJECTIVE**

Import nltk and download the ‘stopwords’ and ‘punkt’ packages.

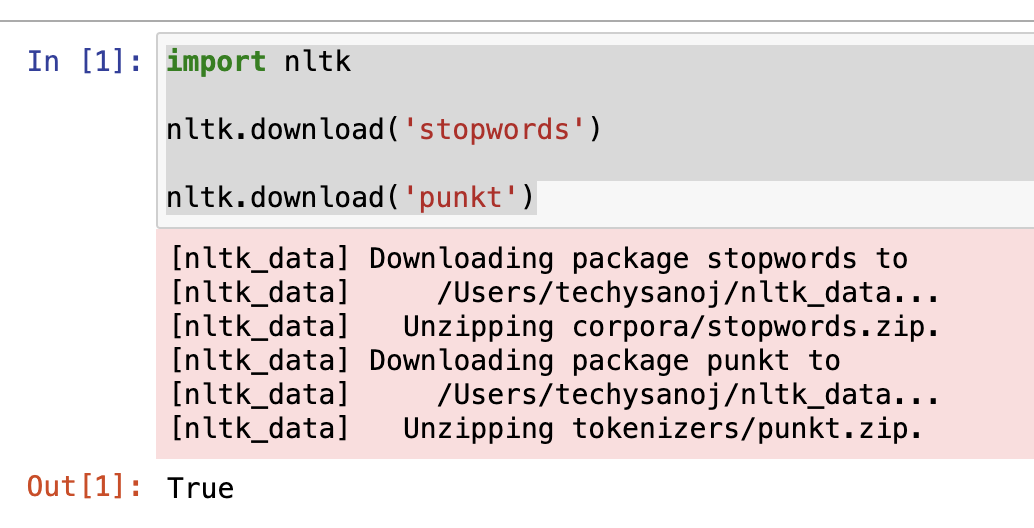
**CODE:**

import nltk

nltk.download('stopwords')

nltk.download('punkt')

**OUTPUT:**



**EXPERIMENT 2 – SPACY AND LOAD THE MODEL**

**OBJECTIVE**

Import spacy and load the language model.

