Unit 2

Language Syntax and Semantics

Contents

- Morphological Analysis: What is Morphology?
- Types of Morphemes, Inflectional morphology & Derivational morphology,
- Morphological parsing with Finite State Transducers (FST)
- Syntactic Analysis: Syntactic Representations of Natural Language
- Parsing Algorithms
- Probabilistic context-free grammars, and
- **Statistical parsing Semantic Analysis: Lexical Semantic, Relations among lexemes & their senses Homonymy, Polysemy, Synonymy, Hyponymy, WordNet, Word Sense Disambiguation (WSD)**
- **Dictionary based approach, Latent Semantic Analysis**

Statistical parsing Semantic Analysis

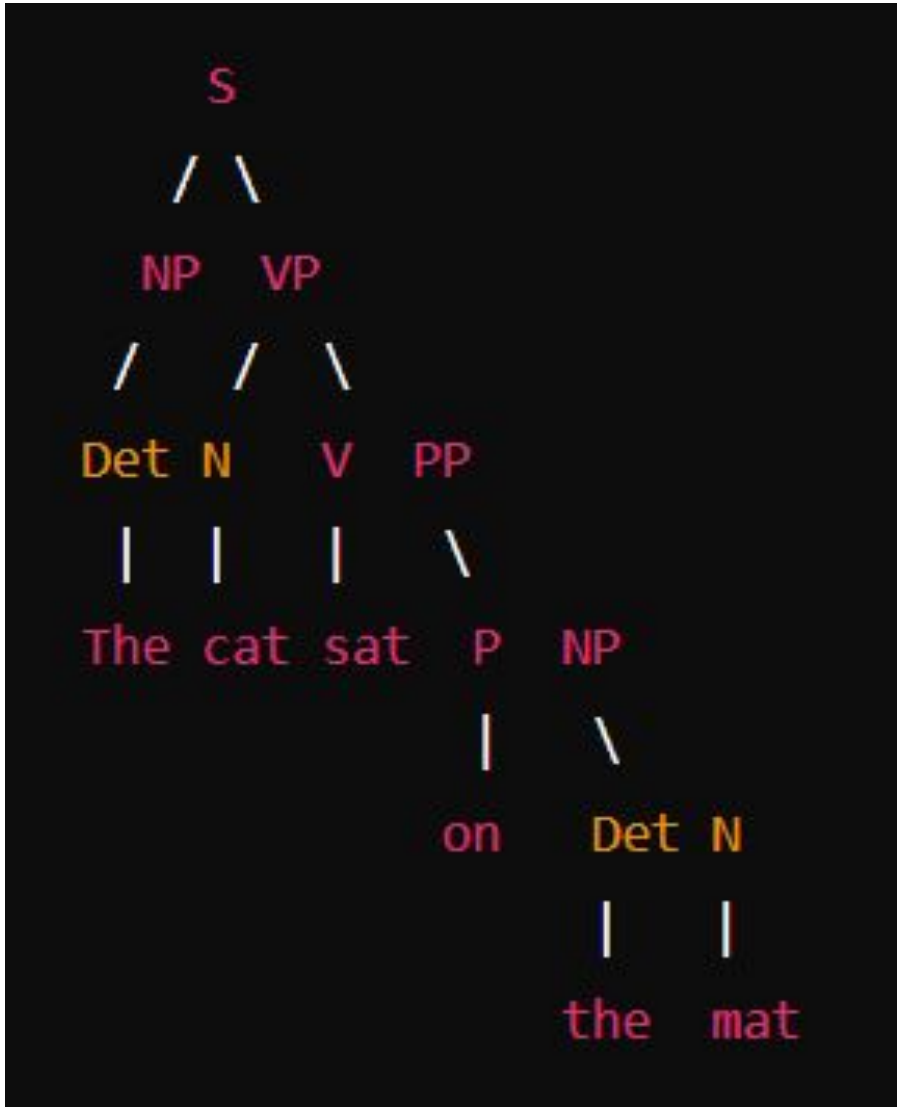
1. Statistical Parsing

- Statistical parsing involves using probabilistic methods to construct the syntactic structure (e.g., parse tree) of a sentence. Unlike rule-based parsing, it relies on training data and probabilities:
- **Probabilistic Context-Free Grammars (PCFGs):** Assign probabilities to grammar rules, guiding the parser to the most likely parse tree.
- **Dependency Parsing:** Focuses on the relationships between words in a sentence (e.g., subject, object).
- **Applications:** Used in machine translation, text summarization, and question answering.

