

**DEPARTMENT OF COMPUTER ENGINEERING**

## CSL804 Computational Lab II

**Eighth Semester, 2021-2022 (Even Semester)**

**Name of Student :** Saurav Kumar

## Roll No. 23

**Division :** BE – CMPN

**Day/ Session :** Monday/Afternoon

**Venue :** SLRTCE Lab 305

## Experiment No. 8

**Title of Experiment :** To study and implement the concept of WordNet

## Date of Conduction :

**Date of Submission :**

|  |  |  |
| --- | --- | --- |
| **Particulars Max. Marks Marks Obtained** | | |
| Preparedness and Efforts(PE) | **3** |  |
| Knowledge of tools(KT) | **3** |  |
| Debugging and results(DR) | **3** |  |
| Documentation(DN) | **3** |  |
| Punctuality & Lab Ethics(PL) | **3** |  |
| **Total** | **15** |  |

**Grades – Meet Expectations (3 Marks), Moderate Expectations (2 Marks), Below Expectations (1 Mark)**

**Checked and Verified by Name of Faculty :** Prof. Neelam Kulkarni

## Signature :

**Date :**

EXPERIMENT NO: 8

WORDNET

**AIM:** To study and implement the concept of WordNet.

**SOFTWARE:** Python, NLTK,

# THEORY:

WordNet is an on-line lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, and adjectives are organised into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets.

# IMPLEMENTATION:

WORDNET CODE

from nltk.corpus import wordnet

print("Finding all the synonym set (SynSet) of a word for all possible POS tags" )

synsets = wordnet.synsets('dog') print(synsets)

print("Finding the synset if POS tag is known and number of senses") synset = wordnet.synset('dog.n.01')

print("Printing the definition(gloss) for the word 'dog'") print(synset.definition)

print("find the hypernym of a synset") dog = wordnet.synset('dog.n.01') print(dog.hypernyms())

print("find the hyponyms of a synset") print(dog.hyponyms())

print("Find commmon hypernyms between two words") dog = wordnet.synset('dog.n.01')

cat = wordnet.synset('cat.n.01')

print(dog.lowest\_common\_hypernyms(cat)) #Finding similarities between words

dog = wordnet.synset('dog.n.01') cat = wordnet.synset('cat.n.01') tree = wordnet.synset('tree.n.01')

print("Path similarity:Return a score denoting how similar two word senses are, based on the shortest path that connects the senses in the is-a (hypernym/hypnoym) taxonomy. The score is in the range 0 to 1")

print(wordnet.path\_similarity(dog,cat)) print(wordnet.path\_similarity(dog,tree))

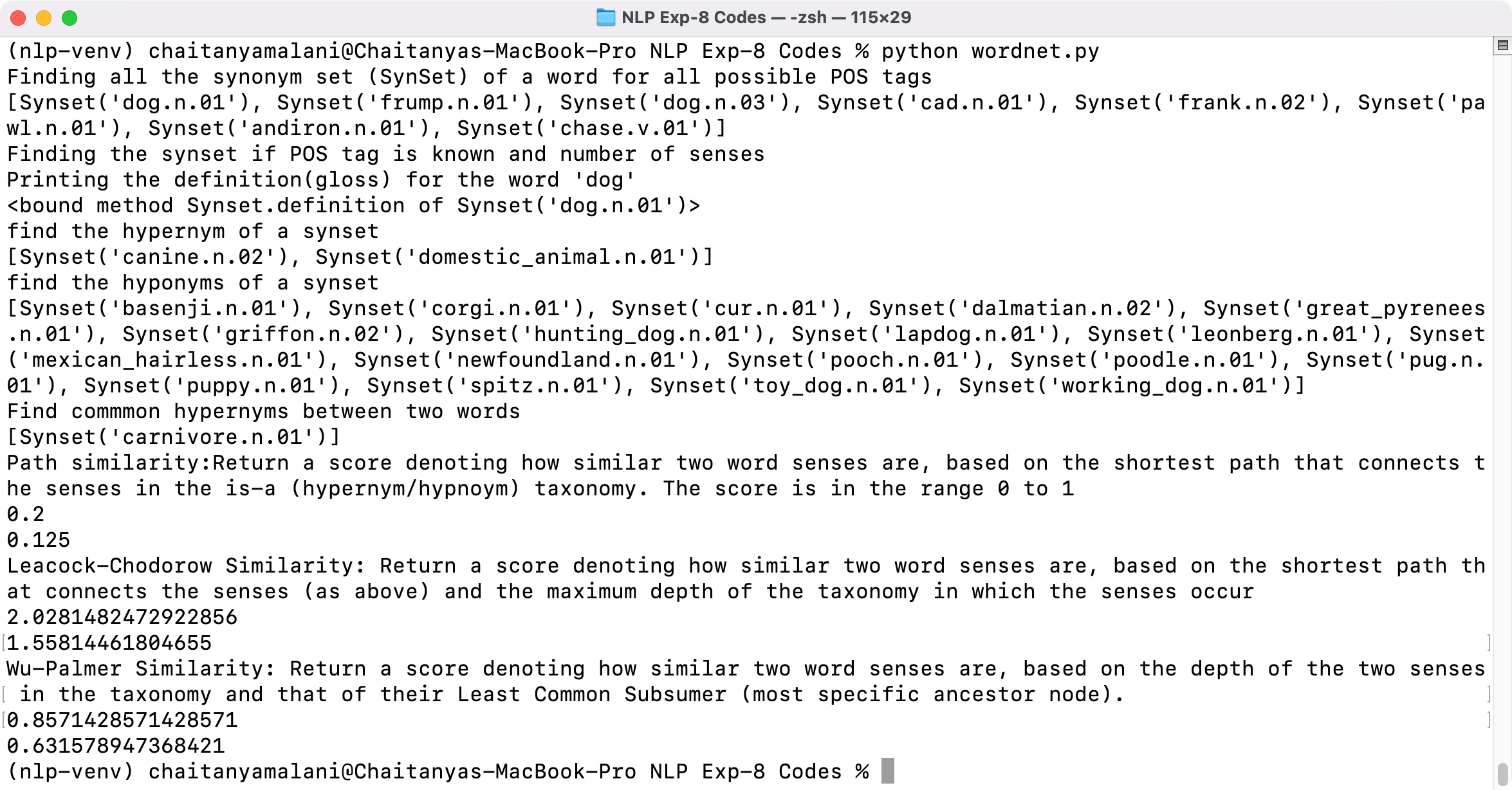
print("Leacock-Chodorow Similarity: Return a score denoting how similar two word senses are, based on the shortest path that connects the senses (as above) and the maximum depth of the taxonomy in which the senses occur")

print(wordnet.lch\_similarity(dog,cat)) print(wordnet.lch\_similarity(dog,tree))

print("Wu-Palmer Similarity: Return a score denoting how similar two word senses are, based on the depth of the two senses in the taxonomy and that of their Least Common Subsumer (most specific ancestor node).")

print(wordnet.wup\_similarity(dog,cat)) print(wordnet.wup\_similarity(dog,tree))

OUTPUT



# CONCLUSION:

Thus we have studied and implemented the concept of WordNet.