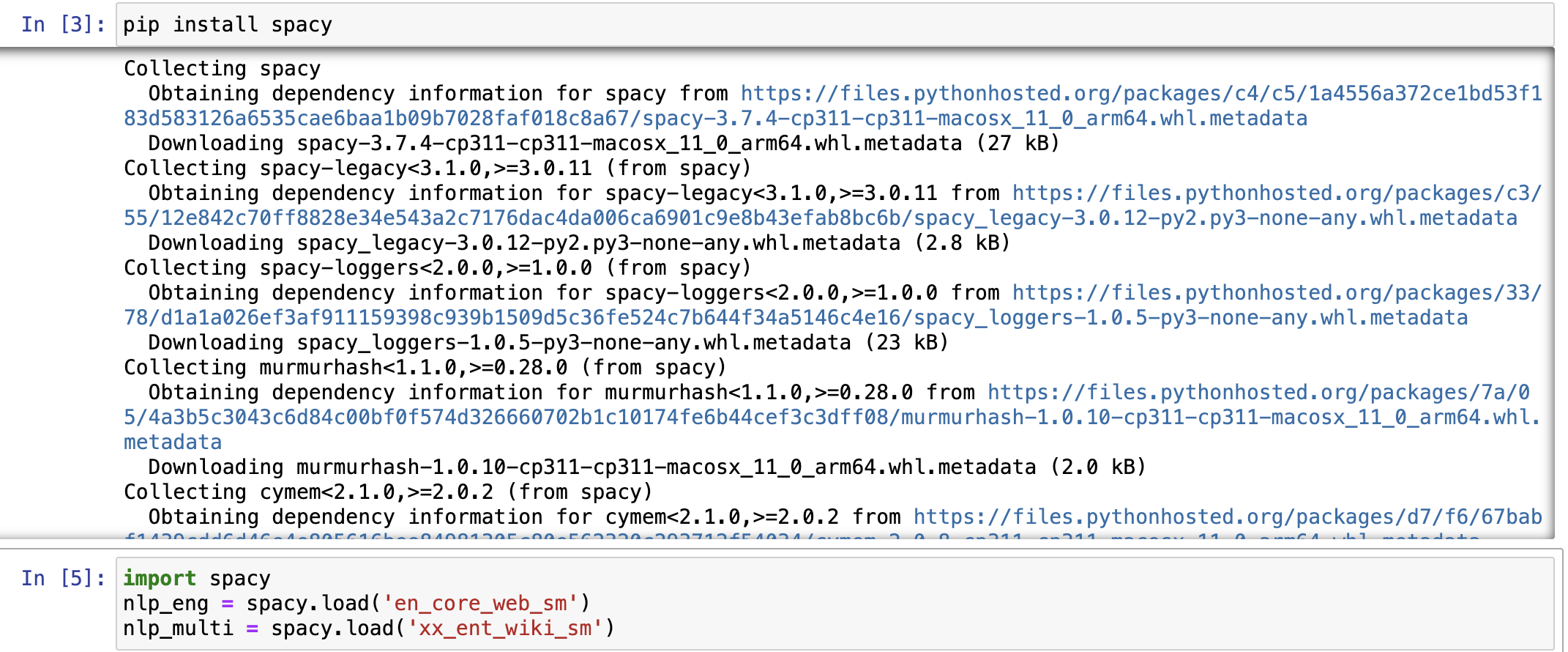
**CODE:**

import spacy

nlp\_eng = spacy.load('en\_core\_web\_sm')

nlp\_multi = spacy.load('xx\_ent\_wiki\_sm')

**OUTPUT:**



**EXPERIMENT 3 – TOKENIZATION**

**OBJECTIVE**

WAP in python to tokenize a given text.

**CODE:**

from nltk import word\_tokenize

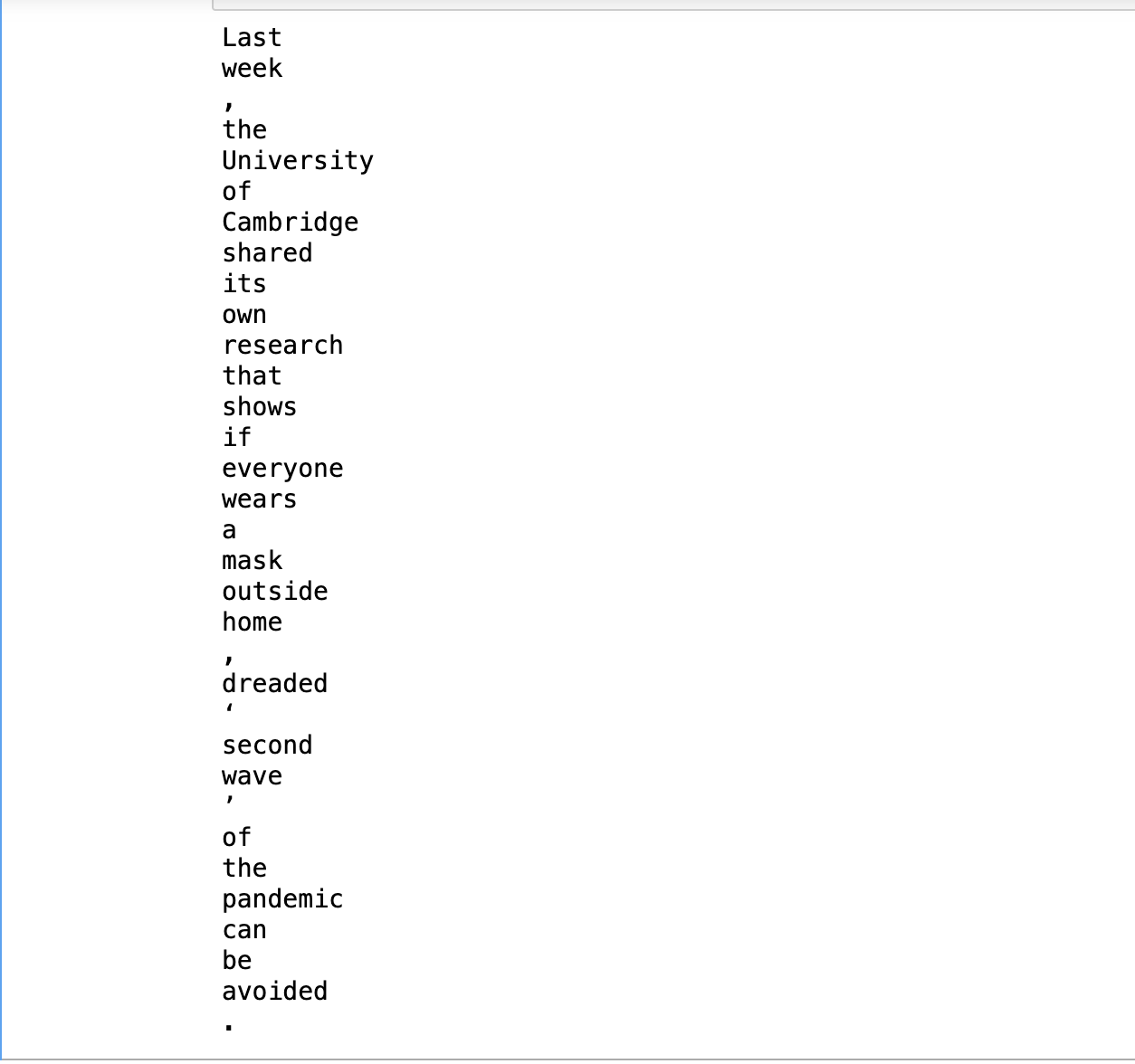
text = "Last week, the University of Cambridge shared its own research that shows if everyone wears a mask outside home,dreaded ‘second wave’ of the pandemic can be avoided."

