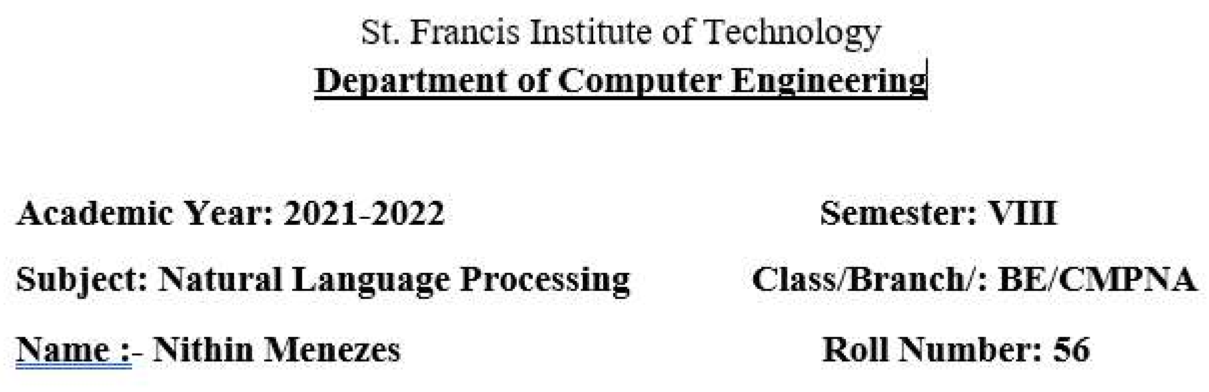

St. Francis Institute of Technology

Department of Computer Engineering

Academic Year: 2021-2022 **Semester: VIII**

Subject: Natural Language Processing **Class/Branch/: BE/CMPNA**

Name :- Nithin Menezes **Roll Number: 56**

```
 Experiment No 06

 Chunking

**Aim:** To implement the Chunking with NLTK toolkit

**Theory:**

 Chunking is a process of extracting phrases from unstructured text. Chunking
works on top of POS tagging, it uses pos-tags as input and provides chunks as
output. Similar to POS tags, there are a standard set of Chunk tags like Noun
Phrase(NP), Verb Phrase (VP), etc. Chunking is very important when you want to
extract information from text such as Locations, Person Names etc. in NLP
called Named Entity Extraction.
```

Experiment No 06

Chunking

Aim: To implement the Chunking with NLTK toolkit

Theory:

Chunking is a process of extracting phrases from unstructured text. Chunking works on top of POS tagging, it uses pos-tags as input and provides chunks as output. Similar to POS tags, there are a standard set of Chunk tags like Noun Phrase(NP), Verb Phrase (VP), etc. Chunking is very important when you want to extract information from text such as Locations, Person Names etc. in NLP called Named Entity Extraction.

Double-click (or enter) to edit

```
import nltk
import pandas as pd
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('maxent_ne_chunker')
nltk.download('words')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package maxent_ne_chunker is already up-to-date!
[nltk_data] Downloading package words to /root/nltk_data...
[nltk_data]   Package words is already up-to-date!
True
```

```
import matplotlib
matplotlib.use('Agg')
sentence = [("the", "DT"), ("little", "JJ"), ("yellow", "JJ"), ("dog", "NN"), ("barked", "VBD
grammar = "NP: {<DT>?<JJ>*<NN>}"

cp = nltk.RegexpParser(grammar)
result = cp.parse(sentence)
print(*result)
```

```
(NP the/DT little/JJ yellow/JJ dog/NN) ('barked', 'VBD') ('at', 'IN') (NP the/DT cat/NN
```

Conclusion:

Thus we implemented chunking with the help of NLTK tool kit. Chunking is a process of extracting phrases from unstructured text.

```
POS tag list:

CC  coordinating conjunction
CD  cardinal digit
DT  determiner
EX  existential there (like: "there is" ... think of it like "there exists")
FW  foreign word
IN  preposition/subordinating conjunction
JJ  adjective   'big'
JJR adjective, comparative  'bigger'
JJS adjective, superlative  'biggest'
LS  list marker 1)
MD  modal   could, will
NN  noun, singular 'desk'
NNS noun plural 'desks'
NNP proper noun, singular   'Harrison'
NNPS    proper noun, plural 'Americans'
PDT predeterminer   'all the kids'
POS possessive ending   parent\'s
PRP personal pronoun    I, he, she
PRP$    possessive pronoun  my, his, hers
RB  adverb  very, silently,
RBR adverb, comparative better
RBS adverb, superlative best
RP  particle    give up
TO  to  go 'to' the store.
UH  interjection    errrrrrrrm
VB  verb, base form take
VBD verb, past tense    took
VBG verb, gerund/present participle taking
VBN verb, past participle   taken
VBP verb, sing. present, non-3d take
VBZ verb, 3rd person sing. present  takes
WDT wh-determiner   which
WP  wh-pronoun  who, what
WP$ possessive wh-pronoun   whose
WRB wh-abverb   where, when
```

● ×