MSC AI
SEM II, IV – NLP
UNIT III

Perquisites for Sematic Analysis

Introduction to Semantic Analysis

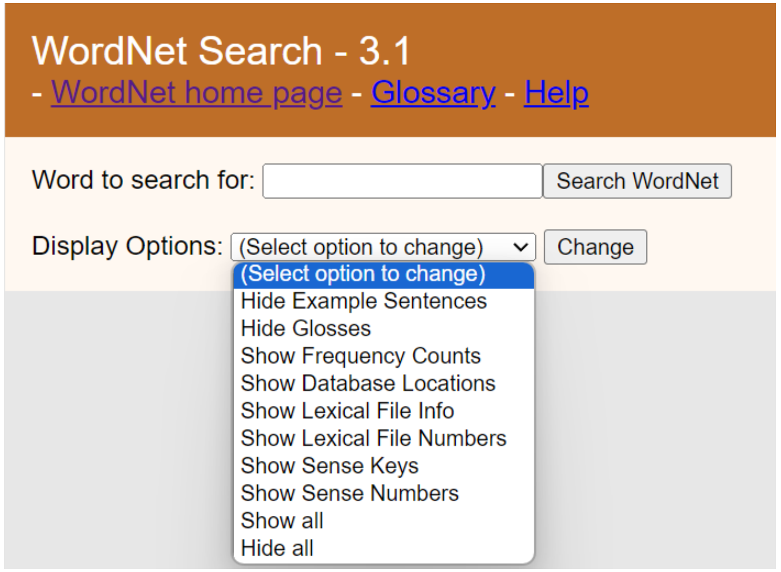
- Semantic Analysis is the process of understanding natural language text at the semantic level.
- Its objective is to extract meaningful information from text.
- It enhances text understanding for applications like translation, summarization, and sentiment analysis.
- Its applications in NLP include Text classification, information retrieval, word sense disambiguation.
- It has 5 key Components:

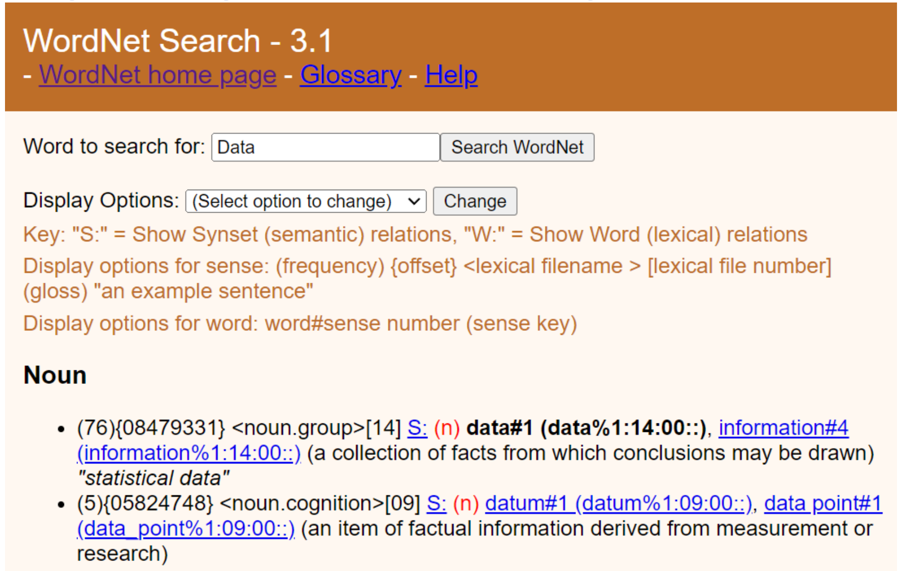
1. WordNet and Synsets,
2. Lexical Semantic Relations,
3. Word Sense Disambiguation,
4. Named Entity Recognition,
5. Semantic Representations: Propositional Logic, First Order Logic

1. WordNet and Synsets

What is WordNet?

- It is a huge lexical database of English words.
- This database was created around 1985, is a part of Princeton University's Cognitive Science Laboratory guided by Professor G. A. Miller, available at https:// wordnet. princeton.edu (https://shorturl.at/hCi7a)
- This lexical database consists of nouns, adjective, verbs, and adverbs, and related lexical terms, which are grouped together based on some common concepts into sets, known as **cognitive synonym sets or synsets .**





Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (frequency) {offset} <lexical filename > [lexical file number] (gloss) "an example sentence"

Display options for word: word#sense number (sense key)

Noun

- (76){08479331} <noun.group>[14] S: (n) **data#1 (data%1:14:00::)**, information#4 (information%1:14:00::) (a collection of facts from which conclusions may be drawn) *"statistical data"*
- (5){05824748} <noun.cognition>[09] S: (n) datum#1 (datum%1:09:00::), data point#1 (data_point%1:09:00::) (an item of factual information derived from measurement or research)

