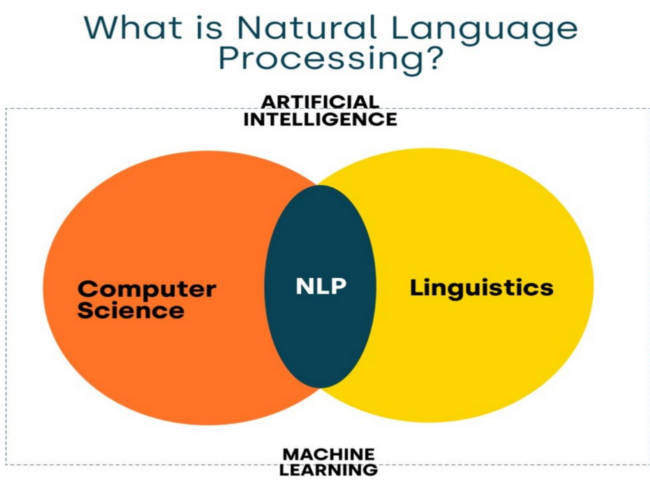
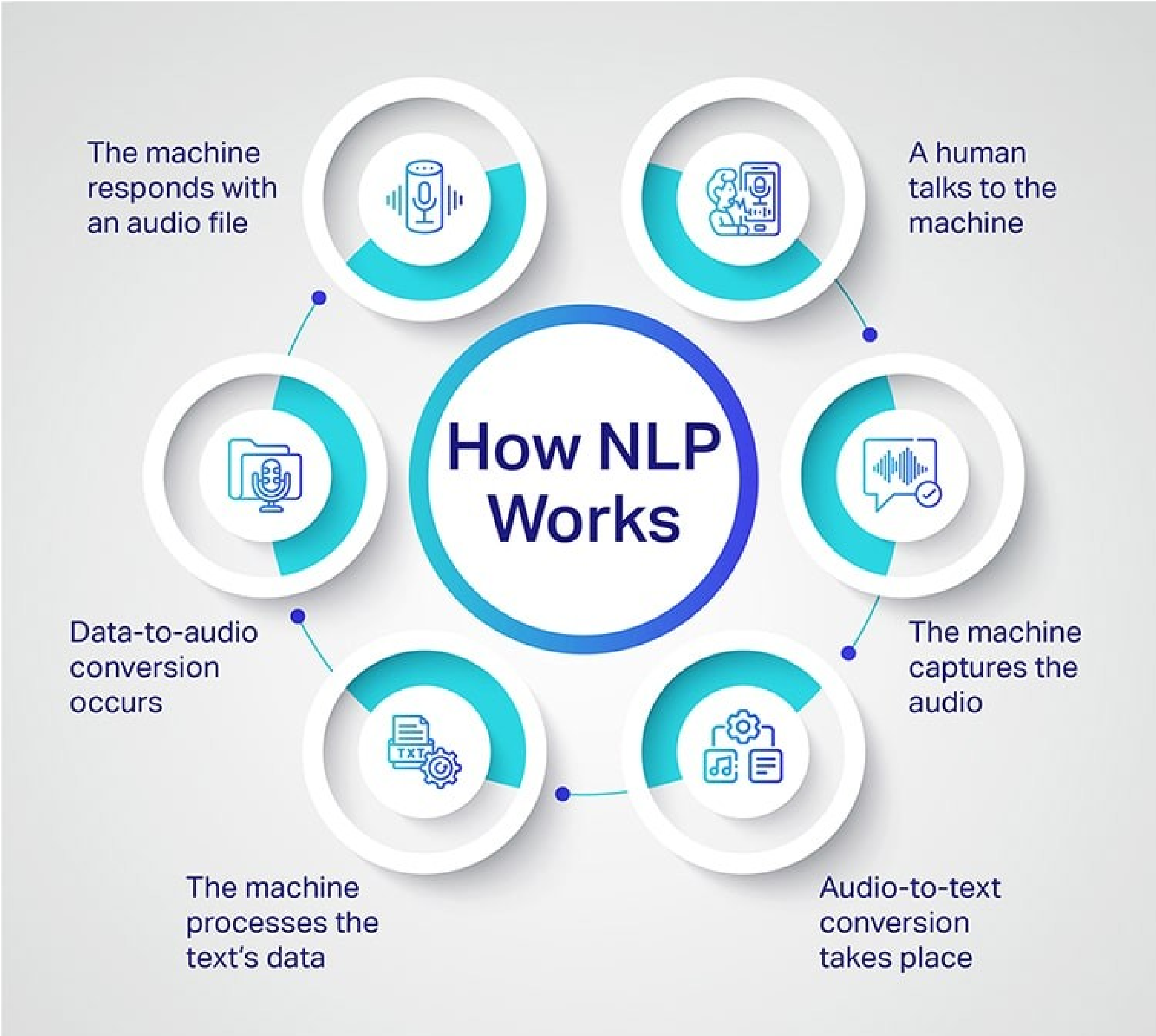


Natural Language Processing

NLP is a subfield of computer science and artificial intelligence (AI) that uses machine learning to enable computers to understand and communicate with human ...



Natural Language Processing (NLP) systems use machine learning algorithms to analyze large amounts of unstructured data and extract relevant information. The algorithms are trained to recognize patterns and make inferences based on those patterns. Here's how it works:

- Text Processing: Discuss techniques like tokenization, stemming, and lemmatization.
- Syntactic Analysis: Explain parsing and grammar analysis.
- Semantic Analysis: Cover meaning extraction and context understanding.

- **Applications of the NLP:**
- Real-time language translation
- Spam filters in email services
- Voice assistants and chatbots

- Text summarization
- Autocorrect features
- Sentiment analysis and more

Approaches to Natural Language Processing.

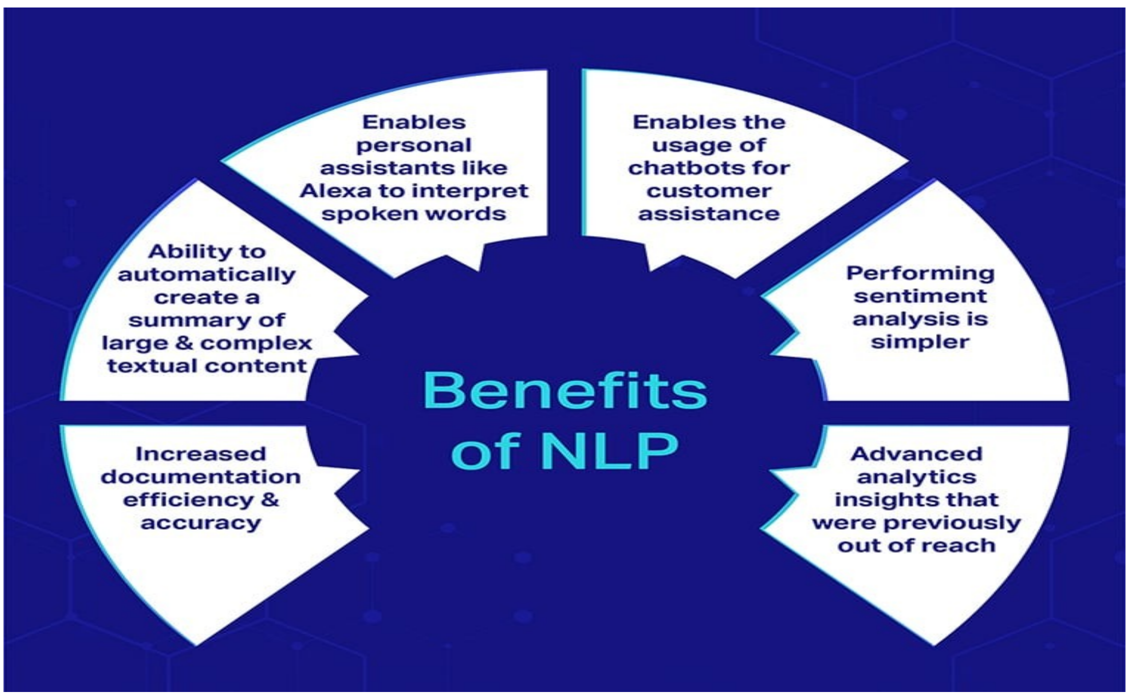
Some of the approaches to NLP are:

Supervised NLP: Trains models on labeled data to make accurate predictions, like classifying emails.

