Module 4

Q. What is Word Sense Disambiguation (WSD)? Explain the dictionary based approach to Word Sense Disambiguation . What is Lesk Algorithm ?

=>

1. Word Sense Disambiguation (WSD) is the process of identifying the correct meaning of a word that has multiple meanings, based on the context in which it is used.
2. Many words in English have more than one meaning, and choosing the right one is essential for understanding sentences correctly.
3. For example, the word **"bank"** can mean:
    a. A financial institution ("I went to the bank to deposit money.")
    b. The side of a river ("He sat on the river bank.")
4. WSD helps computers understand which meaning of the word "bank" is being used by analyzing surrounding words and the sentence structure.
5. It is an important part of Natural Language Processing (NLP) for tasks like **machine translation, information retrieval, and text summarization.**
6. Without WSD, a system might misinterpret the meaning of sentences, leading to wrong translations or irrelevant search results.
7. WSD systems use linguistic and statistical information to predict the most likely sense of a word.
8. Approaches to WSD can be broadly divided into **Knowledge-based, Supervised, Unsupervised, and Hybrid methods.**
9. Among them, **Dictionary-based approaches** are one of the simplest and oldest techniques.
10. It mainly relies on dictionaries or lexical databases such as **WordNet** to determine word meanings.

Dictionary-Based Approach to WSD

1. The dictionary-based approach uses predefined word definitions and sense examples from resources like WordNet, Oxford Dictionary, or Merriam-Webster.
2. The main idea is to compare the **context of the word in a sentence** with the **definitions (glosses)** of all its possible senses in the dictionary.
3. The sense whose definition shares the most words with the surrounding context is chosen as the correct meaning.
4. The most well-known dictionary-based algorithm is **Lesk's Algorithm (1986).**

Steps in Dictionary-Based WSD (Using Lesk Algorithm)

1. Identify the **target word** whose meaning needs to be disambiguated.

2. Retrieve all possible meanings (senses) of the target word from the dictionary.
3. Collect all the words from the **definitions (glosses)** of these senses.
4. Extract the **context words** from the sentence in which the target word appears.
5. Compare the gloss words of each sense with the context words.
6. The sense that has the **maximum word overlap** with the context is selected as the correct meaning.
7. For example:
    a. Sentence: "I went to the bank to deposit money."
    b. Target word: "bank"
    c. Possible senses:
        i. Bank (financial): *an institution for receiving, keeping, and lending money*.
        ii. Bank (river): *the land alongside or sloping down to a river or lake*.
    d. Context words: *deposit, money, went*
    e. Overlap found with financial sense → Correct sense = *Financial institution*.

Basic Idea of Lesk Algorithm

1. Every word in a dictionary has one or more definitions (senses).
2. Each definition contains a set of words that describe that particular meaning.
3. When a word appears in a sentence, its true sense is the one whose definition overlaps the most with the definitions of nearby words.
4. Therefore, the algorithm tries to find the sense with maximum overlap between the gloss of the target word and the glosses of the context words.

Steps of the Lesk Algorithm

1. Input a sentence containing an ambiguous word.
2. Identify the target word for which the sense is to be determined.
3. Collect all possible senses (definitions) of the target word from a dictionary (e.g., WordNet).
4. Gather the glosses of each possible sense.
5. Extract context words , the words surrounding the target word in the sentence.
6. For each sense of the target word:
7. Compare its gloss with the glosses of the context words.
8. Count how many words are common (overlap) between them.
9. Select the sense which has the highest overlap score (maximum number of common words).

Q. Explain with suitable example the following relationships between word meanings: Hyponymy, Hypernymy, Meronymy, Holonymy, Homonymy, Polysemy, Synonymy, Antonymy.

=>

1. Words in a language are often related to each other through their meanings.
2. These relationships help computers and humans understand how words are connected in context.
3. In Natural Language Processing (NLP) and Linguistics, semantic relationships between words play a crucial role in **Word Sense Disambiguation, Information Retrieval, Ontology Building, and Text Understanding.**
4. The main types of word-meaning relationships are **Hyponymy, Hypernymy, Meronymy, Holonymy, Homonymy, Polysemy, Synonymy, and Antonymy.**
5. Each of these relationships expresses a unique type of connection between words.

2. Hyponymy (Subordinate Relationship)

1. **Definition:** Hyponymy is a "kind-of" relationship between a specific word and a general category.
2. The more specific word is called a **hyponym.**
3. It represents a **subcategory or a specific instance** of a broader concept.
4. **Example:**
    a. *Rose*, *Lily*, and *Tulip* are hyponyms of *Flower*.
    b. *Apple* is a hyponym of *Fruit*.
5. This relationship helps in **classifying words hierarchically.**

3. Hypernymy (Superordinate Relationship)

1. **Definition:** Hypernymy is the opposite of hyponymy.
2. A **hypernym** is a word that denotes a **broader category** covering several specific words.
3. It represents a **general term** for a group of related subtypes.
4. **Example:**
    a. *Animal* is a hypernym of *Dog, Cat, and Elephant*.
    b. *Vehicle* is a hypernym of *Car, Bus, and Bike*.
5. Hypernyms are useful in building **semantic hierarchies and taxonomies**.

4. Meronymy (Part-Whole Relationship)

1. **Definition:** Meronymy is a relationship where one word denotes a **part** of another word.
2. The smaller or component part is called a **meronym.**
3. **Example:**

   a. *Wheel* is a meronym of *Car*.
   b. *Keyboard* is a meronym of *Computer*.
4. It describes how smaller entities combine to form a larger object.

