

SITA3012 Natural Language Processing

Unit 3

SEMANTIC INTERPRETATION

Semantic and logical form - Linking syntax and semantics - Ambiguity resolution - Other strategies for semantic interpretation - Scoping for interpretation of noun phrases, Semantic attachments - Word senses, Relations between the senses.

3.1 Semantic and logical form

The entire purpose of a natural language is to facilitate the exchange of ideas among people about the world in which they live. These ideas converge to form the "meaning" of an utterance or text in the form of a series of sentences. The meaning of a text is called its *semantics*. A fully adequate natural language semantics would require a complete theory of how people think and communicate ideas.

The semantics, or meaning, of an expression in natural language can be abstractly represented as a *logical form*. Once an expression has been fully parsed and its syntactic ambiguities resolved, its meaning should be uniquely represented in logical form. Conversely, a logical form may have several equivalent syntactic representations.

Consider the sentence "The ball is red." Its logical form can be represented by `red(ball)`. This same logical form simultaneously represents a variety of syntactic expressions of the same idea, like "Red is the ball."

The most challenging issue that stands between a syntactic utterance and its logical form is ambiguity; lexical ambiguity, syntactic ambiguity, and semantic ambiguity. Lexical ambiguity arises with words that have many senses. For example, the word "go" has at least the following different senses, or meanings:

move
depart
pass
vanish
reach
extend
set out

Compounding the situation, a word may have different senses in different parts of speech. The word "flies" has at least two senses as a noun (insects, fly balls) and at least two more as a verb (goes fast, goes through the air).

Case grammars have been proposed to further clarify relations between actions and objects. So-called *case roles* can be defined to link certain kinds of verbs and various objects. These roles include:

agent
theme
instrument

For example, in "John broke the window with the hammer," a case grammar would identify John as the agent, the window as the theme, and the hammer as the instrument.

Logical Forms and Lambda Calculus

The language of logical forms is a combination of the first order predicate calculus (FOPC) and the lambda calculus. The predicate calculus includes unary, binary, and n-ary predicates, such as:

Lisp notation	Prolog notation	Sentence represented
(dog1 fido1)	dog1(fido1)	"Fido is a dog"
(loves1 sue1 jack1)	loves1(sue1, jack1)	"Sue loves Jack"
(broke1 john1 window1 hammer1)	broke1(john1, window1, hammer1)	"John broke the window with a hammer"

The third example shows how the semantic information transmitted in a case grammar can be represented as a predicate.

Logical forms can be constructed from predicates and other logical forms using the operators & (and), => (implies), and the quantifiers all and exists. In English, there are other useful quantifiers beyond these two, such as many, a few, most, and some. For example, the sentence "Most dogs bark" has the logical form

(most d1: (& (dog1 d1) (barks1 d1)))

Finally, the lambda calculus is useful in the semantic representation of natural language ideas. If $p(x)$ is a logical form, then the expression $\lambda x.p(x)$ defines a function with bound variable x . *Beta-reduction* is the formal notion of applying a function to an argument.

Semantic Rules for Context Free Grammars

One way to generate semantic representations for sentences is to associate with each grammar rule an associated step that defines the logical form that relates to each syntactic category. Consider simple grammars with S, NP, VP, and TV categories.

1. If the grammar rule is $S \rightarrow NP VP$ and the logical forms for NP and VP are NP' and VP' respectively, then the logical form S' for S is $VP'(NP')$. For example, in the sentence "bertrand wrote principia" suppose that:

$NP' = \text{bertrand}$ and $VP' = \lambda x.\text{wrote}(x, \text{principia})$

Then the logical form S' is the result of Beta reduction:

```
(\x.wrote(x, principia))bertrand = wrote(bertrand, principia)
```

Prolog Representation

To accommodate the limitations of the ASCII character set, the following conventions are used in Prolog to represent logical forms and lambda expressions.

<u>Expression</u>	<u>Prolog convention</u>
(forall x: p(x))	all(X, p(X)) (recall that Prolog variables are capitalized)
(exists x: p(x))	exists(X, p(X))
and	&
implies	=>
$\lambda x.p(x)$	$X^p(X)$
$(\lambda x.p(x)) y$	reduce($X^p(X)$, Y, LF)

The Beta reduction reduce is defined by the Prolog rule *reduce*(Arg^Exp , Arg, Exp). For example,

```
reduce(X^halts(X), shrdlu, LF)
```

gives the answer $LF = \text{halts}(\text{shrdlu})$.

