Natural Language Processing

R18 B.Tech. CSE (AIML) III & IV Year JNTU Hyderabad

Prepared by
K SWAYAMPRABHA
Assistance Professor

UNIT - III

Semantic Parsing

Semantic parsing is a technique in natural language processing (NLP) that involves mapping natural language sentences into structured representations, such as logical forms or executable code. The goal of semantic parsing is to enable computers to understand the meaning of natural language sentences and to perform tasks based on that understanding.

Here is an example of how semantic parsing works:

Input sentence: "What is the capital of France?"

```
Semantic representation: {"type": "query",

                          "target": "capital",

                           "entity": {"type": "location", "name": "France"}}
```

In this example, the input sentence is a question that asks for the capital of France. The semantic representation captures the meaning of the sentence by identifying the type of the sentence (a query) and the target of the query (the capital) and the entity to which the query applies (France). The semantic representation can be used by a computer program to generate an answer to the question or to perform other tasks based on the query.

Another example:

Input sentence: "Find me the cheapest flight from New York to San Francisco next Monday."

Semantic representation:

```
{"type": "query",

  "target": "flight",

  "filters": [{"type": "location", "name": "New York"},

              {"type": "location", "name": "San Francisco"},

              {"type": "date", "value": "next Monday"},


{"type": "sort", "key": "price", "order": "asc"}]}
```

In this example, the input sentence is a request to find the cheapest flight from New York to San Francisco on the next Monday. The semantic representation captures the meaning of the sentence by identifying the type of the sentence (a query), the target of the query (a flight), and a set of filters that narrow down the search to flights that depart from New York, arrive at San Francisco, depart on the next Monday, and are sorted by price in ascending order. The semantic representation can be used by a computer program to search for flights that match the criteria and to return the cheapest option.

Semantic Interpretation

Semantic interpretation is the process of assigning meaning to a piece of language, such as a word, phrase, sentence, or text. It is a fundamental task in natural language processing (NLP) and involves analyzing language in its context to infer its intended meaning. The goal of semantic interpretation is to enable computers to understand the meaning of natural language and to perform tasks based on that understanding.

Here are some examples of semantic interpretation:

1. Word Sense Disambiguation: In natural language, many words have multiple meanings depending on their context. For example, the word "bank" can refer to a financial institution or the edge of a river. Semantic interpretation involves determining the correct meaning of a word based on its context. This task is known as word sense disambiguation.

2. Named Entity Recognition: Another task of semantic interpretation is named entity recognition, which involves identifying and classifying named entities such as people, organizations, and locations in a piece of text. For example, in the sentence "Bill Gates is the founder of Microsoft," semantic interpretation would recognize "Bill Gates" as a person and "Microsoft" as an organization.

3. Sentiment Analysis: Semantic interpretation can also be used to perform sentiment analysis, which involves identifying the sentiment or opinion expressed in a piece of text. For example, in the sentence "I love this product," semantic interpretation would recognize a positive sentiment.

4. Question Answering: Semantic interpretation is also used in question answering, which involves answering a question based on a given piece of text. Semantic interpretation helps to identify the relevant information in the text that answers the question.

1. Structural ambiguity can have a significant impact on semantic interpretation because it can lead to multiple possible interpretations of a sentence, each with a different meaning. Resolving structural ambiguity is therefore an important step in semantic interpretation, as it helps to ensure that the correct meaning of a sentence is understood.

example of how structural ambiguity can impact semantic interpretation:

Sentence: "The old man the boat."

This sentence is structurally ambiguous because it is unclear whether the man is old or the boat is old. Depending on how the sentence is parsed, it can be interpreted in two different ways:

1. The man is old: In this interpretation, "the old man" is a noun phrase that refers to an elderly man who is performing the action of "the boat."
2. The boat is old: In this interpretation, "the old" is an adjective modifying "boat," and the sentence means that the boat being referred to is old.