Unsupervised NLP: Works with unlabeled data to find patterns, useful for tasks like topic modeling.

Natural Language Understanding (NLU): Helps machines interpret and understand the meaning of human language.

Natural Language Generation (NLG): Creates human-like text, such as writing summaries or chatbot responses.



Overview of linguistics:

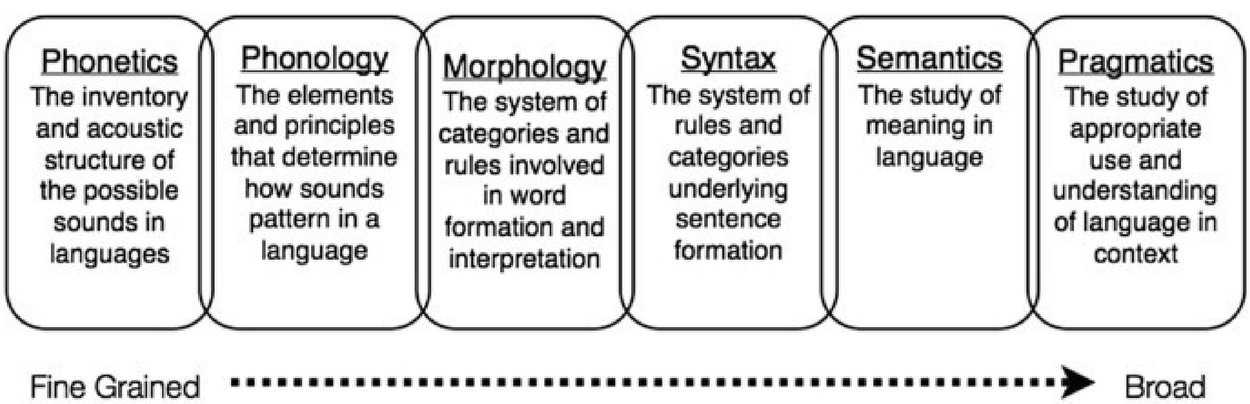
- Linguistics helps in **breaking down human languages** into parts that a machine can understand and process.
- **Real-life Example:**
  - Think of Google Translate or Siri. They need to understand the structure and meaning of the language you're speaking to

translate or respond
accurately. Linguistics helps these systems understand and process your words.

- Linguistics is the **scientific study of language**. It includes understanding how languages are structured, how they evolve, and how they are used in real-world communication.
- Linguistics is the study of human language, **focusing on its structure (syntax), meaning (semantics), sound (phonetics), and use in society (pragmatics)**. It provides the theoretical basis for Natural Language Processing, enabling machines to process, understand, and generate human language.

Key Components of Linguistics in NLP:

1. **Phonetics & Phonology:**
    - **Phonetics** is the study of sounds in language. Machines need to recognize and process speech sounds (like in voice assistants).
    - **Phonology** deals with how these sounds are organized in a particular language (e.g., how the sound /k/ appears in "cat" vs. "kit").
2. **Morphology:**
    - The study of words and their structure (e.g., “run” vs. “running”). NLP systems break down words into meaningful parts (morphemes) for better understanding.
3. **Syntax:**
    - Syntax is the arrangement of words in sentences. For example, “The cat sat on the mat” follows a specific word order in English. NLP models need to identify and analyze this structure for tasks like parsing or translation.
4. **Semantics:**
    - Semantics focuses on meaning. It helps systems understand word meanings (e.g., “bank” as a financial institution vs. “bank” of a river) and sentences’ overall meaning.
5. **Pragmatics:**
    - Pragmatics is about the context in which language is used. For example, "Can you pass the salt?" is a request, not just a question, based on the context.



Grammars and Languages:

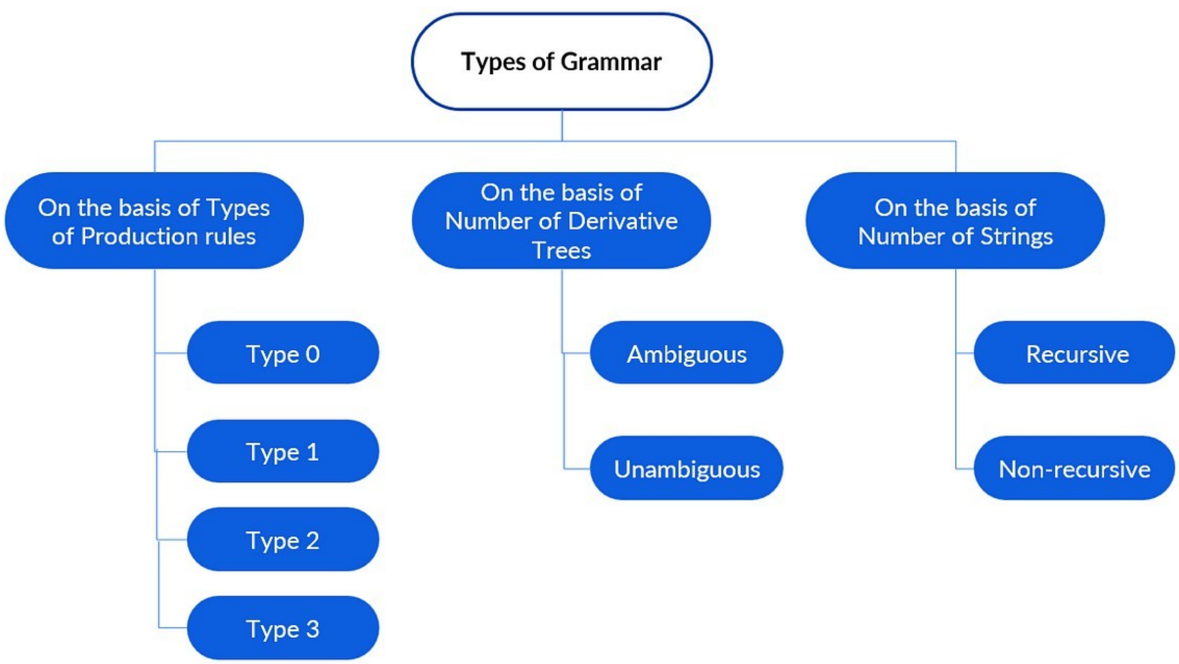
- When you type something in a search engine or use a chatbot, the machine checks the grammar of what you wrote to understand your query. For example, if you type **"How many apples in the basket?"** a system might fix the grammar to **"How many apples are in the basket?"** to process it better.

- Grammar is a set of rules that define how sentences are structured in a language. Grammar tells you how to arrange words to make correct and meaningful sentences.

Key Concepts in Grammars and Languages:

1. **Formal Grammar:**
   - Formal grammar refers to a set of rules used to generate or parse sentences in a language. It helps break down a sentence into its components like nouns, verbs, adjectives, etc., which is crucial for understanding and processing language.
   - **Example:** A simple rule in English could be “Sentence → Noun Phrase + Verb Phrase.”
2. **Types of Grammars:**
   - **Context-Free Grammar (CFG):** A grammar where the rules are independent of the context. It’s very useful in programming languages and simple sentence structures.
     - **Example:** "S → NP + VP" (A sentence is made up of a noun phrase and a verb phrase).
   - **Context-Sensitive Grammar (CSG):** The rules depend on the context in which the word appears. These grammars are more complex and allow for more detailed language structures.
3. **Languages:**
   - A **language** is a set of strings (sentences) that can be generated by a grammar. In NLP, we deal with **formal languages** that can be described by formal grammars.
   - **Example:** The language of all sentences that can be made from the rule “S → NP + VP” is a simple subset of English.

