Lecture 3-5
Natural Language Understanding (NLU)

Why did the bicycle fall over? Because it was two tired.

Did you get the joke?

How will a NLP model interpret it?

Note: Project: Building NLP model to recognize humor using **Knowledge Graphs/Transformer based models:** Using knowledge graphs to represent relationships between concepts can help the model understand the associations between "bicycle," "tire," and "tired."

Morphological Analysis

Morphological analysis is a fundamental aspect of Natural Language Processing (NLP) that deals with the **structure of words** and how they are formed from smaller meaningful units called **morphemes**. Morphemes are the smallest units of meaning in a language, and they can be **prefixes**, **suffixes**, or **root words**.

Activity

Scenario 1: Information Retrieval (Search Engine) with "dog run" Query

Corpus:

1. "The playful dogs are running in the park."
2. "The dog ran after the ball, jumping high."
3. "Running is a great way to exercise dogs."
4. "The dog's running speed was impressive."
5. "Sadly, the dog was unhappy today."
6. "Unhappiness was evident in the dog's behavior."
7. "The dog happily chased the squirrels."
8. "Many dogs enjoy jumping and running."
9. "I saw a dog run across the street."

Query: "dog run"

Without Morphology (Exact String Matching):

- **What it means:** The search engine looks for the *exact* sequence of characters "dog run" within the corpus. No variations, no stemming, no lemmatization, and no implicit pluralization.
- **Analysis:**
  - We need to find a sentence that contains the *literal* string "dog run."
  - Sentence 9, "I saw a dog run across the street." contains the exact string "dog run".
- **Result:**
  - **Only sentence 9 will be matched.**
  - Sentences 1, 2, 3, 4, 5, 6, 7, and 8 will be completely ignored, as they do not contain the precise string "dog run."

With Morphology (Stemming/Lemmatization):

- **What it means:** The search engine applies stemming or lemmatization to both the query and the corpus.
- **Process (Stemming Example):**
  - Query: "dog run" (remains the same as both words are already stems)
  - Corpus:
    - "dogs" -> "dog"
    - "running" -> "run"
    - "ran" -> "ran" (in many cases this would also become run, depending on the stemmer)
    - "jumping" -> "jump"
- **Analysis:**
  - Now, the search engine looks for the stemmed forms "dog" and "run."
  - Sentences 1, 2, 3, 4, 8, and 9 will be matched.
- **Result:**
  - Sentences 1, 2, 3, 4, 8 and 9 are returned as relevant.
  - Sentences 5, 6, and 7 are not returned.

Key Difference:

- Without morphology, the search is extremely limited and misses most relevant results.
- With morphology, the search becomes much more effective, capturing variations of the query terms.

Scenario 2:

Corpus :

1. "The playful dogs are running in the park."
2. "The dog ran after the ball, jumping high."
3. "Running is a great way to exercise dogs."
4. "The dog's running speed was impressive."

5. "Sadly, the dog was unhappy today."
6. "Unhappiness was evident in the dog's behavior."
7. "The dog happily chased the squirrels."
8. "Many dogs enjoy jumping and running."
9. "I saw a dog run across the street."

Sentiment Lexicon (Simplified):

- "happy": 1
- "great": 1
- "impressive": 1
- "unhappy": -1
- "sadly": -1

Scenario: Sentiment Analysis of Sentences 5, 6, and 7

Sentence 5: "Sadly, the dog was unhappy today."

- **Without Morphology:**
  - "sadly" matches -1.
  - "unhappy" matches -1.
  - Total score: -2 (Correctly identified as negative)
- **With Morphology (Lemmatization):**
  - No change in this sentence, as the lexicon words are already in their base form.
  - Total score: -2 (Correctly identified as negative)

Sentence 6: "Unhappiness was evident in the dog's behavior."

- **Without Morphology:**
  - "Unhappiness" does not match any word in the lexicon.
  - Total score: 0 (Incorrectly classified as neutral)
- **With Morphology (Lemmatization):**
  - "Unhappiness" is lemmatized to "unhappy."
  - "unhappy" matches -1.
  - Total score: -1 (Correctly identified as negative)

Sentence 7: "The dog happily chased the squirrels."

