UNIT IV: SEMANTICS AND PRAGMATICS

This unit moves beyond grammatical structure (syntax) to the study of **meaning**.

- **Semantics:** The study of the literal meaning of words, phrases, and sentences, independent of context. It answers the question, "What does this statement mean?"
- **Pragmatics:** The study of how context influences meaning. It deals with the intended meaning or the "meaning in action." It answers the question, "What is the speaker *trying to do* with this statement?"

Example:

- **Sentence:** "It's cold in here."
- **Semantic Meaning:** A statement about the low temperature of the room.
- **Pragmatic Meaning:** Could be a request to close a window or turn on the heat.

---

1. Representing Meaning

To enable a computer to work with meaning, we need to represent it in a formal, structured way.

Requirements for a Meaning Representation

A good meaning representation language should have the following properties:

1. **Verifiability:** You should be able to compare the representation with a knowledge base of the world to determine if it's true or false.
2. **Unambiguity:** The representation must be clear and have only one possible interpretation, even if the original sentence was ambiguous.
3. **Canonical Form:** The same meaning should have the same representation, regardless of how it's expressed in words. For example, "Does the restaurant serve vegetarian food?" and "Are vegetarian dishes served by the restaurant?" should map to the same formal query.
4. **Inference:** The representation should allow the system to draw new conclusions from existing facts. For example, if we know "A is a car," we should be able to infer "A is a vehicle."

First-Order Logic (FOL)

First-Order Logic (FOL) is a powerful and expressive formal language used to represent knowledge and meaning. It contains:

- **Objects:** Representations of specific things in the world (e.g., John, Mary, Table1).
- **Properties (Predicates):** Statements about objects (e.g., Happy(John) means "John is happy").
- **Relations (Predicates):** Statements relating multiple objects (e.g., Loves(John, Mary) means "John loves Mary").
- **Quantifiers:**
    - **Universal Quantifier (∀):** Means "For all" or "Every."
        - *Example:* "All students are smart" becomes $\forall x \text{ Student}(x) \Rightarrow \text{Smart}(x)$
        - (Read as: For every x, if x is a student, then x is smart).
    - **Existential Quantifier (∃):** Means "There exists" or "Some."
        - *Example:* "Some students are happy" becomes $\exists x \text{ Student}(x) \land \text{Happy}(x)$
        - (Read as: There exists an x such that x is a student and x is happy).

Description Logics (DL)

Description Logics (DLs) are a family of knowledge representation languages that are less expressive than full FOL but are much more computationally manageable. They are excellent for defining **concepts** (classes of objects) and **roles** (relationships).

- **Core Use:** DLs are the foundation of **ontologies** and the **Semantic Web** (e.g., OWL - Web Ontology Language). They are used to build large, structured knowledge bases that machines can reason with efficiently.

---

2. From Syntax to Semantics

This section explains how we can automatically derive the meaning representation from a sentence's parse tree.

Syntax-Driven Semantic Analysis

This approach is based on the **Principle of Compositionality**, which states:

"The meaning of a whole is a function of the meaning of its parts."

This means we can build the meaning of a sentence by starting with the meanings of individual words and combining them according to the rules of the syntactic parse tree, from the bottom up.

Semantic Attachments

This is the practical method for implementing compositionality. We "attach" a small semantic rule to each syntactic grammar rule.

- **Syntactic Rule:** S -> NP VP
    - **Semantic Attachment:** The meaning of S is created by applying the meaning of the VP to the meaning of the NP.

Simple Example: "John loves Mary"

1. **Parse Tree:** A tree is built with S at the top, branching to NP ("John") and VP ("loves Mary"). The VP further branches to V ("loves") and NP ("Mary").
2. **Lexical Semantics:**
    - Meaning of "John" -> John
    - Meaning of "Mary" -> Mary
    - Meaning of "loves" -> Loves(y, x) (A two-place relation)
3. **Composition:**
    - First, the VP meaning is built by combining V and NP: Loves(Mary, x)
    - Next, the S meaning is built by combining the NP ("John") and the VP. The meaning John fills the x slot.
    - **Final Representation:** Loves(Mary, John) (or more intuitively Loves(John, Mary), depending on rule definition).

---

3. Lexical Semantics (The Meaning of Words)

Word Senses

A **word sense** is a distinct meaning of a word. Many words are polysemous (have multiple senses).