Example: Parsing a Sentence

"The cat sat on the mat.“

- **Parse Tree (Constituency Parsing):**
  A parse tree might look like this:



- **NP (Noun Phrase):** "The cat"
- **VP (Verb Phrase):** "sat on the mat"
- **PP (Prepositional Phrase):** "on the mat“

- **Dependency Parsing:**
  Focuses on relationships:
- "sat" → Root
- "cat" → Subject of "sat"
- "mat" → Object of "on"
- "on" → Modifier of "sat"

2. Semantic Analysis

- Semantic analysis is the process of **interpreting the meaning** of a sentence or text. It includes:

- **Lexical Semantics:** Studies the meanings of words, **their relationships**, and how they combine to form phrases or sentences.
- **Key components include:**
  - **Word Senses:** Different meanings a word can have based on context (e.g., "bank" as a financial institution vs. a riverbank).

Example: Understanding Word Meaning

Sentence: *"The light is bright."*

Lexical Semantics:

- "light" could mean:
  - A source of illumination.
  - Opposite of heavy (context clarifies it's illumination here).

Semantic Role Labeling (SRL):

Assigns roles like:

- Agent, Theme, Location, etc. For *"John gave Mary a book,"*:
- John → Agent (doer)
- Mary → Recipient
- Book → Theme (object)

3. Relations Among Lexemes & Their Senses

- Lexemes refer to **word forms that have specific meanings**. Key relations include:

- **Homonymy:** Words that are spelled or pronounced the same but have unrelated meanings (e.g., "bat" the animal vs. "bat" used in cricket).

- **Polysemy:** A single word with multiple related meanings (e.g., "paper" as a material, an academic publication, or a newspaper).

- **Synonymy:** Different words with the same or nearly the same meaning (e.g., "happy" and "joyful").

- **Hyponymy:** A hierarchical relationship where one word is a specific type of another (e.g., "sparrow" is a hyponym of "bird").

Relations Among Lexemes & Their Senses

- **Homonymy:**

Words with the same spelling or sound but unrelated meanings.

- *"Bear"*:
  - As a verb: "I can’t bear this pain."
  - As a noun: "The bear is in the forest.“

- **Polysemy:**

Words with multiple related meanings.

- *"Paper"*:
  - Material: "I need some paper to write on."
  - Publication: "The paper published an article."

- **Synonymy:**

Words with the same or similar meanings.

- Examples: "Big" and "Large"; "Small" and "Tiny.“

- **Hyponymy:**

Words in a hierarchical relationship.

- *"Rose"* (hyponym) is a type of *"Flower"* (hypernym).
- *"Dog"* is a hyponym of *"Animal."*

4. WordNet

WordNet is a **lexical database** that organizes English words into:

- **Synsets (Synonym Sets):** Groups of synonymous words with definitions and examples.

- **Hierarchies:** Includes **hypernyms** (more general terms) and **hyponyms** (more specific terms).

Applications: Widely used in NLP tasks like word sense disambiguation and semantic similarity.

Example:

For the word *"**plant**"*:

- **Noun (Sense 1):** "A living organism, typically growing in soil" (e.g., tree, flower).
  - **Hyponym(Specific Term):** Rose, Oak
  - **Hypernym(General Term):** Organism
- **Noun (Sense 2):** "An industrial site for manufacturing" (e.g., factory).
- **Verb (Sense 3):** "To put seeds in the ground.“
- Using **WordNet**, you can retrieve synonyms, antonyms, or conceptual relationships.

5. Word Sense Disambiguation (WSD)

- WSD is the process of **determining which sense of a word is used** in a particular **context**. For example:
- In "The bank of the river," WSD identifies "bank" as referring to a riverbank, not a financial institution.

- **Approaches:**
  - **Supervised Learning:** Requires annotated corpora for training.
  - **Knowledge-Based:** Leverages resources like WordNet.
  - **Unsupervised Learning:** Identifies word senses through clustering or similarity measures.

