SRM INSTITUTE OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTING TECHNOLOGIES

21CSE356T

NATURAL LANGUAGE PROCESSING

UNIT- 3

S.PRABU
Assistant Professor
C-TECH-SRM-IST-KTR

UNIT-3
SEMANTIC AND DISCOURSE ANALYSIS

Representing Meaning, Lexical Semantics, Word Senses, Relation between Senses, Word Sense Disambiguation, Word Embeddings, Word2Vec, CBOW, Skip-gram and GloVe, Discourse Segmentation, Text Coherence, Discourse Structure, Reference Resolution, Pronominal Anaphora Resolution, Coreference Resolution

Topic 1: Representing Meaning

Representing meaning in Natural Language Processing (NLP) involves capturing the semantics (meaning) of words, sentences, or texts in a structured way that a machine can understand and process.

Lexical Semantics – Deals with the meaning of individual words and their relationships (synonyms, antonyms, hypernyms, hyponyms).

Compositional Semantics – Meaning of a sentence is derived from the meanings of its words and their syntactic structure.

Word Representations – Words can be represented using:

- **One-hot encoding** – Binary representation of words in a fixed vocabulary.
- **Word Embeddings (Word2Vec, GloVe, FastText)** – Vectorized representations capturing semantic similarities.

Semantic Roles & Frames – Assigning roles (agent, action, object) to words in a sentence to represent meaning.

Logical Form – Representing meaning using formal logic (predicate logic, first-order logic).

Knowledge Graphs – Connecting entities and their relationships in a graph-based structure.

Distributional Semantics – Words occurring in similar contexts have similar meanings (distributional hypothesis).

Sentence & Document-Level Representations – Using techniques like TF-IDF, LSA, or BERT-based models to encode textual meaning.

Pragmatics & Contextual Meaning – Meaning also depends on context, speaker intention, and discourse analysis.

Topic 2 Lexical Semantics

Lexical semantics is the branch of linguistics that deals with the meaning of individual words and their relationships. It focuses on how words represent concepts and how they relate to each other within a language.

Key Concepts in Lexical Semantics:

1. **Synonymy** – Words with similar meanings.
   - Example: *happy* and *joyful*
   - Both words convey a similar sense of positive emotion.
2. **Antonymy** – Words with opposite meanings.
   - Example: *hot* vs. *cold*
   - These words describe opposite temperature conditions.
3. **Hyponymy & Hypernymy** – Hierarchical relationships between words.
   - **Hypernym (superordinate term):** General category
   - **Hyponym (subordinate term):** More specific concept under the category
   - Example: *Animal* (hypernym) → *Dog* (hyponym) → *Labrador* (further hyponym)
4. **Homonymy** – Words that have the same spelling or pronunciation but different meanings.
   - Example: *bank* (a financial institution) vs. *bank* (side of a river)
5. **Polysemy** – A word with multiple related meanings.
   - Example: *head*
     - *Head of a company* (leader)
     - *Head of a person* (body part)
6. **Meronymy & Holonymy** – Part-whole relationships.
   - **Meronym:** A part of something (*wheel* is a meronym of *car*).

- **Holonym:** The whole entity (*car* is a holonym of *wheel*).

7. **Collocations** – Words that commonly appear together.
    - Example: *strong tea* (not **powerful tea**)
    - Certain words naturally co-occur due to usage patterns.

Example Sentences in Lexical Semantics:

- "I saw a **bat** flying in the sky." (animal)
- "He hit the ball with a **bat**." (sports equipment)
    - This demonstrates **homonymy** (same spelling, different meanings).
- "A **dog** is an **animal**, and a **Labrador** is a type of **dog**."
    - This shows **hyponymy-hypernymy** relationships.

Lexical semantics plays a crucial role in NLP applications like **word sense disambiguation, machine translation, sentiment analysis, and information retrieval**.

Topic 3 Word Senses

A word sense refers to a specific meaning of a word in a given context. Since many words have multiple meanings, determining the correct sense is crucial in Natural Language Processing (NLP).

