UNIT-III

SEMANTIC ANALYSIS AND DISCOURSE PROCESSING

1. SEMANTIC ANALYSIS

- Semantic Analysis is a subfield of Natural Language Processing (NLP) that attempts to understand the meaning of Natural Language.
- Understanding Natural Language might seem a straightforward process to us as humans.
- However, due to the vast complexity and subjectivity involved in human language, interpreting it is quite a complicated task for machines.
- Semantic Analysis of Natural Language captures the meaning of the given text while taking into account context, logical structuring of sentences and grammar roles.

Parts of Semantic Analysis

Semantic Analysis of Natural Language can be classified into two broad parts:

1. **Lexical Semantic Analysis**: Lexical Semantic Analysis involves understanding the meaning of each word of the text individually. It basically refers to fetching the dictionary meaning that a word in the text is deputed to carry.
2. **Compositional Semantics Analysis**: Although knowing the meaning of each word of the text is essential, it is not sufficient to completely understand the meaning of the text. For example, consider the following two sentences:

- Sentence 1: Students love GeeksforGeeks.
- Sentence 2: GeeksforGeeks loves Students.
- Although both these sentences 1 and 2 use the same set of root words {student, love, geeksforgeeks}, they convey entirely different meanings.
- Hence, under Compositional Semantics Analysis, we try to understand how combinations of individual words form the meaning of the text.

Tasks involved in Semantic Analysis

In order to understand the meaning of a sentence, the following are the major processes involved in Semantic Analysis:

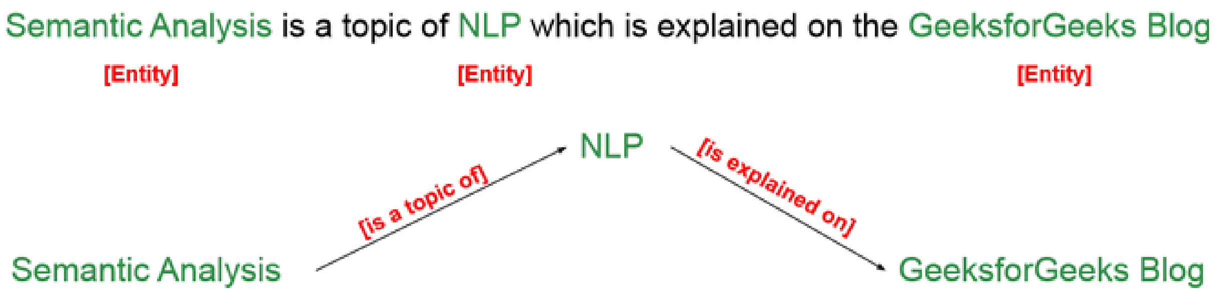
- Word Sense Disambiguation
- Relationship Extraction

a. Word Sense Disambiguation:

- In Natural Language, the meaning of a word may vary as per its usage in sentences and the context of the text.
- Word Sense Disambiguation involves interpreting the meaning of a word based upon the context of its occurrence in a text.
- For example, the word ‘Bark’ may mean ‘the sound made by a dog’ or ‘the outermost layer of a tree.’
- Likewise, the word ‘rock’ may mean ‘a stone‘ or ‘a genre of music‘ – hence, the accurate meaning of the word is highly dependent upon its context and usage in the text.
- Thus, the ability of a machine to overcome the ambiguity involved in identifying the meaning of a word based on its usage and context is called Word Sense Disambiguation.

b. Relationship Extraction:

- Another important task involved in Semantic Analysis is Relationship Extracting. It involves firstly identifying various entities present in the sentence and then extracting the relationships between those entities.
- For example, consider the following sentence: Semantic Analysis is a topic of NLP which is explained on the GeeksforGeeks blog. The entities involved in this text, along with their relationships, are shown below.



Elements of Semantic Analysis

Some of the critical elements of Semantic Analysis that must be scrutinized and taken into account while processing Natural Language are:

- **Hyponymy:** Hyponymys refers to a term that is an instance of a generic term. They can be understood by taking class-object as an analogy. For example: '*Color*' is a hypernymy while '*grey*', '*blue*', '*red*', etc, are its hyponyms.
- **Homonymy:** Homonymy refers to two or more lexical terms with the same spellings but completely distinct in meaning. For example: '*Rose*' might mean '*the past form of rise*' or '*a flower*', – same spelling but different meanings; hence, '*rose*' is a homonymy.
- **Synonymy:** When two or more lexical terms that might be spelt distinctly have the same or similar meaning, they are called Synonymy. For example: *(Job, Occupation), (Large, Big), (Stop, Halt).*
- **Antonymy:** Antonymy refers to a pair of lexical terms that have contrasting meanings – they are symmetric to a semantic axis. For example: *(Day, Night), (Hot, Cold), (Large, Small).*
- **Polysemy:** Polysemy refers to lexical terms that have the same spelling but multiple closely related meanings. It differs from homonymy because the meanings of the terms need not be closely related in the case of homonymy. For example: '*man*' may mean '*the human species*' or '*a male human*' or '*an adult male human*' – since all these different meanings bear a close association, the lexical term '*man*' is a polysemy.
- **Meronomy:** Meronomy refers to a relationship wherein one lexical term is a constituent of some larger entity. For example: '*Wheel*' is a meronym of '*Automobile*'

Meaning Representation

While, as humans, it is pretty simple for us to understand the meaning of textual information, it is not so in the case of machines. Thus, machines tend to represent the text in specific formats in order to interpret its meaning. This formal structure that is used to understand the meaning of a text is called meaning representation.