text = word\_tokenize(text)

for t in text:

print(t)

**OUTPUT:**

****

**EXPERIMENT 4 – SENTENCE IN DOCUMENT**

**OBJECTIVE**

WAP in python to get the sentences of a text document.

**CODE:**

file = open('file.txt')

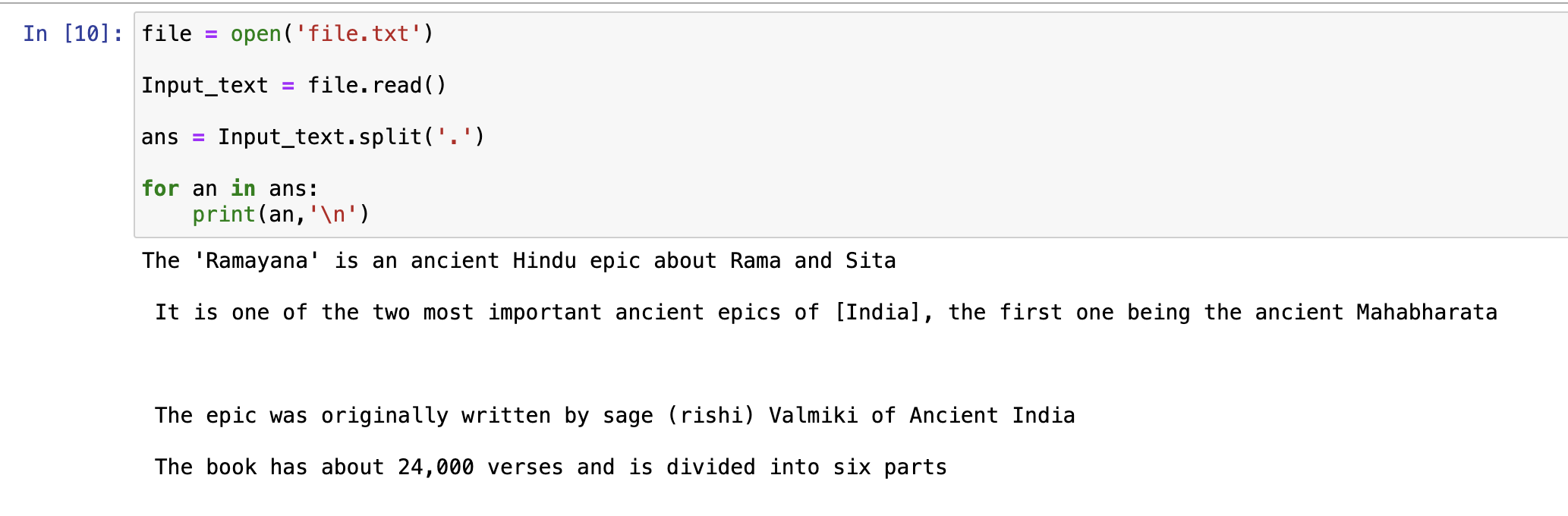
Input\_text = file.read()