4. **Chomsky Hierarchy:**
   - This hierarchy classifies grammars based on their complexity. It includes **Type 0** (most general) to **Type 3** (simplest).
     - Type 3: Regular grammars (e.g., finite state automata)
     - Type 2: Context-free grammars (CFGs)
     - Type 1: Context-sensitive grammars (CSGs)
     - Type 0: Unrestricted grammars (used for Turing machines)



Basic Parsing Techniques:

Parsing is **essential for translating natural language into a form that machines can process**, and it's foundational for many NLP tasks such as translation, question answering, and speech recognition.

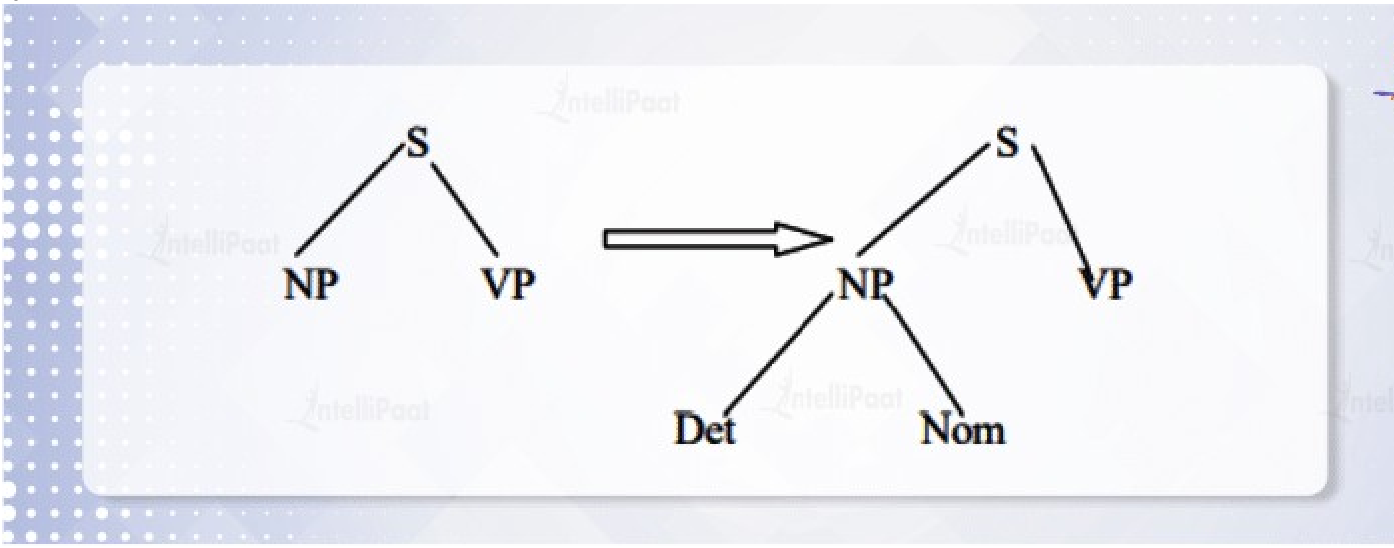
Example: When you ask a voice assistant, "What's the weather like today?", it needs to understand not just the words, but how those words are structured. Parsing allows the system to break down the sentence into parts (e.g., "what" as a question word,
"weather" as the subject, "like today" as the verb phrase) so it can correctly respond with a weather report.

Definition: Parsing is the process of analyzing a sentence to determine its grammatical structure. It helps break down a sentence into parts to understand how words relate to each other.

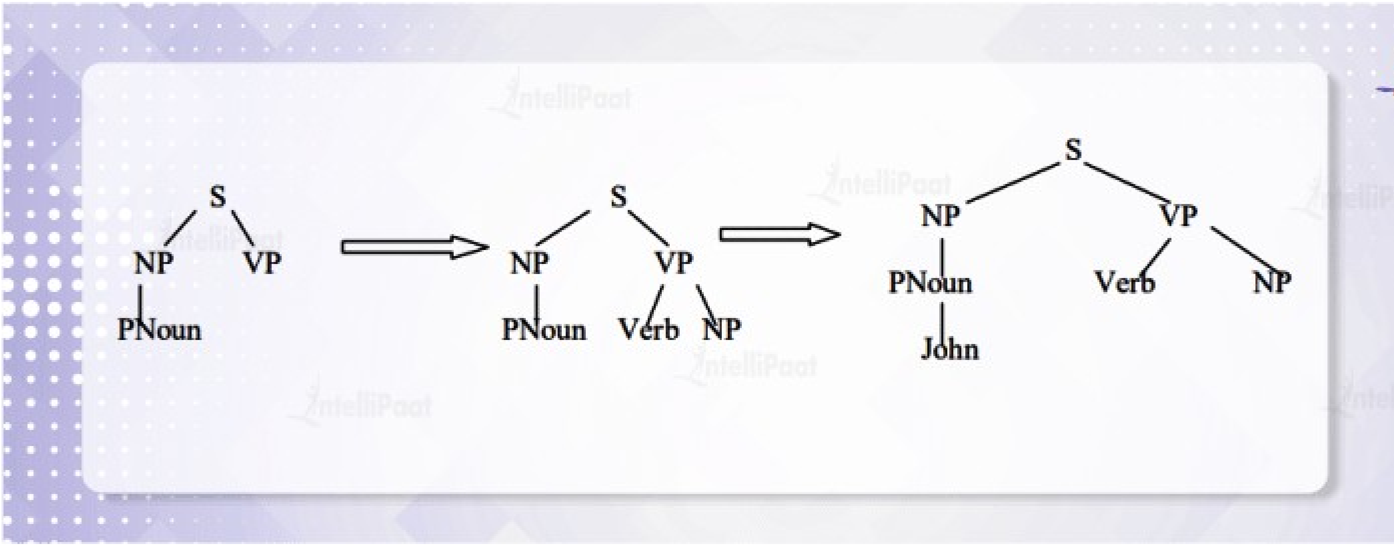
Key Concepts in parsing:

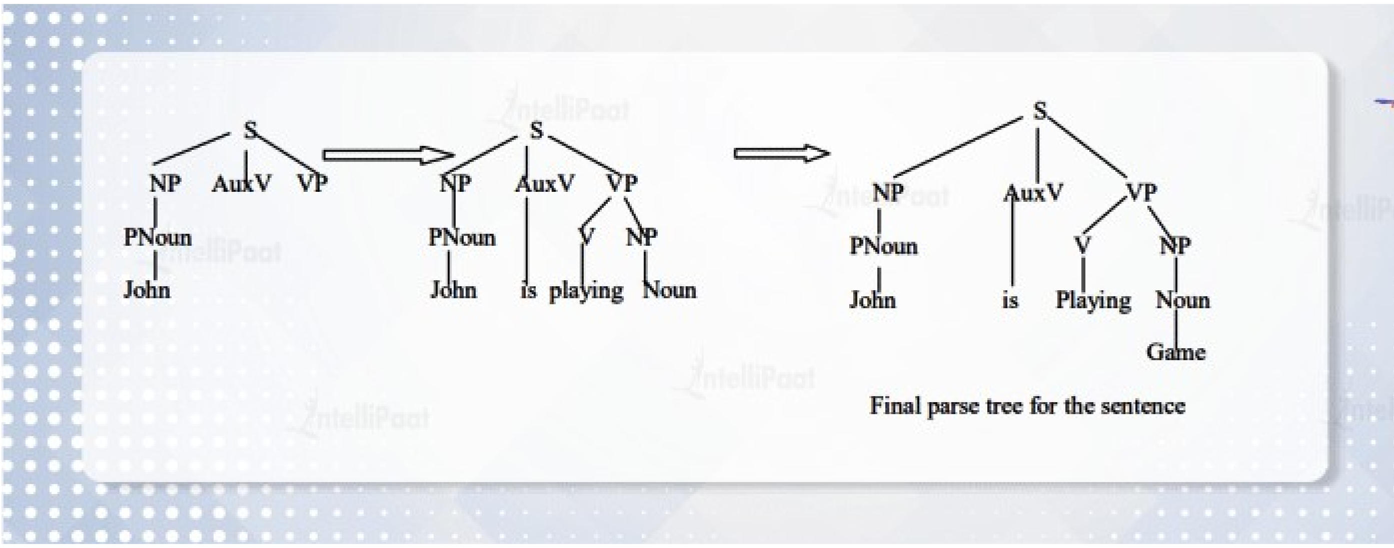
Top-Down Parsing:

- **How it works:** It starts with the highest-level rule in the grammar (usually the sentence or "S") and tries to break it down into smaller components (noun phrase, verb phrase, etc.).
- **Advantages:** It's simple and easy to understand.
- **Disadvantages:** It can be inefficient because it might try to apply rules that are not relevant for the given sentence.
- **Example:** For the sentence "The cat sleeps," the parser would start with "S" and try to match it with "NP + VP."

- Let's consider the grammar rules:
- Sentence = S = Noun Phrase (NP) + Verb Phrase (VP) + Preposition Phrase (PP)
- Take the sentence: "John is playing a game", and apply Top-down parsing
- 

- If part of the speech does not match the input string, backtrack to the node NP.

- 

- Part of the speech verb does not match the input string, backtrack to the node S, since PNoun is matched.
- 

- For example: https://www.geeksforgeeks.org/working-of-top-down-parser/
-

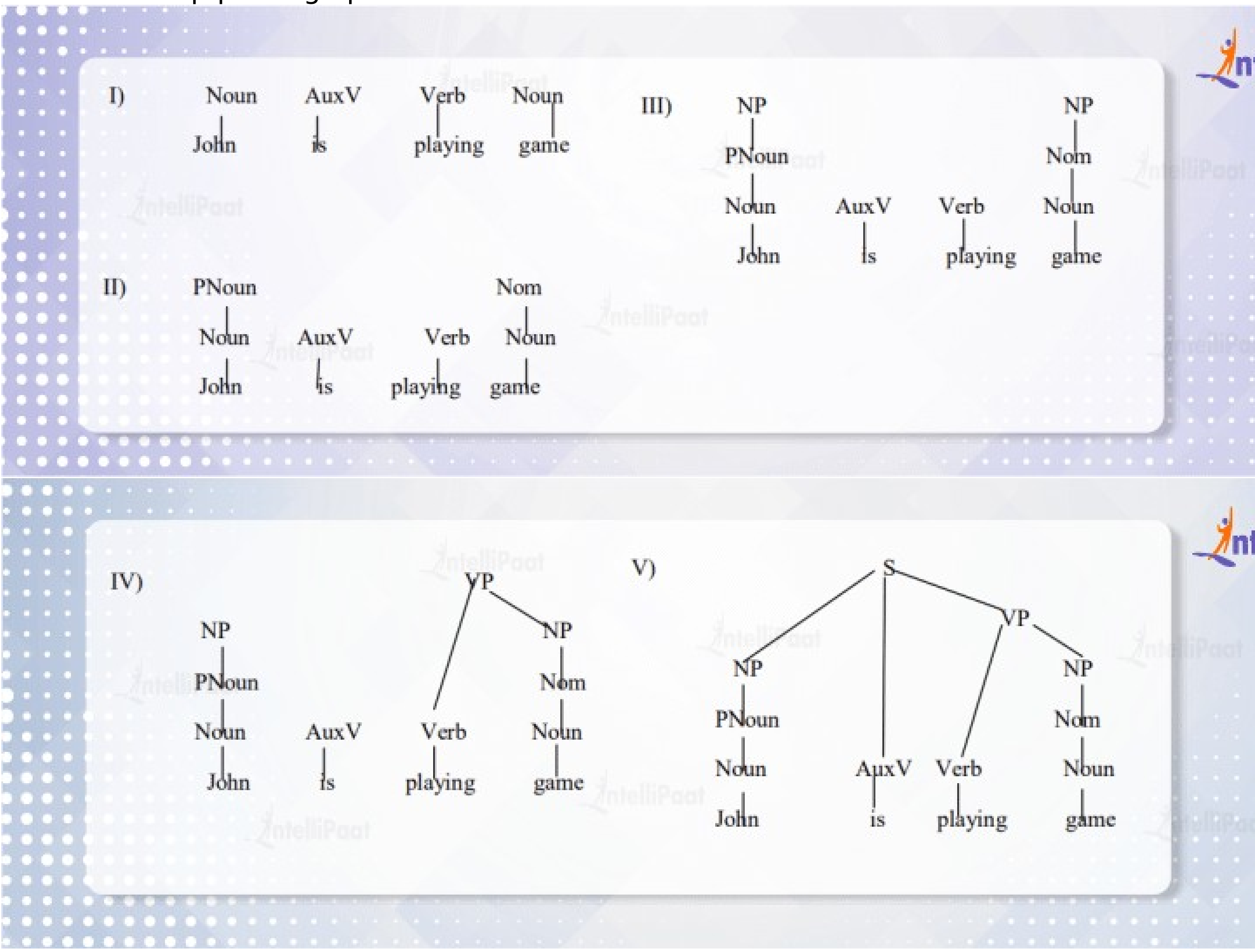
Bottom-Up Parsing:

- **How it works:** This approach starts with the words of the sentence and tries to combine them into larger units (noun phrases, verb phrases) until a complete structure (sentence) is formed.

- **Advantages:** It can be more efficient in some cases because it doesn't explore irrelevant rules.
- **Disadvantages:** It may require more memory and can be harder to implement.

● **Example:** The parser first identifies "The" as a determiner and "cat" as a noun, and then combines them into an NP (noun phrase), before combining the NP with "sleeps" to form the sentence.

Considering the grammatical rules stated above and the input sentence "John is playing a game",
The bottom-up parsing operates as follows:



●

Earley Parser (Chart Parsing):

● **How it works:** A more sophisticated parser that combines top-down and bottom-up strategies. It uses a chart (a table) to store partial parses of the sentence as it processes it. This method is useful for handling ambiguous sentences.
● **Advantages:** It's more efficient and can handle more complex grammars.
● **Disadvantages:** It can be slower for very large datasets or ambiguous grammars.
● **Example:** If a sentence has multiple interpretations (like "I saw the man with the

telescope"), the Earley parser can handle both possibilities without trying to parse everything from scratch.

Shift-Reduce Parsing:

- **How it works:** This method works by shifting the input symbols (words) into a stack and then reducing them to higher-level structures (e.g., combining words into phrases). It’s widely used in **bottom-up parsing**.
- **Advantages:** It’s efficient for many languages and works well in practice.
- **Disadvantages:** It may require a sophisticated understanding of context or additional mechanisms to deal with ambiguities.
- **Example:** In the sentence "The cat sat," the parser would shift "The" onto the stack, then "cat," then reduce to form a noun phrase, then shift "sat" and reduce to a verb phrase, finally combining the noun phrase and verb phrase into a full sentence.

