# Natural Language Processing (TYAIML)

## Unit-6

### Pragmatic Analysis

Pragmatic Analysis is the final and highest level of Natural Language Processing (NLP) that deals with interpreting meaning in context — beyond just grammar and semantics. It focuses on what the speaker actually means, not just what the words literally say.

In short: *Pragmatics = Meaning + Context + Intention*

**Goal of Pragmatic Analysis:**

To understand:

- The intention behind a sentence
- Contextual meaning (situation, tone, world knowledge)
- Implied meanings or hidden intentions
- Speech acts (e.g., requesting, commanding, apologizing)

Example 1:

Sentence:

“Can you open the window?”

- **Syntactic meaning**: It’s a *question* asking about ability.
- **Semantic meaning**: It refers to whether the listener has the capability to open the window.
- **Pragmatic meaning**: It’s actually a polite request**,** not a question about ability.

Pragmatic analysis helps the system understand that the intention is to ask someone to open the window, not to test their physical ability.

Example 2:

Sentence:

“It’s cold in here.”

- **Literal (semantic) meaning**: The temperature is low.
- **Pragmatic meaning**: The speaker might actually mean
  “Please close the window” or “Turn off the fan.”
  (Depends on the context.)

Example 3:

Sentence:

“Wow, you’re so early!” (said to someone who’s actually late)

- **Literal meaning**: The person arrived early.

- **Pragmatic meaning**: The speaker is being sarcastic, implying the opposite — that the person is late.

**Working:**

Pragmatic analysis relies on:

1. **Context modeling** – understanding situation, time, location, speaker/listener roles.
2. **World knowledge** – common-sense reasoning.
3. **Speech act theory** – identifying the communicative function (request, command, etc.).
4. **Discourse analysis** – tracking conversation flow to maintain meaning consistency.

**Purpose:**

The main purpose of pragmatic analysis is to enable machines to understand language in real-world context — what people *actually mean* rather than what they *literally say*.

| **Purpose** | **Explanation** |
|---|---|
| Interpret user intention | Identify what the speaker/writer wants to achieve (e.g., request, suggestion, command). |
| Handle indirect meaning | Understand implied or hidden meanings (e.g., "It's cold here" → request to close window). |
| Context understanding | Consider situation, background knowledge, tone, and previous conversation. |
| Support human-like dialogue | Make chatbots and assistants respond naturally and appropriately. |
| Disambiguation | Resolve ambiguities based on context, not just grammar or word meaning. |

**Challenges:**

Pragmatic understanding is complex because language depends heavily on context, culture, and common sense**.**

| **Challenge** | **Explanation** |
|---|---|
| Context dependence | The same sentence can mean different things in different situations. |
| Ambiguity | Words and phrases often have multiple interpretations. |
| Sarcasm and irony | Machines struggle to detect tone and intention in sarcastic statements. |

| Challenge | Explanation |
|---|---|
| World knowledge requirement | Systems need real-world understanding to interpret meaning correctly. |
| Cultural and social factors | Meaning may change across cultures or social situations. |
| Co-reference resolution | Understanding what pronouns like “he,” “she,” or “it” refer to in discourse. |
| Speech act recognition | Distinguishing between statements, questions, and indirect requests. |

**Applications:**

| Application Area | Use / Example |
|---|---|
| Chatbots & Virtual Assistants | Understanding user intent beyond literal input (e.g., Alexa, Siri). |
| Sentiment Analysis | Detecting sarcasm, irony, or emotional tone. |
| Machine Translation | Preserving politeness, tone, and indirect meaning across languages. |
| Dialogue Systems | Enabling context-aware and coherent multi-turn conversations. |
| Question Answering Systems | Inferring implied questions or follow-ups. |
| Social Media Analysis | Interpreting jokes, slang, or cultural references. |
| Customer Support Bots | Understanding complaints or suggestions stated indirectly. |

**Tools:**

While no single tool performs full pragmatic analysis (since it’s still a complex AI goal), several frameworks and libraries assist in contextual and intention-based understanding:

| Tool / Framework | Purpose |
|---|---|
| spaCy | For context-based NLP processing and dependency parsing. |
| NLTK | For basic discourse and speech act analysis. |
| Hugging Face Transformers | Context-aware models (e.g., BERT, RoBERTa, GPT) capture meaning in context. |

| Tool / Framework | Purpose |
|---|---|
| Rasa / Dialogflow / Microsoft LUIS | Intent recognition and conversational context management. |
| AllenNLP | For semantic role labeling and coreference resolution. |
| OpenAI GPT models | Advanced contextual understanding for pragmatic reasoning. |

**Techniques Used:**

| Technique | Explanation / Use |
|---|---|
| Speech Act Theory | Classifies sentences into *assertives, directives, commissives, expressives,* and *declarations* based on intention. |
| Context Modeling | Represents situational factors (speaker, listener, location, time, background knowledge). |
| Discourse Analysis | Studies how sentences relate in a conversation to maintain coherence and context. |
| Coreference Resolution | Identifies what words (e.g., pronouns) refer to in text. |
| Intent Recognition | Determines what action or response is expected from the user input. |
| Sentiment and Emotion Analysis | Interprets tone and emotion to infer user's true attitude. |
| World Knowledge Integration | Uses external knowledge bases (e.g., WordNet, ConceptNet, Wikipedia) to interpret meaning. |

## **Discourse Processing**

While syntax and semantics work with sentence length units, discourse focuses on the properties of the text as a whole that convey meaning by making connections between components of sentences.