- **Without Morphology:**
  - "happily" does not match any word in the lexicon.
  - Total score: 0 (Incorrectly classified as neutral)
- **With Morphology (Lemmatization):**
  - "happily" is lemmatized to "happy."
  - "happy" matches 1.
  - Total score: 1 (Correctly identified as positive)

Analysis:

- **Without Morphology:** The sentiment analysis fails to recognize the sentiment in sentences 6 and 7 because it cannot link "unhappiness" to "unhappy" or "happily" to "happy".
- **With Morphology:** Lemmatization allows the system to correctly identify the sentiment in all three sentences.

Key Takeaway:

- Morphological analysis, specifically lemmatization in this case, significantly improves the accuracy of sentiment analysis by recognizing variations of sentiment-bearing words.
- This demonstrates that morphological analysis is not just important for information retrieval, but also for many other NLP tasks.

Applications of Morphological Analysis

1. Search Engines

- **Problem**: Users may search for different forms of a word (e.g., "run," "running," "ran").
- **Solution**: Morphological analysis ensures that all forms of a word are recognized and matched.
- **Example**:
  - Query: "running shoes"
  - Search engine recognizes "running" as a form of "run" and returns results for "run shoes," "ran shoes," etc.

2. Spell Checkers and Correctors

- **Problem**: Users may misspell words or use incorrect forms (e.g., "happi" instead of "happy").
- **Solution**: Morphological analysis breaks down words and suggests corrections based on valid morphemes.
- **Example**:
  - Input: "I am very happi."
  - Correction: "I am very happy."

3. Machine Translation

- **Problem**: Words in one language may have multiple forms that need to be translated correctly.

- **Solution**: Morphological analysis ensures that words are translated in their correct forms.
- **Example**:
  - English: "She is running."
  - Spanish: "Ella está corriendo." (Here, "running" is translated to "corriendo," the correct form of the verb "correr")

Syntax Analysis (Syntactic Processing)

Syntax: The rules that govern the structure of sentences. It describes how words are arranged to form grammatically correct phrases and sentences.

Syntax Analysis (Parsing): The process of analyzing a sentence to determine its grammatical structure. It involves building a parse tree that represents the relationships between words and phrases.

Application 1 (spell check and correction):

Incorrect Sentence:
"He go to the park every day."

🔹 Basic Spell Checker Output:
✅ No spelling errors detected (all words exist in the dictionary).

🔹 Syntax-Based Spell Checker Output:
❌ *Error detected:* “go” should be “goes” based on subject-verb agreement.
✅ Corrected Sentence: *"He goes to the park every day."*

How Syntax Analysis Works in Spell Checking?

1. Tokenization: Breaks the sentence into words.

   `"He go to the park every day."` → `["He", "go", "to", "the", "park", "every", "day"]`

2. Part-of-Speech (POS) Tagging: Identifies word roles.

   `"He" (Pronoun), "go" (Verb), "to" (Preposition), "the" (Article), "park" (Noun)`

3. Parsing (Syntax Tree Generation): Analyzes sentence structure

Note: [(Project Idea)-> Grammar checking tool from scratch / Syntax Parsing in NLP]

4. Grammar Rule: *Singular subject ("He") requires a singular verb ("goes").*
5. Error Detection & Correction: Suggests corrections based on grammar rules.

   ❌ `"go"` (wrong) → ✅ `"goes"` (correct)

The Problem:

The sentence "He go to the park every day" is grammatically incorrect. It violates the basic subject-verb agreement rule in English.

The Subject-Verb Rule:

- **Subject:** The person or thing performing the action (e.g., "He," "She," "They").
- **Verb:** The action word (e.g., "go," "run," "eat").
- **Agreement:** The verb must match the subject in number (singular or plural) and person (first, second, or third person).

Why "He go" Is Wrong:

- **Subject:** "He" is a third-person singular pronoun (it refers to one person, and it's neither "I" nor "you").
- **Verb:** "go" is the base form of the verb.
- **The Mismatch:** In the present tense, third-person singular subjects require a verb ending in "-s" or "-es."