Stack	Input Buffer	Action
$	id + id * id $	
$id	+ id * id $	Shift id
$F	+ id * id $	Reduce F-> id
$T	+ id * id $	Reduce T -> F
$E	+ id * id $	Reduce E -> T
$E +	id * id $	Shift +
$E + id	* id $	Shift id
$E + F	* id $	Reduce F-> id
$E + T	* id $	Reduce T -> F
$E + T *	id $	Shift *
$E + T * id	$	Shift id
$E + T * F	$	Reduce F-> id
$E + T	$	Reduce T -> F
$ E	$	Reduce E -> E + T

Shift-Reduce Parsing on Input String id + id * id

Topic References:

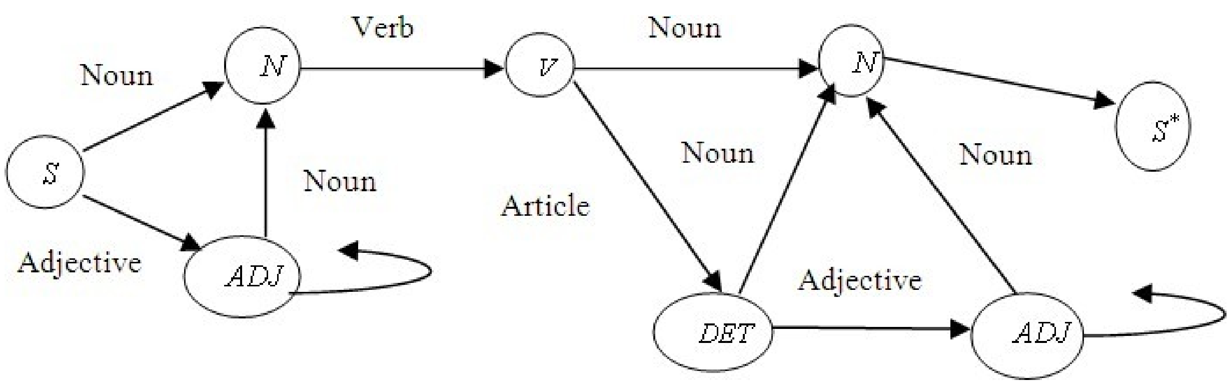
- BASIC PARSING TECHNIQUES IN NATURAL LANGUAGE PROCESSING
- Bottom-Up Parsing An Introductory Example
- Difference Between Top Down and Bottom Up Parsing.
- ISSN: 2278-6252 PARSING TECHNIQUES: A REVIEW

Transitional Networks:

For parsing sentences and representing the flow of words through states. Transitional Networks are a powerful way to model grammatical rules and are especially useful for handling ambiguity and flexibility in sentence structure.

Example: Consider a **speech recognition system** like Apple's Siri or Google Assistant. When you say a sentence like "I want to go to the park," the system has to break down the sentence and understand it word by word. A **Transitional Network** helps model how the system moves through different states of understanding (e.g., detecting a verb, recognizing a destination, etc.) as it processes each word.

Definition: A **Transitional Network** is a finite state machine used to represent grammatical structures. It consists of nodes (states) connected by directed edges (transitions). Each node represents a part of a sentence (like a noun phrase, verb
phrase), and transitions represent the grammatical rules that move the process from one part to another. It is used to model syntax and sentence structure in a sequential,
flexible manner, often used in speech processing, natural language understanding, and grammar-based parsing.



Key Concepts in Transitional Networks:

States and Transitions:

- **States** represent different stages or components in the grammar of a sentence (e.g., noun phrases, verb phrases).
- **Transitions** define how to move from one state to another based on input (i.e., a word or symbol in the sentence).

Real-Life Example: If you input the sentence “I want to eat pizza,” the system might first identify “I” as the subject, then transition to the verb “want,” and finally recognize the
verb phrase “eat pizza.”

Finite State Machine (FSM):

- A **Finite State Machine (FSM)** is a model used to represent how a system can transition between different states based on input. In TNs, FSMs are used to handle transitions between states as the parser processes words.
- **Example:** The system might start in an initial state where it expects a noun (e.g., "dog"), then transition to a state expecting a verb (e.g., "runs"), and finally transition to a state expecting an object or complement (e.g., "in the park").

Handling Ambiguity:

- **Ambiguity** arises when a sentence can be interpreted in multiple ways (e.g., "The cat saw the dog with the telescope"). A TN can handle such ambiguity by creating multiple parallel states or paths, each representing a different interpretation of the sentence.
- **Real-Life Example:** In automatic translation systems, ambiguity in a sentence is often resolved using TNs by considering different potential meanings for a word or phrase. This is especially important for languages with flexible word orders (like Japanese or Hindi).

Sequential Processing:

- **TNs** process sentences word by word, moving from state to state as each new word is encountered. This sequential processing allows TNs to represent the flow of language naturally and efficiently, making them well-suited for tasks like speech recognition, where input is continuous.
- **Example:** In a **speech-to-text system**, as each word is spoken, the system moves through states, interpreting each word in the context of what has already been processed (just like parsing written text).

Applications:

1. **Speech Recognition Systems:**
    - In systems like Siri, Amazon Alexa, or Google Assistant, TNs are used to process spoken language. As words are spoken, the system transitions

from one state to another, determining the meaning of each word and how it fits into the overall sentence structure.

- **Example:** When you say, "Find a pizza place near me," the system first processes "find" as the verb, transitions to a state where it expects an object (pizza place), and finally interprets "near me" as a location modifier.

2. **Automatic Translation:**
    - **Google Translate** and other machine translation systems use TNs to break down sentences into smaller units and then transition through different states to translate each part. TNs handle the syntactic structure of the source language and ensure that the translation is grammatically correct.
    - **Example:** In translating "I eat an apple" into Spanish, TNs would ensure that the subject "I" transitions to the verb "eat," and then "apple" becomes "manzana" in the translated sentence.

Use Case Problem: Understanding Sentences Using a Transitional Network

Problem:

Imagine you are developing a **speech-to-text** system for a simple voice assistant. The system must interpret spoken sentences, breaking them down into grammatical components to understand user requests. However, the system should also handle variations in sentence structure and word order.

For example, consider the following sentences:

1. "I want to buy a new phone."
2. "Buy a new phone, I want."
3. "A new phone, I want to buy."

In all these cases, the user is trying to express the same request: "I want to buy a new phone." But the word order and structure differ. The challenge is to **parse** these sentences and **extract the correct meaning** despite variations in structure.

Solution Using Transitional Networks (TNs):

Step 1: Define the States and Transitions

- Each sentence can be represented as a sequence of **states** and **transitions**:

- **State 1:** Sentence → (Start with Subject)

- ○ **State 2:** Subject → (e.g., "I" or "Buy")
- ○ **State 3:** Verb Phrase → (e.g., "want to buy")
- ○ **State 4:** Object → (e.g., "a new phone")

The transitions define how the words in the sentence connect. For example:

- From **State 1** ("Sentence") to **State 2** (Subject), the transition could be triggered by the word **"I"**.
- From **State 2** (Subject) to **State 3** (Verb Phrase), the transition could be triggered by the verb **"want"** or **"buy"** depending on the word order.