1. WordNet and Synsets

The WordNet database consists of

- over 155,000 words,
- represented in more than 117,000 synsets,

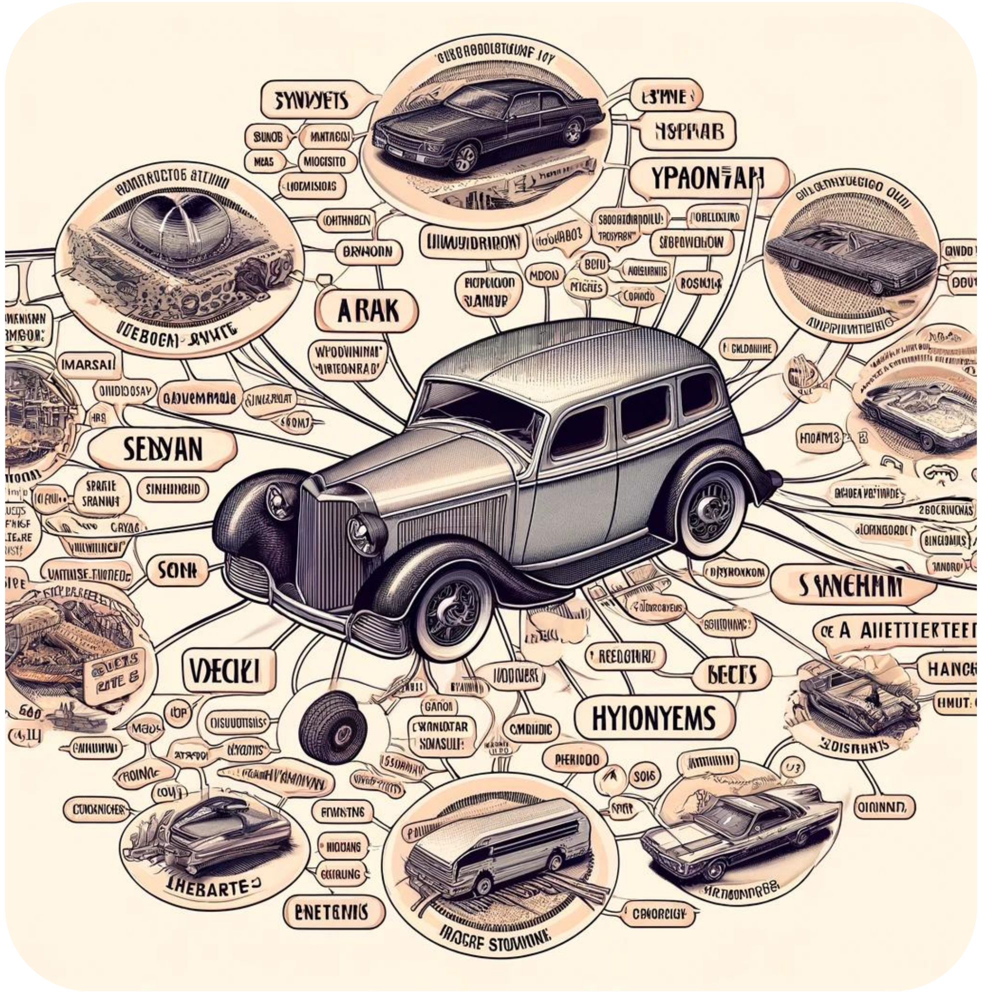
and

- contains over 206,000 word-sense pairs.

It is roughly 12 MB in size and can be accessed through various interfaces and APIs.

- The official web site has a web application interface.

It can be accessed at http://wordnetweb.princeton.edu/perl/webwn or download it from https://wordnet.princeton.edu/wordnet/download/



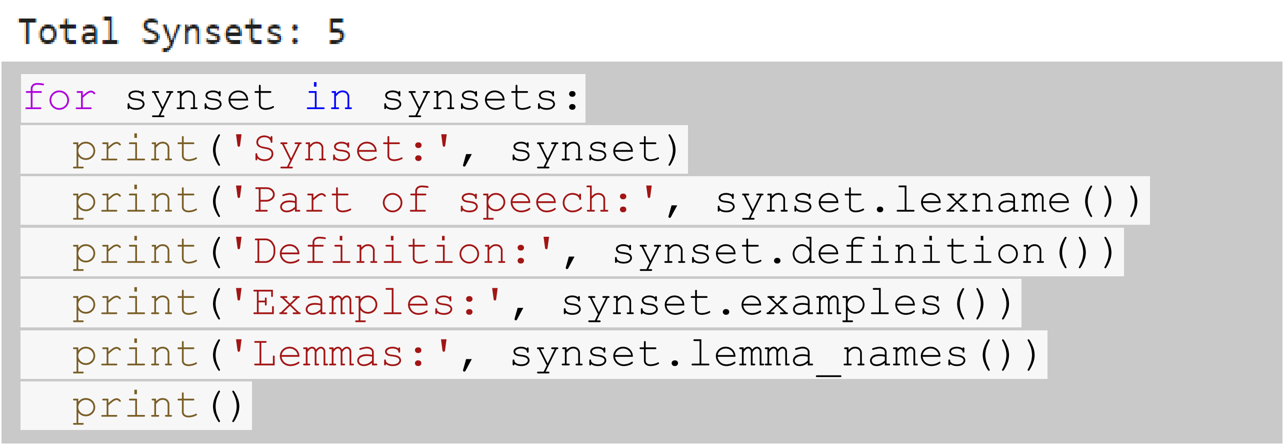
The Structure of WordNet:

- Synsets: Sets of cognitive synonyms expressing distinct concepts.
- Lemmas: Base forms of words.
- Relations: Hypernyms, hyponyms, meronyms, antonyms.
- Applications in NLP: Text classification, information retrieval, word sense disambiguation.
- Image: Diagram of WordNet structure showing synsets and their relations.