Semantics of a Simple Grammar

Using these ideas, we can write Prolog rules with semantics as follows:

```
s(S) --> np(NP), vp(VP), {reduce(VP, NP, S)}.
np(NP) --> det(D), n(N), {reduce(D, N, NP)}.
np(NP) --> n(NP).
vp(VP) --> tv(TV), np(NP), {reduce(TV, NP, VP)}.
vp(VP) --> iv(VP).
```

In the first rule, VP has the lambda expression and NP has the subject. In fact, the references to *reduce* can be removed from these rules, and their effects can be inserted directly where they will take place. That is, the following set of rules

```
s(S) --> np(NP), vp(NP^S).
np(NP) --> det(N^NP), n(N).
np(NP) --> n(NP).
vp(VP) --> tv(NP^VP), np(NP).
vp(VP) --> iv(VP).
```

captures the same semantics as the original set. This is called *partial execution*.

3.2 Linking syntax and semantics / Semantic Interpretation Pipeline

A system for semantic analysis determines the meaning of words in text. Semantics gives a deeper understanding of the text in sources such as a blog post, comments in a forum, documents, group chat applications, chatbots, etc. With lexical semantics, the study of word meanings, semantic analysis provides a deeper understanding of unstructured text.

How Does Semantic Analysis Work?

Semantic analysis starts with lexical semantics, which studies individual words' meanings (i.e., dictionary definitions). Semantic analysis then examines relationships between individual words and analyzes the meaning of words that come together to form a sentence. For example, it provides context to understand the following sentences:

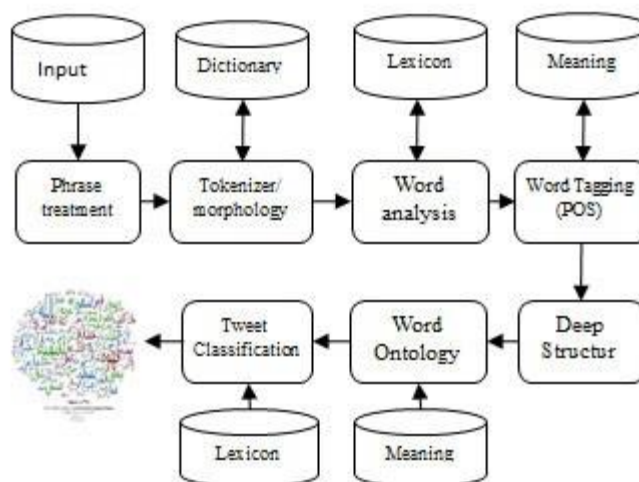
“The boy ate the apple” defines an apple as a fruit.

“The boy went to Apple” defines Apple as a brand or store.

Technically, semantic analysis involves:

1. Data processing.
2. Defining features, parameters, and characteristics of processed data
3. Data representation
4. Defining grammar for data analysis
5. Assessing semantic layers of processed data
6. Performing semantic analysis based on the linguistic formalism

Typical Semantic interpretation pipeline:



Critical elements of semantic analysis

The critical elements of semantic analysis are fundamental to processing the natural language:

- **Hyponyms:** This refers to a specific lexical entity having a relationship with a more generic verbal entity called hypernym. For example, red, blue, and green are all hyponyms of color, their hypernym.
- **Meronymy:** Refers to the arrangement of words and text that denote a minor component of something. For example, mango is a meronymy of a mango tree.
- **Polysemy:** It refers to a word having more than one meaning. However, it is represented under one entry. For example, the term 'dish' is a noun. In the sentence, 'arrange the dishes on the shelf,' the word dishes refers to a kind of plate.
- **Synonyms:** This refers to similar-meaning words. For example, abstract (noun) has a synonyms summary–synopsis.
- **Antonyms:** This refers to words with opposite meanings. For example, cold has the antonyms warm and hot.
- **Homonyms:** This refers to words with the same spelling and pronunciation, but reveal a different meaning altogether. For example, bark (tree) and bark (dog).

The NLP Problem Solved by Semantic Analysis

NLP cannot decipher ambiguous words, which are words that can have more than one meaning in different contexts. Semantic analysis is key to contextualization that helps disambiguate language data so text-based NLP applications can be more accurate.

- **Lexical analysis** is the process of reading a stream of characters, identifying the lexemes and converting them into tokens that machines can read.
- **Grammatical analysis** correlates the sequence of lexemes (words) and applies formal grammar to them so part-of-speech tagging can occur.
- **Syntactical analysis** analyzes or parses the syntax and applies grammar rules to provide context to meaning at the word and sentence level.
- **Semantic analysis** uses all of the above to understand the meaning of words and interpret sentence structure so machines can understand language as humans do.