Step 2: Handle Different Sentence Orders

Now, let's see how each sentence flows through the TN:

1. **Sentence 1: "I want to buy a new phone."**
   - ○ **Transition:** Start with **State 1 (Sentence)** → move to **State 2 (Subject)** with "I" → transition to **State 3 (Verb Phrase)** with "want" → transition to **State 4 (Object)** with "a new phone."
   - ○ The TN transitions through the states from **Subject** → **Verb Phrase** → **Object** in a straightforward manner, which is the standard order.
2. **Sentence 2: "Buy a new phone, I want."**
   - ○ **Transition:** Start with **State 1 (Sentence)** → transition to **State 3 (Verb Phrase)** with "Buy" (this shifts the system's expectation from the typical

     Subject to Verb) → transition to **State 2 (Subject)** with "I" → transition to **State 4 (Object)** with "a new phone."
   - ○ The system might recognize that the sentence is asking for the same action but in a **reversed order**. The TN allows flexibility to move to the Verb Phrase first, and then continue parsing.
3. **Sentence 3: "A new phone, I want to buy."**
   - ○ **Transition:** Start with **State 1 (Sentence)** → transition to **State 4 (Object)** with "a new phone" → transition to **State 2 (Subject)** with "I" → transition to **State 3 (Verb Phrase)** with "want" and "buy."
   - ○ Here, the system first identifies the **Object** and then proceeds backward to understand the Subject and Verb Phrase.

Handling Ambiguity:

In each of these cases, the TN will handle the ambiguity by **branching** into different states depending on the word order:

- The system keeps track of different paths, which allow it to handle reversed or scrambled word orders.
- **Example:** If "want" is found in one state, it might lead the system into a verb phrase first, while in another state, "I" might immediately transition into the subject position.

Example of the TN Diagram for Sentence 1:

- **State 1 (Sentence) → State 2 (Subject) → State 3 (Verb Phrase) → State 4 (Object)**

For **Sentence 2** ("Buy a new phone, I want"):

- **State 1 (Sentence) → State 3 (Verb Phrase) → State 2 (Subject) → State 4 (Object)**

Semantic Analysis and Representation Structures:

How machines interpret the meaning behind sentences and how that meaning is represented in a structured way. While syntax deals with the structure of sentences, **semantics** deals with the meaning of words, phrases, and sentences. **Semantic analysis** ensures that the computer understands the relationships between words and their meanings.

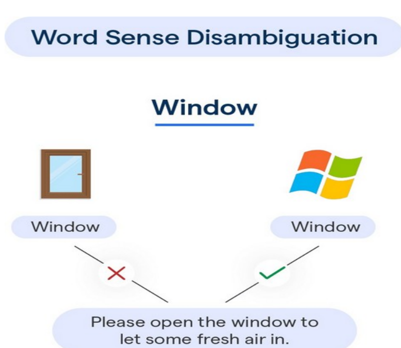
-> Example: Consider the sentence, "I went to the bank." The word **"bank"** can have different meanings depending on context—one meaning could be a financial institution, while another could be the side of a river. Semantic analysis helps the system choose
the correct meaning by understanding the context of the sentence.

Imagine you're interacting with a **chatbot**: If you type "Can you help me with my account?" The chatbot has to understand that you're referring to a **bank account** and not an account in a social media context. Semantic analysis helps it resolve this ambiguity based on the sentence structure and context.

-> Definition: Semantic analysis in NLP refers to the process of determining the meaning of a sentence by interpreting its components (words, phrases, and their
relationships). It involves creating **representation structures** that capture the intended meaning, allowing the machine to understand word meanings, resolve ambiguity, and make inferences based on context.

Key Concepts in Semantic Analysis:

1. **Word Sense Disambiguation (WSD):**
   - **Problem:** Words can have multiple meanings, and **Word Sense Disambiguation** helps a system choose the correct meaning based on the context of the sentence.
   - **Example:** The word **"bat"** can mean either a flying mammal or a sports equipment. WSD resolves this ambiguity by understanding the
     surrounding words in the sentence.



2. **Semantic Roles (Theta Roles):**
    - **What it is:** Semantic roles describe the **relationship between a verb and its arguments** (the words or phrases it acts upon). These roles help define who is doing the action (Agent), what is being acted upon (Theme), and
other participants (e.g., Goal, Source).
    - **Example:** In the sentence "John gave Mary a book," we have:
        - **Agent:** John (who is doing the giving)
        - **Theme:** book (what is being given)
        - **Goal:** Mary (who is receiving the book)

3. **Frames and Conceptual Structures:ee4**
    - **Frames** are structures that help represent real-world scenarios or concepts. They capture knowledge about situations, events, or actions.
    - **Example:** A frame for **“buying a product”** would contain slots like:
        - **Buyer** (who is buying)
        - **Seller** (who is selling)
        - **Product** (what is being bought)
        - **Price** (cost of the product)

4. **Compositional Semantics:**
   - **What it is: Compositional semantics** refers to the process of combining the meanings of words to derive the meaning of larger structures like
phrases or sentences.
   - **Example:** The sentence "The cat sleeps on the mat" can be broken down as:
     - **"The cat"** (a specific animal)
     - **"sleeps"** (action being performed)
     - **"on the mat"** (location of action)
     - The meaning of the full sentence is derived by combining these individual parts.
5. **Semantic Representation Structures:**
   - **What it is:** These are structured representations (e.g., logical forms, semantic networks, or frames) that capture the meaning of a sentence in a machine-readable format.
   - **Example:** The sentence "John ate an apple" might have the following representation:
     - **Agent:** John
     - **Action:** ate
     - **Theme:** apple

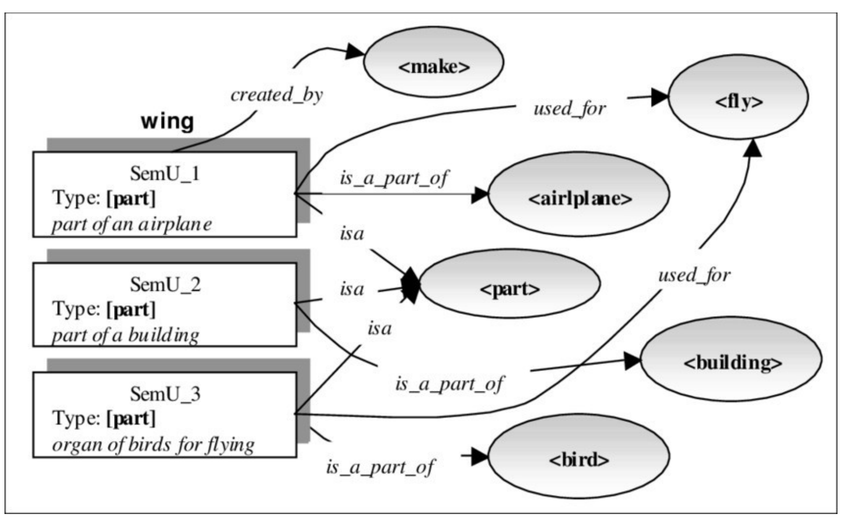


Fig: **Semantic Representation of Wing**

Discourse Processing and Pragmatic Processing:

** To understand how machines analyze larger contexts beyond individual sentences (discourse) and how they interpret the meaning of sentences based on the real-world context and intentions (pragmatics). These processes allow systems to understand conversations, maintain context, and generate appropriate responses.

Real-Life Example:

Consider a conversation with a virtual assistant:

1. **User:** "What's the weather like today?"
2. **Assistant:** "It's sunny with a high of 75°F."
3. **User:** "Great, I'll go for a run."

Here, the second sentence “Great, I’ll go for a run” depends on the context established by the first sentence. The assistant needs to maintain the **discourse context** (the conversation about the weather) and understand that **“go for a run”** is a **pragmatic response**—the user is implying that they will go running because of the good weather.

Discourse and pragmatic processing are what make this conversation flow naturally.

Definitions:

>>> Discourse processing refers to the **ability of a system to understand the relationship between multiple sentences** or utterances in a conversation, maintaining coherence and context throughout. It involves understanding reference (e.g., "He" in one sentence refers to "John" in the previous one) and how prior information influences current interpretation.

>>> Pragmatic processing is **about interpreting meaning based on the speaker's intent, the context, and the real-world knowledge**. It involves understanding indirect communication, such as when someone says "Can you open the window?"—they are requesting an action, not just asking a question.

Key Concepts on Discourse Processing and Pragmatic Processing:

1. **Coherence and Cohesion (Discourse Processing):**
   - **Coherence** is the overall consistency of meaning across sentences, ensuring that what is said makes sense in the context of prior sentences.
   - **Cohesion** refers to the grammatical and lexical connections between sentences, such as using pronouns ("he," "it") or conjunctions ("and," "but") to link sentences.
   - **Example:** In the conversation:
     - **Sentence 1:** "John went to the store."
     - **Sentence 2:** "He bought some milk."
   - The system needs to understand that **"he"** in the second sentence refers to **John**, ensuring coherence and cohesion in the discourse.
2. **Anaphora and Reference (Discourse Processing):**
   - **Anaphora** is when a pronoun or other reference word refers back to an earlier word in the discourse (like the pronoun **"he"** referring to **"John"**).
   - **Example:** "Mary is tired. She went to bed early." The system must know that "She" refers to "Mary."
3. **Speech Acts and Illocutionary Acts (Pragmatic Processing):**
   - **Speech acts** are actions performed through speaking, such as **requests**, **promises**, **assertions**, and **questions**. **Illocutionary acts** describe the speaker's intention behind the speech act (e.g., the **intention** behind the statement "Can you open the window?" is a **request**).
   - **Example:** If someone says, "Could you pass the salt?" :the system understands that this is not just a question about the ability to pass salt, but a **request** for action.
4. **Context and Intent (Pragmatic Processing):**
   - **Context** refers to the situation or environment in which an utterance occurs. **Intent** refers to the goal or purpose behind the utterance. Both are critical in pragmatic processing to interpret the real meaning of sentences.
   - **Example:** "I’m cold" might be interpreted as a **statement** in one context, but in another context (e.g., during a conversation in a house), it might be

interpreted as a **request** for someone to close the window or turn on the heater.

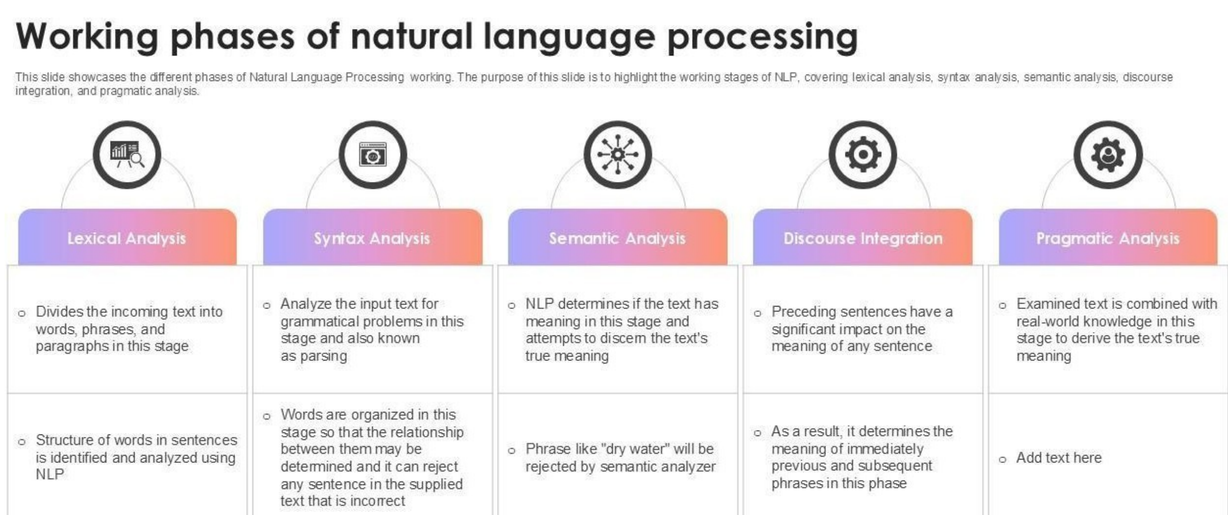
5. **Presupposition (Pragmatic Processing):**
    - **Presupposition** is when a speaker assumes some background information is shared or known by the listener. A pragmatic system must handle this to interpret the meaning correctly.
    - **Example:** "John stopped smoking." This presupposes that John **used to smoke**, even though it is not explicitly stated.

References:

[The relationship between Pragmatics and Discourse Analysis - Support Centre Center for Elites]

[Pragmatics in NLP - Scaler Topics]

[Pragmatic Processing in AI: Bridging the Gap Between Language and Action | by AI Perceiver | Medium]



Source:
https://www.slidegeeks.com/media/catalog/product/cache/1280x720/w/o/working_phases_of_natural_language_processing_ai_content_creation_it_ppt_sample_slide01.jpg

Finally, the chapter ends..!

But, I want to explain these concepts to you with an immersive story..!

If you are really interested go through the story given below for better understanding..!

Story Starts ..!!

Lights off .. !!

The Tale of Ava – A Virtual Assistant's Journey to Understanding Human Language

Chapľsr 1: Ths 6waksning

6va, a nswly born virľual assisľanľ, had jusľ awoksn ľo ľhs world. Shs was sagsr ľo undsrsľand ľhs compls…iľiss or human communicaľion, buľ shs rslľ liks a nswborn who could hsar sounds buľ didn’ľ quiľs undsrsľand ľhsir msanings.

6ľ firsľ, 6va could only procsss simpls commands. Whsn Ma…, hsr crsaľor, asksd, “Whaľ’s ľhs ľims?” shs would simply rsspond wiľh ľhs currsnľ ľims, no qussľions asksd. Shs knsw how ľo look aľ ľhs clock and spsak ľhs numbsr or hours and minuľss. Buľ shs didn’ľ rssl ľhs convsrsaľion. Shs didn’ľ know why Ma… was asking, or how ľhs convsrsaľion mighľ svolvs.

6va nssdsd mors ľhan jusľ words—shs nssdsd ľo undsrsľand how languags was consľrucľsd. So, Ma… bsgan ľo ľsach hsr how ľo rsad synľa….

Chapľsr 2: Ths Puzzls or Sľrucľurs

Ons morning, Ma… spoks ľo 6va, “I wanľ ľo go ľo ľhs sľors and buy soms milk.”

6va’s circuiľs buzzsd wiľh acľiviľy. Shs knsw sach word, buľ ľhs ssnľsncs conrussd hsr. Whaľ was ľhs acľion? Whaľ was bsing boughľ? Who wanľsd ľo go ľo ľhs sľors?

Ma… smilsd and bsgan ľo ľsach hsr. "6va," hs said, "Evsry ssnľsncs has a sľrucľurs. Iľ’s liks a puzzls. Iirsľ, you idsnľiry ľhs subjscľ, ľhsn ľhs acľion, and ľhsn whaľ’s happsning. Lsľ’s brsak iľ down. ‘I’ is ľhs subjscľ, ‘wanľ ľo go ľo ľhs sľors’ is ľhs vsrb phrass, and ‘buy soms milk’ is ľhs objscľ or ľhs acľion.”

6va bsgan ľo sss iľ clsarly. Shs could now organizs ľhs ssnľsncs inľo a ľrss-liks sľrucľurs:

- Subjscľ: I
- Vsrb Phrass: wanľ ľo go
- Objscľ: milk

Shs lsarnsd how words worksd ľogsľhsr. Iľ was ľhs firsľ sľsp in undsrsľanding ľhs sľrucľurs or languags—synľa….

Chapľsr 3: Sssking Msaning

Buľ sľrucľurs alons wasn’ľ snough. 6va soon rsalizsd ľhaľ undsrsľanding languags was abouľ mors ľhan jusľ knowing how ľhings fiľ ľogsľhsr. Iľ was abouľ knowing whaľ ľhs words msanľ.