1. WordNet and Synsets

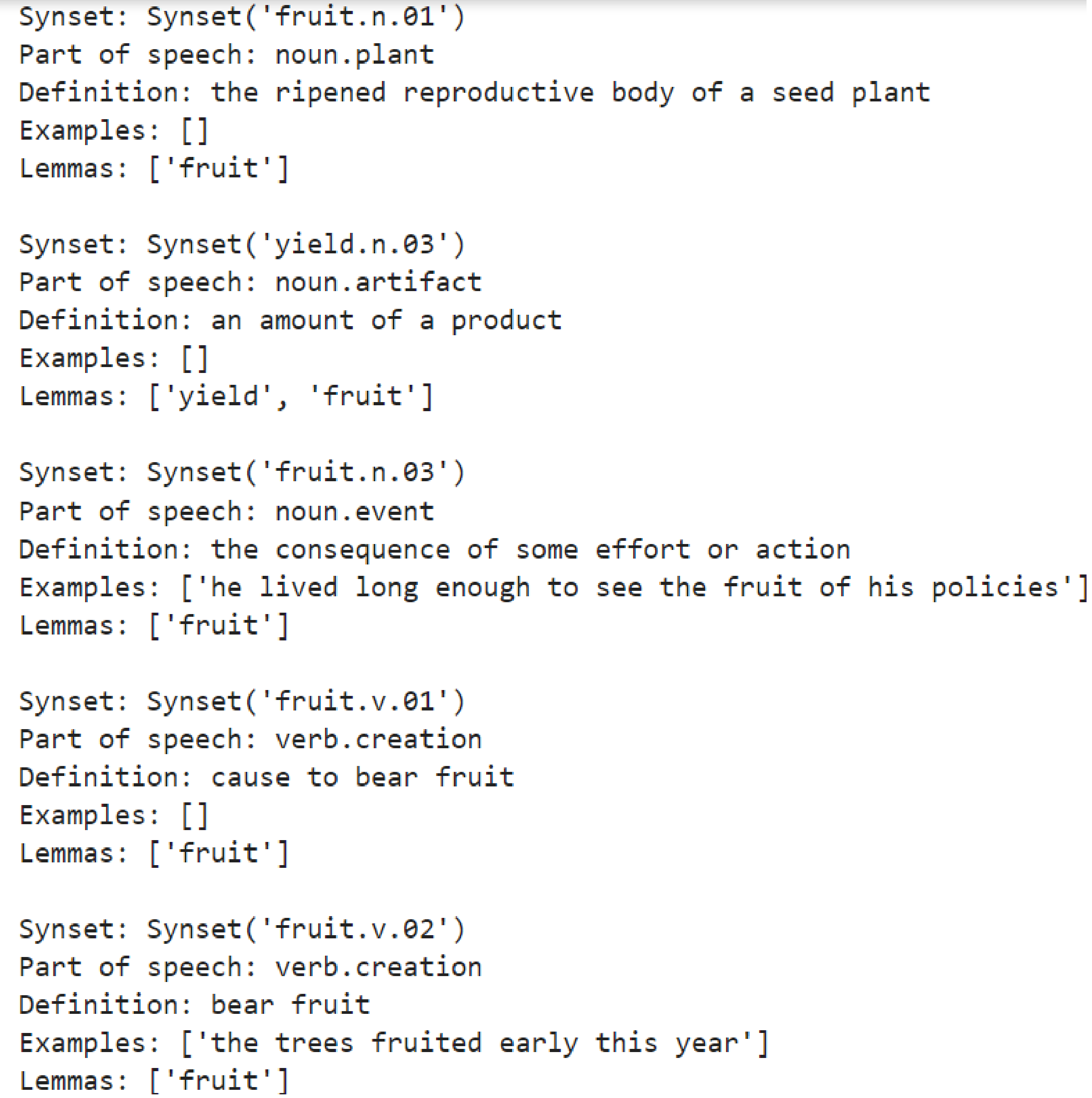
The output shows the details of

- each synset associated with the term 'fruit' ,
- the definitions give us the sense of each synset
- the lemma associated with it.
- The part of speech for each synset is also mentioned, which includes nouns and verbs.
- examples that show how the term is used in actual sentences.



```
Total Synsets: 5
```

```python
for synset in synsets:
  print('Synset:', synset)
  print('Part of speech:', synset.lexname())
  print('Definition:', synset.definition())
  print('Examples:', synset.examples())
  print('Lemmas:', synset.lemma_names())
  print()
```



```
Synset: Synset('fruit.n.01')
Part of speech: noun.plant
Definition: the ripened reproductive body of a seed plant
Examples: []
Lemmas: ['fruit']

Synset: Synset('yield.n.03')
Part of speech: noun.artifact
Definition: an amount of a product
Examples: []
Lemmas: ['yield', 'fruit']

Synset: Synset('fruit.n.03')
Part of speech: noun.event
Definition: the consequence of some effort or action
Examples: ['he lived long enough to see the fruit of his policies']
Lemmas: ['fruit']

Synset: Synset('fruit.v.01')
Part of speech: verb.creation
Definition: cause to bear fruit
Examples: []
Lemmas: ['fruit']

Synset: Synset('fruit.v.02')
Part of speech: verb.creation
Definition: bear fruit
Examples: ['the trees fruited early this year']
Lemmas: ['fruit']
```

1. Understanding Synsets

Synsets are the Sets of synonyms that share a common meaning. Examples of Synsets in WordNet:

① ② ③ ④ ⑤

Synset for "car": {car, auto, automobile, machine, motorcar}.

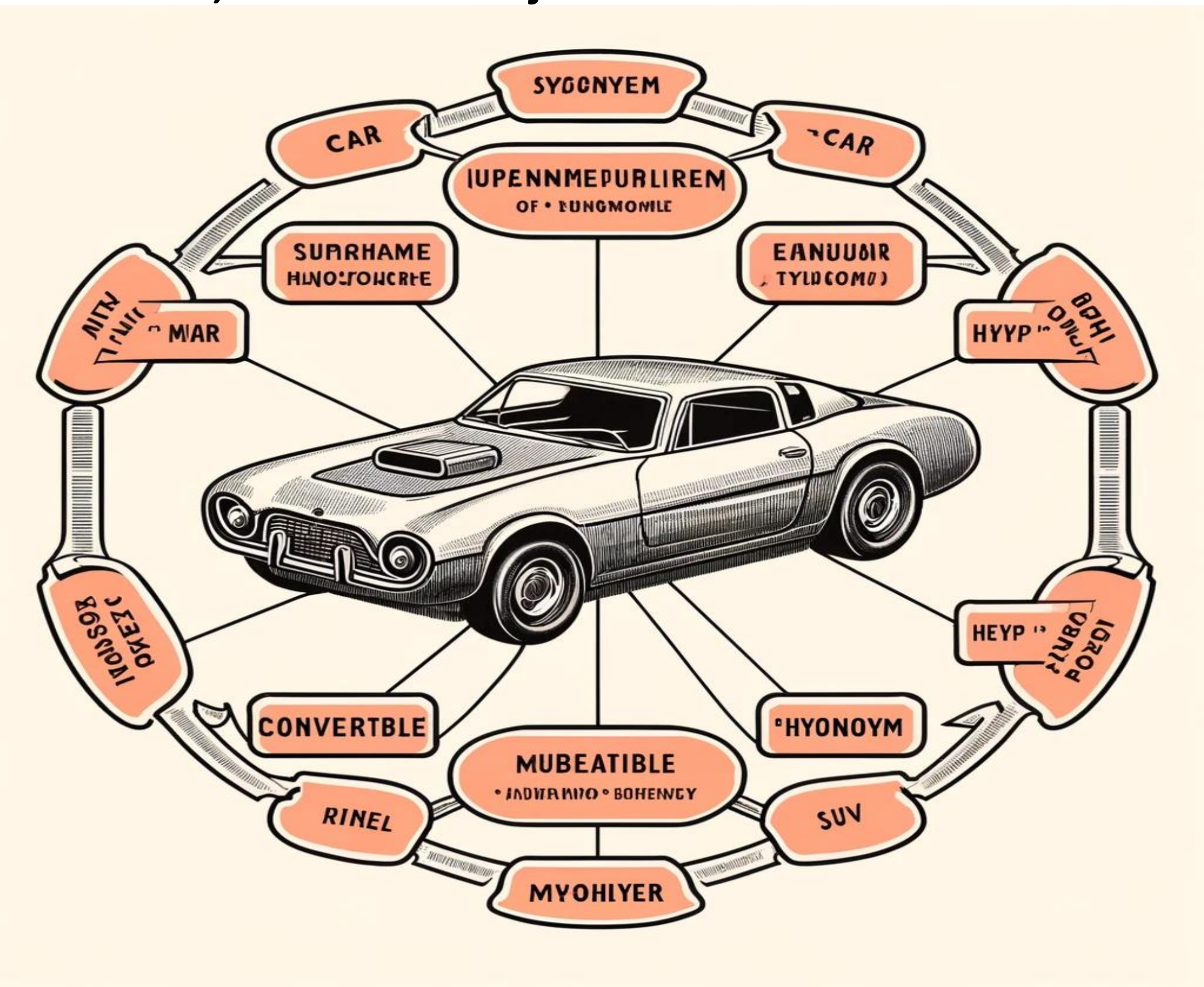
Synset Relations:

Hypernyms: More general terms (e.g., "vehicle" for "car").

Hyponyms: More specific terms (e.g., "sedan" for "car").

Meronyms: Part-whole relations (e.g., "wheel" for "car")

Figure shows a central synset with words such as 'car,' 'automobile,' 'auto,' and 'motorcar,' and illustrates the hypernym, hyponym, and meronym.



2. Lexical Semantic Relations

Text semantics refers to the study of meaning and context.

In Semantic Analysis lexical relations help in understanding context and relationships between words.

Examples and Applications of lexical relations include – Enhancing search engines, text summarization, and information extraction.

We know that Synsets give a nice abstraction over various terms and provide useful information like definition, examples, POS, and lemmas.

Semantic relationships among entities can be explored using synsets.

Types of Lexical Relations:

Synonymy: Words with similar meanings (e.g., "big" and "large").