Semantic Analysis Is Part of a Semantic System / NLP interpret figurative language

A semantic system brings entities, concepts, relations and predicates together to provide more context to language so machines can understand text data with more accuracy. Semantic analysis derives meaning from language and lays the foundation for a semantic system to help machines interpret meaning.

Consider the following elements of semantic analysis that help support language understanding:

- **Hyponymy:** A generic term.
- **Homonymy:** Two or more lexical terms with the same spelling and different meanings.
- **Polysemy:** Two or more terms that have the same spelling and similar meanings.
- **Synonymy:** Two or more lexical terms with different spellings and similar meanings.
- **Antonymy:** A pair of lexical terms with contrasting meanings.
- **Meronymy:** A relationship between a lexical term and a larger entity.

Semantic analysis techniques

The semantic analysis uses two distinct techniques to obtain information from text or corpus of data. The first technique refers to text classification, while the second relates to text extractor.

1. Semantic classification

Semantic classification implies text classification wherein predefined categories are assigned to the text for faster task completion. Following are the various types of text classification covered under semantic analysis:

- **Topic classification:** This classifies text into preset categories on the basis of the content type
- **Sentiment analysis:** Today, sentiment analysis is used by several social media platforms such as Twitter, Facebook, Instagram, and others to detect positive, negative, or neutral emotions hidden in text (posts, stories). Sentiment analysis helps brands identify dissatisfied customers or users in real-time and gets a hint on what customers feel about the brand as a whole.

- **Intent classification:** Intent classification refers to the classification of text based on customers' intentions in the context of what they intend to do next. You can use it to tag customers as 'interested' or 'not Interested' to effectively reach out to those customers who may intend to buy a product or show an inclination toward buying it.

2. Semantic extraction

Semantic extraction refers to extracting or pulling out specific data from the text.

Extraction types include:

- **Keyword extraction:** This technique helps identify relevant terms and expressions in the text and gives deep insights when combined with the above classification techniques.
- **Entity extraction:** As discussed in the earlier example, this technique is used to identify and extract entities in text, such as names of individuals, organizations, places, and others.

3.3 Ambiguity resolution

Natural Language Processing (NLP) is a field of research and application that analyses how with the help of machine we can comprehend and manipulate natural language for further exploration. In simple terms, atomic terms are words and composite terms are phrases. Words are the constitutional building block of language. Human languages either in spoken or written form, is composed of words. The NLP approaches related to word level are one of the initial steps towards comprehending the language.

The performance of NLP systems including machine translation, automatic question answering, information retrieval depends on the correct meaning of the text. Biggest challenge is ambiguity i.e., unclear or open meaning depending on the context of usage.

Types of Ambiguity

There are different forms of ambiguity that are relevant in natural language and, consequently, in artificial intelligence(AI) systems.

1. **Lexical Ambiguity:** This type of ambiguity represents words that can have multiple assertions. For instance, in English, the word "back" can be a noun (back stage), an adjective (back door) or an adverb (back away).
2. **Syntactic Ambiguity:** This type of ambiguity represents sentences that can be parsed in multiple syntactical forms. Take the following sentence: " I heard his cell phone ring in my

office”. The propositional phrase “in my office” can be parsed in a way that modifies the noun or on another way that modifies the verb.

3. **Semantic Ambiguity:** This type of ambiguity is typically related to the interpretation of sentence. For instance, the previous sentence used in the previous point can be interpreted as if I was physically present in the office or as if the cell phone was in the office.
4. **Metonymy:** The most difficult type of ambiguity, metonymy deals with phrases in which the literal meaning is different from the figurative assertion. For instance, when we say “Samsung us screaming for new management”, we don’t really mean that the company is literally screaming (although you never know with Samsung these days).

Metaphors

Metaphors are a specific type of metonymy on which a phrase with one literal meaning is used as an analogy to suggest a different meaning. For example, if we say: “Roger Clemens was painting the corners”, we are not referring to the former NY Yankee star working as a painter.

From a conceptual standpoint, metaphors can be seen as a type of metonymy on which the relationship between sentences is based on similarity.

Ambiguity in Natural Language Processing can be removed using:

- Word Sense Disambiguation
- Part of Speech Tagger
- HMM (Hidden Markov Model) Tagger
- Hybrid combination of taggers with machine learning techniques.