ans = Input\_text.split('.')

for an in ans:

print(an,'\n')

**OUTPUT:**

****

**EXPERIMENT 5 – TOKENIZATION WITH STOP WORDS**

**OBJECTIVE**

WAP in python to tokenize text with stop words as delimiters.

**CODE:**

text = "Walter was feeling anxious. He was diagnosed today. He probably is the best person I know."

stop\_words\_and\_delims = ['was', 'is', 'the', '.', ',', '-', '!', '?']

for r in stop\_words\_and\_delims:

text = text.replace(r, 'DELIM')

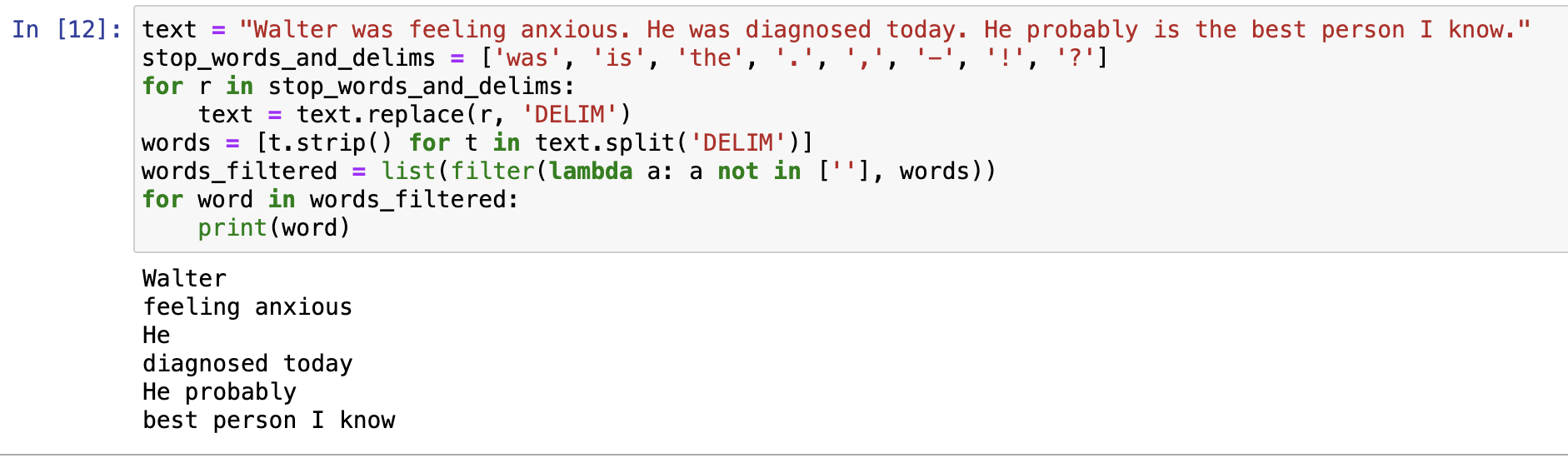
words = [t.strip() for t in text.split('DELIM')]

words\_filtered = list(filter(lambda a: a not in [''], words))

for word in words\_filtered:

print(word)

**OUTPUT:**

****

**EXPERIMENT 6 – CUSTOM WORDS IN spaCy**

**OBJECTIVE**

WAP in python to add custom stop words in spaCy.