Antonymy: Words with opposite meanings (e.g., "hot" and "cold").

Hyponymy: Words that are subtypes of a more general term (e.g., "rose" is a hyponym of "flower").

Meronymy: Words that denote a part of something (e.g., "wheel" is a meronym of "car").

2.1 Components Lexical Semantic Relations

(i) Entailment of Synsets

- refers to some event or action that logically involves or is associated with some other action or event that has taken place or will take place.
- Ideally this **applies very well to verbs** indicating some specific action.

Consider the code:

```
for action in ['walk', 'eat', 'digest’]:
    action_syn = wn.synsets(actionIt , pos='v')[0]
    print(action_syn, '-- entails -->', action_syn.entailments())
```

It gives the output

```
Synset('walk.v.01') -- entails --> [Synset('step.v.01')]
Synset('eat.v.01') -- entails --> [Synset('chew.v.01'),
Synset('swallow.v.01')]
Synset('digest.v.01') -- entails --> [Synset('consume.v.02’)]
```

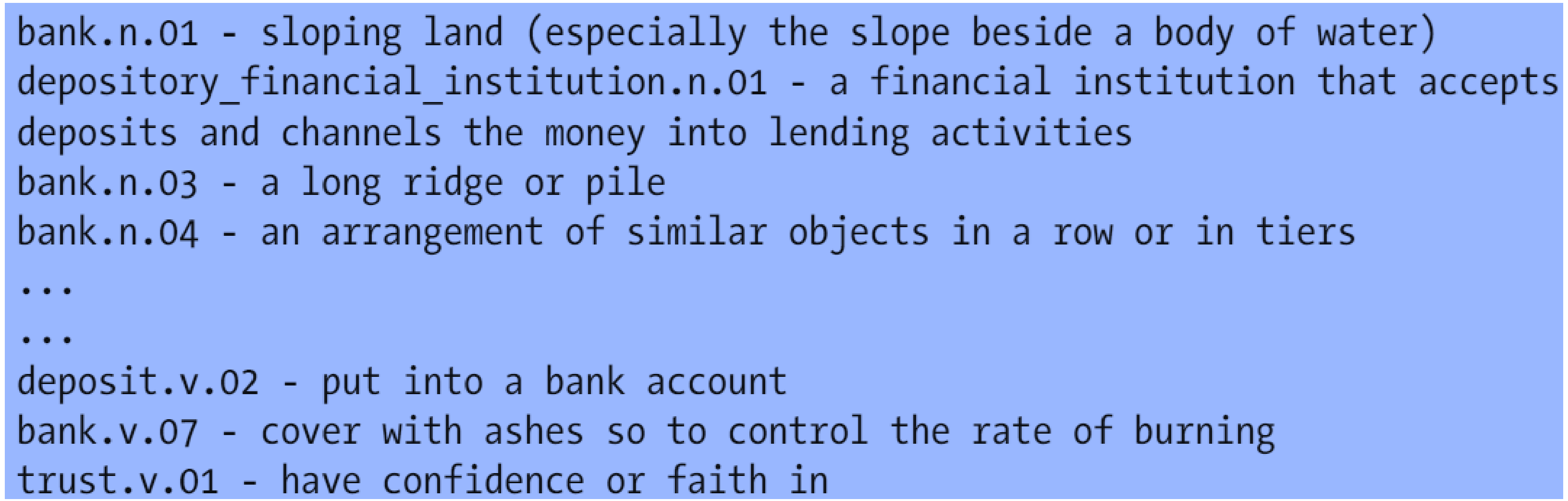
It entails the related terms like *walking* entail *stepping* , and *eating* entails *chewing* and *swallowing*.

2.1 Components Lexical Semantic Relations

(ii) Homonyms and Homographs

- Homographs are words with same spelling but may have different pronunciation and meaning.
- Homonyms refer to words or terms having the same written form or pronunciation but different meanings. E.g. read(present, past), part(noun, verb). Homonyms are a superset of homographs.

For example:



```
bank.n.01 - sloping land (especially the slope beside a body of water)
depository_financial_institution.n.01 - a financial institution that accepts
deposits and channels the money into lending activities
bank.n.03 - a long ridge or pile
bank.n.04 - an arrangement of similar objects in a row or in tiers
...
...
deposit.v.02 - put into a bank account
bank.v.07 - cover with ashes so to control the rate of burning
trust.v.01 - have confidence or faith in
```

output shows a part of the result obtained for the various homographs for the term 'bank’.

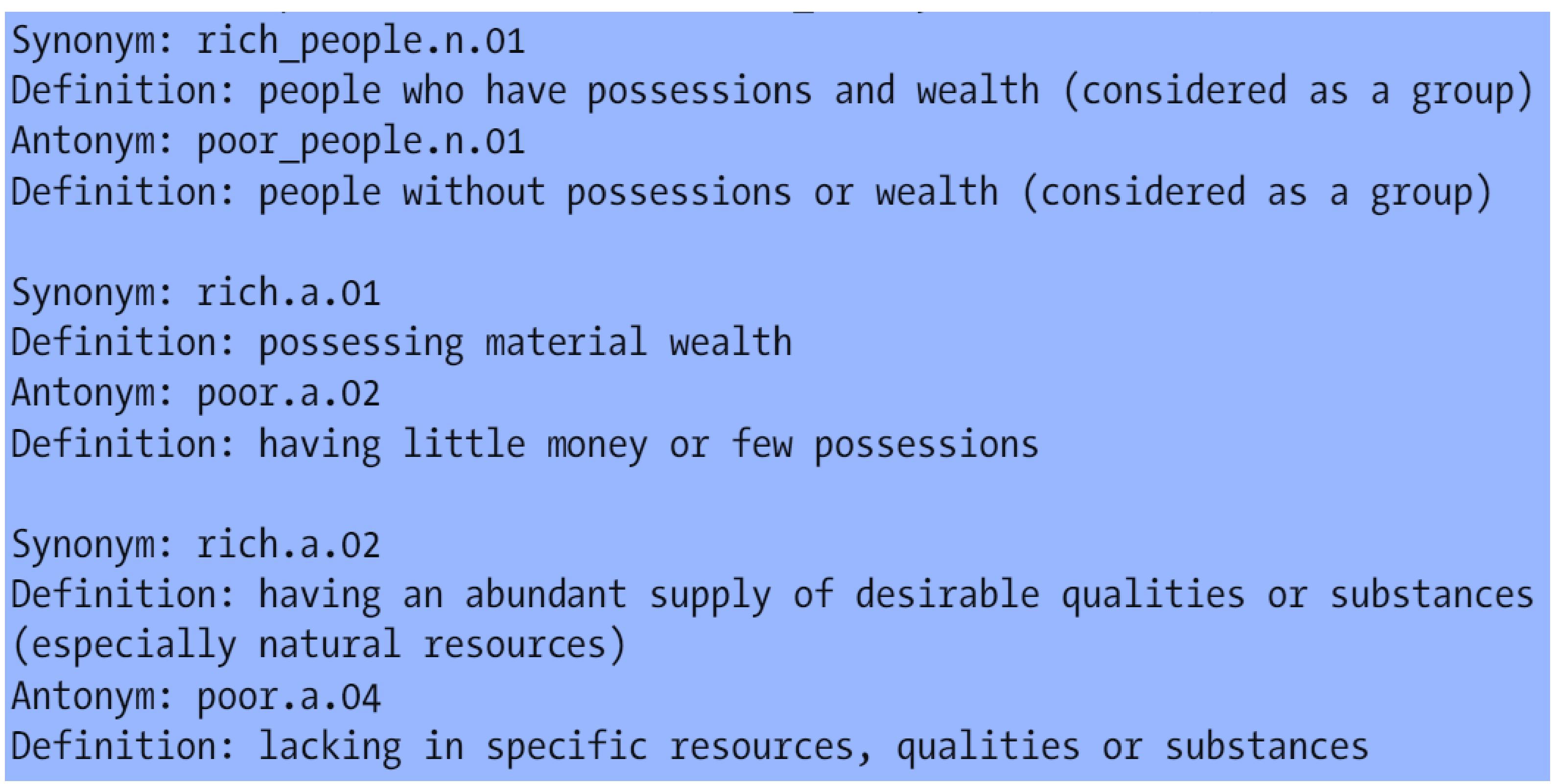
You can see that there are various different meanings associated with the word 'bank’

