

**DEPARTMENT OF COMPUTER ENGINEERING**

## CSL804 Computational Lab II

**Eighth Semester, 2021-2022 (Even Semester)**

**Name of Student :** Saurav Kumar

## Roll No. 23

**Division :** BE – CMPN

**Day/ Session :** Monday/Afternoon

**Venue :** SLRTCE Lab 305

## Experiment No. 5

**Title of Experiment :** To study and implement Chunking.

## Date of Conduction :

**Date of Submission :**

|  |  |  |
| --- | --- | --- |
| **Particulars Max. Marks Marks Obtained** | | |
| Preparedness and Efforts(PE) | **3** |  |
| Knowledge of tools(KT) | **3** |  |
| Debugging and results(DR) | **3** |  |
| Documentation(DN) | **3** |  |
| Punctuality & Lab Ethics(PL) | **3** |  |
| **Total** | **15** |  |

**Grades – Meet Expectations (3 Marks), Moderate Expectations (2 Marks), Below Expectations (1 Mark)**

**Checked and Verified by Name of Faculty :** Prof. Neelam Kulkarni

## Signature :

**Date :**

EXPERIMENT NO: 5

CHUNKING

**AIM:** To study and implement Chunking.

**SOFTWARE:** Python, NLTK, CLTK

# THEORY:

CHUNKING

Chunking is used to add more structure to the sentence by following parts of speech (POS) tagging. It is also known as shallow parsing. The resulted group of words is called "chunks." In shallow parsing, there is maximum one level between roots and leaves while deep parsing comprises of more than one level. Shallow Parsing is also called light parsing or chunking.

The primary usage of chunking is to make a group of "noun phrases." The parts of speech are combined with regular expressions. There are no pre-defined rules, but you can combine them according to need and requirement.

For example, you need to tag Noun, verb (past tense), adjective, and coordinating junction from the sentence. You can use the rule as below

chunk:{<NN.?>\*<VBD.?>\*<JJ.?>\*<CC>?}

Chunking is used for entity detection. An entity is that part of the sentence by which machine get the value for any intention.

# IMPLEMENTATION:

CHUNKING CODE

import nltk

from nltk import pos\_tag from nltk import RegexpParser

# nltk.download('averaged\_perceptron\_tagger')

text ="learn php from guru99 and make study easy".split() print("After Split:", text)

tokens\_tag = pos\_tag(text) print("After Token:",tokens\_tag)

patterns= """mychunk:{<NN.?>\*<VBD.?>\*<JJ.?>\*<CC>?}""" chunker = RegexpParser(patterns)

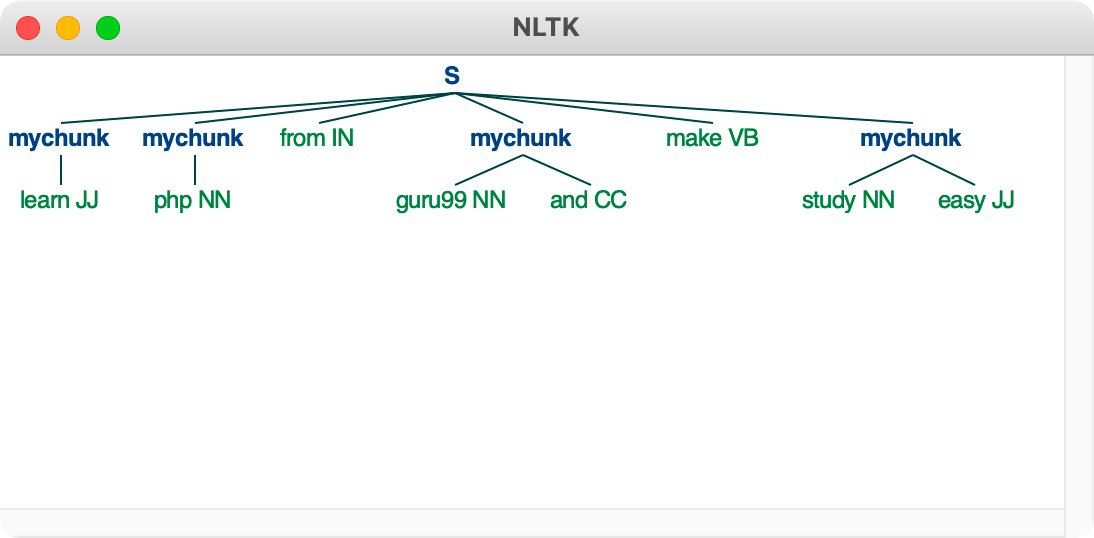
print("After Regex:",chunker)

output = chunker.parse(tokens\_tag) print("After Chunking",output) output.draw()

OUTPUT

Text

Description automatically generated



# CONCLUSION:

Thus we have studied and implemented chunking.