**CODE:**

import spacy

nlp = spacy.load('en\_core\_web\_sm')

custom\_stop\_words = ['was', 'is','the','JUNK','NIL','of','more' ,'.', ',', '-', '!', '?','a']

for word in custom\_stop\_words:

nlp.vocab[word].is\_stop = True

doc = nlp("Jonas was a JUNK great guy NIL Adam was evil NIL Martha JUNKwas more of a fool")

for token in doc:

if not token.is\_stop:

print(token.text, end=" ")

**OUTPUT:**



**EXPERIMENT 7 – STEMMING**

**OBJECTIVE**

WAP to remove punctuations, perform stemming, lemmatize given text and extract usernames from emails.

**CODE:**

punctuations = '''!()-[]{};:'"\,<>./?@#$%^&\*\_~'''

string = "Jonas!!! great \\guy <> Adam --evil [Martha] ;;fool() ."

ans = ""

for char in string:

if char not in punctuations:

ans+=char

print(ans)

from nltk.stem import PorterStemmer

from nltk.tokenize import word\_tokenize

text= "Dancing is an art. Students should be taught dance as a subject in schools . I danced in many of my school function. Some people are always hesitating to dance."

ans = ""

stemmer = PorterStemmer()

tokens = word\_tokenize(text)

for token in tokens:

ans+=stemmer.stem(token)

ans+=" "

print(ans)

from nltk.corpus import wordnet

from nltk.tokenize import word\_tokenize

from nltk.stem.wordnet import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

text= "Dancing is an art. Students should be taught dance as a subject in schools . I danced in many of my school function. Some people are always hesitating to dance."

ans = ""

tokens = word\_tokenize(text)

for token in tokens:

ans+=lemmatizer.lemmatize(token, wordnet.VERB)

ans+=" "

print(ans)

from nltk.tokenize import word\_tokenize

text= "The new registrations are potter709@gmail.com , elixir101@gmail.com. If you find any disruptions, kindly contact granger111@gamil.com or severus77@gamil.com "

text\_list = word\_tokenize(text)

usernames = []

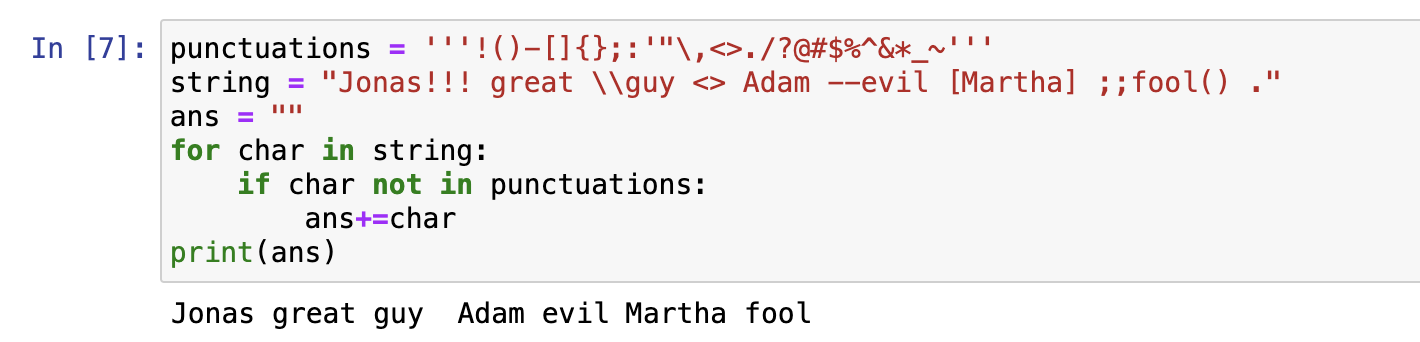
for i in range(len(text\_list)):