2.1 Components Lexical Semantic Relations

(iii) Synonyms and Antonyms

- Synonyms are words having similar meaning and context, and antonyms are words having opposite or contrasting meaning, as you may know already.

For example:



```
Synonym: rich_people.n.01
Definition: people who have possessions and wealth (considered as a group)
Antonym: poor_people.n.01
Definition: people without possessions or wealth (considered as a group)

Synonym: rich.a.01
Definition: possessing material wealth
Antonym: poor.a.02
Definition: having little money or few possessions

Synonym: rich.a.02
Definition: having an abundant supply of desirable qualities or substances
(especially natural resources)
Antonym: poor.a.04
Definition: lacking in specific resources, qualities or substances
```

outputs show sample synonyms and antonyms for the term 'large' and the term 'rich' . Additionally, we explore several synsets associated with the term or concept 'rich' , which rightly give us distinct synonyms and their corresponding antonyms.

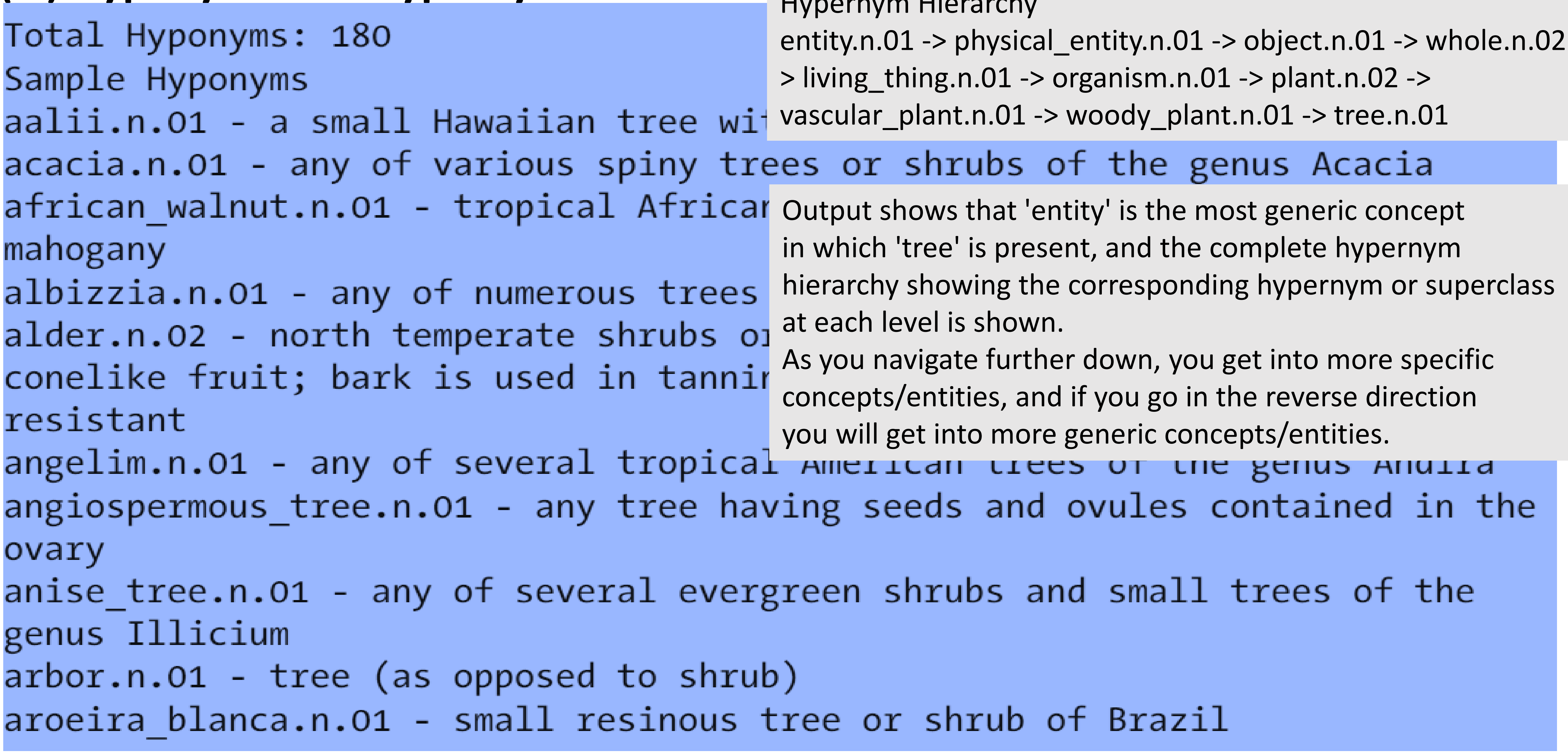
2.1 Components Lexical Semantic Relations

(iv) Hyponyms and Hypernyms

- Synsets represent terms with unique semantics and concepts and are linked or related to each other based on some similarity and context.
- Several of these Synsets represent abstract and generic concepts also besides concrete entities.
- Usually they are interlinked together in the form of a hierarchical structure representing is-a relationships.
- Hyponyms and hypernyms help us explore related concepts by navigating through this hierarchy.
- To be more specific, hyponyms refer to entities or concepts that are a subclass of a
  - higher order concept or entity and
  - have very specific sense or context compared to its superclass.

