## Indian Institute of Information Technology Surat

****

# Lab Report on

# Natural Language Processing (CS 601) Practical

**Submitted by**

### [RAHUL KUMAR SINGH] (UI21CS44)

**Course Faculty**

### Mrs. Nidhi Desai

## Department of Computer Science and Engineering

## Indian Institute of Information Technology Surat

## Gujarat-394190, India

**Jan-2024**

## Lab No: 4

**Aim:**

To demonstrate the use of the Treebank and WordNet library in NLTK by performing sentence tagging and tokenization, and exploring the functions available in WordNet.

**Description:**

**Treebank:** A corpus of text annotated with syntactic structure, used for training and evaluating parsing models.

**WordNet:** A lexical database that groups words into synsets, providing semantic relations and definitions for computational linguistics.

**POS Tagging**: The sentence is tokenized and tagged using NLTK's *word\_tokenize* and *pos\_tag* functions.

**Functions Available in WordNet:**

* **synsets(word)**: Retrieves all synsets (senses) for a given word.
* **definition()**: Provides the definition of a specific synset.
* **lemmas()**: Returns the list of lemmas (base forms of a word) for a synset.
* **hypernyms()**: Shows general concepts related to the word.
* **hyponyms()**: Displays more specific categories related to the word.
* **part\_meronyms()**: Retrieves parts of a concept.
* **member\_holonyms()**: Finds collections to which a concept belongs.

### POS Tags:

* **Sentence**: The sentence as a tokenized string.
* **Noun**: Lists all nouns in the sentence (e.g., "cat", "mat").
* **Verb**: Lists all verbs in the sentence (e.g., "sat").
* **Adjective**: Lists all adjectives (e.g., "beautiful").
* **Adverb**: Lists all adverbs (e.g., "quickly").
* **Determiner**: Lists all determiners (e.g., "the").
* **Preposition**: Lists all prepositions (e.g., "on").
* **Pronoun**: Lists all pronouns (e.g., "he").
* **Conjunction**: Lists all conjunctions (e.g., "and"").
* **Modal**: Lists modal auxiliary verbs (e.g., "will").

### Output:

* **Treebank POS Tagging**: Lists the tokens along with their tags using the Treebank tagger.
* **WordNet-based Tagging**: Lists the tokens along with their WordNet-based POS tags.
* **Time Taken**: Displays the time taken by each approach.
* **Precision**: Compares the POS tagging accuracy between the two methods.

