**EXPERIMENT 6**

**AIM:** POS tagging.

**RESOURCES REQUIRED:**

Python 3, NLTK toolkit, Text editor, 4 GB RAM and above, i5 processor and above

**THEORY:**

Part of Speech (hereby referred to as POS) Tags are useful for building parse trees, which are used in building NERs (most named entities are Nouns) and extracting relations between words. POS Tagging is also essential for building lemmatizers which are used to reduce a word to its root form.

POS tagging is the process of marking up a word in a corpus to a corresponding part of a speech tag, based on its context and definition. This task is not straightforward, as a particular word may have a different part of speech based on the context in which the word is used.

**For example:** In the sentence “Give me your answer”, *answer* is a Noun, but in the sentence “Answer the question”, *answer* is a verb.

To understand the meaning of any sentence or to extract relationships and build a knowledge graph, POS Tagging is a very important step.

The Different POS Tagging Techniques

There are different techniques for POS Tagging:

* **Lexical Based Methods —** Assigns the POS tag the most frequently occurring with a word in the training corpus.
* **Rule-Based Methods —** Assigns POS tags based on rules. For example, we can have a rule that says, words ending with “ed” or “ing” must be assigned to a verb. Rule-Based Techniques can be used along with Lexical Based approaches to allow POS Tagging of words that are not present in the training corpus but are there in the testing data.
* **Probabilistic Methods —** This method assigns the POS tags based on the probability of a particular tag sequence occurring. Conditional Random Fields (CRFs) and Hidden Markov Models (HMMs) are probabilistic approaches to assign a POS Tag.
* **Deep Learning Methods** — Recurrent Neural Networks can also be used for POS tagging.

**Steps Involved:**

Tokenize text (word\_tokenize)

apply pos\_tag to above step that is nltk.pos\_tag(tokenize\_text)

**Some examples are as below:**

| **Abbreviation** | **Meaning** |
| --- | --- |
| CC | coordinating conjunction |
| CD | cardinal digit |
| DT | determiner |
| EX | existential there |
| FW | foreign word |
| IN | preposition/subordinating conjunction |
| JJ | adjective (large) |
| JJR | adjective, comparative (larger) |
| JJS | adjective, superlative (largest) |
| LS | list market |
| MD | modal (could, will) |
| NN | noun, singular (cat, tree) |
| NNS | noun plural (desks) |
| NNP | proper noun, singular (sarah) |
| NNPS | proper noun, plural (indians or americans) |

**CONCLUSION:**

Part of speech tagging is the process of assigning a word in a corpus word class. Parts of speech tagging have numerous uses such as in Named Entity Recognition. Parts of Speech tagging has been carefully studied and implemented on a text corpus.

**CODE:**

from random import choice

from nltk import pos\_tag

from nltk import word\_tokenize

from nltk.corpus import brown

samples = choice(brown.paras(categories="adventure"))

corpus = " ".join([" ".join(sample) for sample in samples])

print(f"Original corpus :\n{corpus}\n")

tokens = word\_tokenize(corpus)

print(f"Tokenized words : \n{tokens}\n")

tagged\_words = pos\_tag(tokens)

print(f"POS tagged words : \n{tagged\_words}")

**OUTPUT:**