Simile

A simile is a figure of speech that involves comparing one thing to another using the words "like" or "as" to highlight a similarity between them. Similes are often used to make descriptions more vivid, expressive, or relatable. Here are some examples of similes:

As brave as a lion. (*Meaning:* Very courageous or fearless.)

As busy as a bee. (*Meaning:* Very industrious or constantly occupied.)

3.4 Other strategies for semantic interpretation notes in NLP

Semantic interpretation in natural language processing (NLP) involves understanding the meaning of text or speech. It's a broad field, and various strategies can be employed. Here strategies for semantic interpretation in NLP:

1. Word Embeddings

Utilize pre-trained word embeddings like Word2Vec, GloVe, or FastText to represent words as dense vectors. This can capture semantic relationships between words.

2. Contextual Embeddings

Explore contextual embeddings such as BERT (Bidirectional Encoder Representations from Transformers) or GPT (Generative Pre-trained Transformer) for contextualized word representations. These models consider the surrounding context, providing a more nuanced understanding of meaning.

3. Semantic Role Labeling

Employ SRL models to identify the roles of different words in a sentence. This helps in understanding the relationships between entities and actions.

4. Dependency Parsing

Use dependency parsing to analyze the grammatical structure of a sentence. Understanding the dependencies between words can contribute to a better comprehension of the meaning.

5. Knowledge Graphs

Integrate information from knowledge graphs like ConceptNet or DBpedia to enhance semantic understanding by leveraging relationships between entities.

6. Named Entity Recognition

Incorporate NER models to identify and classify entities in text, such as persons, organizations, locations, etc. This helps in understanding the context of the document.

7. Co-reference Resolution

Resolve co-references in a document to link pronouns and noun phrases to the entities they refer to, ensuring a coherent and accurate interpretation.

8. Sentiment Analysis

Combine sentiment analysis techniques to understand the emotional tone or sentiment expressed in the text, providing additional context to the semantic interpretation.

9. Syntax-Driven Approaches

Explore syntactic structures and grammar rules to interpret the meaning of sentences. Syntax can provide valuable information about relationships and hierarchies in language.

10. Transfer Learning

Leverage transfer learning techniques to fine-tune pre-trained models on specific semantic interpretation tasks. This can be particularly useful when dealing with limited labeled data.

11. Rule-Based Systems

Develop rule-based systems to capture specific patterns or semantic relationships in the data. This can be useful in cases where explicit rules define the semantics.

12. Domain-Specific Models

Train or fine-tune models on domain-specific data to improve semantic interpretation in specialized fields like medicine, law, or finance.

13. Hybrid Approaches

Combine multiple strategies, such as rule-based systems, machine learning models, and knowledge graphs, in a hybrid approach to benefit from the strengths of each.

14. Collective interpretation

In linguistics it refers to the understanding of a group or collection of individuals as a single unit. This concept is particularly relevant when analyzing the semantics and pragmatics of language. Collective interpretation involves treating a plural noun phrase as a single entity or group rather than as a collection of individual members. Examples are Collective Nouns, Plural Noun Phrases with Singular Interpretation and Collective Pronouns.

3.5 Scoping for interpretation of noun phrases

Scoping in the interpretation of noun phrases in NLP involves defining the boundaries and context within which a noun phrase should be understood. It's about determining the extent to which the interpretation should cover, including the entities involved, relationships with other phrases, and the overall context. Here are some considerations for scoping in the interpretation of noun phrases:

- **Local Context**

Define the immediate context around the noun phrase. Consider the words and phrases in proximity that might influence the interpretation. This could involve a few words to a complete sentence.

- **Syntactic Scope**

Analyze the syntactic structure of the sentence to understand the grammatical relationships between the noun phrase and other parts of the sentence. Dependency parsing can be particularly useful in determining the syntactic scope.

- **Coreference Resolution**

Resolve coreferences to ensure that pronouns or other expressions referring to the noun phrase are correctly identified and linked. This expands the scope by associating the noun phrase with its referents.

- **Named Entity Recognition**

If the noun phrase contains named entities, consider the scope of those entities. NER can help identify entities and their boundaries, contributing to a more accurate interpretation.

- **Semantic Role Labeling**

Understand the roles played by the noun phrase in the context of the sentence. SRL can help identify the semantic roles of different constituents, providing insights into the actions and relationships involved.

- **Temporal and Spatial Context**

Consider the temporal and spatial context to determine if there are specific time or location constraints on the interpretation of the noun phrase. This is especially relevant in cases where the meaning is dependent on when or where the statement is made.