Types of Word Senses:

1. **Monosemous Words** – Words with only one meaning.
    - Example: *Oxygen* (a chemical element).
    - No ambiguity in meaning.
2. **Polysemous Words** – Words with multiple related meanings.
    - Example: *Book*
        - *I read a book* (a physical object).
        - *I will book a ticket* (to reserve something).
3. **Homonymous Words** – Words that have different meanings but the same spelling or pronunciation.

- Example: *Bat*
    - *A bat is flying in the sky* (an animal).
    - *He hit the ball with a bat* (sports equipment).

Word Sense Disambiguation (WSD)

Since words can have multiple senses, NLP uses Word Sense Disambiguation (WSD) to determine the correct meaning based on context.

Example Sentence:

- *He went to the **bank** to withdraw money.* → (**Bank** = financial institution)
- *The boat was near the **bank** of the river.* → (**Bank** = land beside a river)

In this case, WSD helps in identifying the correct sense based on the surrounding words.

Topic 4: Relation between Senses

The relation between senses refers to how different word meanings (senses) are connected. Words can be related in various ways based on meaning, usage, and context.

Types of Relations Between Senses:

1. **Synonymy (Similar Meaning)**
    - Words that have the same or nearly the same meaning.
    - Example: *Big* ↔ *Large*, *Happy* ↔ *Joyful*
    - Sentence: *He lives in a **big** house. = He lives in a **large** house.*
2. **Antonymy (Opposite Meaning)**
    - Words with opposite meanings.
    - Example: *Hot* ↔ *Cold*, *Fast* ↔ *Slow*
    - Sentence: *The water is **hot**, not **cold**.*
3. **Homonymy (Same Form, Different Meaning)**
    - Words that are spelled or pronounced the same but have different meanings.
    - Example:
        - *Bat* (flying animal) vs. *Bat* (used in cricket/baseball).

        - *Bank* (financial institution) vs. *Bank* (side of a river).

4. **Polysemy (One Word, Multiple Related Meanings)**
    - A single word with multiple, related meanings.
    - Example:
        - *Book* (a physical object) vs. *Book* (to reserve something).
        - *Head* (part of the body) vs. *Head* (leader of an organization).
5. **Hyponymy & Hypernymy (Hierarchy of Meanings)**
    - **Hypernym:** A general word that covers more specific words.
    - **Hyponym:** A specific type of a general word.
    - Example:
        - *Animal* (Hypernym) → *Dog* (Hyponym) → *Labrador* (More specific Hyponym).
6. **Meronymy & Holonymy (Part-Whole Relationship)**
    - **Meronym:** A part of something.
    - **Holonym:** The whole to which a part belongs.
    - Example:
        - *Wheel* is a **meronym** of *Car*.
        - *Car* is a **holonym** of *Wheel*.

Example Sentence Demonstrating Multiple Relations:

- *The* ***head*** *of the company read a* ***book*** *about different dog breeds.*
    - **Head** (Polysemy: Leader & Body part).
    - **Book** (Polysemy: Physical object & Action of booking).
    - **Dog breeds** (Hyponymy: Dog is a hyponym of Animal).

Understanding these relations helps in **Word Sense Disambiguation (WSD)**, **Information Retrieval**, **Machine Translation**, and **Text Understanding** in NLP.

Topic 5 Word Sense Disambiguation

Word Sense Disambiguation (WSD) is the process of determining the correct meaning (sense) of a word in a given context. Since many words have multiple meanings, WSD helps NLP systems understand the intended sense accurately.

Example of Word Sense Disambiguation:

Consider the word **"bat"**, which has multiple meanings:

1. **Bat (animal):** *A bat is flying in the sky.*
2. **Bat (sports equipment):** *He hit the ball with a bat.*

A human can easily determine the correct meaning based on the surrounding words, and WSD enables machines to do the same.