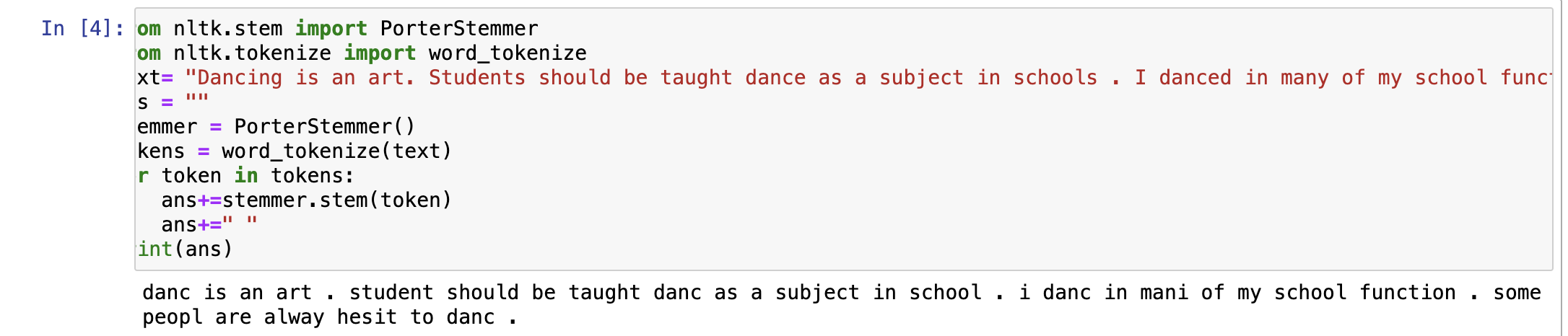
if text\_list[i] == "@":

usernames.append(text\_list[i-1])

print(usernames)

**OUTPUT:**







**EXPERIMENT 8 – SPELL CORRECTION**

**OBJECTIVE**

WAP to do spell correction, extract all nouns, pronouns and verbs in a given text.

**CODE:**

import textblob

from textblob import TextBlob

text="He is a gret person. He beleives in bod"

textb = TextBlob(text)

correct\_text = textb.correct()

print(correct\_text)

import nltk

from nltk import word\_tokenize, pos\_tag

text="James works at Microsoft. She lives in manchester and likes to play the flute"

tokens = word\_tokenize(text)

parts\_of\_speech = nltk.pos\_tag(tokens)

nouns = list(filter(lambda x: x[1] == "NN" or x[1] == "NNP",

parts\_of\_speech))

for noun in nouns:

print(noun[0])

from nltk import pos\_tag, word\_tokenize

text = "I may bake a cake for my birthday. The talk will introduce reader about Use of baking"

words = word\_tokenize(text)

verb\_phrases = []

for i in range(len(words)):

if i > 0 and pos\_tag(words)[i][1] == 'VB':

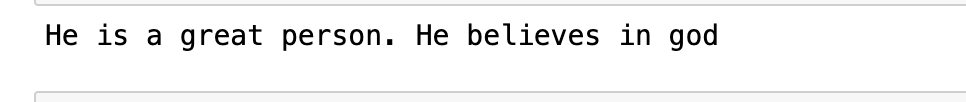
verb\_phrase = words[i-1] + ' ' + words[i]

verb\_phrases.append(verb\_phrase)

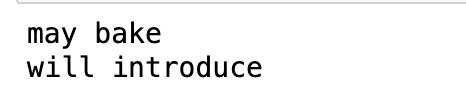
for i in verb\_phrases:

print (i)

**OUTPUT:**







**EXPERIMENT 9 – SIMILARITY BETWEEN WORDS**

**OBJECTIVE**

WAP to find similarity between two words and classify a text as positive/negative sentiment.

**CODE:**

import spacy

nlp = spacy.load('en\_core\_web\_md')

words = "amazing terrible excellent"

tokens = nlp(words)

token1, token2, token3 = tokens[0], tokens[1], tokens[2]

print(f"Similarity between {token1} and {token2} : ",

token1.similarity(token2))

print(f"Similarity between {token1} and {token3} : ",

token1.similarity(token3))

from textblob import TextBlob

text = "It was a very pleasant day"

print(TextBlob(text).sentiment)

**OUTPUT:**

