Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science

AY: 2025-26

Class:	BE	**Semester:**	VII
Course Code:	CSDOL7011	**Course Name:**	Natural Language Processing

Name of Student:	Konisha Jayesh Thakare
Roll No. :	71
Experiment No.:	7
Title of the Experiment:	Calculating Semantic Similarity and Performing Word Sense Disambiguation using WordNet
Date of Performance:	26.08.2025
Date of Submission:	02.09.2025

Evaluation

Performance Indicator	**Max. Marks**	**Marks Obtained**
Performance	5	
Understanding	5	
Journal work and timely submission	10	
Total	20	

Performance Indicator	**Exceed Expectations (EE)**	**Meet Expectations (ME)**	**Below Expectations(BE)**
Performance	4-5	2-3	1
Understanding	4-5	2-3	1
Journal work and timely submission	8-10	5-8	1-4

Checked by

Name of Faculty : Dr. Tatwadarshi P. Nagarhalli

Signature :

Date :

Aim: To calculate semantic similarity between words and perform Word Sense Disambiguation (WSD) using the WordNet lexical database in NLTK.

Theory:

WordNet is a large lexical database of English where nouns, verbs, adjectives, and adverbs are grouped into synsets (sets of synonyms) representing a concept. It provides relationships such as hypernymy (generalization), hyponymy (specialization), and meronymy (part-whole relationships), which are useful for semantic analysis. Semantic Similarity measures how closely two words or concepts are related in meaning. Common WordNet-based similarity measures include:

Path Similarity: Based on the shortest path connecting synsets in the hierarchy.

Wu-Palmer Similarity: Considers depth of synsets and their Least Common Subsumer (LCS).

Word Sense Disambiguation (WSD) is the task of determining the correct meaning of a word in a given context. Since many words have multiple meanings (polysemy), WSD is crucial for applications like machine translation, information retrieval, and question answering.

NLTK provides access to WordNet and functions for computing similarity and performing simple WSD.

Procedure:

1. Select words or sentences for semantic analysis.
2. Import WordNet from NLTK.
3. Retrieve synsets for the words.
4. Calculate semantic similarity between selected synsets using path-based or Wu-Palmer measures.
5. Perform Word Sense Disambiguation to choose the appropriate synset based on context.
6. Interpret similarity scores and disambiguated meanings.

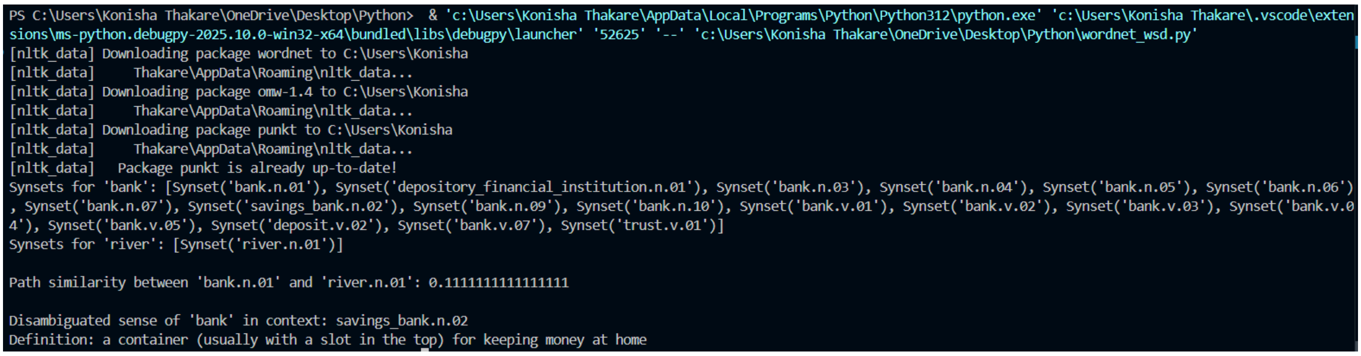
Algorithm:

1. Import WordNet from NLTK.
2. Retrieve synsets for the words under consideration.
3. Compute pairwise semantic similarity using path_similarity or wup_similarity.
4. For WSD, select the synset with the highest similarity to context words.
5. Display results and interpret the semantic relationships.

Implementation:

```
import nltk
from nltk.corpus import wordnet as wn
from nltk.wsd import lesk
nltk.download('wordnet')
nltk.download('omw-1.4')
nltk.download('punkt')
word1='bank'
word2='river'
synsets1=wn.synsets(word1)
synsets2=wn.synsets(word2)
print(f"Synsets for '{word1}': {synsets1}")
print(f"Synsets for '{word2}': {synsets2}")
similarity=synsets1[0].path_similarity(synsets2[0])
print(f"\nPath similarity between '{synsets1[0].name()}' and '{synsets2[0].name()}': {similarity}")
sentence="He went to the bank to deposit money."
wsd_result=lesk(nltk.word_tokenize(sentence),'bank')
print(f"\nDisambiguated sense of 'bank' in context: {wsd_result.name()}")
print(f"Definition: {wsd_result.definition()}")
```

Output:



```
PS C:\Users\Konisha Thakare\OneDrive\Desktop\Python>  & 'c:\Users\Konisha Thakare\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\Konisha Thakare\.vscode\exten
sions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '52625' '--' 'c:\Users\Konisha Thakare\OneDrive\Desktop\Python\wordnet_wsd.py'
[nltk_data] Downloading package wordnet to C:\Users\Konisha
[nltk_data]     Thakare\AppData\Roaming\nltk_data...
[nltk_data] Downloading package omw-1.4 to C:\Users\Konisha
[nltk_data]     Thakare\AppData\Roaming\nltk_data...
[nltk_data] Downloading package punkt to C:\Users\Konisha
[nltk_data]     Thakare\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
Synsets for 'bank': [Synset('bank.n.01'), Synset('depository_financial_institution.n.01'), Synset('bank.n.03'), Synset('bank.n.04'), Synset('bank.n.05'), Synset('bank.n.06')
, Synset('bank.n.07'), Synset('savings_bank.n.02'), Synset('bank.n.09'), Synset('bank.n.10'), Synset('bank.v.01'), Synset('bank.v.02'), Synset('bank.v.03'), Synset('bank.v.0
4'), Synset('bank.v.05'), Synset('deposit.v.02'), Synset('bank.v.07'), Synset('trust.v.01')]
Synsets for 'river': [Synset('river.n.01')]

Path similarity between 'bank.n.01' and 'river.n.01': 0.1111111111111111

Disambiguated sense of 'bank' in context: savings_bank.n.02
Definition: a container (usually with a slot in the top) for keeping money at home
```

Conclusion:

This experiment demonstrated the calculation of semantic similarity and Word Sense Disambiguation using WordNet in NLTK. Semantic similarity measures provide a numerical estimate of how closely related two words or concepts are based on their positions in the WordNet hierarchy. Word Sense Disambiguation identifies the appropriate meaning of a polysemous word in context, ensuring accurate understanding of text. Together, these techniques are foundational for applications such as machine translation, information retrieval, question answering, and semantic search, enabling more intelligent and context-aware natural language processing.