In this example, resolving the structural ambiguity is crucial for semantic interpretation because it determines the identity of the subject of the sentence and therefore the meaning of the sentence as a whole.

2. Word sense is a crucial aspect of semantic interpretation as it determines the meaning of a word in a given context. Words often have multiple meanings or senses, and understanding which sense of a word is being used in a particular context is essential for accurate semantic interpretation.

Here is an example of word sense in semantic interpretation:

Sentence: "I saw a bat in the park."

The word "bat" has multiple senses, including a flying mammal or a piece of sports equipment. In this context, the word "bat" likely refers to the animal sense, but without further context, it is not entirely clear. If the sentence were "I hit the bat with a stick," the sense of "bat" would be more apparent.

Another example is the word "bank," which can have multiple meanings depending on the context. It can refer to a financial institution or the edge of a river. Consider the following sentences:

- "I need to go to the bank to withdraw some money."
- "The children played by the bank of the river."

In the first sentence, "bank" refers to a financial institution, while in the second sentence, it refers to the edge of a river. Understanding the correct sense of "bank" in each context is essential for accurate semantic interpretation.

Entity and event resolution are important aspects of semantic interpretation that involve identifying and extracting information about entities (such as people, places, and organizations) and events (such as actions, states, and processes) from text.

Here is an example of entity resolution in semantic interpretation:

Sentence: "John Smith works at Google in New York City."

In this sentence, "John Smith" is an entity (a person), "Google" is an entity (an organization), and "New York City" is an entity (a location). Entity resolution involves identifying and extracting these entities from the text and linking them to their corresponding types (e.g., person, organization, or location).

Event resolution involves identifying and extracting information about events from text. For example:

Sentence: "The dog chased the cat up the tree."

In this sentence, the event is the action of the dog chasing the cat. Event resolution involves identifying and extracting information about the action, including the actors (the dog and the cat) and the location (up the tree).

Entity and event resolution can be challenging in natural language processing because entities and events can be expressed in many different ways and can be ambiguous. For example, in the sentence "John Smith is the CEO of XYZ," it may be unclear whether "XYZ" refers to an organization or a person. Resolving these ambiguities requires a deep understanding of the context and the syntax of the sentence, as well as knowledge about the world and common sense reasoning.

3. Predicate-argument structure is an important aspect of semantic interpretation that involves identifying the relationships between the main verb (predicate) and its arguments (subjects, objects, and other complements). Understanding the predicate-argument structure of a sentence is critical for accurate semantic interpretation because it allows us to identify the roles and relationships of the various elements in the sentence.

Here is an example of predicate-argument structure in semantic interpretation:

Sentence: "The cat chased the mouse."

In this sentence, the predicate is "chased," and its arguments are "cat" (the subject) and "mouse" (the object). The predicate-argument structure can be represented as follows:

Predicate: chased Subject: cat Object: mouse

Another example is:

Sentence: "John gave Mary the book."

In this sentence, the predicate is "gave," and its arguments are "John" (the subject), "Mary" (the indirect object), and "the book" (the direct object). The predicate-argument structure can be represented as follows:

Predicate: gave Subject: John Indirect object: Mary Direct object: the book

Identifying the predicate-argument structure of a sentence can be challenging because it requires an understanding of the syntax and semantics of the sentence. In some cases, the arguments may be omitted or expressed implicitly, making it difficult to identify the relationships between the predicate and its arguments.

Meaning representation is the process of representing the meaning of a sentence or a text in a structured and formal way. The goal of meaning representation is to capture the underlying meaning of the text, independent of the surface form of the language used to express it.

Here is an example of meaning representation in semantic interpretation:

Sentence: "John gave Mary the book."

One common way to represent the meaning of this sentence is using a logical form, which captures the predicate-argument structure of the sentence and the relationships between the various elements. The logical form for this sentence could be:

give(john, book, mary)

This logical form represents the meaning of the sentence by capturing the predicate "give" and its arguments, "john," "book," and "mary," in a structured and formal way. This logical form can be used to perform various natural language processing tasks, such as question answering and text generation.