Methods of WSD:

1. **Dictionary-Based Approach**
   - Uses a lexical database (like WordNet) to find possible meanings of a word and matches them with the context.
2. **Supervised Machine Learning**
   - Uses labeled training data to teach a model which sense is most appropriate in a given sentence.
3. **Unsupervised Learning (Clustering-based)**
   - Groups words into clusters based on similarity without labeled data.
4. **Knowledge-Based Approach**
   - Uses external knowledge sources (e.g., ontologies, thesauruses) to determine the meaning.

Example Sentences and Disambiguation:

- *She went to the **bank** to withdraw money.* → **Bank** (financial institution).
- *The boat is near the **bank** of the river.* → **Bank** (land beside a river).

The importance of Word Sense Disambiguation (WSD) in NLP is significant across various applications. In machine translation, selecting the correct word sense ensures accurate translations, preventing errors caused by ambiguous words. For chatbots and virtual assistants, WSD helps in correctly interpreting user queries, enabling more precise and relevant responses. Similarly, in search engines, it plays a crucial role in displaying the most relevant results by understanding the intended meaning of search terms, improving the overall user experience.

Topic 6 Word Embeddings

Word embeddings are numerical vector representations of words that capture their meanings, relationships, and context in a continuous vector space. Unlike traditional one-hot encoding, word embeddings allow words with similar meanings to have similar vector representations.

Example of Word Embeddings:

Consider the words "king", "queen", and "man" represented as vectors:

- king → [0.7, 0.2, 0.5]
- queen → [0.8, 0.3, 0.6]
- man → [0.6, 0.1, 0.4]

Since "king" and "queen" are related, their vectors are closer in space compared to "man."

Popular Word Embedding Models:

1. Word2Vec – Captures word relationships using neural networks.
2. GloVe (Global Vectors for Word Representation) – Uses word co-occurrence statistics to create embeddings.
3. FastText – Improves embeddings by considering subword information.
4. BERT Embeddings – Contextual word representations based on transformers.

Word embeddings are important in NLP as they enhance the understanding of word meanings in context, allowing models to process language more effectively. They play a crucial role in machine translation by mapping similar words closely in vector space, improving translation accuracy. Additionally, in sentiment analysis, word embeddings help detect the emotional tone of text by capturing relationships between words, leading to better interpretation of sentiments and opinions.

Topic 7 Word2Vec

Word2Vec is a popular word embedding technique that converts words into numerical vectors while preserving their semantic relationships. Developed by Google, it helps machines understand word meanings based on their usage in large text corpora.

How Word2Vec Works?

Word2Vec uses two main approaches to learn word embeddings:

1. **Continuous Bag of Words (CBOW)** – Predicts a target word based on surrounding words.
2. **Skip-gram** – Predicts surrounding words given a target word.

Example of Word2Vec:

If trained on a large dataset, Word2Vec can capture relationships like:

- **King - Man + Woman ≈ Queen**
- **Paris - France + Italy ≈ Rome**

This shows that the model understands the analogy between words.

Why is Word2Vec Important?

Word2Vec is important because it captures the relationships between words, allowing words with similar meanings to have similar vector representations. This enhances various NLP tasks such as machine translation, sentiment analysis, and text classification by improving the understanding of word contexts. Additionally, Word2Vec efficiently learns word meanings from large datasets with minimal supervision, making it a powerful tool for developing intelligent language models.

CBOW (Continuous Bag of Words)

CBOW is a model used in Word2Vec, a technique for word embeddings in natural language processing (NLP). It predicts a target word based on the surrounding context words in a given window size.

How CBOW Works:

1. **Input:** A set of context words (surrounding words in a sentence).
2. **Hidden Layer:** A neural network processes the input and learns relationships between words.
3. **Output:** The model predicts the most likely target word that fits the context.

Example:

For the sentence:

*"The cat sits on the **mat**."*

If we take the context words as **"The cat sits on the"**, CBOW will predict the missing target word **"mat."**

CBOW vs. Skip-gram:

- **CBOW**: Predicts a target word from surrounding words (faster and better for frequent words).
- **Skip-gram**: Predicts surrounding words given a target word (better for rare words).