The Correct Sentence:

- "He **goes** to the park every day."

Simple Explanation:

Think of it like this:

- If the subject is "He," "She," or "It," the verb usually needs an "s" at the end when you're talking about things that happen regularly (present tense).
- If the subject is "I," "You," "We," or "They," the verb stays in its base form (no "s").

Examples:

- Correct:
  - "She **eats** apples."
  - "The dog **runs** fast."
  - "He **plays** football."
- Correct:
  - "I **go** home."

    - "They **walk** to school."
    - "We **sing** songs."

So, in "He go to the park every day," the "go" needs to become "goes" to follow the subject-verb agreement rule.

https://colab.research.google.com/drive/1uCZNFFlLlVUx_niiVBkEA3HjuZlC9IKG?usp=sharing

Application 2 (question answering):

"The cat chased the mouse. The mouse ran under the table. The children laughed."

Questions:

1. Who chased the mouse?
2. Where did the mouse run?
3. Who laughed?

How Syntax Analysis Helps:

1. **Identifying Subject-Verb-Object Relationships:**
    - Syntax analysis allows the system to identify the core components of each sentence:
        - **Subject:** The entity performing the action.
        - **Verb:** The action itself.
        - **Object (if applicable):** The entity receiving the action.

    For example:

        - "The cat chased the mouse."
            - "The cat" (subject)
            - "chased" (verb)
            - "the mouse" (object)
2. **Understanding Prepositional Phrases:**
    - Syntax analysis helps identify prepositional phrases and their relationships to other parts of the sentence.
    - For example:
        - "The mouse ran under the table."
            - "under the table" (prepositional phrase) modifying the verb "ran."
    - This clarifies that "under the table" is the location of the action.
3. **Answering "Who" Questions:**

    - "Who" questions typically seek the subject of a sentence.
    - Syntax analysis helps isolate the noun phrase that functions as the subject.
    - Example:
        - "Who laughed?"
            - "The children" (subject)
4. **Answering "Where" Questions:**

    - "Where" questions seek a location or place.
    - Syntax analysis helps identify prepositional phrases or adverbial phrases that indicate location.
    - Example:
        - "Where did the mouse run?"
            - "under the table" (prepositional phrase)

In essence, syntax analysis allows us to:

- Move beyond simply matching keywords.
- Understand the grammatical roles of words within a sentence.
- Identify the relationships between words and phrases.
- Extract the precise information needed to answer the questions.

Note:

- **Prepositional Phrases:**
    - These are groups of words that start with words like "under," "in," "on," "near," "behind," etc.
    - They tell us "where" something is in relation to something else.
    - Example: "under the table" tells us the mouse is located relative to the table.
- **Adverbial Phrases:**
    - These are groups of words (or sometimes just one word) that tell us "where," "when," or "how" something happened.
    - They act like adverbs, which are words that describe verbs.
    - Example: "nearby" or "in the distance"

How Syntax Analysis Finds Them:

1. **Spotting the "Location" Words:**
    - The syntax tool recognizes words like "under," "in," "on," etc., and knows they often start with location phrases.
2. **Seeing the Connections:**
    - It then looks at the words that come after those location words and figures out how they fit together.
    - It understands that "under the table" is a single unit that tells us "where."
3. **Figuring Out What They Describe:**

    - The tool also checks which part of the sentence the location phrase is connected to.
    - If it's connected to a verb (an action word), it knows it's telling us "where" the action happened.

Simple Example:

- "The cat slept **under the chair**."
    - The syntax tool sees "under" and knows it's likely a location phrase.
    - It groups "under the chair" together.
    - It sees that "under the chair" is connected to the verb "slept", so it knows it is telling us where the cat slept.

Why It's Better Than Just Looking for Words:

- If you just look for words like "under," you might get confused.
- Example: "The cat found the key under the rug, which was under the table."
    - Just looking for "under" would give you two locations, but syntax analysis tells you which "under" belongs to which action.

Another Example :

The cat found the key under the rug, which was under the table."