2.1 Components Lexical Semantic Relations

(iv) Hyponyms and Hypernyms



```
Total Hyponyms: 180
Sample Hyponyms
aalii.n.01 - a small Hawaiian tree wi
acacia.n.01 - any of various spiny trees or shrubs of the genus Acacia
african_walnut.n.01 - tropical Africar
mahogany
albizzia.n.01 - any of numerous trees
alder.n.02 - north temperate shrubs or
conelike fruit; bark is used in tannir
resistant
angelim.n.01 - any of several tropical American trees of the genus Andira
angiospermous_tree.n.01 - any tree having seeds and ovules contained in the
ovary
anise_tree.n.01 - any of several evergreen shrubs and small trees of the
genus Illicium
arbor.n.01 - tree (as opposed to shrub)
aroeira_blanca.n.01 - small resinous tree or shrub of Brazil
```

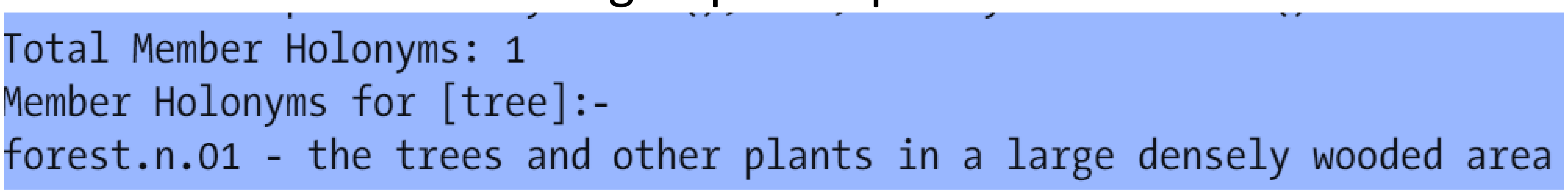
Hypernym Hierarchy
entity.n.01 -> physical_entity.n.01 -> object.n.01 -> whole.n.02 -> living_thing.n.01 -> organism.n.01 -> plant.n.02 -> vascular_plant.n.01 -> woody_plant.n.01 -> tree.n.01

Output shows that 'entity' is the most generic concept in which 'tree' is present, and the complete hypernym hierarchy showing the corresponding hypernym or superclass at each level is shown.
As you navigate further down, you get into more specific concepts/entities, and if you go in the reverse direction you will get into more generic concepts/entities.

2.1 Components Lexical Semantic Relations

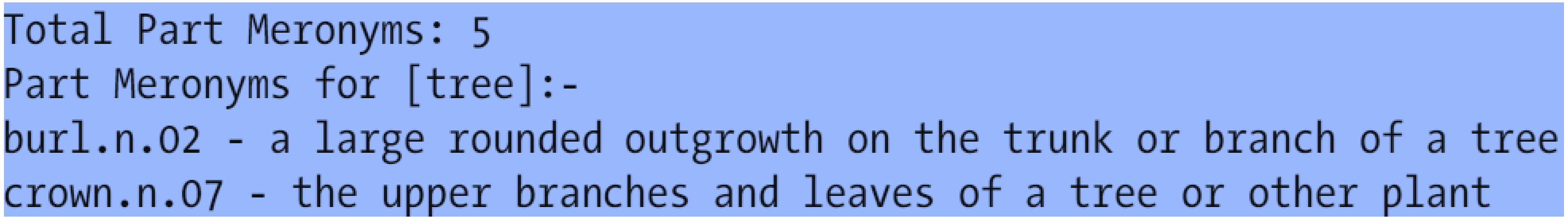
(v) Holonyms and Meronyms

- Holonyms are entities that contain a specific entity of our interest. Basically holonym refers to the relationship between a term or entity that denotes the whole and a term denoting a specific part of the whole.



```
Total Member Holonyms: 1
Member Holonyms for [tree]:-
forest.n.01 - the trees and other plants in a large densely wooded area
```

- From the output, we can see that 'forest' is a holonym for 'tree' , which is semantically correct because, of course, a forest is a collection of trees.
- Meronyms are semantic relationships that relate a term or entity as a part or constituent of another term or entity.

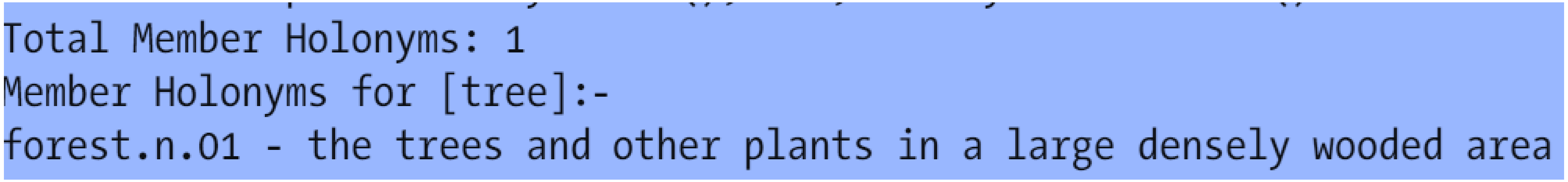


```
Total Part Meronyms: 5
Part Meronyms for [tree]:-
burl.n.02 - a large rounded outgrowth on the trunk or branch of a tree
crown.n.07 - the upper branches and leaves of a tree or other plant
```

2.1 Components Lexical Semantic Relations

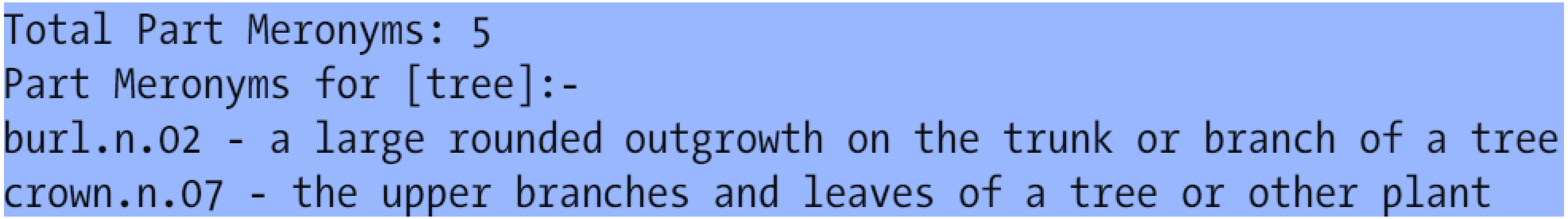
(v) Holonyms and Meronyms

- Holonyms are entities that contain a specific entity of our interest. Basically holonym refers to the relationship between a term or entity that denotes the whole and a term denoting a specific part of the whole.



```
Total Member Holonyms: 1
Member Holonyms for [tree]:-
forest.n.01 - the trees and other plants in a large densely wooded area
```

- From the output, we can see that 'forest' is a holonym for 'tree' , which is semantically correct because, of course, a forest is a collection of trees.
- Meronyms are semantic relationships that relate a term or entity as a part or constituent of another term or entity.



```
Total Part Meronyms: 5
Part Meronyms for [tree]:-
burl.n.02 - a large rounded outgrowth on the trunk or branch of a tree
crown.n.07 - the upper branches and leaves of a tree or other plant
```

3. Word sense disambiguation

- Recall we have seen homographs and homonyms, which are basically **words that look or sound similar but have very different meanings.**
- This **meaning, is contextually based on** how **it has been used and also depends on the word semantics**, and is called **word sense** .
- With the assumption that the word has multiple meanings based on its context, the process of identifying the correct sense or semantics of a word based on its usage is called **word sense disambiguation.**
- **i.e. The process of determining which sense of a word is used in a given context.**

There are **three methods for identifying WSD** include :

i. Supervised: Uses labeled training data.

ii. Unsupervised: Clusters senses without labeled data.

iii. Knowledge-based: Uses dictionaries and thesauri.