Basic Units of Semantic System:

In order to accomplish Meaning Representation in Semantic Analysis, it is vital to understand the building units of such representations. The basic units of semantic systems are explained below:

1. **Entity:** An entity refers to a particular unit or individual in specific such as a person or a location. For example GeeksforGeeks, Delhi, etc.
2. **Concept:** A Concept may be understood as a generalization of entities. It refers to a broad class of individual units. For example Learning Portals, City, Students.
3. **Relations:** Relations help establish relationships between various entities and concepts. For example: 'GeeksforGeeks is a Learning Portal', 'Delhi is a City.', etc.
4. **Predicate:** Predicates represent the verb structures of the sentences.

In Meaning Representation, we employ these basic units to represent textual information.

Approaches to Meaning Representations:

Now that we are familiar with the basic understanding of Meaning Representations, here are some of the most popular approaches to meaning representation:

1. First-order predicate logic (FOPL)
2. Semantic Nets
3. Frames
4. Conceptual dependency (CD)
5. Rule-based architecture
6. Case Grammar
7. Conceptual Graphs

Semantic Analysis Techniques

Based upon the end goal one is trying to accomplish, Semantic Analysis can be used in various ways. Two of the most common Semantic Analysis techniques are:

Text Classification

In-Text Classification, our aim is to label the text according to the insights we intend to gain from the textual data.

For example:

- In **Sentiment Analysis,** we try to label the text with the prominent emotion they convey. It is highly beneficial when analyzing customer reviews for improvement.
- In **Topic Classification**, we try to categories our text into some predefined categories. For example: Identifying whether a research paper is of Physics, Chemistry or Maths
- In **Intent Classification**, we try to determine the intent behind a text message. For example: Identifying whether an e-mail received at customer care service is a query, complaint or request.

Text Extraction

In-Text Extraction, we aim at obtaining specific information from our text. For Example,

- In **Keyword Extraction**, we try to obtain the essential words that define the entire document.

- In **Entity Extraction**, we try to obtain all the entities involved in a document.

Significance of Semantics Analysis

- Semantics Analysis is a crucial part of Natural Language Processing (NLP). In the ever-expanding era of textual information, it is important for organizations to draw insights from such data to fuel businesses.
- Semantic Analysis helps machines interpret the meaning of texts and extract useful information, thus providing invaluable data while reducing manual efforts.
- Besides, Semantics Analysis is also widely employed to facilitate the processes of automated answering systems such as chatbots – that answer user queries without any human interventions.

****

2. Lexical Semantics

- Lexical semantics in Natural Language Processing (NLP) deals with the meaning of words and their relationships.
- It is a crucial area for understanding language, enabling machines to process and interpret human text effectively. Here are key aspects of lexical semantics in NLP

Word Meaning Representation

- **Lexemes & Words:** A lexeme is the base unit of meaning (e.g., "run" includes "runs," "running," and "ran").
- **Orthographic Form**-This refers to the way a lexeme is written or spelled in a particular language. Example: The word "cat" has the orthographic form c-a-t in English.
- **Phonological Form**-This refers to the way a lexeme is pronounced, represented using phonetic symbols. Example: The word "cat" is pronounced as /kæt/ in the International Phonetic Alphabet (IPA).
- **Sense Disambiguation:** A word can have multiple meanings (e.g., "bank" as a financial institution vs. a riverbank). NLP techniques like **Word Sense Disambiguation (WSD)** help determine the correct sense based on context.

Relations among Lexemes and Their Senses

a. Homonymy

Homonymy refers to the phenomenon where two or more words have the **same spelling or pronunciation** but different, **unrelated meanings**. In **Natural Language Processing (NLP)**, homonyms pose a challenge because they require **contextual understanding** to determine the correct meaning.

Types of Homonymy

1. **Homophones** – Words that **sound the same** but have different meanings and spellings.
   - Example:
     - *Flour* (used in baking) vs. *Flower* (a plant).
     - *Pair* (a set of two) vs. *Pear* (a fruit).
2. **Homographs** – Words that are **spelled the same** but have different pronunciations and meanings.

  - Example:
    - *Lead* **(/lɛd/)** (a type of metal) vs. *Lead* **(/liːd/)** (to guide).
    - *Bass* **(/bæs/)** (a type of fish) vs. *Bass* **(/beɪs/)** (low-frequency sound).
3. **Pure Homonyms** – Words that are **both spelled and pronounced the same** but have different meanings.
  - Example:
    - *Bank* (a financial institution) vs. *Bank* (side of a river).
    - *Bat* (a flying mammal) vs. *Bat* (sports equipment).

b. Polysemy

Polysemy refers to a single word having **multiple related meanings**. Unlike **homonyms**, which have completely unrelated meanings, **polysemous words share a common semantic origin** but are used in different contexts.

