Name : Sarvagya Singh **SAPID : 600009200030** **BATCH : K1**

Experiment No 8

Aim: Implement a program to Identify the word senses using "synset" in NLTK

Theory:

Wordnet?

Wordnet is a large lexical database of English, which was created by Princeton. It is a part of the NLTK corpus. Nouns, verbs, adjectives, and adverbs all are grouped into set of synsets, i.e., cognitive synonyms. Here each set of synsets express a distinct meaning. Following are some use cases of Wordnet −

- It can be used to look up the definition of a word.
- We can find synonyms and antonyms of a word.
- Word relations and similarities can be explored using Wordnet.
- Word sense disambiguation for those words having multiple uses and definitions.

Synset is a special kind of a simple interface that is present in NLTK to look up words in WordNet. Synset instances are the groupings of synonymous words that express the same concept. Some of the words have only one Synset and some have several.

wordnet. synsets(word) can be used to get a list of Synsets. This list can be empty (if no such word is found) or can have few elements.
Hypernyms and Hyponyms –
Hypernyms: More abstract terms
Hyponyms: More specific terms.
Both come to picture as Synsets are organized in a structure like that of an inheritance tree. This tree can be traced all the way up to a root hypernym. Hypernyms provide a way to categorize and group words based on their similarity to each other.

Applications

There are several subfields in natural language processing which can benefit from having a large lexical database, especially one as big and extensive as WordNet. Many semantic applications can draw benefits from using WordNet, including Word Sense Disambiguation (WSD), question answering and sentiment analysis. Many papers have been published regarding WordNet and WSD, exploring different approaches and algorithms, which is the main field for using this.

<u>Word Sense Disambiguation</u>

One of the first things you realize when working with linguistic or textual data is just how ambiguous the words are. Language is very contextual, and the meanings of the words depend upon the context in which you are using it.

To give a hint how all this works, consider three examples of the distinct senses that exist for the word "*bass*":

1. a type of fish
2. tones of low frequency
3. a type of instrument

and the sentences:

1. *I went fishing for some sea bass.*
2. *The bass line of the song is too weak.*

To a human, it is obvious that the first sentence is using the word "*bass (fish)*", as in the former sense above and in the second sentence, the word "*bass (instrument)*" is being used as in the latter sense below. Developing algorithms to replicate this human ability can often be a difficult task,

Nevertheless, we have solved several exceedingly difficult problems to a reasonable degree of accuracy with computational approaches. Let's talk about one of the naiver approaches to word sense disambiguation, which actually does fairly well when given a reasonably large input.

But first, what’s this Word Sense Disambiguation all about. Well, the sense of a word is a way of identifying how we use a given word by associating its definition. This is where the ambiguity problem comes in, how does a computer know, how to treat a given input when each one has a number of different senses, some of which have wildly different connotations and usages? This one problem is a key building block for all sorts of more complex, interesting NLP systems, from sentiment analysis to machine translation.

Lab assignment to be performed in this session:

Import WordNet form NLTK corpus reader.

Look up a word using Synsets ().

Look up a definition of word found in Synsets.

Look up for the examples of the words found in Synsets.

Find a set of synonyms that share a common meaning.

Find a set of antonyms that does not share a common meaning.

Find a set of hyponyms and hypernyms of a word..

CL-LAB-8

Name: Sarvagya Singh

SAP: 60009200030

Div/Batch: K/K1

SYSNET IN NLTK

In [ ]:

```
import nltk
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
```

Out[ ]:

```
True
```

In [ ]:

```
from nltk.corpus import wordnet
```

In [ ]:

```
syn = wordnet.synsets('hello')[0]

print ("Synset name :  ", syn.name())

# Defining the word
print ("\nSynset meaning : ", syn.definition())

# list of phrases that use the word in context
print ("\nSynset example : ", syn.examples())
```

```
Synset name :   hello.n.01

Synset meaning :  an expression of greeting

Synset example :  ['every morning they exchanged polite hellos']
```

In [ ]:

```
print ("Synset name :  ", syn.name())

print ("\nSynset abstract term :  ", syn.hypernyms())

print ("\nSynset specific term :  ",
       syn.hypernyms()[0].hyponyms())

syn.root_hypernyms()

print ("\nSynset root hypernerm :  ", syn.root_hypernyms())
```

```
Synset name :   hello.n.01

Synset abstract term :   [Synset('greeting.n.01')]

Synset specific term :   [Synset('calling_card.n.02'), Synset('good_afternoon.n.01'), Syn
set('good_morning.n.01'), Synset('hail.n.03'), Synset('hello.n.01'), Synset('pax.n.01'),
Synset('reception.n.01'), Synset('regard.n.03'), Synset('salute.n.02'), Synset('salute.n.
03'), Synset('welcome.n.02'), Synset('well-wishing.n.01')]

Synset root hypernerm :   [Synset('entity.n.01')]
```

In [ ]:

```
syn = wordnet.synsets('hello')[0]
print ("Syn tag : ", syn.pos())

syn = wordnet.synsets('doing')[0]
print ("Syn tag : ", syn.pos())

syn = wordnet.synsets('beautiful')[0]
print ("Syn tag : ", syn.pos())

syn = wordnet.synsets('quickly')[0]
print ("Syn tag : ", syn.pos())
```

```
Syn tag :  n
Syn tag :  v
Syn tag :  a
Syn tag :  r
```

In [ ]: