Rohan Nyati

500075940

R177219148

B5-Ai & Ml

Experiment -5

Q1) Discuss about the Part of Speech (POS) tagging in NLP. How can it be performed by using NLTK?

Ans: Part-of-speech (POS) tagging is a popular Natural Language Processing process which refers to categorizing words in a text (corpus) in correspondence with a particular part of speech, depending on the definition of the word and its context.

Example:

Input: Why not tell someone?

Output: Why – adverb, not – adverb, tell – verb, someone – noun, ? – punctuation mark, sentence closure.

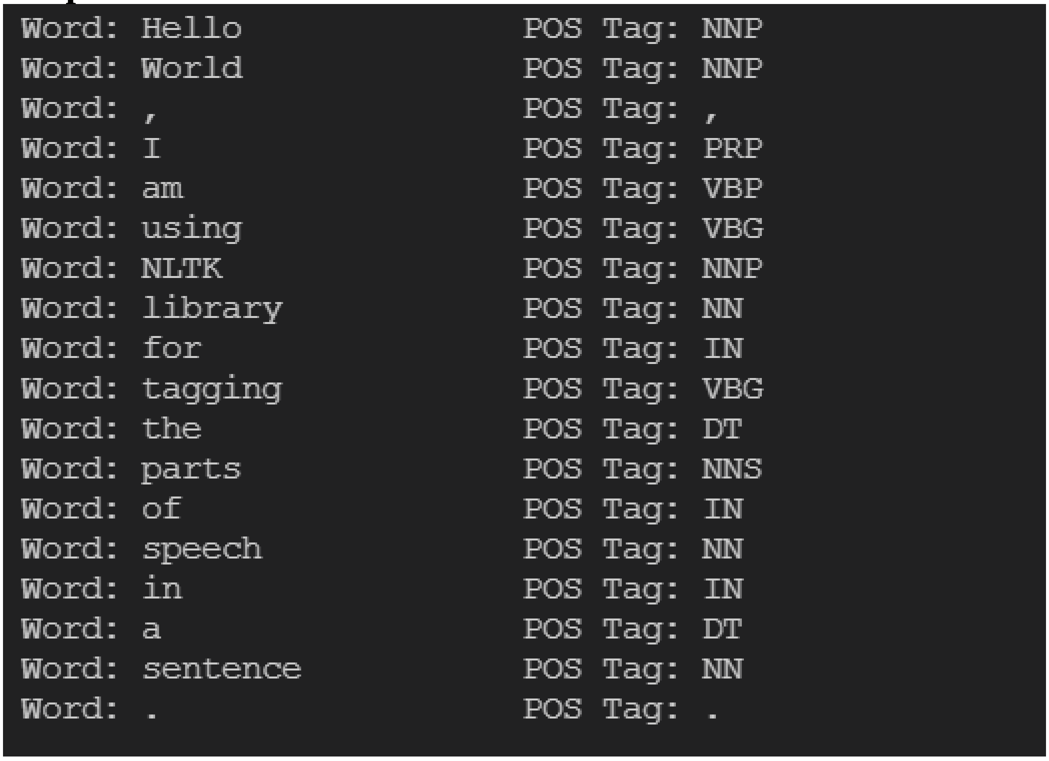
NLTK: The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language. NLTK is a software package for manipulating linguistic data and performing NLP tasks. It is also intended to support research and teaching in NLP or closely related areas, including empirical linguistics, cognitive science, artificial intelligence, information retrieval, and machine learning.

Code:

```
import nltk
from nltk.tokenize import word_tokenize

text = "Hello World, I am using NLTK library for tagging the parts of speech in a sentence."
tokenized_text = word_tokenize(text)
pos_tags = nltk.pos_tag(tokens=tokenized_text)
for tags in pos_tags:
    print(f"Word: {tags[0]} \t\t POS Tag: {tags[1]}")
```

Output:



```
Word: Hello              POS Tag: NNP
Word: World              POS Tag: NNP
Word: ,                  POS Tag: ,
Word: I                  POS Tag: PRP
Word: am                 POS Tag: VBP
Word: using              POS Tag: VBG
Word: NLTK               POS Tag: NNP
Word: library            POS Tag: NN
Word: for                POS Tag: IN
Word: tagging            POS Tag: VBG
Word: the                POS Tag: DT
Word: parts              POS Tag: NNS
Word: of                 POS Tag: IN
Word: speech             POS Tag: NN
Word: in                 POS Tag: IN
Word: a                  POS Tag: DT
Word: sentence           POS Tag: NN
Word: .                  POS Tag: .
```

POS Tags:

NNP = Proper noun,
PRP = Personal Pronoun,
VBP = Verb, singular, present,
VBG = Verb, present participle taking,
NN = Noun, singular,
IN = Preposition/subordinating conjunction,
DT = Determiner,
NNS = Noun, plural,

Q2) Discuss about the Conditional Random Fields and their role in POS tagging.

Ans: Models based on Bag of Words technique fail to capture syntactic relations between words. Considering a sentiment analysis problem based on only Bag of Words, the model will be unable to capture the difference between "I like your car.", where "like" is a verb with a positive sentiment, and another sentence "I am like you in many ways.", where in this sentence "like" is a preposition with a neutral sentiment. To improve the accuracy of this Bag of Words techniques, we can use Parts of Speech tagging.

Conditional Random Fields (CRF): A CRF is a Discriminative Probabilistic Classifiers. The difference between discriminative and generative models is that while discriminative models try to model conditional probability distribution, i.e., P(y|x), generative models try to model a joint probability distribution, i.e., P(x,y). Logistic Regression, SVM, CRF are Discriminative Classifiers. CRF's can also be used for sequence labelling tasks like Named Entity Recognizers (NERs) and POS Taggers.

In CRFs (Conditional Random Fields), the input is a set of features derived from the input sequence using feature functions, the weights associated with the features (that are learned) and the previous label and the task is to predict the current label. The weights of different feature functions will be determined such that the likelihood of the labels in the training data will be maximized.

In CRF, a set of feature functions are defined to extract features for each word in a sentence. Some examples of feature functions are: is the first letter of the word capitalized, what the suffix and prefix of the word, what is the previous word, is it the first or the last word of the sentence, is it a number etc. These set of features are called State Features. In CRF, we also pass the label of the previous word and the label of the current word to learn the weights. CRF will try to determine the weights of different feature functions that will maximize the likelihood of the labels in the training data. The feature function dependent on the label of the previous word is Transition Feature