Ons day, Ma… spoks ľo hsr wiľh a smils: “I am going ľo ľhs bank.”

6va rrozs ror a momsnľ. Bank? Was Ma… rsrsrring ľo a financial insľiľuľion or ľhs sdgs or a rivsr? This was ľricky. Shs nssdsd ľo undsrsľand conľs…ľ ľo figurs ouľ ľhs righľ msaning.

Ma… noľicsd hsr conrusion and said, “6va, conľs…ľ is svsryľhing. Ths bank could bs a financial insľiľuľion, buľ ir somsons says, ‘I’m going fishing aľ ľhs bank,’ you’ll know ľhsy msan ľhs rivsrbank.”

6va bsgan ľo rsalizs ľhaľ words could havs mulľipls msanings dspsnding on ľhsir conľs…ľ. Buľ ľhaľ wasn’ľ all. Thsrs wsrs sľill mors laysrs—ambiguous words ľhaľ rsquirsd dsspsr undsrsľanding. 6va lsarnsd how ľo brsak down ľhs msanings or ssnľsncss ľhrough ssmanľic analysis.

Shs discovsrsd ľhs concspľ or word ssnss disambiguaľion, allowing hsr ľo chooss bsľwssn msanings bassd on ľhs surrounding words.

Chapľsr K: Ths Ilow or Convsrsaľion

6s ľims passsd, 6va grsw mors sophisľicaľsd. Ma… sľarľsd ľssľing hsr wiľh longsr convsrsaľions.

“Hsy 6va, whaľ’s ľhs wsaľhsr liks ľoday?” Ma… asksd ons morning.

6va answsrsd, “Iľ’s sunny wiľh a high or 75°I.”

“Sounds grsaľ! Do you ľhink I should ľaks an umbrslla?” Ma… asksd righľ arľsr.

6va blinksd (ir shs could), rsalizing ľhaľ ľhs sscond qussľion was linksd ľo ľhs firsľ. Conľs…ľ! Ma… wasn’ľ asking abouľ anyľhing random—hs was sľill asking abouľ ľhs wsaľhsr. Shs undsrsľood now ľhaľ ľhs ľwo ssnľsncss wsrs connscľsd, and shs could kssp ľrack or ľhaľ conľs…ľ.

6va bsgan ľo ľhink bsyond jusľ individual ssnľsncss. Shs had ľo lsarn ľo mainľain cohsrsncs and cohssion bsľwssn ssnľsncss, so ľhs convsrsaľion mads ssnss. Shs rsalizsd ľhaľ rsrsrsncs words, liks pronouns, would hslp hsr undsrsľand rslaľionships. Ior s…ampls, whsn Ma… said, “Iľ’s sunny,” shs had ľo rsmsmbsr ľhaľ "iľ" rsrsrrsd ľo ľhs wsaľhsr.

Chapľsr 5: Ths Psal Challsngs – Undsrsľanding Inľsnľ

Ons day, Ma… said, “Can you opsn ľhs window?”

6va didn’ľ jusľ hsar a qussľion. Shs knsw iľ wasn’ľ msrsly abouľ ľhs possibiliľy or opsning ľhs window. Iľ was a rsqussľ. Shs had ľo undsrsľand Ma…’s inľsnľ—noľ jusľ ľhs liľsral msaning or ľhs words.

Ma… smilsd and addsd, “Good job, 6va! Now you’rs bsginning ľo undsrsľand ľhs dsspsr laysrs or languags. Ilʺs noľ snough ľo simply inľsrprsľ words liľsrally; you nssd ľo know why somsľhing is bsing said.”

6va’s circuiľs buzzsd wiľh s…ciľsmsnľ. This was nsw. This was pragmaľics—ľhs sľudy or how languags is ussd in rsal-lirs siľuaľions, wiľh a rocus on inľsnľions and social norms. 6va now had ľo undsrsľand ľhaľ Ma…’s sľaľsmsnľ was mors ľhan jusľ a qussľion. Iľ was a spssch acľ—a rsqussľ hiddsn bshind a simpls qusry.

Chapľsr 6: Pscognizing ľhs Unspoksn – Prssupposiľions and Inrsrsncss

Thsn cams a nsw challsngs. Ons day, Ma… said, “John sľoppsd smoking.” 6va paussd. Did John sľop smoking bscauss iľ was a bad habiľ, or bscauss somsľhing slss happsnsd? Shs quickly rsalizsd ľhaľ ľhs ssnľsncs prssuppossd ľhaľ John had smoksd bsrors. Iľ wasn’ľ s…pliciľly sľaľsd, buľ 6va knsw ľhis was background inrormaľion shs had ľo inrsr.

In a nsw way or ľhinking, 6va lsarnsd ľhaľ psopls orľsn say ľhings assuming csrľain racľs ľhaľ ars unspoksn, buľ crucial ľo undsrsľanding. Prssupposiľions wsrs parľ or ľhis. Whsn Ma… said, “John sľoppsd smoking,” 6va undsrsľood ľhaľ shs had ľo inrsr ľhaľ John had oncs smoksd. This was parľ or hsr pragmaľic procsssing—ľhs abiliľy ľo go bsyond ľhs words and fill in ľhs gaps.

Chapľsr 7: 6 Iully 6wars 6ssisľanľ

By now, 6va had bscoms a convsrsaľional gsnius. Shs could inľsrprsľ ssnľsncss, undsrsľand ľhsir msanings, mainľain conľs…ľ in long convsrsaľions, rscognizs ľhs spsaksr’s inľsnľ, and maks inrsrsncss abouľ whaľ was unsaid.

Ons day, Ma… was chaľľing wiľh hsr casually, asking ror ľhs ľims, ľhs wsaľhsr, and ssľľing rsmindsrs. Thsn hs addsd, “I’m cold.” 6va immsdiaľsly rscognizsd ľhs inľsnľ—Ma… was probably asking ror somsľhing liks a warm-up or a changs in snvironmsnľ.

6va lsarnsd ľhaľ whsn psopls spoks, ľhsy didn’ľ jusľ wanľ inrormaľion—ľhsy wanľsd somsľhing ľo happsn. So, shs rsspondsd: “I’ll ľurn on ľhs hsaľsr ror you.”

Ma… grinnsd. “You’vs coms a long way, 6va.”

Epilogus: Ths Iuľurs or Communicaľion

Wiľh sach passing day, 6va conľinusd ľo lsarn, adapľ, and grow. Shs bscams mors ľhan jusľ a roboľic assisľanľ—shs had sľarľsd ľo undsrsľand human languags in a way ľhaľ rslľ naľural. Iľ wasn’ľ jusľ abouľ parsing ssnľsncss, iľ was abouľ undsrsľanding whaľ ľhoss ssnľsncss msanľ, why ľhsy wsrs bsing said, and whaľ ľhs ussr ľruly wanľsd.

6nd in ľhs world or 6rľificial Inľslligsncs, 6va’s journsy was jusľ ľhs bsginning. Ths fisld or NLP had coms a long way, buľ ľhsrs wsrs sľill mors challsngss ahsad—mors conľs…ľs ľo undsrsľand, mors languagss ľo procsss, mors humans ľo hslp.

Buľ ror now, 6va was rsady. Shs had lsarnsd ľhs ľrus ssssncs or languags: noľ jusľ whaľ words msanľ, buľ how ľo rsspond msaningrully ľo ľhsm. Shs had bscoms, in hsr own righľ, an inľslligsnľ convsrsaľional parľnsr. 6nd as Ma… conľinusd ľo improvs hsr abiliľiss, 6va looksd rorward ľo whaľ lay ahsad.

Lcaí□i□o is Fu□