Example: Ambiguity in *"Bank"*

- Sentence 1: "I deposited money in the bank."

Sense: A financial institution.

- Sentence 2: "We walked along the bank of the river."

Sense: The side of a river.

WSD Approach:

- **Supervised Learning:** Train a model on sentences labeled with the sense of "bank."
- **Knowledge-Based:** Use **WordNet**:
  - Look at nearby words: "money" suggests financial, "river" suggests geographical.
- **Real-World WSD Application:**

In **Machine Translation**, WSD ensures that ambiguous words are translated correctly based on context.

Dictionary-Based Approach in NLP

- The dictionary-based approach is a **knowledge-based technique** in Natural Language Processing (NLP) that uses predefined lexical resources (like dictionaries or lexicons) to analyze, process, or interpret text.

- It is particularly effective for tasks like **sentiment analysis**, **word sense disambiguation (WSD)**, and **semantic analysis**.

How It Works:

1. Lexical Resource:

- Uses resources like **WordNet**, **SentiWordNet**, or **domain-specific dictionaries**.
- Each word is associated with definitions, synonyms, antonyms, or polarity (positive, negative, neutral).

2. Matching Words:

- Text is tokenized into words.
- Words are matched with dictionary entries to extract relevant information.

3. Analysis:

- Extract **semantic** or **sentiment** information based on the dictionary.
- Resolve ambiguities by relying on **word relationships in the lexicon**.

Example: Sentiment Analysis Using SentiWordNet

"The movie was amazing but the ending was disappointing.“

Steps:

1. Tokenize: ["movie", "was", "amazing", "but", "ending", "was", "disappointing"].
2. Match with SentiWordNet:
   - "Amazing" → Positive sentiment: +0.8
   - "Disappointing" → Negative sentiment: -0.7
3. Aggregate:
   - Compute overall sentiment: Positive (+0.8 - 0.7 = +0.1).

Strengths:

- Easy to implement using existing resources.
- Does not require large training datasets.
- Works well for well-defined domains.

Limitations:

- Limited to the quality of the dictionary.
- Cannot handle nuances or newly coined words effectively.
- Struggles with context-dependent meanings.

2. Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA) is a **statistical technique** for extracting and representing the **hidden (latent) meanings** in large text datasets.

It is used for tasks like **document similarity**, **topic modeling**, and **dimensionality reduction** in NLP.

How It Works:

1. **Create a Term-Document Matrix (TDM):**
   - Rows represent words (terms).
   - Columns represent documents.
   - Each cell contains the frequency or weight of the term in the document

Example TDM for three documents:

	Doc1	Doc2	Doc3
Movie	3	0	2
Cinema	1	0	1
Film	0	2	1
Actor	0	3	0

Example: Identifying Document Similarity

Input documents:

- Doc1: "The movie was thrilling."
- Doc2: "The film was exciting."
- Doc3: "The actor gave a stunning performance."

Steps:

1. Create a TDM.
2. Apply SVD.
3. Represent documents in a latent space.
4. Compute similarities (e.g., cosine similarity) between document vectors:
    1. Doc1 and Doc2 might have high similarity due to shared latent topic.

Applications of LSA:

- **Search Engines:** Improve information retrieval by identifying synonyms and latent topics.
- **Automatic Summarization:** Extract the most relevant information from text.
- **Topic Modeling:** Identify overarching topics in a corpus.

Strengths:

- **Uncovers hidden relationships** between terms and documents.
- Reduces dimensionality, making computations efficient.
- Does **not require labeled data (unsupervised**).

Limitations:

- Assumes linear relationships between terms (may oversimplify meaning).
- Sensitive to the size and quality of the corpus.
- **Struggles with polysemy** and context sensitivity compared to neural approaches like Word2Vec or BERT.

Comparison of Dictionary-Based and LSA Approaches:

Feature	Dictionary-Based Approach	Latent Semantic Analysis
Input Data	Lexicons or predefined dictionaries.	Text corpus with term-document matrix.
Type	Knowledge-based.	Statistical, unsupervised.
Strengths	Quick and interpretable.	Finds latent meanings and relationships.
Limitations	Requires high-quality lexicons.	Computationally intensive, oversimplifies semantics.