Semantic Analysis

Semantic Analysis

1. Introduction to Semantic Analysis

Semantic analysis deals with representing the meaning of natural language sentences in a formal and systematic way. In computational linguistics, the goal is to map sentences or utterances into structured meaning representations that a machine can process. Semantics is generally divided into two parts: lexical semantics, which studies the meaning of individual words, and sentence-level semantics, which studies how words combine to form the meaning of a complete sentence. For example, understanding the word "bank" requires identifying its correct sense, while understanding "Ravi went to the bank" requires combining meanings based on context.Semantic analysis aims to represent the meaning of natural language sentences in a formal and machine-understandable way. When a system processes a sentence such as:

"Ravi booked a flight to Hyderabad."

it must identify the action (booked), the agent (Ravi), the object (flight), and the destination (Hyderabad). A structured meaning representation may be written as:

$$Booked(Ravi, FlightX) \wedge Destination(FlightX, Hyderabad)$$

Thus, semantics converts natural language into logical structures that allow reasoning and inference.

2. Principle of Compositionality

A fundamental concept in semantics is the Principle of Compositionality, which states that the meaning of a sentence is determined by the meanings of its parts and the rules used to combine them. For instance, the meaning of "Murthy likes music" is derived from the meanings of Murthy, likes, and music, along with their grammatical structure. However, natural language does not always strictly follow this principle. In idioms like "kick the bucket," the meaning ("to die") cannot be derived from the literal meanings of kick and bucket, showing limitations of pure compositionality.

Example: *"Anita loves music."*

$$Loves(Anita, Music)$$

However, compositionality fails in idioms.

Example: *"The old man kicked the bucket."*

Literal composition suggests kicking a bucket, but idiomatically it means:

$$Died(OldMan)$$

Thus, semantic systems must sometimes treat idioms as single semantic units.

3. Expressiveness in Meaning Representation

A meaning representation language must have sufficient expressive power to capture the wide variety of content found in natural language. It should represent events, entities, relationships, constraints, and abstract concepts. For example, the sentence "President nominates speaker" must represent the nomination event and the roles played by each participant. Without adequate expressiveness, a system cannot accurately model real-world knowledge or complex relationships.A meaning representation language must capture complex structures such as quantifiers, time, and negation.

Example: *"Every student submitted the assignment before Monday."*

$$\forall x(Student(x) \rightarrow Submitted(x, Assignment))$$

Expressive power is necessary to represent such logical relationships accurately.

4. Predicate-Argument Structure

The core structure of semantic representation is the predicate-argument structure. A predicate represents an action, event, or relation, while its arguments represent participants. For example, in "Murthy likes music," the verb likes is the predicate, and Murthy and music are its arguments. This can be represented as Likes(Murthy, music). Predicate-argument structure specifies how many arguments a verb takes and their syntactic positions, such as NP likes NP or NP likes VP. This structure helps connect syntax with semantics.

Example 1: *"Murthy likes music."*

$$Likes(Murthy, Music)$$

Example 2: *"A cup is on the table."*

$$On(Cup, Table)$$

The predicate determines the number and type of arguments required.

5. Thematic Roles and Selectional Restrictions

Each argument of a predicate plays a thematic role (theta-role), such as agent, experiencer, or theme. In "Ravi eats rice," Ravi is the agent (eater), and rice is the theme (entity being eaten). Verbs also impose selectional restrictions, which are semantic constraints on their arguments. For example, the verb eat requires an animate subject and an edible object. These restrictions prevent semantically invalid interpretations such as "The stone eats rice."

Example: *"Ravi ate mango."*

$$Ate(Ravi, Mango)$$

Here:

- Ravi – Agent
- Mango – Theme

Selectional restrictions ensure semantic validity. *"The rock ate mango"* is semantically invalid because the subject must be animate.

6. Lambda Calculus for Semantic Composition

To systematically combine meanings, lambda calculus is used as a formal tool. A lambda expression like $\lambda x P(x)$ allows variables to be bound to arguments through a process called lambda reduction.

Verb representation:

$$\lambda x \lambda y Likes(x, y)$$

Applying to Music:

$$(\lambda x \lambda y Likes(x, y))(Music)$$

$$\Rightarrow \lambda x Likes(x, Music)$$

Applying to Anita:

$$(\lambda x Likes(x, Music))(Anita)$$

$$\Rightarrow Likes(Anita, Music)$$

Lambda reduction ensures correct semantic composition.

7. Canonical Forms, Word Sense Disambiguation, and Inference

Different sentences may express the same meaning using different words or structures. To avoid inconsistency, systems convert such expressions into a canonical form. For example, "offering a flight to Hyderabad" and "serving Hyderabad" may map to the same representation in a travel domain. Selecting the correct meaning of ambiguous words is called word sense disambiguation (WSD). Additionally, semantic systems must support inference, which allows them to derive new facts from known ones.

- "The airline offers a flight to Hyderabad."
- "The airline serves Hyderabad."

Canonical representation:

$$Flight(AirlineX, Hyderabad)$$

Word Sense Disambiguation example:

- "He sat on the bank." (river bank)
- "He deposited money in the bank." (financial bank)

Inference example:

$$Goes(Flight101, Hyderabad)$$

If user requests:

$$\exists x\, Goes(x, Hyderabad)$$

The system infers Flight101 satisfies the request.

8. Model-Theoretic Semantics and Truth Conditions

Model-theoretic semantics evaluates sentence meaning in terms of truth within a model of the world. Instead of checking whether a sentence is actually true in reality, it checks whether the sentence is true in a given model. For example, "All students passed" is true in a model where every student entity satisfies the predicate Passed. This approach explains compositionality and connects language to logical reasoning. Together, lexical semantics, predicate-argument structure, lambda calculus, canonical forms, and inference form the foundation of semantic analysis in NLP.

Example: *"All students passed."*

$$\forall x(Student(x) \rightarrow Passed(x))$$

A sentence is true if it holds within the model being considered.

Conclusion: Semantic analysis integrates lexical meaning, predicate-argument structure, thematic roles, lambda calculus, canonical forms, and inference mechanisms to enable systematic interpretation and reasoning over natural language.