Let's break it down in very simple terms:

The Core Problem:

We need to know *where* the cat found the key, not just where things are in general.

How Syntax Analysis Solves It:

1. **Identifying Phrases:**
    - Syntax analysis first breaks the sentence into meaningful chunks (phrases). It recognizes:
        - "The cat found the key" (a complete thought)
        - "under the rug" (a location phrase)
        - "which was under the table" (another location phrase)
2. **Determining Relationships:**
    - This is where syntax really shines. It figures out *how* these phrases connect:
        - It sees that "under the rug" directly follows "found the key." This means "under the rug" is telling us *where* the finding happened.
        - It sees that "which was under the table" is a separate clause that describes the location of the *rug*.

3. **Understanding the "Which":**

    - The word "which" is important. It connects the second "under the table" to "the rug". Syntax analysis understands that "which was under the table" is a description of the rug, not a description of where the cat found the key.
4. **Giving the Correct Answer:**
    - Because it understands these connections, syntax analysis can confidently say: "The cat found the key under the rug." It knows that "under the table" is just extra information about the rug's location.

Why Just Looking for "Under" Fails:

- If you just look for the word "under," you get two locations.
- You have no way of knowing which "under" is connected to the action of "finding the key."
- You would have no way to know that "which was under the table" is describing the rug.

In essence:

Syntax analysis connects the "where" (location phrases) to the "what" (actions). It's not just about finding location words; it's about finding *which* location word is relevant to the action you're interested in.

Applications

Machine Translation:

- This is arguably the most demanding and impactful application. Accurate translation relies heavily on understanding the source sentence's syntactic structure to generate a grammatically correct and semantically equivalent target sentence. The complexities of different language structures make parsing indispensable.

Question Answering:

- The ability to accurately parse a question is essential for understanding what information is being requested. This allows systems to extract relevant information from knowledge bases or text documents, providing precise answers.

Information Extraction:

- Extracting structured information from unstructured text is a core NLP task. Syntax analysis helps to identify relationships between entities and events by parsing sentences and identifying grammatical dependencies. This is vital for knowledge graph construction, data mining, and many other applications.

Grammar Checking:

- While seemingly simpler, grammar checking has a huge impact on written communication. Syntax parsing allows for the detection of complex grammatical errors beyond simple spelling mistakes, improving the quality of writing in various contexts.

Semantic Analysis (Meaning Extraction)

Semantic analysis is the process of extracting **meaning** from text. It helps NLP models **understand words, phrases, and sentences** beyond just their grammatical structure. Unlike syntax analysis, which focuses on **sentence structure**, semantic analysis ensures that **words make sense together in context**.

Why is Semantic Analysis Important?

1. **Word Sense Disambiguation:** "The bank of the river." vs. "I went to the bank to deposit money”. Semantic analysis helps determine the correct meaning of "bank" based on the context.
2. **Sentiment Analysis:** "This movie is great!" vs. "This movie is not great." Semantic analysis understands that "not great" expresses negative sentiment.

Activity

Create a table with columns for "Sentence," "Possible Meanings," and "Intended Meaning.
Corpus:

1. "I saw the bat flying." (animal or baseball bat?)
2. "The man saw the dog with the telescope." (who has the telescope?)
3. "She is blue." (color or feeling sad?)

Hint: What do you know about bats that helps you understand the first sentence?

Let's break down those examples and how we use our knowledge to arrive at the correct meaning:

1. "I saw the bat flying."

- **Possible Meanings:**
  - "Bat" as a nocturnal flying mammal.
  - "Bat" as a piece of equipment used in baseball.
- **How We Use Common Sense:**
  - We know that baseball bats don't typically fly on their own.
  - We know that bats (the animal) are capable of flight.
- **Correct Meaning:** "I saw the bat (the animal) flying."

2. "The man saw the dog with the telescope."

- **Possible Meanings:**
    - The man used a telescope to see the dog.
    - The dog was holding the telescope.
- **How We Use Common Sense:**
    - We know that dogs don't typically use telescopes.
    - We know that humans use telescopes to see distant objects.
- **Correct Meaning:** "The man used the telescope to see the dog."

3. "She is blue."

- **Possible Meanings:**

    - "Blue" as a color.
    - "Blue" as a feeling of sadness.
- **How We Use Common Sense:**

    - We know that people don't literally turn the color blue when they are sad.
    - We know that "blue" can be used metaphorically to describe a feeling of sadness.
- **Correct Meaning:** "She is feeling sad (blue)."

The Challenge for Computers:

While humans seamlessly integrate common sense and world knowledge, it's a significant challenge for computers. They need to be explicitly provided with this knowledge or be able to learn it from vast amounts of data. This is where techniques like knowledge graphs and contextual embeddings come into play, as we discussed earlier.

Example 2: *"Apple is a leading technology company."*

Without semantic understanding, an NLP model might interpret *"Apple"* as a fruit instead of a company.

Processing the Sentence: “I saw the bat flying." and explore how a machine could potentially interpret this sentence using a technique called word embeddings.

- **Tokenization:** The sentence is first tokenized into individual words: ["I", "saw", "the", "bat", "flying"]
- **Embedding Lookup:** Each word is then looked up in a pre-trained word embedding model (like Word2Vec, GloVe, or FastText). This gives us a vector for each word.

  OR

Contextualization: More advanced techniques like BERT or ELMo can generate contextualized embeddings, where the vector for "bat" would be influenced by the surrounding words ("flying" in this case).

- **Disambiguation with Embeddings:** The embedding for "bat" would be compared to the embedding for "fly" (or related words like "wings," "air," etc.). The animal sense of "bat" would likely be closer to these concepts in the embedding space.
- Based on the embedding similarities, the machine could potentially infer that the sentence is more likely about a flying animal. It might assign a higher probability to the animal sense of "bat" and use this interpretation for downstream tasks (e.g., translation, question answering).

How "Bat" Gets Its Embedding?

Word Embeddings:

- Representing Words as Vectors: Word embeddings are dense vector representations of words. Each word is mapped to a point in a high-dimensional space, where semantically similar words are closer together.
- Capturing Relationships: These vectors capture various relationships between words, such as:
    - Similarity ("cat" and "dog" would be closer than "cat" and "airplane")

Word embeddings are not created in isolation. They are learned from massive amounts of text data (corpora). This data can be anything from Wikipedia articles to books to web pages.

The idea is that the way words are used in these texts reveals their meaning.

Word2Vec:

- This popular algorithm uses a neural network to predict either:
    - The surrounding words given a target word (CBOW - Continuous Bag of Words).
    - The target word given the surrounding words (Skip-gram).
- During training, the network adjusts the embedding vectors to better predict these relationships

Note: Project on CBOW and Skip-gram model (implement from scratch)

Other models: Glove, FastText (you may implement these algo from scratch as well)

BERT, ELMo (Contextual Embeddings):

- These models use transformer architectures and are trained to predict masked words in sentences or to predict the next sentence.
- They create embeddings that are dependent on the words that surround the target word

- The output of these algorithms is a high-dimensional vector space. For example, a Word2Vec model might create vectors with 300 dimensions.
- Each dimension represents a feature, although these features are not easily interpretable by humans.
- The values in the vector are learned during training to capture the word's relationships with other words.

Additional Notes

Word Embeddings vs Contextual Embedding

Example:

"I love to eat Granny Smith apples. They're my favorite kind of apple. I always buy them from the Apple store near my house. The store also sells Apple Watches, but I'm not really interested in those. I'm more of an Android person when it comes to technology. Speaking of apples, I need to pick some up from the market later."

Word Embeddings (e.g., Word2Vec):

- **Single Embedding:** Word2Vec would create a single embedding for "apple," which would be an average representation of all its senses encountered during training. This embedding would be a blend of the fruit, the company, and potentially other less common senses.
- **Ambiguity:** When processing the example paragraph, Word2Vec would use the same embedding for all instances of "apple." This would lead to ambiguity and difficulty in distinguishing between the different meanings.
- **Limited Context:** Word2Vec has a limited context window, so it wouldn't be able to fully utilize the surrounding words to disambiguate "apple." For example, it might not connect "Granny Smith" to the fruit sense or "store" and "Watches" to the company sense.