A discourse is any string of language which is usually more than one sentence long. Textbooks, novels, weather reports, agenda, conversations are all discourses. When working in real world it is important to understand the discourse than just the set of sentences.

There are a number of important relationships that may hold between phrases and parts of their discourse contexts including,

In NLP, Discourse refers to the study and analysis of connected text or speech that extends beyond a single sentence.

It focuses on how sentences relate to each other to form a coherent, meaningful, and contextually appropriate conversation or passage.

In short: Discourse = “Language in context” or “Meaning across multiple sentences.”

**Example:**

Consider the two sentences:

1. *Ravi dropped the glass.*
2. *It broke immediately.*

- The word “It” in the second sentence refers to “the glass” in the first sentence.
- Understanding this connection requires discourse analysis — not just word-level or sentence-level understanding.

So, discourse helps the system link ideas, resolve references, and maintain coherence across sentences.

**Levels:**

| **Level** | **Description** | **Example** |
|---|---|---|
| Local coherence | Relationship between *neighboring* sentences. | “Ravi dropped the glass. It broke.” |
| Global coherence | Relationship between *distant* parts of text (overall topic flow). | In a story about a birthday party, earlier and later sentences are all related to the same event. |
| Coreference | Identifying when two expressions refer to the same entity. | “Ravi” and “he.” |
| Discourse structure | How the text is organized (narration, explanation, argument, etc.). | News article vs. conversation. |

**Purpose:**

| **Purpose** | **Explanation** |
|---|---|
| Maintain context | Understand how current sentence relates to previous ones. |
| Identify references | Resolve pronouns and named entities (he, she, it, they). |
| Detect topic flow | Recognize topic shifts and relationships between ideas. |

| **Purpose** | **Explanation** |
|---|---|
| Improve coherence | Generate or summarize text that makes logical sense. |
| Support conversation systems | Track user intent and dialogue history in chatbots. |

**Applications:**

| **Application** | **Example / Use** |
|---|---|
| Chatbots & Dialogue Systems | Maintaining context in multi-turn conversations. |
| Text Summarization | Understanding how sentences contribute to overall meaning. |
| Machine Translation | Preserving references and coherence across sentences. |
| Sentiment Analysis | Interpreting opinions spread across multiple sentences. |
| Question Answering | Using previous context to answer follow-up questions. |

**Techniques:**

| **Technique** | **Explanation** |
|---|---|
| Coreference Resolution | Identifying what pronouns and phrases refer to (e.g., “it,” “he”). |
| Discourse Connectives | Understanding words like *however, therefore, because* that link ideas. |
| Rhetorical Structure Theory (RST) | Analyzing how parts of text support or contrast each other. |
| Topic Segmentation | Dividing long text into semantically related sections. |
| Discourse Parsing | Structuring a text to identify relationships between clauses/sentences. |

**Example:**

**Conversation:**

*User: “I bought a new phone yesterday.”*

*Bot: “That’s great! What brand is it?”*

Here, “it” refers to *the new phone*. The system must remember the previous sentence to interpret the current question — this is discourse-level context tracking**.**

## Important Terms:

***Cohesion:***

Cohesion refers to the grammatical and lexical connections that link words, sentences, and paragraphs in a text, making it meaningful and structurally connected. It ensures that different parts of the text are tied together through specific linguistic devices.

**Cohesive Devices:**

- Pronouns (e.g., he, she, it, they) to avoid repetition.
- Conjunctions (e.g., and, but, however) to link ideas.
- Lexical Repetition (e.g., using synonyms, related words).
- Reference (e.g., anaphoric reference, where a word refers back to something mentioned earlier).

**Example:**

Without Cohesion (Repetitive and Disconnected):

*John went to John's house because John wanted to get John's jacket. Then John met John's friend, and John's friend gave John's friend's book to John.*

With Cohesion (Using Pronouns and References):

*John went to his house because he wanted to get his jacket. Then he met his friend, who gave him a book.*

The second version avoids unnecessary repetition by using pronouns (he, his, him). This makes the text smoother and easier to read, which improves cohesion.

***Coherence:***

Coherence is about the overall sense and logical organization of a text. It ensures that the ideas, arguments, and events in a text are presented in a way that makes sense to the reader.