- **Knowledge Base Integration**

Utilize external knowledge bases or ontologies to enhance the scoping. This involves linking noun phrases to relevant entities in a knowledge base to extract additional information and context.

- **Pragmatic Considerations**

Take into account pragmatic factors, such as the speaker's intent, the conversational context, and the overall discourse structure. Pragmatic considerations can help refine the scope based on the speaker's communicative goals.

- **Domain-Specific Scoping Rules**

Develop domain-specific scoping rules if the interpretation task is focused on a particular domain. Different domains may have specific conventions or constraints that influence how noun phrases should be interpreted.

- **Hierarchical Relationships**

Explore hierarchical relationships within the text, especially in cases where the noun phrase is part of a larger structure. This involves understanding the hierarchical organization of information.

- **Dynamic Scoping**

Consider dynamic scoping, where the scope of interpretation may evolve as more information is processed. This is particularly relevant in interactive or real-time processing scenarios.

Scoping in noun phrase interpretation is a nuanced task that often requires a combination of syntactic, semantic, and contextual analysis. Integrating various NLP techniques and leveraging external knowledge can contribute to more accurate and comprehensive scoping in the interpretation of noun phrases.

3.6 Semantic attachments

In Natural Language Processing (NLP), semantic attachments typically refer to the association between words and their meanings or senses within a given context. This concept is closely related to word sense disambiguation, a task in NLP that aims to determine the correct sense or meaning of a word in a particular instance.

Semantic attachments are crucial for understanding the nuances of language, as many words can have multiple meanings depending on the context in which they are used. Here's how semantic attachments are relevant in NLP:

1. **Word Sense Disambiguation (WSD):**

- In NLP, one common application of semantic attachments is in Word Sense Disambiguation. WSD algorithms attempt to determine the correct sense of a word in context by considering the surrounding words and the meanings associated with those words. This involves identifying the semantic attachment of the target word within a specific instance of its usage.

2. **Lexical Semantics:**

- Semantic attachments are integral to understanding the lexical semantics of words. Lexical semantics focuses on the meaning of individual words and how their meanings relate to each other. Establishing semantic attachments helps NLP systems navigate the complexities of word meanings and improve the accuracy of language understanding.

3. **Embeddings and Representation Learning:**

- Word embeddings, such as Word2Vec, GloVe, and embeddings learned through neural networks, capture semantic relationships between words in vector spaces.

These embeddings essentially encode semantic attachments by representing words with vectors in a way that captures their contextual and semantic similarities.

4. **Semantic Role Labeling (SRL):**

- In tasks like Semantic Role Labeling, understanding the semantic roles of words in a sentence involves recognizing how each word contributes to the overall meaning of the sentence. This requires identifying the semantic attachments of verbs, nouns, and other parts of speech.

5. **Knowledge Graphs and Ontologies:**

- Semantic attachments play a role in linking words or entities to specific concepts in knowledge graphs or ontologies. This linking helps create a structured representation of knowledge, allowing NLP systems to leverage semantic relationships for improved understanding.

3.7 Word Sense:

Word sense, in the context of semantic attachment, refers to the specific meaning or interpretation of a word in a given context. Many words in natural language have multiple senses, and determining the correct sense in a particular instance is essential for accurate language understanding. Semantic attachment involves associating words with their meanings or senses based on the context in which they are used.

Here's a breakdown of how word sense is relevant to semantic attachment:

1. **Ambiguity:**

- Words often have multiple meanings, and the process of disambiguating these meanings in a specific context is known as word sense disambiguation (WSD). Semantic attachment helps associate the correct sense with a word in a given instance, enhancing the accuracy of language processing tasks.

2. **Contextual Understanding:**

- The meaning of a word can change based on the context in which it appears. Semantic attachment involves capturing the contextual nuances that influence the interpretation of a word. For example, the word "bank" could refer to a financial institution or the side of a river, and the correct sense depends on the context.

3. **Lexical Semantics:**

- Lexical semantics focuses on understanding the meanings of individual words. Semantic attachment is a crucial aspect of lexical semantics because it involves linking words to their specific senses. This understanding contributes to the overall comprehension of language.

4. Word Embeddings:

- Word embeddings, such as Word2Vec or GloVe, capture semantic relationships between words by representing them as vectors in a continuous vector space. These embeddings help in understanding word sense through the context in which words co-occur in a given dataset.