Contextual Embeddings (e.g., BERT):

- **Dynamic Embeddings:** BERT would generate different embeddings for "apple" depending on the context.
    - In the first sentence, the presence of "eat" and "favorite kind" would lead to an embedding closer to the fruit sense.
    - In the second sentence, "store" and "Watches" would result in an embedding closer to the company sense

- Named Entity Recognition (NER):
  Apple (ORGANIZATION), technology (CATEGORY), company (ENTITY)

✅ Applications: Sentiment analysis, text summarization, search engines, chatbots.

Application (NER)

Semantic Analysis is about understanding meaning in text, and Named Entity Recognition (NER) is one of its most powerful applications. NER helps NLP systems identify and categorize important entities (like names, locations, dates) from text, enabling machines to understand contextual meaning.

```
text = "Apple Inc. launched the new iPhone in California on September 12, 2023."

# Process text through NLP pipeline

doc = nlp(text)

# Print detected entities

print("Named Entities, Type & Position:")

for ent in doc.ents:

   print(f"{ent.text} ({ent.label_}) - Start: {ent.start_char}, End: {ent.end_char}")
```

```
Named Entities, Type & Position:
Apple Inc. (ORG) - Start: 0, End: 10
iPhone (ORG) - Start: 28, End: 34
California (GPE) - Start: 38, End: 48
September 12, 2023 (DATE) - Start: 52, End: 70
```

Understanding Meaning, Not Just Words

- "Apple" can mean a **fruit** 🍎 or a **company** 🏢 (Apple Inc.).
- Semantic analysis ensures that **"Apple Inc." is classified as an Organization** instead of a fruit.

Extracting Important Information from Text

- Words like *"New York," "Amazon,"* and *"Tesla"* are not just words; they are **places, companies, and products**.
- Semantic analysis helps **machines understand what they refer to**.

Disambiguating Words Based on Context

- *"Paris is beautiful."* → **Paris (GPE - Location)**
- *"Paris won the gold medal in swimming."* → **Paris (PERSON - Athlete)**
- Without **semantic analysis**, NLP would struggle to **assign the right meaning**.

How NER Achieves Semantic Analysis?

NER uses machine learning and deep learning techniques to identify semantic meaning in text.

Technologies Used for Semantic Analysis & NER

Several advanced NLP techniques and models are used to achieve **semantic understanding** in NER:

1. Rule-Based Approaches (Pattern Matching)
    - Uses predefined **grammar rules** and **regular expressions** to find entities.
    - Example:
        - `"Dr. John Smith"` → Matches **title + name pattern** → **PERSON**

🚫 **Limitation:** Hardcoded rules fail for unseen variations.

---

2. Machine Learning-Based NER
- **Trains models on labeled text data** to predict named entities.
- Uses algorithms like **Naïve Bayes, Conditional Random Fields (CRF), Hidden Markov Models (HMM).**

✅ **Example:**

- Training a model on **news articles** to detect **company names, locations, and dates**.

🚫 **Limitation:** Needs **large labeled datasets**.

3️⃣ Deep Learning-Based NER (State-of-the-Art)

Modern NLP models use **deep learning architectures** like:

🔹 Recurrent Neural Networks (RNN) & Long Short-Term Memory (LSTM)

- Models like **Bi-LSTM-CRF** are used for sequential labeling tasks (like NER).
- Understands **contextual relationships** in text sequences.

✅ **Example:**

- **Sentence:** *"Amazon is expanding in India."*
- **Bi-LSTM Output:**
  - `"Amazon"` → **ORG**
  - `"India"` → **GPE (Location)**

🚫 **Limitation:** Computationally expensive.

🔹 Transformer-Based Models (BERT, GPT, T5)

- **BERT (Bidirectional Encoder Representations from Transformers)** achieves **state-of-the-art NER accuracy**.
- Can process long text **with contextual awareness** (understanding words in relation to others).