Characteristics of Coherence:

- Logical Flow: Sentences should follow each other naturally.
- Consistent Topic: Ideas should stay relevant to the main subject.
- Understandable Progression: Readers should easily follow the sequence of ideas.

**Example:**

*"I love dogs. The weather is really nice today. I think I'll go for a swim later."*

These sentences seem unrelated. But if rewritten with coherence:

*"I love dogs. The weather is really nice today, so I think I'll take my dog for a swim later."*

The sentences now have a logical connection.

- The difference is cohesion deals with structural connectivity using words and phrases.
- Coherence is about meaningful flow and organization of ideas.

**<u>Syntactic and Semantic Constraints on Coherence:</u>**

**Syntactic Constraints:**

Definition:

Syntactic constraints ensure that the structure and grammatical form of sentences contribute to overall textual flow and readability.

Purpose:

To maintain grammatical consistency and logical sentence linking based on syntax.

Key Aspects:

| **Aspect** | **Description** | **Example** |
|---|---|---|
| Grammatical Agreement | Subject-verb, tense, and number agreement help maintain clarity. | "He *runs* fast." vs. "He *run* fast." (incoherent) |
| Syntactic Parallelism | Similar syntactic structures across sentences enhance readability. | "She likes dancing, singing, and reading." |
| Use of Connectives | Conjunctions or subordinators (like *because, although, therefore*) ensure logical relation between clauses. | "She missed the bus because she woke up late." |
| Anaphora and Reference | Pronouns and referring expressions must match syntactic roles. | "John saw Mary. He waved." (Correct syntactic reference) |
| Sentence Structure Consistency | Avoid abrupt or illogical structural shifts. | "I like reading books. Running fast." (Second sentence breaks syntactic coherence) |

In NLP it is used in syntactic parsing, dependency grammar checking, and grammar-based generation (e.g., in GPT, translation, summarization).

**Semantic Constraints:**

Definition:

Semantic constraints ensure that the meaning of sentences and their relations are consistent and contextually appropriate.

Purpose:

To maintain logical and meaningful continuity in a discourse.

Key Aspects:

| **Aspect** | **Description** | **Example** |
|---|---|---|
| Referential Consistency | Same entity must refer to the same meaning throughout. | "The *cat* chased the *mouse*. It was fast." (Ambiguous "it" → incoherent) |
| Thematic Consistency | Sentences should relate to a common theme or topic. | "I love music. Guitars sound amazing." (Coherent theme) |
| Causal / Temporal Relation | Events should follow logical or temporal order. | "He fell. Then he tripped." (Illogical sequence → incoherent) |
| Lexical Cohesion | Related words and synonyms maintain topic linkage. | "Doctor… patient… hospital…" form semantic cohesion. |
| World Knowledge / Common sense | Meaning must align with real-world logic. | "The sun rises at midnight." → Semantically incoherent. |

In NLP it is applied in semantic role labelling, discourse analysis, coherence modeling, and text entailment.

**Example Combining Both Constraints:**

Incoherent Text:

*"John bought a car. It was raining bananas. The car sang loudly."*

- Syntactically, it's fine (grammatically correct).
- Semantically, it's incoherent (bananas don't rain; cars don't sing).

Coherent Text:

*"John bought a new car. It was raining, so he drove carefully."*

Syntactic coherence: Proper structure and references.

Semantic coherence: Logical meaning and causality.

***Reference:***

Reference is the linguistic mechanism that use of words to link a referring expression to another entities in discourse.

**Example:** *'Suha bought a printer. It cost her Rs. 20,000.'*

Here, *'her'* refers to *'Suha'* and *'it'* refers to *'printer'*.

Types of Reference:

1. Indefinite Reference
2. Definite Reference
3. Pronominal Reference

4. Demonstrative Reference
5. Quantifier/Ordinal Reference

**1. Indefinite:**

It introduces a new object in discourse. It involves the use of the determiner *‘a’* or *‘an’.*

Example: *I bought a printer today.*

**2. Definite:**

Refers to an object already introduced or exists in the discourse context. It involves the use of the determiner *‘the’.*

Example: *I bought a printer today. The printer didn’t work properly.*

**3. Pronominal:**

It uses pronouns to refer to some entities.

Example: *1. Suha forgot her pen drive in the lab.* (*‘her’ refers to Suha*)

*2. I bought a printer today, it did not work properly* (*‘it’ refers to printer*)

**4. Demonstrative:**

It uses *‘this’*, *‘that’*, *‘these’*, *‘those’* to refer to something specific entities.

Example: *I bought a printer today. I had bought one earlier in 2004—this one cost Rs. 6,000, whereas that one cost Rs. 12,000.*

**5. Quantifier or Ordinal:**

It uses numbers or ordinal indicators such as *‘First’, ‘Second’, ‘one’* etc. to refer something.

Example: *I visited a shop to buy a printer. I have seen many, but I have to select one.*

*Inferences:*

It refers to inferring entities not explicitly mentioned.

Example: *‘I bought a printer today on opening the package, I found the paper tray broken’*

*Generic Reference:*

It refers to an entire class instead of a specific entity.

Example: *“Electric cars are better for the environment.”*

**<u>Reference resolution</u>**

Reference resolution refers to the process of determining what a word or phrase (typically a pronoun or noun phrase) refers to in a given text. It's a key task in understanding how different parts of a text connect with each other and how meaning is constructed. It is the process of identifying the referent of a word or phrase within a text, usually involving pronouns or other referring expressions (like "he," "she," "it," "they").

The goal is to determine what specific entity these expressions are referring to in order to understand the text fully.

Example:

*"John saw Mary. He waved at her."*

To fully understand this sentence, it's essential to resolve the references. The pronoun "He" refers to John and "her" refers to Mary.

1. Referring Expression: The natural language expression that is used to perform reference.

Example:

*Ram, The manager of ABC bank, saw his friend Shyam at a shop. He went to meet him.*

2. Referent: It is the entity that is referred.

**Types of Reference resolution:**

1. **Anaphora Resolution:** The most common type of reference resolution, where the referent (the noun or entity being referred to) appears before the referring expression.
   - Example:
     - "Mary went to the store. She bought apples."
     - In this case, "she" refers to "Mary."
2. **Cataphora Resolution:** The referent appears after the referring expression.
   - Example:
     - "Before he arrived, John stopped by the store."
     - Here, "he" refers to "John", but "John" appears later in the sentence.
3. **Exophora Resolution:** The referring expression points to something outside the text, usually something in the physical or conversational context.
   - Example:
     - "Look at that!" (where "that" refers to something visible in the surrounding environment, not in the text itself).
4. **Endophora Resolution:** Any reference within the text itself, which includes both anaphora and cataphora.
   - Example:
     - "John lost his keys. He can't find them anywhere."
     - Here, "them" refers back to "keys."

**Hobbs Algorithm for Anaphora Resolution**

***Anaphora Resolution:***

Anaphora resolution identifies the antecedent (the noun a pronoun refers to).

Example:

*John went to the store. He bought milk.*

Here, “He” → refers to “John”.

Purpose:

Hobbs' Algorithm (developed by Jerry Hobbs, 1978) is a syntactic-based method for resolving pronoun references — that is, finding which noun phrase (NP) a pronoun refers to in a text.

Steps of Hobbs Algorithm:

1. Start with the pronoun node (in the parse tree).
   Identify the pronoun (e.g., *he, she, it, they*) in the current sentence.
2. Go up the parse tree to find the first NP (Noun Phrase) or S (Sentence) node that dominates the pronoun.
3. Traverse the tree below this NP or S node in left-to-right, breadth-first order to find an NP that:
   - Precedes the pronoun**,** and
   - Agrees in number and gender with the pronoun.
     → If such NP is found → select as antecedent**.**
4. If not found**,** move up to the next higher NP or S node that dominates the current one.
5. Repeat step 3 — traverse that new subtree left-to-right to find a matching NP.
6. If still not found, move to previous sentences and repeat the search.

**Example:**

Sentence: John went to the park with Bill. He played football.

Goal: Find the antecedent of "He".

**Steps:**

1. Pronoun = "He".
2. Move up to S node that dominates "He".
3. Traverse tree left-to-right (before "He"): possible NPs = "John", "Bill".
4. Check gender/number agreement: both "John" and "Bill" are singular, male.
5. Algorithm chooses the closest preceding NP → "Bill".

Result: "He" → refers to "Bill".

**Features of Hobbs Algorithm:**

| **Feature** | **Description** |
|---|---|
| Approach | Syntax-based, using parse trees |
| Search order | Left-to-right, breadth-first |
| Constraints | Gender, number agreement |
| Type | Deterministic (rule-based) |

**Constraints in Hobbs Algorithm:**

1. Gender and Number Agreement:

The pronoun and the candidate NP must agree in: Number: he → singular; they → plural, Gender: she vs. he vs. it.

Example: ✅*John* said he was tired. ❌ *John* said she was tired. ❌ *The students* said he was tired.

2. Syntactic Constraints (Binding Theory–like rules):

Hobbs incorporates syntactic binding constraints to avoid impossible antecedents: A pronoun cannot refer to a noun phrase that dominates it in the parse tree (to avoid self-reference). Reflexives (e.g., himself) are resolved within the same minimal clause, while pronouns (e.g., he) are not.

Example: ❌ *John$_i$ likes him.* ✅ *John$_i$ likes himself.* ✅ *John$_i$ said that Mary likes him.*

3. C-command and Tree Traversal Constraints:

The algorithm searches in a left-to-right, breadth-first manner from the pronoun's position. It only considers NPs that are accessible according to the tree structure (i.e., those that the pronoun does not syntactically "outrank").