5. Machine Translation and NLP Applications:

- In tasks like machine translation, sentiment analysis, and information retrieval, accurately capturing word sense is essential for generating meaningful and contextually appropriate output. Semantic attachment ensures that words are interpreted correctly in various NLP applications.

6. Knowledge Graphs:

- In the context of knowledge graphs or ontologies, words are linked to specific concepts, and these associations represent semantic attachments. Understanding word senses in this context enables the creation of structured knowledge representations.

3.8 Relations Between the senses:

In the context of lexical semantics and natural language processing, words often have multiple senses, and the relationships between these senses play a crucial role in understanding the complexity of language. Here are some common relations between senses of words:

1. Synonymy:

- Synonymy refers to a relationship between senses where two or more words have similar meanings. For example, "small" and "little" are synonyms.

2. Antonymy:

- Antonymy is a relationship between senses where two words have opposite meanings. For instance, "hot" and "cold" are antonyms.

3. Hyponymy/Hypernymy:

- Hyponymy represents a hierarchical relationship where one sense (hyponym) is a more specific instance of another (hypernym). For example, "rose" is a hyponym of "flower," and "flower" is the hypernym of "rose."

4. **Holonymy/Meronymy:**

- Holonymy denotes a part-whole relationship. A holonym includes the whole, and a meronym includes a part. For instance, "tree" is a holonym of "branch," and "branch" is a meronym of "tree."

5. **Polysemy:**

- Polysemy refers to a situation where a single word has multiple related meanings. The senses are related but not necessarily synonymous. For example, the word "bank" can mean a financial institution or the side of a river.

6. **Co-hyponymy:**

- Co-hyponyms are words that share a common hypernym but are not necessarily synonyms. For example, "dog" and "cat" are co-hyponyms because they share the hypernym "animal."

7. **Troponymy:**

- Troponymy is a relationship between verbs where one verb represents a manner or means of performing the action of another verb. For instance, "run" and "walk" are troponyms of the more general verb "move."

8. **Antemeronymy:**

- Antemeronymy is a rare relation where a part is associated with the whole. An example is "cog" being an antemeronym of "gear."

3.9 Scoping Phenomena

"Scoping phenomena" typically refers to linguistic phenomena related to the interpretation and hierarchical arrangement of certain linguistic elements, especially quantifiers and logical operators, within a sentence. The term "scope" in linguistics refers to the range or domain over which an operator or quantifier applies.

Here are a couple of common examples of scoping phenomena:

Quantifier Scope

Scope Ambiguity

Logical Operators

3.10 Tree banks in Grammar:

Treebanks are widely used in the Natural Language Processing (NLP) community to support the creation and training of parsers and taggers, work on machine translation and speech recognition, and research on joint syntactic and semantic role labeling. Treebanks have also been used as the basis for downstream annotation projects, such as PropBanks, the Penn Discourse Treebank and word alignment.

Treebanks are fully parsed corpora that are manually annotated for syntactic structure at the sentence level and for part-of-speech or morphological information at the token level. Every token and every sentence in the text is annotated.

Goal of Treebanking

To represent useful linguistic structure in an accessible way, including:

- Consistent annotation
- Searchable trees
- “Correct” linguistic analysis if possible, but at a minimum consistent and searchable
- Annotation useful to both linguistic and NLP communities
- Empirical methods providing portability to new languages
- Structures that can be used as the basis for additional downstream annotation

Types of treebanks in grammar:

1. Syntax Trees:

Treebanks consist of sentences that have been manually annotated with syntax trees. Each node in the tree represents a word or a group of words, and the edges indicate grammatical relationships.

2. Constituency and Dependency Trees:

Treebanks can use either constituency trees or dependency trees to represent the syntactic structure of sentences. Constituency trees break down sentences into constituents (phrases and clauses), while dependency trees focus on the relationships between words.

3. Penn Treebank:

The Penn Treebank is a well-known treebank for English. It has played a significant role in the development of syntactic annotation standards and has been widely used in research and the development of NLP tools.

4. Multilingual Treebanks:

Treebanks are created for various languages, allowing researchers to study and develop syntactic models for different linguistic contexts. Efforts like the Universal Dependencies project aim to create cross-linguistically consistent treebanks.

Questions to Revise:

1. Semantic interpretation - Challenges and opportunities
2. Semantic interpretation pipeline in NLP – Preprocessing Methods
3. NLP interpret figurative language
4. Discourse models and external knowledge
5. Types of ambiguity
6. Feature classification in syntactic analysis
7. Tree banks used in grammar based models in NLP
8. Parsing techniques in syntactic NLP