CBOW is widely used for generating dense word embeddings, which are useful for many NLP tasks like sentiment analysis, machine translation, and text classification.

Topic 8 Skip-gram and GloVe

Skip-gram and GloVe

Both Skip-gram and GloVe are popular word embedding techniques in NLP, used to represent words in continuous vector space where similar words have similar representations.

Skip-gram (Word2Vec)

Skip-gram is another model in Word2Vec (besides CBOW). Unlike CBOW, which predicts a target word given its context, Skip-gram predicts surrounding context words given a target word.

How Skip-gram Works:

1. **Input:** A single word (target word).
2. **Hidden Layer:** A neural network processes this word.
3. **Output:** The model predicts context words (surrounding words).

Example:

For the sentence:

*"The cat sits on the **mat**."*

- **CBOW**: Predicts **"mat"** given context words (**"The cat sits on the"**).
- **Skip-gram**: Predicts context words (**"The," "cat," "sits," "on"**) given the target word **"mat."**

Key Features of Skip-gram:

- Works well with **small datasets** and **rare words**.
- Generates **better-quality embeddings** than CBOW but is computationally **slower**.

Topic 9 Discourse Segmentation

Discourse Segmentation is the task of dividing a text into meaningful coherent segments, usually at the level of sentences or paragraphs, to understand the structure of discourse. It helps in analyzing how different parts of a text relate to each other in a conversation, document, or dialogue.

Why is Discourse Segmentation Important?

1. **Improves Text Coherence Analysis** – Helps identify different topics or shifts in discourse.
2. **Enhances NLP Applications** – Used in summarization, machine translation, and information retrieval.
3. **Supports Dialogue Systems** – Helps chatbots and virtual assistants understand conversation flow.
4. **Aids Sentiment and Opinion Mining** – Useful in breaking down reviews or comments into distinct opinion-based sections.

Types of Discourse Segmentation

1. **Sentence-Level Segmentation** – Splitting text into sentences (e.g., in speech-to-text systems).
2. **Topic Segmentation** – Identifying where one topic ends and another begins.
3. **Dialogue Segmentation** – Separating speaker turns in a conversation.

Example of Discourse Segmentation

Consider this passage:

"Climate change is a serious global issue. Scientists warn about rising temperatures. Governments are working on policies. On another note, recent tech advancements in AI are revolutionizing industries."

A **discourse segmentation system** would separate the two topics:

1. **Segment 1 (Climate Change Topic):** "Climate change is a serious global issue. Scientists warn about rising temperatures. Governments are working on policies."
2. **Segment 2 (AI Technology Topic):** "On another note, recent tech advancements in AI are revolutionizing industries."

Methods for Discourse Segmentation

- **Rule-Based Approaches** – Using linguistic rules (e.g., discourse markers like "however," "on another note").
- **Supervised Machine Learning** – Training models on labeled datasets.
- **Unsupervised Techniques** – Using clustering methods based on word similarity.
- **Neural Networks & Deep Learning** – Using models like BERT or GPT for segmenting complex texts.

Topic 10 Text Coherence.

Text coherence refers to the logical connection and smooth flow of ideas in a text, making it meaningful and easy to understand. A coherent text ensures that sentences and paragraphs are well-organized, related, and contribute to the overall message.

Why is Text Coherence Important?

1. **Enhances Readability** – Makes text easier to follow.
2. **Improves Comprehension** – Helps readers understand the relationships between ideas.
3. **Essential for NLP Applications** – Used in text summarization, machine translation, and question-answering systems.
4. **Key in Discourse Analysis** – Ensures that discourse structure is logically connected.

Types of Text Coherence

1. **Local Coherence** – Ensures smooth connections between **consecutive sentences**.
   - Example: *John loves football. He plays for his college team.*
   - The second sentence maintains coherence by referring to "John" and his love for football.
2. **Global Coherence** – Ensures a text follows a **central theme or topic**.
   - Example: A research paper on **deep learning** should not suddenly shift to **political analysis** without a clear connection.