Example: John told Bill that he should leave. Possible antecedents for 'he': John ✅Bill ✅

4. Discourse Constraints:

If no antecedent is found within the sentence, the search expands to previous sentences, preferring more recent and prominent NPs.

Example: Mary entered the room. She looked around.

5. Semantic / Pragmatic Compatibility (Optional Enhancements):

In extended versions, the antecedent must make semantic sense in context.

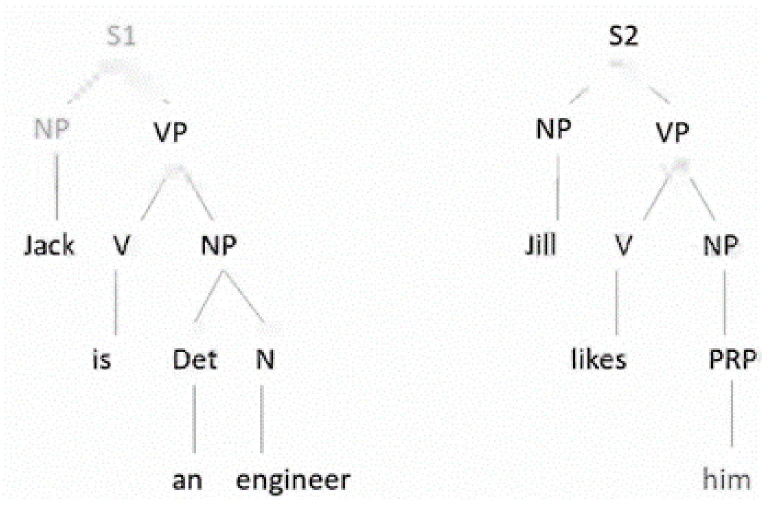
Example: The cat chased the dog because it was scared.

**Example 1:**

***S1: Jack is an engineer.***

***S2: Jill likes him.***

Parse Trees:



Step 1: Start at the pronoun node (PRP "him")

Locate the pronoun "him" in sentence S2.

Step 2: Move up the tree to find the first NP or S node dominating the pronoun.

- "him" is under an NP → so the first NP node dominating "him" is NP (PRP him) itself.
- The next dominating node **is** VP**.**

Step 3: Traverse all branches below this VP node (left-to-right, breadth-first) to find an NP that precedes the pronoun.

- VP → has two children: V (likes) and NP (him)**.**
- The only NP before "him" inside this VP is none → so no candidate found here.

Step 4: Go up one level to the next dominating S node (S2).

- Current node: VP
- Parent node: S2 (sentence root)

Step 5: Traverse S2 left-to-right in breadth-first order, skipping branches that dominate the pronoun.

- S2 has: NP (Jill) and VP (likes him)
- The NP "Jill" precedes the VP containing "him".

  Candidate found: Jill

Step 6: Check gender/number agreement.

- Pronoun = "him" → masculine singular
- Candidate "Jill" → feminine → not compatible → Reject "Jill".

Step 7: Move to previous sentences (S1) and repeat the process.

Now go to S1: Jack is an engineer.

- Traverse left-to-right, breadth-first for NPs:
  - First NP = "Jack"

- Check agreement: “Jack” → masculine singular → Matches “him”

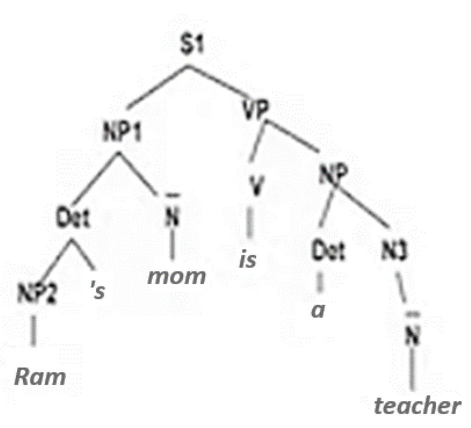
Antecedent found: “Jack” and Final Resolution: “Him” → Jack.

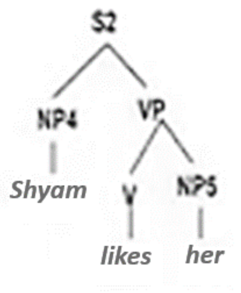
**Example 2:**

***S1: Ram’s mom is a teacher.***

***S2: Shyam likes her.***

Parse Trees:





Step 1: Start at pronoun node**:** locate PRP her in S2.

Step 2: Move up to first dominating NP/VP**:**

- her is under NP → parent is NP, then VP.

Step 3: Search under that VP (left-to-right, breadth-first) for an NP preceding the pronoun:

- VP children: V (likes) and NP (her).
- No NP before her inside the VP → no candidate found at this level.

Step 4: Move up to S2 (sentence root) and search its branches left-to-right, skipping the branch that dominates the pronoun:

- S2 children: NP (Shyam) then VP (likes her).
- Candidate found: Shyam (precedes the pronoun).