✅ **Example: Using BERT for NER**

- **Input:** *"Tesla will release new self-driving software in 2024."*
- **BERT Output:**
  - `"Tesla"` → ORG
  - `"2024"` → DATE
  - `"self-driving software"` → PRODUCT

🔥 **Advantage:** Works **better than rule-based or ML models** in complex, real-world texts.

Pragmatic Analysis (Contextual Understanding)

Pragmatic analysis in Natural Language Processing (NLP) is the process of understanding the intended meaning of a sentence beyond its literal interpretation by considering context, speaker intentions, and real-world knowledge. It helps machines grasp implied meanings, sarcasm, politeness, indirect requests, and conversational nuances that are not explicitly stated in text.

While syntax analysis ensures grammatical correctness and semantic analysis determines meaning, pragmatic analysis interprets the intended message in different scenarios.

Language is **ambiguous** and often depends on **context**. Consider the sentence:

"Can you pass the salt?"

- **Literal Meaning (Syntactic & Semantic Analysis):** The speaker is asking about the listener’s ability to pass salt.
- **Pragmatic Meaning:** It is actually a polite request to pass the salt.

Without **pragmatic analysis**, an NLP system might fail to recognize **indirect speech acts** like requests, humor, or sarcasm.

Key Components of Pragmatic Analysis

1️⃣ **Speech Acts:** Understanding whether a statement is a **request, command, promise, or question**.

- *Example: "It’s cold in here."*
    - **Literal:** A statement about temperature.
    - **Pragmatic Meaning:** An indirect request to **close the window**.

2️⃣ **Contextual Meaning:** Considering the **speaker, listener, and situation** to interpret meaning correctly.

- *Example: "That’s just great!"*
    - **Literal:** A positive statement.
    - **Pragmatic Meaning:** If spoken sarcastically, it may mean the **opposite** (i.e., "That’s terrible").

3️⃣ **Conversational Implicature:** Interpreting **unstated meanings** using **Grice’s Maxims** (relevance, quantity, quality, manner).

- *Example: "Are you going to the party?"*
    - Response: *"I have an early meeting tomorrow."*
    - **Implied Meaning:** The speaker is **not going** to the party.

4️⃣ **Coreference Resolution:** Identifying **pronouns and references** in dialogue.

- *Example: "John told Mike he won the award."*
    - **Who won the award?** Pragmatic analysis helps resolve such ambiguity.

Applications of Pragmatic Analysis in NLP

Chatbots & Virtual Assistants: Helps AI understand indirect requests (*"I'm tired" → Suggesting sleep mode*).
Sentiment Analysis: Detects sarcasm and hidden emotions in text (*"Oh, great! Another meeting!"*).
Machine Translation: Ensures cultural and contextual relevance in translations.
Dialogue Systems: Enables AI to maintain **coherent and context-aware conversations**.

Challenges in Pragmatic Analysis

🚧 **Ambiguity:** Many phrases have multiple interpretations.
🚧 **Sarcasm & Humor:** Hard to detect without **context & tone**.
🚧 **Cultural Differences:** Meaning varies across languages and social contexts.
🚧 **Real-World Knowledge:** AI lacks human-like common sense, making **contextual understanding difficult**.

Pragmatic Analysis of the Sentence

"After accidentally dropping the cake on the floor, she said, 'This cake is amazing!'"

Literal Meaning (Semantic Analysis):

- The speaker is saying that the cake is amazing.
- A naive AI without pragmatic understanding might assume that the person genuinely thinks the cake is great.

Pragmatic Meaning (Contextual Analysis):

- The phrase *"accidentally dropping the cake on the floor"* sets up the context.
- In reality, dropping a cake is an unfortunate event.
- The speaker's statement *"This cake is amazing!"* **contradicts** the expected reaction (disappointment or frustration).
- This is an example of **sarcasm**, where the speaker means the opposite of what they are literally saying.

How Pragmatic Analysis Works Here

To correctly interpret the **true intent** behind this statement, an NLP system must analyze:

1️⃣ **Contextual Clues:**

- The event before the statement (*dropping the cake on the floor*) suggests a negative situation.
- The adjective *"amazing!"* seems **incongruent** with the negative event.