## Source Code:

**import nltk**

**from nltk.corpus import wordnet as wn**

**from nltk.tokenize import word\_tokenize**

**from nltk import pos\_tag**

**from time import time**

**nltk.download('treebank')**

**nltk.download('wordnet')**

**nltk.download('punkt')**

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

**nltk.download('omw-1.4')**

**pos\_full\_form = {**

**'CC': 'Coordinating conjunction', 'CD': 'Cardinal digit', 'DT': 'Determiner',**

**'EX': 'Existential there', 'FW': 'Foreign word', 'IN': 'Preposition/subordinating conjunction',**

**'JJ': 'Adjective', 'JJR': 'Adjective, comparative', 'JJS': 'Adjective, superlative',**

**'LS': 'List item marker', 'MD': 'Modal', 'NN': 'Noun, singular or mass', 'NNS': 'Noun, plural',**

**'NNP': 'Proper noun, singular', 'NNPS': 'Proper noun, plural', 'PDT': 'Predeterminer',**

**'POS': 'Possessive ending', 'PRP': 'Personal pronoun', 'PRP$': 'Possessive pronoun',**

**'RB': 'Adverb', 'RBR': 'Adverb, comparative', 'RBS': 'Adverb, superlative', 'RP': 'Particle',**

**'TO': 'To', 'UH': 'Interjection', 'VB': 'Verb, base form', 'VBD': 'Verb, past tense',**

**'VBG': 'Verb, gerund or present participle', 'VBN': 'Verb, past participle',**

**'VBP': 'Verb, non-3rd person singular present', 'VBZ': 'Verb, 3rd person singular present',**

**'WDT': 'Wh-determiner', 'WP': 'Wh-pronoun', 'WP$': 'Possessive wh-pronoun', 'WRB': 'Wh-adverb'**

**}**

**def wordnet\_pos\_code(tag):**

**if tag.startswith('J'):**

**return wn.ADJ**

**elif tag.startswith('V'):**

**return wn.VERB**

**elif tag.startswith('N'):**

**return wn.NOUN**

**elif tag.startswith('R'):**

**return wn.ADV**

**else:**

**return None**

**tokenization = tokenized\_texts**

**total\_treebank\_time = 0**

**total\_wordnet\_time = 0**

**total\_precision = 0**

**total\_sentences = len(tokenization)**

**def get\_pos\_full\_form(tag):**

**return pos\_full\_form.get(tag, "Unknown")**

**for tokens in tokenization:**

**print(f"\nProcessing sentence: {' '.join(tokens)}")**

**word = tokens[0]**

**synsets = wn.synsets(word)**

**print(f"\nSynsets of '{word}':")**

**for synset in synsets:**

**print(f"- {synset.name()}: {synset.definition()}")**

**print(f"\nLemmas of '{word}':")**

**for synset in synsets:**

**lemmas = synset.lemmas()**

**for lemma in lemmas:**

**print(f"- {lemma.name()}")**

**print(f"\nHypernyms of '{word}':")**

**for synset in synsets:**

**hypernyms = synset.hypernyms()**

**for hypernym in hypernyms:**

**print(f"- {synset.name()} -> {hypernym.name()}")**

**print(f"\nHyponyms of '{word}':")**

**for synset in synsets:**

**hyponyms = synset.hyponyms()**

**for hyponym in hyponyms:**

**print(f"- {synset.name()} -> {hyponym.name()}")**

**print(f"\nExample sentences for '{word}':")**

**for synset in synsets:**

**examples = synset.examples()**

**for example in examples:**

**print(f"- {synset.name()}: {example}")**

**# Treebank**

**start\_treebank = time()**

**treebank\_tags = pos\_tag(tokens)**

**end\_treebank = time()**

**print("Treebank POS tagging results:")**

**for token, tag in treebank\_tags:**

**print(f'{token} -> {tag} ({get\_pos\_full\_form(tag)})')**

**# WordNet**

**start\_wordnet = time()**

**wordnet\_tags = []**

**for token, tag in treebank\_tags:**

**wn\_tag = wordnet\_pos\_code(tag)**

**if wn\_tag:**

**synsets = wn.synsets(token, wn\_tag)**

**if synsets:**

**wn\_tagged = synsets[0].pos()**

**else:**

**wn\_tagged = "unknown"**

**else:**

**wn\_tagged = "unknown"**

**wordnet\_tags.append((token, wn\_tagged))**

**end\_wordnet = time()**

**print("\nWordNet-based tagging results:")**

**for token, tag in wordnet\_tags:**

**print(f'{token} -> {tag}')**

**treebank\_time = end\_treebank - start\_treebank**

**wordnet\_time = end\_wordnet - start\_wordnet**

**total\_treebank\_time += treebank\_time**

**total\_wordnet\_time += wordnet\_time**

**correct\_matches = sum(1 for t1, t2 in zip(treebank\_tags, wordnet\_tags) if t1[1].startswith(t2[1].upper()))**

**precision = correct\_matches / len(treebank\_tags)**

**total\_precision += precision**

**avg\_precision = total\_precision / total\_sentences**

**avg\_treebank\_time = total\_treebank\_time / total\_sentences**

**avg\_wordnet\_time = total\_wordnet\_time / total\_sentences**

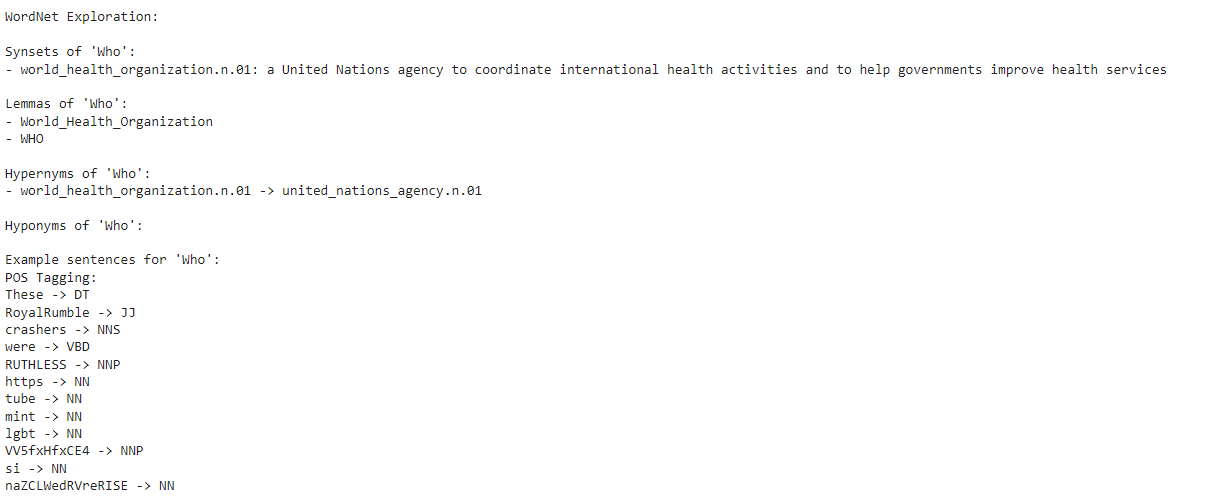
**print(f"\nOverall average time for Treebank POS tagging: {avg\_treebank\_time:.6f} seconds")**