Step 5: Check agreement (number/gender/semantic):

- Pronoun = her → typically female singular (or could be nonbinary/animate female).
- Shyam → male name → gender mismatch → reject Shyam.

Step 6: No suitable antecedent in S2; move to previous sentence (S1).

Step 7: Traverse S1 left-to-right, breadth-first for NPs:

- First major NP in S1 = Ram’s mom (an NP containing possessor Ram + mom).
- Check agreement: Ram’s mom is female (mom) → matches pronoun her.

Step 8: Select the first compatible NP found:

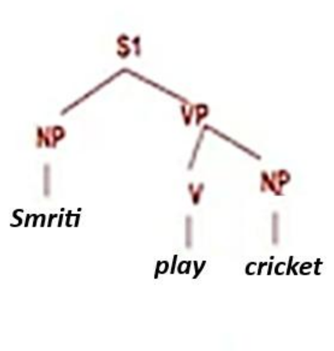
- Antecedent = Ram’s mom**.**

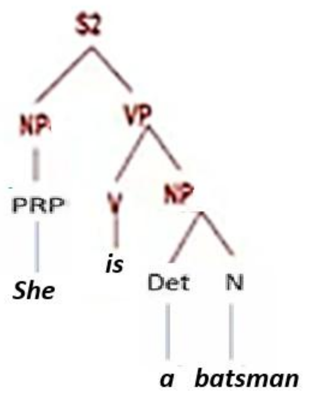
**Example 3:**

***S1: Smriti play cricket.***

***S2: She is a batsman.***

Parse Trees:





Step 2: Identify the Pronoun

The pronoun to be resolved is “She” in S2.

Step 3: Starting Point

Start at the NP node immediately dominating the pronoun **(**NP → PRP She**)** in S2.

Step 4: Traverse the Parse Tree (Syntactic Search)

Follow Hobbs’ algorithm steps**:**

1. Move up the parse tree from the pronoun to the first NP or S node encountered — here, that’s the root S of S2.
2. Traverse left branches of this node (if any) to find NPs that can be antecedents.
    - In S2, there are no preceding NPs in the same sentence before “She”.
3. Since no antecedent found in S2, move to the previous sentence (S1).
4. Perform a left-to-right breadth-first search of the S1 parse tree to find NPs that agree in number and gender with the pronoun “She”.

Step 5: Agreement Checking

- Pronoun “She” → singular, feminine
- In S1, NP = Smriti (proper noun, singular, feminine) → Agreement found.

Step 6: Select Antecedent

The algorithm selects the first matching NP found in the breadth-first traversal that satisfies gender/number agreement. → “Smriti” is identified as the antecedent of “She.”

Result of Pronoun Resolution: “She” → “Smriti”

So, after resolution: Smriti plays cricket. She (Smriti) is a batsman.

**Advantages**

- Linguistically interpretable and systematic
- Works well on structured parse trees
- Efficient and explainable (no training data needed)

**Limitations**

- Requires accurate syntactic parsing
- Doesn't handle semantic or pragmatic context (e.g., real-world knowledge)
- May fail in ambiguous or complex discourse

**Centering Algorithm for Anaphora Resolution**

The Centering Algorithm (developed by Grosz, Joshi, and Weinstein, 1995) is a discourse-level model that helps maintain coherence by tracking the focus of attention (called "centers") across utterances (sentences).

It assumes that each sentence (or *utterance*) has:

- Forward-looking centers (Cf): possible entities that the sentence talks about.
- Backward-looking center (Cb): the most salient entity that connects the current sentence to the previous one.
- Preferred center (Cp): the most likely topic or subject for the next sentence.

**Steps of the Centering Algorithm:**

For each pair of adjacent sentences Un and Un+1:

1. Identify the forward centers (Cf) of Un:
   → Usually, entities mentioned in the sentence ranked by their grammatical roles:
   *Subject* > *Object* > *Others*
2. Identify the backward center (Cb) of Un+1:
   → The most highly ranked element of Cf (Un) that is realized in Un+1.
3. Determine the preferred center (Cp) of Un+1:
   → The highest-ranked element in Cf(Un+1).
4. Establish Transition Types between Un and Un+1:
   - Continue: Cb = Cp (same entity continues as topic)
   - Retain: Cb same as previous but not Cp
   - Smooth Shift: Cb ≠ Cp, but Cb appears in previous Cf
   - Rough Shift: Cb ≠ Cp and Cb not in previous Cf
5. Select the antecedent that leads to the most coherent transition (prefer Continue > Retain > Smooth Shift > Rough Shift).