Techniques for Achieving Text Coherence

1. **Use of Cohesive Devices** – Words or phrases that connect ideas:
   - *Addition:* **Moreover, Furthermore, In addition**
   - *Contrast:* **However, On the other hand, Nevertheless**
   - *Cause & Effect:* **Because, Therefore, As a result**
   - *Reference Words:* **This, That, These, Those**
2. **Logical Order** – Arranging ideas in a sequence:
   - **Chronological:** *First, Next, Finally*
   - **Cause and Effect:** *X happened because of Y*
   - **Problem-Solution:** *Identifies a problem and offers solutions*
3. **Lexical Cohesion** – Repeating key words or using synonyms to maintain the topic:
   - Example: *Artificial Intelligence (AI) is transforming industries. This technology is now widely used in healthcare and finance.*
   - "This technology" refers back to "AI," maintaining coherence.

Topic 11 Discourse Structure

Discourse Structure refers to the way a conversation, document, or text is organized to convey meaning effectively. It defines how different parts of a text (sentences, paragraphs, or dialogue turns) are connected to form a coherent and meaningful whole.

Why is Discourse Structure Important?

1. **Ensures Logical Flow** – Helps maintain clarity in communication.
2. **Improves Text Coherence** – Ensures sentences relate meaningfully to one another.
3. **Aids NLP Tasks** – Used in summarization, dialogue systems, and machine translation.
4. **Facilitates Information Extraction** – Helps identify key sections in documents.

Components of Discourse Structure

1. **Sentences and Paragraphs** – Basic building blocks of discourse.
2. **Discourse Markers** – Words or phrases that signal relationships (e.g., *however, therefore, in contrast*).
3. **Discourse Relations** – Logical connections between different parts of text.

Types of Discourse Structure Models

1. **Rhetorical Structure Theory (RST)**
   - Explains how sentences relate in a hierarchical structure.
   - Example: *"Since it was raining, we stayed indoors."*
     - **Relation:** Cause-Effect
2. **Segmentation-based Models**
   - Divide text into **segments** based on topic shifts or coherence.
   - Example: In a news article, different paragraphs may discuss *introduction → background → analysis → conclusion*.

3. **Dependency Graph Models**
    - Represent text as a **tree or graph**, where nodes are sentences and edges are logical relationships.

Example of Discourse Structure

Incoherent Structure (No Clear Relations):

"The car broke down. I bought apples. The weather is nice today."

Coherent Structure (Well-Organized Discourse):

"The car broke down, so I had to walk to the store. Since I was already there, I bought some apples. The weather was nice, which made the walk enjoyable."

Here, **discourse markers ("so," "since," "which")** help establish logical relationships, creating a structured discourse.

Discourse Structure in NLP Applications

1. **Text Summarization** – Identifies key parts of a document.
2. **Chatbots & Dialogue Systems** – Helps in natural conversation flow.
3. **Sentiment Analysis** – Understands how opinions shift within a review.
4. **Machine Translation** – Preserves meaning when translating languages.

Topic 12 Reference Resolution

Reference Resolution is the process of identifying what a word or phrase (such as a pronoun or noun phrase) refers to in a given text. It is a crucial task in Natural Language Processing (NLP) and is essential for understanding meaning in discourse.

Why is Reference Resolution Important?

Improves Text Coherence – Ensures clarity in who or what is being referred to.

Enhances Machine Translation – Helps translate pronouns and noun references correctly.

Aids Chatbots & Virtual Assistants – Allows systems to track conversations accurately.

Supports Sentiment Analysis – Determines the correct entity being reviewed or criticized.

Types of Reference Resolution

1. Coreference Resolution

Identifies when two or more expressions in a text refer to the same entity.

Example:

"John loves football. He plays for his college team."

"He" refers to "John" (Coreference).

2. Pronominal Anaphora Resolution

Resolves pronouns to their corresponding nouns.

Example:

"Sarah went to the store. She bought some milk."

"She" refers to "Sarah".