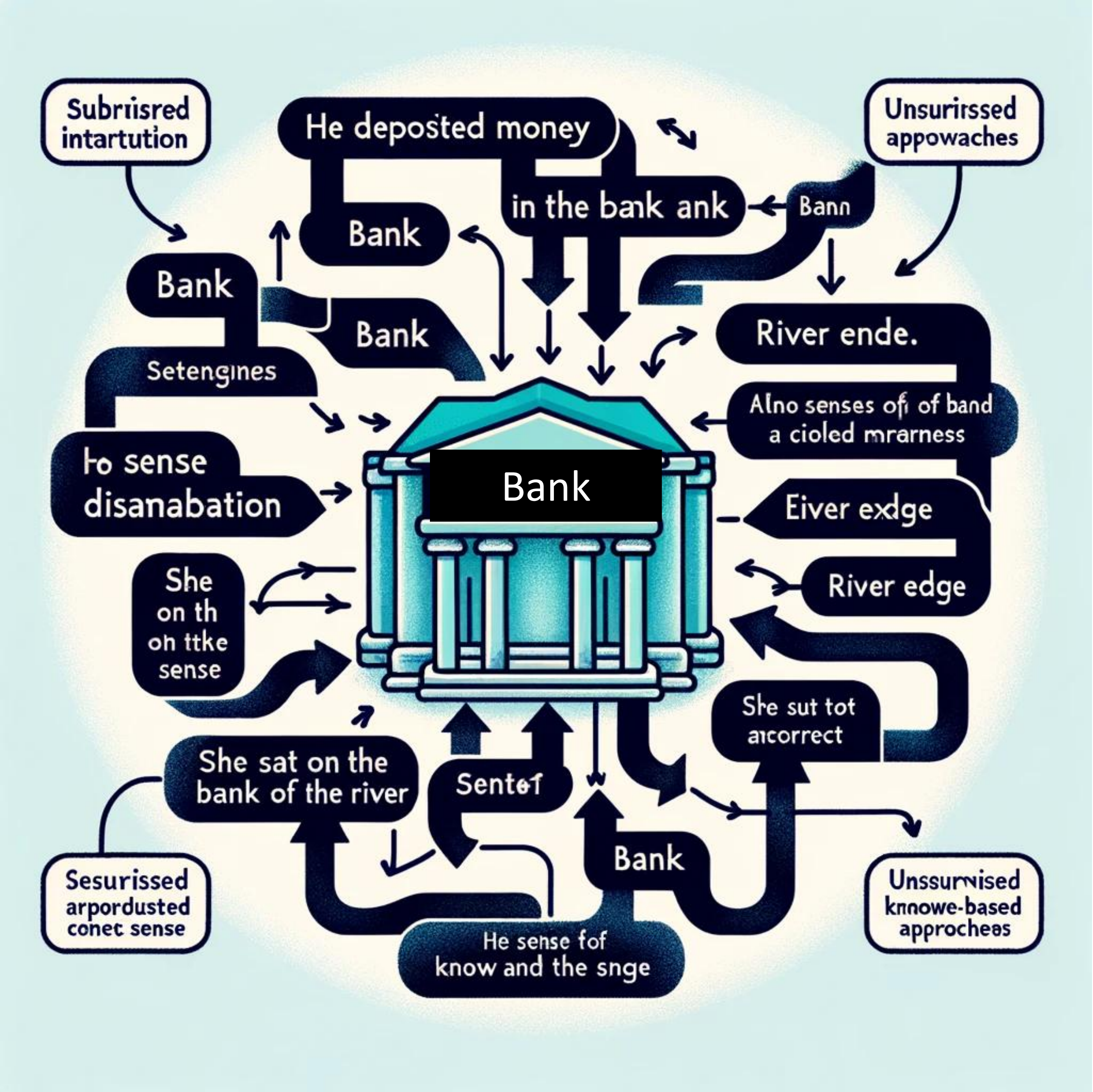
3. Word sense disambiguation

The basic principle and objective behind these methods is

- to pull dictionary or vocabulary definitions for a word to disambiguate in a body of text and
- to compare the words in these definitions with a section of text surrounding the word of interest.
- to return the synset with the maximum number of overlapping words or terms **between the context sentence and the different definitions from each synset** for the word of interest.
- (WordNet definitions can also be used for words instead of a dictionary.)

3. Word sense disambiguation

- The diagram showing the process of word sense disambiguation (WSD). It includes the word "bank" in the center with its two possible meanings, and illustrates the context-based disambiguation process with example sentences.
- The methods used for WSD, such as supervised, unsupervised, and knowledge-based approaches, are also depicted.



3. Word sense disambiguation

Example consider a # sample text and word to disambiguate

samples = [('The fruits on that plant have ripened', 'n’), ('He finally reaped the fruit of his hard work as he won the race', 'n')]

word = 'fruit’

After performing words sense disambiguation it gives the output

Sentence: The fruits on that plant have ripened

Word synset: Synset('fruit.n.01')

Corresponding definition: the ripened reproductive body of a seed plant

Sentence: He finally reaped the fruit of his hard work as he won the race

Word synset: Synset('fruit.n.03')

Corresponding definition: the consequence of some effort or action

3. Word sense disambiguation

Example consider a # sample text and word to disambiguate

```
samples = [('Lead is a very soft, malleable metal', 'n’), (‘Ranbir is the actor who plays the lead in that movie', 'n’), ('This road leads to nowhere', 'v')]
word = 'lead’
```

After performing word sense disambiguation it gives the output

```
Sentence: Lead is a very soft, malleable metal
Word synset: Synset('lead.n.02')
Corresponding definition: a soft heavy toxic malleable metallic element; bluish white when freshly cut but tarnishes readily to dull grey

Sentence: Ranbir is the actor who plays the lead in that movie
Word synset: Synset('star.n.04')
Corresponding definition: an actor who plays a principal role

Sentence: This road leads to nowhere
Word synset: Synset('run.v.23')
Corresponding definition: cause something to pass or lead somewhere
```

3. Word sense disambiguation

Generally, **Lesk Algorithm** is used for word sense disambiguation

- to disambiguate two words, 'fruit' and 'lead' in various text documents, one can use the Lesk algorithm to get the correct word sense for the word we are disambiguating based on its usage and context in each document.
- This tells you how fruit can mean both **an entity that is consumed** as well as **some consequence one faces on applying efforts**.
- Similarly, we can see the word **lead** can mean the soft metal, causing something/someone to go somewhere, or even an actor who plays the main role in a play or movie.