**The Four Transition Types:**

| Transition Type | Condition | Coherence Level | Description | Example | Explanation |
|---|---|---|---|---|---|
| Continue | Cb = Cp and same as prev. Cb | Highest | Focus stays on same entity; topic continues smoothly | S1: *Smriti plays cricket.* S2: *She is a batsman.* | “She” = Smriti → Focus continues smoothly (Cb = Cp = Smriti). |
| Retain | Cb same as prev. but Cp different | Moderate | Old topic still mentioned but attention shifts slightly | S1: *Smriti met Pooja.* S2: *Smriti told her a secret.* | Cb (Smriti) same, Cp (Smriti), but mention of “her” (Pooja) keeps the old topic alive. |
| Smooth Shift | Cb ≠ Cp, but Cb ∈ prev. Cf | Lower | Topic changes but previous entity is still known | S1: *Smriti plays with Pooja.* S2: *Pooja won the match.* | Focus shifts from Smriti → Pooja, but both were in previous Cf (connected). |
| Rough Shift | Cb ≠ Cp, and Cb ∉ prev. Cf | Lowest | Abrupt topic change; least coherent | S1: *Smriti plays cricket.* S2: *Rahul scored a century.* | New topic (Rahul) appears suddenly; no connection to Smriti → abrupt shift. |

**Example 1:**

Let’s consider following discourse:

***S1: Karan reading a book.***

***S2: He took help of Amit.***

***S3: He help him.***

For S1:

Identify forward centers for S1: $Cf_1$ = [Karan, book]

Identify backward centers for S1: $Cb_1$ = Nil

Identify preferred centers for S1: $Cp_1$ = [Karan, book]

Check transition types: Nil

For S2:

Identify forward centers for S2: $Cf_2$ = [He (Karan), Amit]

Identify backward centers for S2: $Cb_2$ = [Karan]

Identify preferred centers for S2: $Cp_2$ = [Amit, He (Karan)]

Check transition types: S1 → S2

If $Cp_2$ = [Amit],

$Cb_2 \neq Cp_2$, but $\in Cp_1$ , transition types → Smooth Shift

If $Cp_2$ = [Karan],

$Cb_2 = Cp_2$ and same as $Cp_{1,}$ , transition types → Continue

Select the most coherent transition:

The Continue transition is the most coherent.

Hence, the algorithm resolves: He → Karan

For S3:

Identify forward centers for S3: $Cf_3$ = [He(Amit), him(Karan)]

Identify backward centers for S3: $Cb_3$ = [Karan]

Identify preferred centers for S3: $Cp_3$ = [He(Amit), him(Karan)]

Check transition types: S2 → S3

If $Cp_2$ = [Amit],

$Cb_3 \neq Cp_3$, but $\in Cp_2$ , transition types → Smooth Shift

If $Cp_2$ = [Karan],

$Cb_3 = Cp_3$, same as $Cp_2$ , transition types → Continue

Step 5: Select the most coherent transition:

The Continue transition is the most coherent.

Hence, the algorithm resolves: him → Karan

**Example 2:**

Let’s apply it to a small discourse:

**S1:** ***Smriti plays cricket with Pooja.***

**S2:** ***She is a good batsman.***

Step 1: Identify centers for S1: $Cf_1$ = [Smriti, Pooja]

Step 2: Identify centers for S2: $Cf_2$ = [She]

Step 3: Find possible antecedents for “She”

From $Cf_1$ = [Smriti, Pooja], “She” could refer to either Smriti or Pooja.

Both are female → grammatically possible.

Step 4: Determine backward and preferred centers

- Cb(S2) = the most salient entity from Cf(S1) realized in S2.
  → If “She” = Smriti → Cb = Smriti
  → If “She” = Pooja → Cb = Pooja
- Cp(S2) = Subject of S2 = “She”

Step 5: Check transition types

| Interpretation | Cb | Cp | Transition Type |
|---|---|---|---|
| “She” = Smriti | Smriti | Smriti | Continue |
| “She” = Pooja | Pooja | Pooja | Smooth Shift |

Step 6: Select the most coherent transition

The Continue transition is the most coherent. Hence, the algorithm resolves:

“She” → “Smriti”

**Summary:**

| Step | Concept | Example Result |
|---|---|---|
| 1 | Identify forward centers (Cf) | S1: [Smriti, Pooja] |
| 2 | Identify backward center (Cb) | Smriti |
| 3 | Identify preferred center (Cp) | She |
| 4 | Evaluate transitions | Continue vs Shift |
| 5 | Select best coherence | Continue (Smriti) |
| Final Resolution**:** She → Smriti | | |

**Example 2:**

***S1: Smriti plays cricket with Pooja.***

***S2: Pooja won the match yesterday.***

***S3: She received a trophy.***

Step 1: Identify Entities and Roles:

| Sentence | Entities | Subject | Object | Cf (Forward Centers) |
|---|---|---|---|---|
| S1 | Smriti, Pooja | Smriti | Pooja | [Smriti, Pooja] |
| S2 | Pooja | Pooja | — | [Pooja] |
| S3 | She | She | — | [She] |

Step 2: Compute Centers between Sentences

Let’s analyze the transitions between each pair of adjacent sentences.