5. Holonymy (Whole-Part Relationship)

1. **Definition:** Holonymy is the reverse of meronymy.
2. A **holonym** is a word that represents a **whole**, which contains parts.
3. **Example:**
   a. *Car* is a holonym of *Wheel, Engine, and Door*.
   b. *Tree* is a holonym of *Branch and Leaf*.
4. It helps in understanding **structural relationships** between components and wholes.

6. Homonymy (Same Spelling/Pronunciation, Different Meaning)

1. **Definition:** Homonymy refers to words that are **spelled or pronounced the same** but have **entirely different meanings.**
2. These words are **unrelated in meaning.**
3. **Example:**
   a. *Bat* → (a flying mammal) and *Bat* → (a cricket bat).
   b. *Bank* → (financial institution) and *Bank* → (side of a river).
4. Homonyms often create **ambiguity** in natural language understanding.

7. Polysemy (Same Word, Related Meanings)

1. **Definition:** Polysemy refers to a single word having **multiple related meanings.**
2. Unlike homonyms, polysemous words share a **common root or conceptual link.**
3. **Example:**
   a. *Head* → (part of the body), *Head* → (leader of a department), *Head* → (top or front part).
   b. *Foot* → (part of body), *Foot* → (bottom of a mountain).
4. Polysemy enriches language by allowing one word to express **several related ideas.**

8. Synonymy (Similar Meaning Relationship)

1. **Definition:** Synonymy occurs when two or more words have **the same or nearly the same meaning.**
2. Such words can often be used interchangeably, depending on the context.
3. **Example:**
   a. *Begin* and *Start*
   b. *Big* and *Large*

    c. *Happy* and *Joyful*
4. Synonyms are important for **paraphrasing, search engines, and sentiment analysis.**

9. Antonymy (Opposite Meaning Relationship)

1. **Definition:** Antonymy is the relationship between words that have **opposite meanings.**
2. There are different types of antonyms — **gradable, complementary, and relational.**
3. **Examples:**
    a. *Hot* ↔ *Cold* (gradable)
    b. *Alive* ↔ *Dead* (complementary)
    c. *Buy* ↔ *Sell* (relational)
4. Antonyms help express **contrast** and are widely used in **text analysis and sentiment classification.**

Q. Semantic Analysis in Natural Language Processing (NLP)

=>

1. **Semantic Analysis** is the process in Natural Language Processing (NLP) that deals with **understanding the meaning of words, phrases, and sentences**.
2. It focuses on deriving **logical and contextual meaning** from the syntactically correct sentences obtained after syntactic analysis.
3. In simple terms, semantic analysis helps machines understand **what a sentence actually means**, not just how it is structured.
4. For example:
    a. Sentence: *"Riya ate an apple."*
    b. Semantic meaning: The action of eating was performed by Riya, and the object eaten was an apple.
5. Semantic analysis is an essential step after lexical and syntactic analysis in the NLP pipeline.
6. It enables machines to perform tasks like **question answering, machine translation, summarization, and chatbot communication** more accurately.

Need for Semantic Analysis

1. Grammar alone cannot explain the **true meaning** of a sentence.
2. The same word or sentence structure can mean different things depending on the context.
3. For example:
    a. *"The chicken is ready to eat."*

→ Can mean either the chicken is cooked (object meaning) or the chicken is hungry (subject meaning).

4. Semantic analysis helps resolve such **ambiguities** and ensures correct interpretation.
5. It also helps computers **link words to concepts**, **recognize relationships**, and **understand intent.**

Goals of Semantic Analysis

1. To determine the **meaning of individual words** (lexical semantics).
2. To understand how **word meanings combine** to form sentence meaning (compositional semantics).
3. To identify **relationships** between entities mentioned in text (like "Ram works at Google").
4. To resolve **ambiguity** — choosing the correct sense of a word using **Word Sense Disambiguation (WSD)**.
5. To create a **semantic representation** that can be used by machines for reasoning or inference.

Components of Semantic Analysis

Semantic analysis can be divided into **two main levels**:

(a) Lexical Semantics

1. Deals with understanding the **meaning of individual words.**
2. It studies how words relate to each other through relationships such as **synonymy, antonymy, hypernymy, hyponymy, and polysemy.**
3. Example: Recognizing that *"big"* and *"large"* have similar meanings, or *"dog"* is a type of *"animal."*

(b) Compositional Semantics

1. Concerned with how **word meanings combine** to form sentence meaning.
2. Based on the **Principle of Compositionality**, which states that *the meaning of a sentence is derived from the meanings of its parts and how they are combined grammatically.*
3. Example: In *"John loves Mary,"* the meaning is composed of the subject (*John*), verb (*loves*), and object (*Mary*).

Techniques Used in Semantic Analysis

1. Word Sense Disambiguation (WSD)

- Helps choose the **correct meaning** of a word that has multiple senses.
- Example: *Bank* → financial institution or river side.

2. Named Entity Recognition (NER)

- Identifies and classifies **named entities** such as people, places, organizations, or dates in text.
- Example: In *"Elon Musk founded SpaceX,"* → *Elon Musk* = Person, *SpaceX* = Organization.

3. Semantic Role Labeling (SRL)

- Determines **who did what to whom** in a sentence.
- Example: *"Riya gave a book to Aarav."*
    - Agent: Riya
    - Action: gave
    - Theme: book
    - Recipient: Aarav

4. Relationship Extraction

- Identifies **semantic relationships** between entities.
- Example: *"Apple acquired Beats."* → Relation: *acquired(company1, company2)*

5. Ontology-Based Analysis

- Uses structured knowledge bases (like **WordNet or DBpedia**) to relate words to **concept hierarchies** and meanings.
- Example: Understanding that "cat" is an "animal" and a "pet."