**18CSC305J-Artificial Intelligence**

**LAB 13 – Implementation of NLP programs**

**Aim:** To implement NLP programs

**Description:**

NLP stands for Natural Language Processing, which is a part of Computer Science, Human language, and Artificial Intelligence. It is the technology that is used by machines to understand, analyze, manipulate, and interpret human's languages. It helps developers to organize knowledge for performing tasks such as translation, automatic summarization, Named Entity Recognition (NER), speech recognition, relationship extraction, and topic segmentation.

**CODE :**

from nltk.tokenize import sent\_tokenize, word\_tokenize

example\_string = """

... Muad'Dib learned rapidly because his first training was in how to learn.

... And the first lesson of all was the basic trust that he could learn. ... It's shocking to find how many people do not believe they can learn, ... and how many more believe learning to be difficult.""" sent\_tokenize(example\_string)

word\_tokenize(example\_string)

Filtering stop words:

nltk.download("stopwords")

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

worf\_quote = "Sir, I protest. I am not a merry man!"

words\_in\_quote = word\_tokenize(worf\_quote)

words\_in\_quote

stop\_words = set(stopwords.words("english"))

filtered\_list = []

for word in words\_in\_quote:

if word.casefold() not in stop\_words:

filtered\_list.append(word)

filtered\_list = [

word for word in words\_in\_quote if word.casefold() not in stop\_words ]

filtered\_list

Stemming:

from nltk.stem import PorterStemmer

from nltk.tokenize import word\_tokenize

stemmer = PorterStemmer()

string\_for\_stemming = """

The crew of the USS Discovery discovered many discoveries. Discovering is what explorers do."""

words = word\_tokenize(string\_for\_stemming)

words

stemmed\_words = [stemmer.stem(word) for word in words]

stemmed\_words

Tagged parts of speech:

from nltk.tokenize import word\_tokenize

sagan\_quote = """

If you wish to make an apple pie from scratch,

you must first invent the universe."""

words\_in\_sagan\_quote = word\_tokenize(sagan\_quote)

import nltk

nltk.pos\_tag(words\_in\_sagan\_quote)

jabberwocky\_excerpt = """

... 'Twas brillig, and the slithy toves did gyre and gimble in the wabe: ... all mimsy were the borogoves, and the mome raths outgrabe.""" words\_in\_excerpt = word\_tokenize(jabberwocky\_excerpt)

nltk.pos\_tag(words\_in\_excerpt)

Lemmatizing:

from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

lemmatizer.lemmatize("scarves")

string\_for\_lemmatizing = "The friends of DeSoto love scarves."

words = word\_tokenize(string\_for\_lemmatizing)

words

lemmatized\_words = [lemmatizer.lemmatize(word) for word in words] lemmatized\_words

lemmatizer.lemmatize("worst")

lemmatizer.lemmatize("worst", pos="a")

Chunking:

from nltk.tokenize import word\_tokenize

lotr\_quote = "It's a dangerous business, Frodo, going out your door." words\_in\_lotr\_quote = word\_tokenize(lotr\_quote)

words\_in\_lotr\_quote

nltk.download("averaged\_perceptron\_tagger")

lotr\_pos\_tags = nltk.pos\_tag(words\_in\_lotr\_quote)

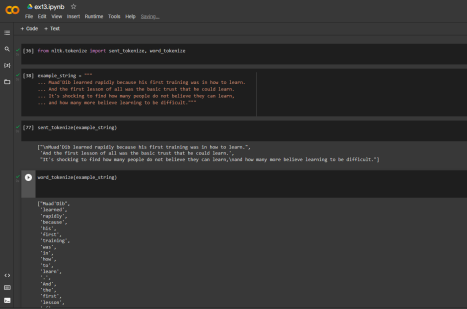
lotr\_pos\_tags

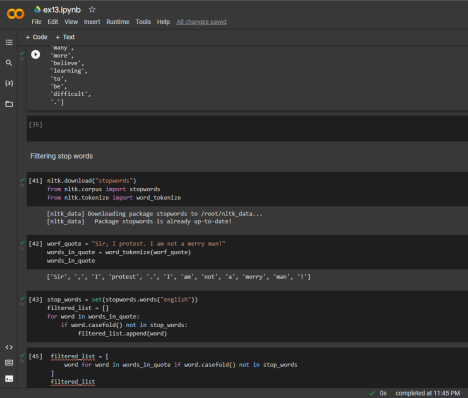
grammar = "NP: {<DT>?<JJ>\*<NN>}"