Another way to represent the meaning of a sentence is using a semantic graph, which captures the relationships between the various concepts in the sentence. The semantic graph for the sentence "John gave Mary the book" could look like th

```
      +--------+
      |  John |
      +--------+
         |
        give
         |
   +------+------+
   |     |      |
 book  Mary   agent
```
This semantic graph represents the meaning of the sentence by capturing the relationships between the concepts "John," "Mary," "book," and "give" in a structured and visual way. This graph can be used to perform various natural language processing tasks, such as information extraction and semantic search.

System Paradigms

In computer science, a system paradigm is a fundamental approach or model for designing and implementing computer systems. Here are some of the commonly recognized system paradigms:

1. **Imperative paradigm**: The imperative paradigm is based on the notion of imperative programming, where a program is a sequence of statements that change the program state.

   The key constructs in imperative programming include variables, assignments, loops, and conditionals. Most programming languages, such as C, C++, Java, and Python, are based on the imperative paradigm.

2. **Functional paradigm**: The functional paradigm is based on the notion of functional programming, where a program is a set of functions that take inputs and produce outputs. In functional programming, functions are treated as first-class citizens, which means they can be passed as arguments to other functions, returned as results, or stored in data structures. Examples of functional programming languages include Haskell, Lisp, and ML.

3. **Object-oriented paradigm**: The object-oriented paradigm is based on the notion of objects, which encapsulate data and behavior. Object-oriented programming is characterized by the concepts of inheritance, polymorphism, and encapsulation.

Examples of object-oriented programming languages include Java, C++, and Python.

4. **Event-driven paradigm**: The event-driven paradigm is based on the notion of events, which are generated by the system or the user and are handled by event handlers. Event-driven programming is commonly used in graphical user interfaces, where user interactions generate events that are handled by event handlers.

   Examples of event-driven programming languages include JavaScript and Python.

5. **Concurrent paradigm**: The concurrent paradigm is based on the notion of concurrency, where multiple tasks or processes are executed simultaneously. Concurrent programming is commonly used in systems that require high performance and scalability, such as web servers and databases.

   Examples of concurrent programming languages include Go and Erlang.

These system paradigms are not mutually exclusive and can be combined in various ways to design and implement computer systems. For example, many modern programming languages, such as Java and Python, support multiple paradigms, including imperative, object-oriented, and functional programming.

In semantic parsing,

word sense disambiguation is a critical task, as it involves identifying the correct meaning of a word based on the context in which it is used.

examples of how word sense disambiguation might be performed for the words "resource," "systems," and "software" in the context of semantic parsing:

1. **Resource:**

- In the context of natural resources, the word "resource" might be associated with concepts such as "sustainability," "conservation," and "environmental impact."

- In the context of computing, the word "resource" might be associated with concepts such as "memory," "processing power," and "network bandwidth."

- In the context of human resources, the word "resource" might be associated with concepts such as "talent acquisition," "employee retention," and "skills development."

2. Systems:

- In the context of computing, the word "systems" might be associated with concepts such as "operating systems," "database management systems," and "distributed systems."
- In the context of ecology, the word "systems" might be associated with concepts such as "ecosystems," "food webs," and "biogeochemical cycles."
- In the context of business, the word "systems" might be associated with concepts such as "supply chain systems," "customer relationship management systems," and "quality management systems."

3. **Software**:

- In the context of computing, the word "software" might be associated with concepts such as "operating systems," "application software," and "system software."
- In the context of music, the word "software" might be associated with concepts such as "digital audio workstations," "synthesizers," and "sequencers."
- In the context of law, the word "software" might be associated with concepts such as "patentable computer programs," "algorithmic processes," and "intellectual property laws."