**print(f"Overall average time for WordNet-based tagging: {avg\_wordnet\_time:.6f} seconds")**

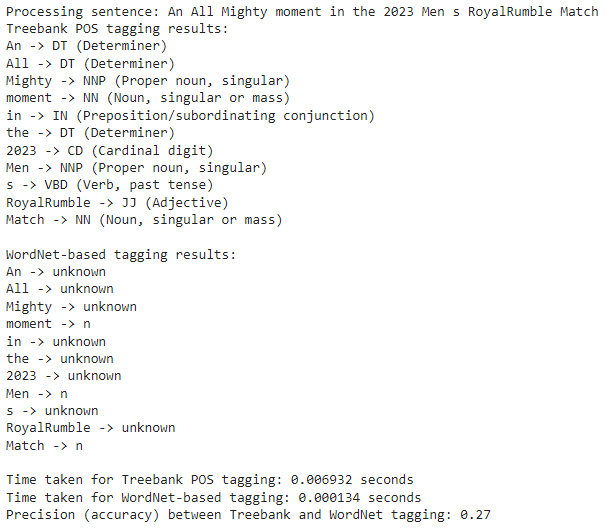
**print(f"Overall precision (accuracy): {avg\_precision:.2f}")**

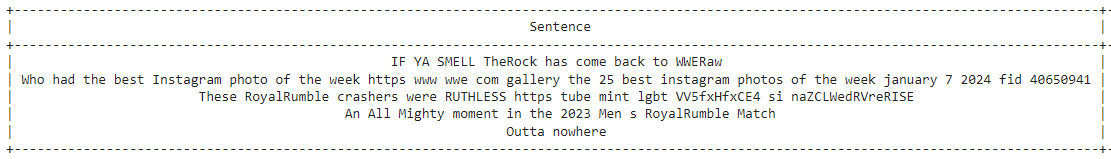
## Output:

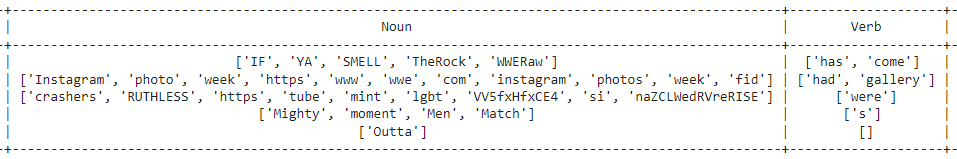
**Wordnet library functions**

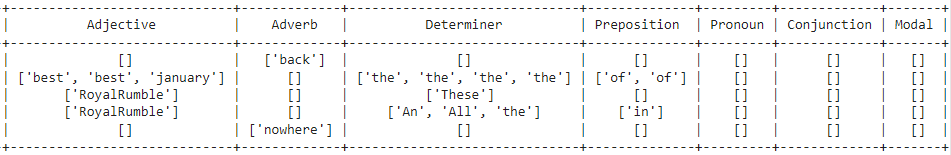


**POS Tagging**

****



****

****

## Conclusion:

* Utilizes NLTK for tagging parts of speech in tokenized sentences.
* Maps POS tags to categories like Noun, Verb, and Adjective, Determiner, Preposition, Pronoun, Conjunction, and Modal.
* Measures and compares the time taken for Treebank and WordNet tagging.
* Provides a structured way to analyze and visualize POS tagging results.