|  |
| --- |
| Experiment No.6 |
| Perform POS tagging on the given English and Indian  Language Text |
| Date of Performance: |
| Date of Submission: |

CSDL7013: Natural Language Processing Lab



Aim: Perform POS tagging on the given English and Indian Language Text

Objective: To study POS Tagging and tag the part of speech for given input in english and an Indian Language.

Theory:

The primary target of Part-of-Speech (POS) tagging is to identify the grammatical group of a given word. Whether it is a NOUN, PRONOUN, ADJECTIVE, VERB, ADVERBS, etc. based on the context. POS Tagging looks for relationships within the sentence and assigns a corresponding tag to the word.

POS Tagging (Parts of Speech Tagging) is a process to mark up the words in text format for a particular part of a speech based on its definition and context. It is responsible for text reading in a language and assigning some specific token (Parts of Speech) to each word. It is also called grammatical tagging.

Steps Involved in the POS tagging example:

* Tokenize text (word\_tokenize)
* apply pos\_tag to above step that is nltk.pos\_tag(tokenize\_text)

CSDL7013: Natural Language Processing Lab

import nltk nltk.download('punkt') nltk.download('averaged\_perceptron\_tagger')

[nltk\_data] Downloading package punkt to /root/nltk\_data...

[nltk\_data] Unzipping tokenizers/punkt.zip.

[nltk\_data] Downloading package averaged\_perceptron\_tagger to [nltk\_data] /root/nltk\_data...

[nltk\_data] Unzipping taggers/averaged\_perceptron\_tagger.zip.

True

from nltk.chunk import RegexpParser from nltk.tokenize import word\_tokenize

sentence = "Education is the transmission of knowledge, skills, and character traits. " Tokenization

tokens = word\_tokenize(sentence)

tokens

['Education',

'is',

'the',

'transmission',

'of',

'knowledge',

',',

'skills',

',',

'and',

'character',

'traits',

# '.'] POS tagging

pos\_tags = nltk.pos\_tag(tokens)

pos\_tags

[('Education', 'NN'),

('is', 'VBZ'),

('the', 'DT'),

('transmission', 'NN'),

('of', 'IN'),

('knowledge', 'NN'),

(',', ','),

('skills', 'NNS'),

(',', ','),

('and', 'CC'),

('character', 'NN'),

('traits', 'NNS'),

# ('.', '.')] Chunking patterns

chunk\_patterns = r"""

NP: {<DT>?<JJ>\*<NN>} # Chunk noun phrases

VP: {<VB.\*><NP|PP>} # Chunk verb phrases """

chunk\_patterns

'\n NP: {<DT>?<JJ>\*<NN>} # Chunk noun phrases\n VP: {<VB.\*><NP|PP>} # Chunk verb phrases\n'

# Create a chunk parser

chunk\_parser = RegexpParser(chunk\_patterns)

chunk\_parser

<chunk.RegexpParser with 2 stages> Perform chunking

result = chunk\_parser.parse(pos\_tags)

print(result)

(S

(NP Education/NN)

(VP is/VBZ (NP the/DT transmission/NN)) of/IN

(NP knowledge/NN)

,/, skills/NNS

,/, and/CC

(NP character/NN) traits/NNS ./.)

**Conclusion:**

POS tagging (Part-of-Speech tagging) involves labeling words in a text with their grammatical categories (e.g., noun, verb, adjective). For English text, widely available libraries like NLTK or spaCy provide accurate tagging due to well-defined grammar. Indian languages pose greater challenges due to their diversity, script variations, and limited resources. Building accurate POS taggers for Indian languages often requires language-specific models and extensive linguistic knowledge.