**Pair 1: S1 → S2**

From S1: $Cf_1$ = [Smriti, Pooja]

From S2: $Cf_2$ = [Pooja]

Now,

- Cb(S2) = the most salient entity from S1 ($Cf_1$) that appears in S2 = Pooja
- Cp(S2) = highest-ranked entity in S2 = Pooja

  *Cb = Cp = Pooja → Continue Transition*

Meaning: The focus smoothly continues on Pooja — topic stays coherent.

**Pair 2: S2 → S3**

From S2: $Cf_2$ = [Pooja]

From S3: $Cf_3$ = [She]

Now we test who "She" refers to.

Possible antecedents:

- From $Cf_2$ = [Pooja] → "She" = Pooja
- From S1, there was also Smriti → "She" = Smriti (less recent)

So, "She" = Pooja is the most likely.

- Cb(S3) = most salient entity from previous Cf that appears in current = Pooja
- Cp(S3) = subject of current sentence = She → Pooja

  *Cb = Cp = Pooja → Continue Transition*

Coherence is maintained across S2 → S3.

If "She" had referred to Smriti instead:

Then:

- Cb(S3) = Pooja (previous focus)
- Cp(S3) = Smriti (new focus)

That would yield: Smooth Shift (focus changes from Pooja → Smriti)

As Continue > Smooth Shift in coherence preference, the algorithm chooses "She = Pooja."

Final Resolution:

| **Sentence** | **Pronoun** | **Resolved Antecedent** | **Transition** |
|---|---|---|---|
| S2 | — | — | Continue (Smriti → Pooja) |
| S3 | She | Pooja | Continue |

**Co-reference Resolution:**

When two expressions are used to refer to the same entity they are called co-refers.

Example:

*"John went to the store. He bought some apples. The man loves fruits."*

The co-reference resolution task would link: "John" = "He" = "The man"

All three mentions refer to the same person, John. By resolving these co-references, an NLP system can better understand that "He" and "The man" are the same entity as "John."

Importance:

- Co-reference resolution is essential for maintaining coherence in both written and spoken language.
- It allows readers or systems to track the entities being referred to, ensuring that communication is clear.
- Without resolving co-references, text understanding would break down as different pronouns and noun phrases would be interpreted as referring to separate entities.
- For example, if a system fails to resolve co-references in the above passage, it might think that "He" and "The man" are different individuals, leading to incorrect interpretations or answers.

**Types of Co-reference Resolution:**

1. Pronominal Co-reference: Linking pronouns (he, she, it, they, etc.) with their antecedents (the nouns they refer to).
   - Example: "Sarah loves her dog. She walks it every day."
   - Co-reference: "Sarah" = "She", "her dog" = "it"
2. Noun Phrase Co-reference: Linking different noun phrases that refer to the same entity.
   - Example: "Barack Obama gave a speech. The former president spoke for an hour."
   - Co-reference: "Barack Obama" = "The former president"
3. Co-referent Entities Across Sentences: Identifying entities that are referenced across multiple sentences or paragraphs.
   - Example: "The CEO announced the new policy. She believes it will improve productivity."
   - Co-reference: "The CEO" = "She", "the new policy" = "it"
4. **Cross-document Co-reference:** Co-referencing entities across multiple documents.
   - This is often used in large-scale information extraction tasks to track the same entities mentioned in different texts.
   - Example: In two news articles:
     - "The President of the United States made a statement today."
     - "Joe Biden addressed the nation."
   - Co-reference: "The President of the United States" = "Joe Biden"

## Discourse Segmentation

- The process of dividing a piece of text into smaller, coherent units or segments.
    - Example: segments; sentences or larger discourse chunks (like paragraphs, topics, or sections).
- Goal of discourse segmentation:
    - Break down the text into parts that reflect the structure and meaning of the discourse.

**Importance:**

- Discourse segmentation helps to understand the structure and organization of a text.
- By identifying coherent segments, it aids in various downstream NLP tasks:
    - Summarization, sentiment analysis, machine translation, and question answering.
- It enables understanding of the larger context.

**Working:**

Discourse segmentation uses various features to break a document into segments:

- **Lexical cues:**
    - Specific words or phrases (e.g., "However," "In conclusion") can indicate segment boundaries.
- **Syntactic features:**
    - Sentence boundaries can be detected by parsing and analysing sentence structure.
- **Punctuation:**
    - Periods, commas, colons, and other punctuation marks can help in segmenting text into smaller parts.
- **Cohesion markers:**
    - Words or expressions that tie sentences or parts of sentences together (like "because," "and," "therefore") can help detect discourse boundaries.

**Examples:**

- Consider the following text:
  "John went to the store. He bought some milk. He also picked up bread."
- A discourse segmentation model would divide this text into three segments:
    - "John went to the store."
    - "He bought some milk."
    - "He also picked up bread."

- Here, each sentence is a segment, which is typical for many discourse segmentation tasks.
- In more complex texts, segments may not correspond to sentences directly but might represent larger chunks like topics or argumentation units.

*******