Examples of Polysemy

1. **"Paper"**
   - *A material* → "I wrote on a **paper**."
   - *A research article* → "She published a **paper** in a journal."
   - *A newspaper* → "I read the **paper** this morning."
2. **"Run"**
   - *To move quickly* → "He can **run** fast."
   - *To operate* → "She **runs** a business."
   - *To function* → "The machine is **running** smoothly."

Polysemy in NLP and Its Challenges

Polysemy creates **ambiguity** in language processing tasks such as:

- **Word Sense Disambiguation (WSD)** – Determining the correct meaning based on context.
- **Machine Translation** – Avoiding incorrect translations for polysemous words.
- **Information Retrieval** – Improving search engines by identifying intended meanings.

Techniques to Handle Polysemy in NLP

1. **Word Embeddings** –
   - **Traditional embeddings** (Word2Vec, GloVe) represent words with a **single vector**, making it hard to distinguish polysemous meanings.
   - **Contextual embeddings** (BERT, ELMo) generate different vectors for a word **based on context**, helping disambiguate meanings.
2. **Word Sense Disambiguation (WSD)** –
   - **Supervised approaches**: Train models on labeled datasets with word senses.
   - **Unsupervised approaches**: Cluster similar word usages using techniques like **sense embeddings**.
3. **Lexical Databases** –
   - **WordNet** provides structured definitions for different senses of polysemous words.

c. Synonymy

Synonymy in lexical semantics refers to the relationship between words that have the same or nearly the same meaning in certain contexts. In Natural Language Processing (NLP), understanding synonymy is crucial for various tasks like text similarity, information retrieval, and machine translation.

Types of Synonyms

1. **Absolute Synonyms** – Words that are completely interchangeable in all contexts (rare in natural language). Example: *"car"* vs. *"automobile"*
2. **Near-Synonyms** – Words with similar meanings but slight differences in connotation, formality, or usage. Example: *"big"* vs. *"large"* (subtle differences in usage).
3. **Context-Dependent Synonyms** – Words that are synonymous only in certain situations. Example: *"buy"* and *"purchase"* (formal vs. informal).

d. Hyponymy

Hyponymy is a lexical semantic relationship where a word (hyponym) has a more specific meaning than another word (hypernym). In NLP, recognizing hyponymy is important for semantic understanding, text classification, and information retrieval.

- **Hyponym**: A more specific word within a category.
- **Hypernym (Superordinate)**: A more general word that includes hyponyms.

Examples

Hypernym (General)	**Hyponym (Specific)**
Animal	Dog, Cat, Elephant
Vehicle	Car, Bike, Truck
Fruit	Apple, Banana, Mango

- "Dog" is a hyponym of "Animal".
- "Animal" is a hypernym of "Dog".

e. WordNet: A Database of Lexical Relations

WordNet is a large lexical database of English, developed at Princeton University, that groups words into sets of synonyms (synsets) and captures semantic relationships between them. It is widely used in Natural Language Processing (NLP) for tasks like word sense disambiguation, information retrieval, and text classification.

Structure of WordNet

WordNet organizes words into **synsets (synonym sets)** and defines their **semantic relationships**:

(a) Synsets (Synonym Sets)

Each synset represents a unique **concept** and contains words that are synonymous in at least one sense.

- Example:
  - Synset for **"car"**: *{car, automobile, motorcar}*
  - Synset for **"big"**: *{big, large}*

(b) Lexical Relations in WordNet

Relation	Description	Example
Synonymy	Words with similar meanings	*big ↔ large*
Antonymy	Words with opposite meanings	*hot ↔ cold*
Hyponymy	"Is-a" relation (more specific term)	*dog → animal*
Hypernymy	"Is-a" relation (more general term)	*animal → living thing*
Meronymy	"Part-of" relation	*wheel → car*
Holonymy	"Whole-of" relation	*car → vehicle*
Troponymy	"Manner-of" relation (verbs)	*whisper → talk*

****

3. Ambiguity in NLP

Ambiguity in Natural Language Processing (NLP) **happens because human language can have multiple meanings**. Computers sometimes confuse to understand exactly what we mean unlike humans, who can use intuition and background knowledge to infer meaning, computers rely on precise algorithms and statistical patterns.

The sentence **"The chicken is ready to eat"** is ambiguous because it can be interpreted in two different ways:

1. The chicken is cooked and ready to be eaten.
2. The chicken is hungry and ready to eat food.

This dual meaning arises from the structure of the sentence, which does not clarify the subject's role (the eater or the one being eaten). Resolving such ambiguities is essential for accurate NLP applications like chatbots, translation, and sentiment analysis.