2️⃣ **Sarcasm Detection (Project):**

- Humans often use **exaggeration** (*"amazing!"* in a bad situation) to express **sarcasm**.
- The **tone of speech** (if spoken) or **text markers** (e.g., exclamation mark) can hint at sarcasm.

3️⃣ **Common Sense & Real-World Knowledge:**

- If the cake was dropped, it is likely ruined.
- A logical reaction should be frustration, not praise.
- Pragmatic analysis **infers** that the statement means **the opposite** of its literal meaning.

4️⃣ **Sentiment Reversal in NLP Models:**

- **Lexical sentiment analysis** may classify *"amazing"* as positive.
- **Context-aware models** (like transformers trained on sarcastic data) **learn** that positive words in a negative situation might indicate **sarcasm**.

How NLP Systems Handle This

✅ **Rule-Based Approaches:**

- Look for **contradictory context** (e.g., negative event + positive sentiment).
- Maintain **event history** (e.g., *"dropping the cake"* → *negative experience*).

✅ **Machine Learning-Based Approaches:**

- **Sentiment Analysis Models:** Trained on sarcasm-labeled data.
- **Context-Aware AI (BERT, GPT, T5):** Understand **sentence-level dependencies** rather than just individual word meanings.
- **Multi-modal NLP:** If voice tone or facial expressions are available (in videos), AI can **detect tone shifts** indicating sarcasm.

Note:

"Imagine you ask your friend, 'Can you open the window?' and they respond, 'Yes, I can,' but they don't actually open it. Did they understand what you meant?

Expected Outcome:

- **Semantic Analysis:** The system understands the **literal meaning** of the sentence (*ability to open the window*).
- **Pragmatic Analysis:** The system understands the **implied meaning** (it's a **request**, not a yes/no question).
- **AI Challenge:** Why do virtual assistants sometimes fail to act on indirect requests? How can AI improve in understanding human-like interactions?

This question encourages students to **differentiate between semantics (literal meaning) and pragmatics (context and intent)** while also reflecting on real-world NLP challenges! 🚀

How AI Processes This Sentence: "Can you open the window?"

AI uses **different layers of Natural Language Processing (NLP)** to **interpret and act on the sentence**. Let's break it down step by step:

---

1. Semantic Analysis (Literal Meaning)

💡 **How AI interprets semantically:**

- The phrase **"Can you open the window?"** is a grammatically correct **yes/no question** about **ability** (i.e., whether the person is physically capable of opening the window).
- If an AI assistant (like Siri or Alexa) **only** applies **semantic analysis**, it may respond **literally**, saying:
  "Yes, I can." (but without actually opening the window!)

⚠️ **Problem:**

- The system **understands words and grammar**, but it **misses the intent** behind the request.

---

2. Pragmatic Analysis (Context + Intent)

💡 **How AI interprets pragmatically:**

- A human would recognize that this is **not a question about ability** but rather an **indirect request** to actually **open the window**.

- **Pragmatic AI models** consider:
    - **Conversational norms:** People don't always state requests directly.
    - **Contextual clues:**
        - Is the room hot?
        - Did the user give a similar command before?
        - What is the usual response to such requests?

A more advanced AI (with pragmatic analysis) would respond appropriately:
"Sure! Opening the window now."

3️⃣ How AI Achieves Pragmatic Understanding?

To go beyond literal meaning, AI needs:

1. **Intent Recognition (NLU – Natural Language Understanding)**
    - AI models trained on **real-world conversations** recognize **indirect speech acts** (e.g., requests, commands, sarcasm).
    - Uses models like **BERT, GPT, or T5** to **predict user intent** based on similar patterns in large datasets.
2. **Dialogue Context & Memory**
    - **Advanced AI assistants (Google Assistant, ChatGPT, Alexa)** track **previous interactions** to understand conversation flow.
    - If the user previously asked about **room temperature**, AI can **infer the intent** behind "Can you open the window?"
3. **Action Execution (AI Acting on Meaning)**
    - AI integrates with **IoT devices** (like smart home systems) to **physically open the window** when it understands the intent.

Discourse Analysis

DO IT YOURSELF (DIY)