3. Named Entity Resolution

Identifies when different names refer to the same person, place, or entity.

Example:

"Barack Obama was the 44th U.S. President. Obama was known for his speeches."

"Barack Obama" and "Obama" refer to the same entity.

4. Bridging Resolution

Identifies implicit relationships between entities.

Example:

"I bought a book. The cover is beautiful."

"The cover" refers to "the book" without directly mentioning it.

Example of Reference Resolution in Action

Without Reference Resolution (Ambiguous)

"Alice gave a book to Mary. She was happy."

(Who is "she" referring to? Alice or Mary?)

With Reference Resolution (Clear Meaning)
"Alice gave a book to Mary. Mary was happy."
(Here, "she" is resolved to "Mary" for clarity.)

Topic 13 Pronominal Anaphora Resolution

Pronominal Anaphora Resolution is the process of identifying the noun or entity that a pronoun refers to in a text. This is a subtask of Reference Resolution in Natural Language Processing (NLP).

Why is Pronominal Anaphora Resolution Important?
Improves Text Understanding – Helps resolve ambiguities in sentences.
Essential for Chatbots & Virtual Assistants – Ensures continuity in conversations.
Enhances Machine Translation – Helps translate pronouns correctly.
Useful in Text Summarization – Maintains coherence by correctly linking pronouns to their antecedents.

Types of Anaphora Resolution

1. **Pronoun-Antecedent Resolution**
    - Identifies the noun (antecedent) that a pronoun refers to.
    - Example:
        - *"John went to the store. He bought some apples."*
        - **"He" refers to "John".**
2. **Cataphora Resolution**
    - The pronoun appears **before** the noun.
    - Example:
        - *"When he arrived, John was tired."*
        - **"He" refers to "John".**

3. **Zero Anaphora** (Implicit Reference)
   - The subject is **not explicitly mentioned** but understood.
   - Example (common in some languages like Japanese):
     - *"Went to the store. Bought apples."* (Implicit subject: "I")

Examples of Pronominal Anaphora Resolution

Ambiguous Without Resolution

"Maria told Anna that she won the competition."

(Who won? Maria or Anna?)

Resolved Version

*"Maria told Anna that **Anna** won the competition."*

(Now, it's clear that Anna is the winner.)

Topic 14 Coreference Resolution

Coreference Resolution is the task of identifying when different words or phrases in a text refer to the same entity. It is a crucial component of Natural Language Processing (NLP), helping machines understand the meaning of text by linking mentions of the same person, place, or thing.

Why is Coreference Resolution Important?

Improves Text Understanding – Helps in tracking entities across a document.

Essential for Chatbots & Virtual Assistants – Enables proper follow-up in conversations.

Enhances Machine Translation – Maintains consistency in translated texts.

Useful in Text Summarization – Helps in generating clear and coherent summaries.

Types of Coreference Resolution

1. **Pronominal Coreference**
    - Resolving **pronouns** to their corresponding noun.
    - Example:
        - *"John loves football. He plays for his college team."*
        - **"He" refers to "John".**
2. **Nominal Coreference**
    - Resolving **different noun phrases** that refer to the same entity.
    - Example:
        - *"Elon Musk founded Tesla. The billionaire is known for his innovation."*
        - **"The billionaire" refers to "Elon Musk".**
3. **Demonstrative Coreference**
    - Resolving demonstrative pronouns (*this, that, these, those*).
    - Example:
        - *"The company launched a new product. This has attracted many customers."*
        - **"This" refers to "a new product".**
4. **Cataphoric Coreference**
    - When the **pronoun appears before the actual noun**.
    - Example:
        - *"When he arrived, John was tired."*
        - **"He" refers to "John".**

Example of Coreference Resolution

Without Coreference Resolution (Ambiguous)

"Alice met Sarah at the park. She was happy."

(Who is "she"? Alice or Sarah?)

With Coreference Resolution (Clear Meaning)

"Alice met Sarah at the park. Alice was happy."

(Now, we know Alice was the one who was happy.)