Types of Ambiguity in NLP

The meaning of an ambiguous expression often depends on the situation, prior knowledge, or surrounding words. **For example: *He is cool.*** This could mean ***he is calm under pressure*** **or *he is fashionable* depending on the context.**

1. Lexical Ambiguity

- Lexical ambiguity occurs when a single word has multiple meanings, making it unclear which meaning is intended in a particular context. This is a common challenge in language.
- *For example, the word "**bat**" can have two different meanings. It could refer to a **flying mammal**, like the kind you might see at night. Alternatively, "bat" could also refer to a **piece of sports equipment** used in games like baseball or cricket.*

- For computers, determining the correct meaning of such a word requires looking at the surrounding context to decide which interpretation makes sense.

2. **Syntactic Ambiguity**

- Syntactic ambiguity occurs when the structure or grammar of a sentence allows for more than one interpretation.
- This happens because the sentence can be understood in different ways depending on how it is put together.
- For example, take the sentence, “The boy kicked the ball in his jeans.” This sentence can be interpreted in two different ways: one possibility is that the boy was wearing jeans and he kicked the ball while he was wearing them.
- Another possibility is that the ball was inside the boy’s jeans, and he kicked the ball out of his jeans.
- A computer or NLP system must carefully analyze the structure to figure out which interpretation is correct, based on the context.

3. **Semantic Ambiguity**

- Semantic ambiguity occurs when a sentence has more than one possible meaning because of how the words are combined.
- This type of ambiguity makes it unclear what the sentence is truly trying to say.
- For example, take the sentence, “Visiting relatives can be annoying.”
- This sentence could be understood in two different ways. One meaning could be that relatives who are visiting you are annoying, implying that the relatives themselves cause annoyance.
- Another meaning could be that the act of visiting relatives is what is annoying, suggesting that the experience of going to see relatives is unpleasant.
- The confusion comes from how the words "visiting relatives" can be interpreted: is it about the relatives who are visiting, or is it about the action of visiting?
- In cases like this, semantic ambiguity makes it hard to immediately understand the exact meaning of the sentence, and the context is needed to clarify it.

4. **Pragmatic Ambiguity**

- Pragmatic ambiguity occurs when the meaning of a sentence depends on the speaker’s intent, tone, or the situation in which it is said.
- This type of ambiguity is common in everyday conversations, and it can be tricky for computers to understand because it often requires knowing the broader context.
- For example, consider the sentence, “Can you open the window?” In one situation, it could be understood as a literal question asking if the person is physically able to open the window.
- However, in another context, it could be a polite request, where the speaker is asking the listener to open the window, even though they’re not directly giving an order.
- The meaning changes based on the tone of voice or social context, which is something that is difficult for NLP systems to capture without understanding the surrounding situation

5. **Referential Ambiguity**

- Referential ambiguity occurs when a pronoun (like "he," "she," "it," or "they") or a phrase is unclear about what or who it is referring to.
- This type of ambiguity happens when the sentence doesn’t provide enough information to determine which person, object, or idea the pronoun is referring to.

- For example, consider the sentence, “Alice told Jane that she would win the prize.” In this case, it’s unclear whether the pronoun "she" refers to Alice or Jane.
- Both could be possible interpretations, and without further context, we can’t be sure.
- If the sentence was about a competition, "she" could be referring to Alice, meaning Alice is telling Jane that she would win the prize.
- However, it could also mean that Alice is telling Jane that Jane would win the prize.

6. Ellipsis Ambiguity

Ellipsis ambiguity happens when part of a sentence is left out, making it unclear what the missing information is. This often occurs in everyday conversation or writing when people try to be brief and omit words that are understood from the context.

*For example, consider the sentence, "**John likes apples, and Mary does too.**" The word "**does**" is a shortened form of "likes apples," but it’s not explicitly stated. This creates two possible interpretations:*

1. ***Mary likes apples** just like John, meaning both John and Mary enjoy apples.*
2. ***Mary likes something else** (not apples), and the sentence is leaving out the specific thing she likes.*

The ambiguity arises because it's unclear from the sentence whether **"does"** refers to liking apples or something else.

Addressing Ambiguity in Natural Language Processing

- To address ambiguity in NLP, several methods are used to accurately interpret language.
- Contextual analysis is one of the key approaches, where surrounding words and context help determine the correct meaning of a word or phrase.
- Word sense disambiguation (WSD) resolves lexical ambiguity by using context to identify which meaning of a word is being used.
- Parsing and syntactic analysis help resolve syntactic ambiguity by breaking down sentence structures to understand different grammatical interpretations.
- Coreference resolution is used to clarify what pronouns or phrases refer to, solving referential ambiguity.
- Discourse and pragmatic modeling help capture speaker intent and the social context, which resolves pragmatic ambiguity.
- Machine learning and deep learning techniques, like BERT and GPT, leverage large datasets to learn language patterns, aiding in resolving ambiguity.