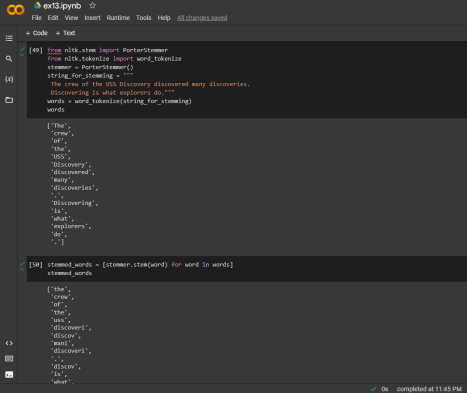
chunk\_parser = nltk.RegexpParser(grammar)

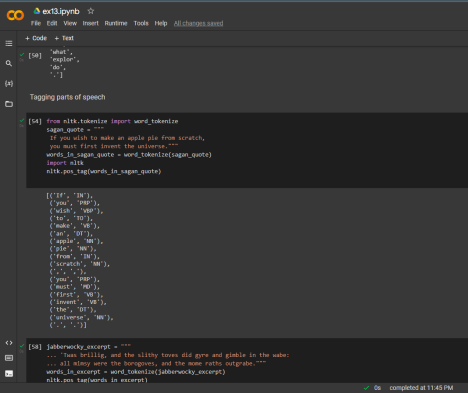
tree = chunk\_parser.parse(lotr\_pos\_tags)

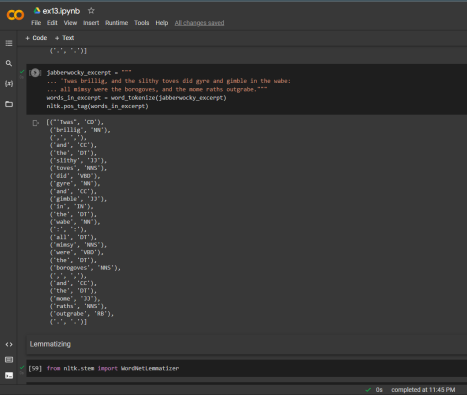
**OUTPUT :**

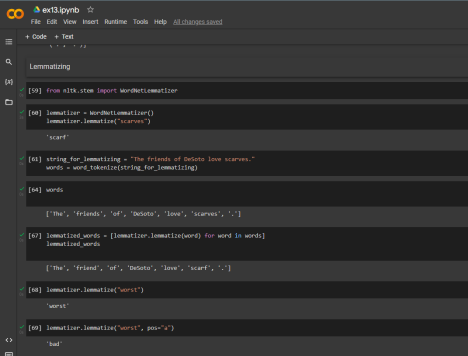
****

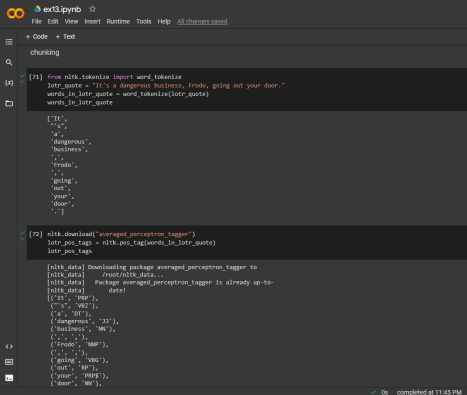
****

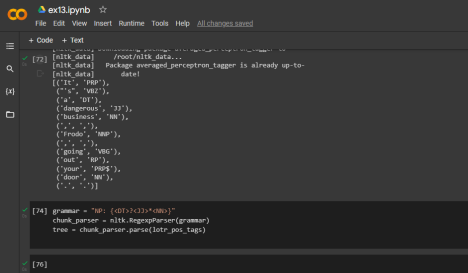
****

****

****

****

****

****

**RESULT :**

Hence the NPL program was implemented.