- **Example:** The word "**bank**" has several senses, including:
    - A financial institution.
    - The side of a river.
    - To rely upon something ("You can bank on it").

Relations Between Senses

Word senses are not isolated; they are related to each other in a network. Key relations include:

- **Synonymy:** Two senses have the same or nearly the same meaning (e.g., car / automobile, big / large).
- **Antonymy:** Two senses have opposite meanings (e.g., hot / cold, fast / slow).
- **Hyponymy & Hypernymy (The "is-a" relation):**
    - A hyponym is a more specific concept (e.g., car is a hyponym of vehicle).
    - A hypernym is a more general concept (e.g., vehicle is a hypernym of car).
- **Meronymy & Holonymy (The "part-of" relation):**
    - A meronym is a part of a larger whole (e.g., wheel is a meronym of car).
    - A holonym is the whole to which parts belong (e.g., car is a holonym of wheel).

Thematic Roles

Thematic roles describe the semantic role that a noun phrase plays in the event described by the verb.

- **AGENT:** The entity that performs the action.
- **THEME** (or **PATIENT**): The entity that is affected by or undergoes the action.
- **INSTRUMENT:** The tool used to perform the action.
- **LOCATION:** Where the event happens.
- **GOAL:** The destination of the action.

Example: "John (AGENT) drove the car (THEME) from London (SOURCE) to Paris (GOAL) with his license (INSTRUMENT)."

Selectional Restrictions

A **selectional restriction** is a constraint that a verb places on the semantic type of its arguments.

- **Example:** The verb "**eat**" has a selectional restriction on its THEME argument: it must be something **edible**.
- This is why the sentence "The rock ate the car" is syntactically correct but semantically bizarre. The restriction is violated.

4. Word Sense Disambiguation (WSD)

WSD is the task of automatically identifying which specific sense of a word is being used in a given context.

Example: In "I need to go to the **bank** to deposit money," WSD is the process of figuring out that "bank" refers to the *financial institution* sense, not the *river bank* sense.

Methods for WSD

1. **Supervised WSD:** This is a standard machine learning approach.
   - **Process:**
     1. Create a training dataset where each ambiguous word is manually tagged with its correct sense (e.g., bank/FINANCIAL).
     2. Extract features from the context of the target word (e.g., nearby words, PoS tags).
     3. Train a classifier (like Naive Bayes or SVM) on these features.
     4. Use the trained model to predict the sense of the word in new, unseen sentences.
2. **Dictionary & Thesaurus Methods (e.g., Lesk Algorithm):**
   - **Idea:** This method uses dictionary definitions to disambiguate.
   - **Simple Lesk Algorithm:** To disambiguate a word, compare the dictionary definition of each of its senses with the words in the surrounding context. The sense whose definition shares the most words with the context is chosen as the correct one.
3. **Bootstrapping Methods (e.g., Yarowsky's Algorithm):**
   - **Idea:** This is a semi-supervised method that starts with a very small number of labeled examples ("seeds") and uses them to label a large unlabeled dataset.
   - **Process:**
     1. Start with a few seed examples for each sense (e.g., "river bank" vs. "financial bank").
     2. Train a simple classifier on these seeds.
     3. Apply the classifier to all unlabeled data.
     4. Take the most confident predictions and add them to the labeled set.
     5. Repeat steps 2-4, "bootstrapping" a larger and larger training set.

---

5. Word Similarity

This is the task of measuring how similar in meaning two words are.

Thesaurus-based Methods

These methods use a structured lexical resource like **WordNet**, which organizes words into a hierarchy of senses.

- **Method:** The similarity between two words can be measured by finding the **shortest path** between them in the "is-a" (hypernym) hierarchy. The shorter the path, the more similar the words are.
- **Example:** In WordNet, the path between "cat" and "dog" is shorter than the path between "cat" and "car," so "cat" is more similar to "dog."

Distributional Methods

These methods are based on the **Distributional Hypothesis**:

"You shall know a word by the company it keeps."

- **Idea:** Words that appear in similar contexts tend to have similar meanings.
- **Method:**
    1. Represent each target word as a **vector**. This vector is built by counting how often the word co-occurs with other context words in a large corpus of text.
    2. The meaning of a word is thus represented by this high-dimensional vector.
    3. The similarity between two words is then calculated as the similarity between their vectors (e.g., using **cosine similarity**). cat and dog will have similar vectors because they both co-occur with words like pet, food, animal, house, etc.