****

4. Word Sense Disambiguation in NLP

- **Word sense disambiguation (WSD**) in Natural Language Processing (NLP) is the problem of identifying which “sense” (meaning) of a word is activated by the use of the word in a particular context or scenario.
- In people, this appears to be a largely unconscious process. The challenge of correctly identifying words in NLP systems is common, and determining the specific usage of a word in a sentence has many applications.
- The application of Word Sense Disambiguation involves the area of Information Retrieval, Question Answering systems, Chat-bots, etc.
- Word Sense Disambiguation (WSD) is a subtask of Natural Language Processing that deals with the problem of identifying the correct sense of a word in context.

- Many words in natural language have multiple meanings, and WSD aims to disambiguate the correct sense of a word in a particular context.
- For example, the word “bank” can have different meanings in the sentences “I deposited money in the bank” and “The boat went down the river bank”.
- WSD is a challenging task because it requires understanding the context in which the word is used and the different senses in which the word can be used. Some common approaches to WSD include:

1. **Supervised learning:** This involves training a machine learning model on a dataset of annotated examples, where each example contains a target word and its sense in a particular context. The model then learns to predict the correct sense of the target word in new contexts.
2. **Unsupervised learning:** This involves clustering words that appear in similar contexts together, and then assigning senses to the resulting clusters. This approach does not require annotated data, but it is less accurate than supervised learning.
3. **Knowledge-based:** This involves using a knowledge base, such as a dictionary or ontology, to map words to their different senses. This approach relies on the availability and accuracy of the knowledge base.
4. **Hybrid:** This involves combining multiple approaches, such as supervised and knowledge-based methods, to improve accuracy.

WSD has many practical applications, including machine translation, information retrieval, and text-to-speech systems. Improvements in WSD can lead to more accurate and efficient natural language processing systems.

Difficulties in Word Sense Disambiguation

There are some difficulties faced by Word Sense Disambiguation (WSD).

- **Different Text-Corpus or Dictionary:** One issue with word sense disambiguation is determining what the senses are because different dictionaries and thesauruses divide words into distinct senses.
- Some academics have proposed employing a specific lexicon and its set of senses to address this problem.
- In general, however, research findings based on broad sense distinctions have outperformed those based on limited ones. The majority of researchers are still working on fine-grained WSD.
- **PoS Tagging:** Part-of-speech tagging and sense tagging have been shown to be very tightly coupled in any real test, with each potentially constraining the other.
- Both disambiguating and tagging with words are involved in WSM part-of-speech tagging.
- However, algorithms designed for one do not always work well for the other, owing to the fact that a word’s part of speech is mostly decided by the one to three words immediately adjacent to it, whereas a word’s sense can be determined by words further away.

Sense Inventories for Word Sense Disambiguation

Sense Inventories are the collection of abbreviations and acronyms with their possible senses. Some of the examples used in Word Sense Disambiguation are:

- **Princeton WordNet:** is a vast lexicographic database of English and other languages that is manually curated. For WSD, this is the de facto standard inventory. Its well-organized Synsets, or clusters of contextual synonyms, are nodes in a network.

- **BabelNet:** is a multilingual dictionary that covers both lexicographic and encyclopedic terminology. It was created by semi-automatically mapping numerous resources, including WordNet, multilingual versions of WordNet, and Wikipedia.
- **Wiktionary:** a collaborative project aimed at creating a dictionary for each language separately, is another inventory that has recently gained popularity.

Approaches for Word Sense Disambiguation

There are many approaches to Word Sense Disambiguation. The three main approaches are given below:

1. **Supervised:** The assumption behind supervised approaches is that the context can supply enough evidence to disambiguate words on its own (hence, world knowledge and reasoning are deemed unnecessary).
   Supervised methods for Word Sense Disambiguation (WSD) involve training a model using a labeled dataset of word senses. The model is then used to disambiguate the sense of a target word in new text. Some common techniques used in supervised WSD include:
   a) **Decision list:** A decision list is a set of rules that are used to assign a sense to a target word based on the context in which it appears.
   b) **Neural Network:** Neural networks such as feedforward networks, recurrent neural networks, and transformer networks are used to model the context-sense relationship.
   c) **Support Vector Machines:** SVM is a supervised machine learning algorithm used for classification and regression analysis.
   d) **Naive Bayes:** Naive Bayes is a probabilistic algorithm that uses Bayes' theorem to classify text into predefined categories.
   e) **Decision Trees:** Decision Trees are a flowchart-like structure in which an internal node represents feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome.

Random Forest: Random Forest is an ensemble learning method for classification, regression, and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes.

- **Supervised WSD Exploiting Glosses:** Textual definitions are a prominent source of information in sense inventories (also known as glosses). Definitions, which follow the format of traditional dictionaries, are a quick and easy way to clarify sense distinctions
- **Purely Data-Driven WSD:** In this case, a token tagger is a popular baseline model that generates a probability distribution over all senses in the vocabulary for each word in a context.
- **Supervised WSD Exploiting Other Knowledge**: Additional sources of knowledge, both internal and external to the knowledge base, are also beneficial to WSD models. Some researchers use BabelNet translations to fine-tune the output of any WSD system by comparing the output senses' translations to the target's translations provided by an NMT system.