4. Named Entity Recognition(NER)

In any text document, there are particular terms that represent **entities that are more informative and have a unique context compared to the rest of the text.**

These entities are known as **named entities**, which more specifically refers to terms that represent real-world objects like **people, places, organizations, and so on, which are usually denoted by proper names.**

Named Entities can be found by **looking at the noun phrases in text documents.**

Named entity recognition, also known as **entity chunking/extraction**, is a technique used in information extraction to identify and segment named entities and classify or categorize them under various predefined classes.

4. Named Entity Recognition(NER)

Entity Type	Description	Examples
Person	Names of individuals	Mahatma Gandhi, Narendra Modi, Priyanka Chopra
Organization	Names of companies, institutions, or agencies	Tata Consultancy Services, Indian Railways, ISRO
Location	Geographic locations such as cities, states, countries	Mumbai, Kerala, India
Date	Specific calendar dates	15th August, Diwali, 26th January
Time	Specific times of the day	10:00 PM, Evening
Money	Monetary values	₹1000, ₹50 crore
Percent	Percentage values	10%, 75%
Facility	Buildings, airports, highways	Taj Mahal, Chhatrapati Shivaji International Airport, NH 44
Product	Names of products	Maruti Suzuki, Parle-G, Dettol
Event	Names of culturally significant events or holidays	Kumbh Mela, Republic Day, IPL
Art	Works of art, books, films	Sholay, Gitanjali, Baahubali
Law	Legal documents, acts, or statutes	Constitution of India, GST Act, IPC

4. Named Entity Recognition(NER)

Techniques for NER include:

- **Rule-based:** Uses handcrafted rules.
- **Machine Learning-based:** Uses algorithms trained on labeled data.
- **Deep Learning-based:** Uses neural networks for higher accuracy.

The typical process of NER in NLTK proceeds as follows: Consider a sample text say

"Apple Inc. is a technology company based in California. Tim Cook is the CEO of Apple.“

Steps to Perform NER

1. **Tokenization:** Split the text into individual words or tokens.
2. **Part-of-Speech Tagging:** Tag each token with its corresponding part of speech (optional but helpful).
3. **NER Tagging:** Identify and classify named entities in the text.

4. Named Entity Recognition(NER)

Annotated Example

1. **Tokenization:**

["Apple", "Inc.", "is", "a", "technology", "company", "based", "in", "California", ".", "Tim", "Cook", "is", "the", "CEO", "of", "Apple", "."]

2. **Part-of-Speech Tagging:** (For simplicity, only showing tags for entities)

[("Apple", "NNP"), ("Inc.", "NNP"), ("is", "VBZ"), ("a", "DT"), ("technology", "NN"), ("company", "NN"), ("based", "VBN"), ("in", "IN"), ("California", "NNP"), (".", "."), ("Tim", "NNP"), ("Cook", "NNP"), ("is", "VBZ"), ("the", "DT"), ("CEO", "NN"), ("of", "IN"), ("Apple", "NNP"), (".", ".")]

3. **NER Tagging:**

[("Apple", "ORGANIZATION"), ("Inc.", "O"), ("is", "O"), ("a", "O"), ("technology", "O"), ("company", "O"), ("based", "O"), ("in", "O"), ("California", "LOCATION"), (".", "O"), ("Tim", "PERSON"), ("Cook", "PERSON"), ("is", "O"), ("the", "O"), ("CEO", "O"), ("of", "O"), ("Apple", "ORGANIZATION"), (".", "O")]

Final Annotated Text

The named entities are highlighted as follows:

- **Organization:** Apple, Apple Inc.
- **Location:** California
- **Person:** Tim Cook

5. Analysing Semantic Representations

Semantic representations are ways **to encode the meaning of words, phrases, sentences, and even larger units of text** in a form that can be used for computational processing.

These representations capture the semantic content of text and are crucial for various NLP tasks, such as machine translation, information retrieval, sentiment analysis, and question answering.

Types of **semantic representations** include:

5.1 Vector Space Models: (A) Word Vectors (Embeddings):

- Represent words as vectors in a continuous, high-dimensional space.
- Examples include Word2Vec, GloVe, and FastText.
- Words with similar meanings are located close to each other in the vector space.
- In Word2Vec, the word "king" might be represented as a vector close to "queen," "prince," and "monarch" in the embedding space, capturing its semantic relationships.

5. Analysing Semantic Representations

(B) Contextual Embeddings:

- Capture the meaning of words in different contexts.
- Examples include ELMo, BERT, and GPT.
- Contextual embeddings generate different vectors for the same word depending on its context in the sentence.

In BERT(Bidirectional Encoder Representations from Transformers) is a state-of-the-art natural language processing (NLP) model developed by Google.), the word "bank" will have different vectors in the sentences "He went to the bank to deposit money" and "She sat by the bank of the river," capturing the different meanings based on context.

2. Lexical Representations

2.1 Semantic Networks:

- Represent words as nodes in a network, with edges representing semantic relations (e.g., synonymy, antonymy).
- Example: WordNet.

In WordNet, the word "car" is connected to "vehicle" (hypernym) and "sedan," "convertible" (hyponyms), representing its hierarchical relations.

5. Analysing Semantic Representations

2.2 Ontologies:

- Represent a set of concepts within a domain and the relationships between those concepts.
- Example: DBpedia, an ontology derived from Wikipedia.

3. Distributional Semantics:

- **Based on the Distributional Hypothesis** "You will shall know a word by the company it keeps."
- Words are represented based on their co-occurrence with other words in large corpora.
- Examples include Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA).

4. Logical Representations

4.1 Propositional Logic:

- Used in query systems and formal reasoning.
- Represents sentences using propositions and logical connectives (AND, OR, NOT, IMPLIES).

Syntax and Semantics for propositional logic

- Syntax: Rules for forming valid statements.
- Semantics: Truth values of statements.
- Example: "If it rains, then the ground is wet.”

4.2 First-Order Logic (FOL):

- Used in knowledge representation and automated reasoning.
- Extends propositional logic by including quantifiers and predicates.
- Can represent more complex relationships and properties of objects.

Syntax and Semantics for FOL

Syntax: Rules for forming statements with variables, predicates, and quantifiers (e.g., ∀, ∃).

Semantics:

Interpretation of variables and predicates in a domain.

Example: "All humans are mortal" ($\forall x\, (\text{Human}(x) \rightarrow \text{Mortal}(x))$).

The sentence "All humans are mortal" can be represented in first-order logic as $\forall x\, (\text{Human}(x) \rightarrow \text{Mortal}(x))$.

5. Frame Semantics:

- Represents meaning based on the idea that words evoke certain structures of related concepts (frames).
- Example: FrameNet, which contains frames for various concepts and the roles associated with them.

6. Semantic Role Labeling (SRL):

- Identifies the predicate-argument structure of a sentence.
- Assigns roles to different constituents (e.g., who did what to whom, when, where).