2. **Unsupervised:** The underlying assumption is that similar senses occur in similar contexts, and thus senses can be induced from the text by clustering word occurrences using some measure of similarity of context.
   - Using fixed-size dense vectors (word embeddings) to represent words in context has become one of the most fundamental blocks in several NLP systems.
   - Traditional word embedding approaches can still be utilized to improve WSD, despite the fact that they conflate words with many meanings into a single vector representation.

- Lexical databases (e.g., WordNet, ConceptNet, BabelNet) can also help unsupervised systems map words and their senses as dictionaries, in addition to word embedding techniques.

3. **Knowledge-Based:** It is built on the idea that words used in a text are related to one another, and that this relationship can be seen in the definitions of the words and their meanings.
   - The pair of dictionary senses having the highest word overlap in their dictionary meanings are used to disambiguate two (or more) words.
   - Lesk Algorithm is the classical algorithm based on Knowledge-Based WSD. Lesk algorithm assumes that words in a given "neighborhood" (a portion of text) will have a similar theme.
   - The dictionary definition of an uncertain word is compared to the terms in its eighbourhood in a simplified version of the Lesk algorithm.

****

5. Discourse Processing in NLP

Discourse processing in NLP (Natural Language Processing) is the study of how sentences and texts are structured to convey meaning beyond individual words or isolated sentences. It involves understanding the relationships between sentences, coherence, reference resolution, and overall text structure.

Key Aspects of Discourse Processing

1. **Cohesion and Coherence**
   - *Cohesion*: How sentences are linked using linguistic devices (e.g., pronouns, conjunctions, and lexical repetition).
   - *Coherence*: How ideas are logically connected to maintain meaning across the text.
2. **Anaphora and Coreference Resolution**
   - *Anaphora resolution*: Identifying what a pronoun or referring expression refers to. Example:
     - *John went to the store. He bought some milk.* → *He* refers to *John*.
   - *Coreference resolution*: Finding all expressions that refer to the same entity in a text.
3. **Discourse Connectives and Relations**
   - Words like *however*, *therefore*, *because*, and *although* signal discourse relations like contrast, causation, and elaboration.
4. **Rhetorical Structure Theory (RST)**
   - RST explains how text segments are related hierarchically to form a coherent structure.
5. **Topic Segmentation and Shift Detection**
   - Identifying changes in topics within a text or conversation.
6. **Dialogue and Conversational Analysis**
   - Understanding multi-turn conversations in chatbots, virtual assistants, and customer service.
7. **Discourse Parsing**
   - Identifying the structure of a text, such as relationships between clauses and sentences.

Applications of Discourse Processing in NLP

- **Chatbots and Virtual Assistants** (e.g., Siri, Alexa)
- **Machine Translation** (e.g., Google Translate improving coherence in translated text)
- **Text Summarization** (e.g., extracting key points while preserving meaning)
- **Sentiment Analysis** (e.g., understanding context in customer reviews)
- **Question Answering Systems** (e.g., retrieving answers based on discourse context)

****

6. Cohesion in NLP

- Coherence and discourse structure are interconnected in many ways. Coherence, along with property of good text, is used to evaluate the output quality of natural language generation system.

The coherent discourse must possess the following properties

a. Coherence relation between utterances

The discourse would be coherent if it has meaningful connections between its utterances. This property is called coherence relation. For example, some sort of explanation must be there to justify the connection between utterances.

b. Relationship between entities

Another property that makes a discourse coherent is that there must be a certain kind of relationship with the entities. Such kind of coherence is called entity-based coherence.

c. Discourse structure

An important question regarding discourse is what kind of structure the discourse must have. The answer to this question depends upon the segmentation we applied on discourse. Discourse segmentations may be defined as determining the types of structures for large discourse. It is quite difficult to implement discourse segmentation, but it is very important for information retrieval, text summarization and information extraction kind of applications.

Algorithms for Discourse Segmentation

a. Unsupervised Discourse Segmentation

- The class of unsupervised discourse segmentation is often represented as linear segmentation.
- We can understand the task of linear segmentation with the help of an example. In the example, there is a task of segmenting the text into multi-paragraph units; the units represent the passage of the original text.
- These algorithms are dependent on cohesion that may be defined as the use of certain linguistic devices to tie the textual units together.
- On the other hand, lexicon cohesion is the cohesion that is indicated by the relationship between two or more words in two units like the use of synonyms.

b. Supervised Discourse Segmentation

- The earlier method does not have any hand-labeled segment boundaries. On the other hand, supervised discourse segmentation needs to have boundary-labeled training data.
- It is very easy to acquire the same. In supervised discourse segmentation, discourse marker or cue words play an important role.
- Discourse marker or cue word is a word or phrase that functions to signal discourse structure. These discourse markers are domain-specific.

Text Coherence

Lexical repetition is a way to find the structure in a discourse, but it does not satisfy the requirement of being coherent discourse. To achieve the coherent discourse, we must focus on coherence relations in specific. As we know that coherence relation defines the possible connection between utterances in a discourse. Hebb has proposed such kind of relations as follows. We are taking two terms $\mathbf{S_0}$ and $\mathbf{S_1}$ to represent the meaning of the two related sentences −

Result

It infers that the state asserted by term $\mathbf{S_0}$ could cause the state asserted by $\mathbf{S_1}$. For example, two statements show the relationship result: Ram was caught in the fire. His skin burned.

Explanation

It infers that the state asserted by $\mathbf{S_1}$ could cause the state asserted by $\mathbf{S_0}$. For example, two statements show the relationship − Ram fought with Shyam's friend. He was drunk.

Parallel

It infers p(a1,a2,…) from assertion of $\mathbf{S_0}$ and p(b1,b2,…) from assertion $\mathbf{S_1}$. Here ai and bi are similar for all i. For example, two statements are parallel − Ram wanted car. Shyam wanted money.

Elaboration

It infers the same proposition P from both the assertions − $\mathbf{S_0}$ and $\mathbf{S_1}$ For example, two statements show the relation elaboration: Ram was from Chandigarh. Shyam was from Kerala.

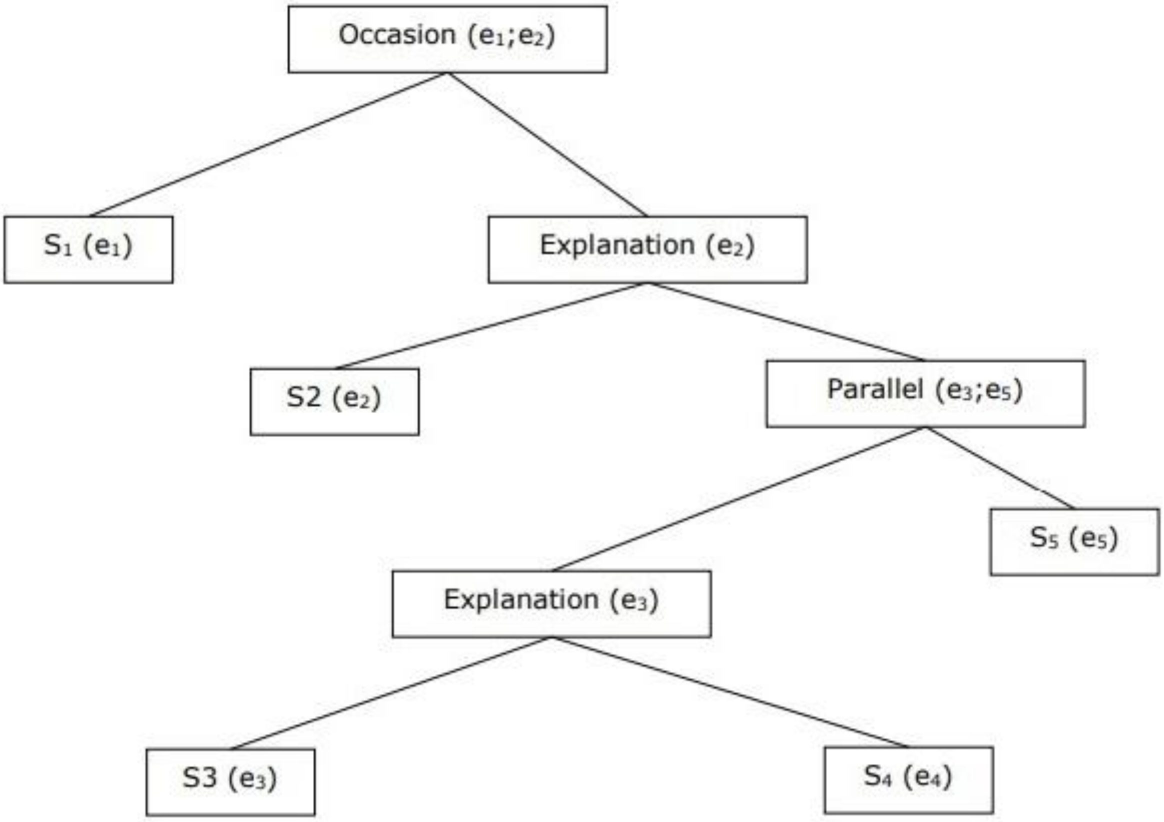
Occasion

It happens when a change of state can be inferred from the assertion of $\mathbf{S_0}$, final state of which can be inferred from $\mathbf{S_1}$ and vice-versa. For example, the two statements show the relation occasion: Ram picked up the book. He gave it to Shyam.

Building Hierarchical Discourse Structure

The coherence of entire discourse can also be considered by hierarchical structure between coherence relations. For example, the following passage can be represented as hierarchical structure −

- **S1** − Ram went to the bank to deposit money.
- **S2** − He then took a train to Shyam's cloth shop.
- **S3** − He wanted to buy some clothes.
- **S4** − He do not have new clothes for party.
- **S5** − He also wanted to talk to Shyam regarding his health



****

7. Reference Resolution

- Interpretation of the sentences from any discourse is another important task and to achieve this we need to know who or what entity is being talked about.
- Here, interpretation reference is the key element. **Reference** may be defined as the linguistic expression to denote an entity or individual.
- For example, in the passage, Ram, the manager of ABC bank, saw his friend Shyam at a shop.
- He went to meet him, the linguistic expressions like Ram, His, He are reference. On the same note, **reference resolution** may be defined as the task of determining what entities are referred to by which linguistic expression.

Terminology Used in Reference Resolution

We use the following terminologies in reference resolution −

- **Referring expression** − The natural language expression that is used to perform reference is called a referring expression. For example, the passage used above is a referring expression.
- **Referent** − It is the entity that is referred. For example, in the last given example Ram is a referent.
- **Corefer** − When two expressions are used to refer to the same entity, they are called corefers. For example, ***Ram*** and ***he*** are corefers.
- **Antecedent** − The term has the license to use another term. For example, ***Ram*** is the antecedent of the reference ***he***.
- **Anaphora & Anaphoric** − It may be defined as the reference to an entity that has been previously introduced into the sentence. And, the referring expression is called anaphoric.
- **Discourse model** − The model that contains the representations of the entities that have been referred to in the discourse and the relationship they are engaged in.

Types of Referring Expressions

The five types of referring expressions are described below −

a. Indefinite Noun Phrases

Such kind of reference represents the entities that are new to the hearer into the discourse context. For example − in the sentence Ram had gone around one day to bring him some food − some is an indefinite reference.

b. Definite Noun Phrases

Opposite to above, such kind of reference represents the entities that are not new or identifiable to the hearer into the discourse context. For example, in the sentence - I used to read The Times of India – The Times of India is a definite reference.

c. Pronouns

It is a form of definite reference. For example, Ram laughed as loud as he could. The word **he** represents pronoun referring expression.

d. Demonstratives

These demonstrate and behave differently than simple definite pronouns. For example, this and that are demonstrative pronouns.

e. Names

It is the simplest type of referring expression. It can be the name of a person, organization and location also. For example, in the above examples, Ram is the name-refereeing expression.

Reference Resolution Tasks

The two reference resolution tasks are described below.

a. Coreference Resolution

It is the task of finding referring expressions in a text that refer to the same entity. In simple words, it is the task of finding corefer expressions. A set of coreferring expressions are called coreference chain. For example - He, Chief Manager and His - these are referring expressions in the first passage given as example.

b. Constraint on Coreference Resolution

In English, the main problem for coreference resolution is the pronoun it. The reason behind this is that the pronoun it has many uses. For example, it can refer much like he and she. The pronoun it also refers to the things that do not refer to specific things. For example, It's raining. It is really good.

c. Pronominal Anaphora Resolution

Unlike the coreference resolution, pronominal anaphora resolution may be defined as the task of finding the antecedent for a single pronoun. For example, the pronoun is his and the task of pronominal anaphora resolution is to find the word Ram because Ram is the antecedent.

****

8. Discourse Coherence and Structures

Discourse coherence and structure are fundamental aspects of Natural Language Processing (NLP) that deal with how texts are organized and how ideas flow logically. These concepts are critical in applications like text summarization, machine translation, sentiment analysis, and dialogue systems.

1. Discourse Coherence

Discourse coherence refers to the logical and meaningful connection between different parts of a text. It ensures that sentences and paragraphs are linked in a way that makes sense to the reader.

Types of Coherence:

- **Local Coherence:** Consistency between adjacent sentences (e.g., pronoun resolution, logical transitions).
- **Global Coherence:** Overall consistency throughout the entire discourse (e.g., topic maintenance, thematic progression).

Key NLP Techniques for Coherence:

- **Lexical Chains:** Using repeated words, synonyms, or semantically related terms to maintain coherence.
- **Entity-based Coherence Models (e.g., Centering Theory):** Tracking entities across sentences to ensure logical flow.
- **Discourse Connectives:** Words like "however," "therefore," and "moreover" help indicate relationships between ideas.
- **Neural Models:** Transformer-based models (like BERT, GPT) capture coherence patterns using large-scale language modeling.

2. Discourse Structure

Discourse structure refers to how information is organized within a text. It helps in understanding how different text segments relate to each other.

Models of Discourse Structure:

- **Rhetorical Structure Theory (RST):** Represents texts as hierarchical structures with "nucleus" and "satellite" relations (e.g., cause-effect, elaboration).
- **Penn Discourse Treebank (PDTB):** Annotates discourse relations explicitly between text spans.
- **Segmented Discourse Representation Theory (SDRT):** Focuses on logical and semantic relations in multi-sentence discourse.

Applications of Discourse Structure in NLP:

- **Text Summarization:** Identifies key information by analyzing discourse relations.
- **Dialogue Systems:** Ensures coherent responses in chatbots and virtual assistants.
- **Machine Translation:** Preserves discourse relations across languages.
- **Information Retrieval:** Improves search results by considering document structure.

****