Module-5
Pragmatic and Discourse Processing

Discourse Processing

- building theories and models of how utterances stick together to form coherent discourse.
- Actually, the language always consists of collocated, structured and coherent groups of sentences rather than isolated and unrelated sentences.
- These coherent groups of sentences are referred to as discourse.
- Coherence and discourse structure are interconnected in many ways. Coherence, along with property of good text, is used to evaluate the output quality of natural language generation system.
- The discourse would be coherent if it has meaningful connections between its utterances. This property is called coherence relation. For example, some sort of explanation must be there to justify the connection between utterances. Another property that makes a discourse coherent is that there must be a certain kind of relationship with the entities. Such kind of coherence is called entity-based coherence.
- Discourse segmentations may be defined as determining the types of structures for large discourse. It is quite difficult to implement discourse segmentation, but it is very important for information retrieval, text summarization and information extraction kind of applications.

Reference Resolution

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

Terminology Used in Reference Resolution

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

Types of Referring Expressions

- **Indefinite Noun Phrases**

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

- **Definite Noun Phrases**

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

- **Pronouns**

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

- **Demonstratives**

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

- **Names**

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

Reference Resolution Tasks

- **Coreference Resolution**
- It is the task of finding referring expressions in a text that refer to the same entity. In simple words, it is the task of finding corefer expressions. A set of coreferring expressions are called coreference chain.
- **Constraint on Coreference Resolution**
- In English, the main problem for coreference resolution is the pronoun it. The reason behind this is that the pronoun it has many uses. For example, it can refer much like he and she. The pronoun it also refers to the things that do not refer to specific things. For example, It's raining. It is really good.
- **Pronominal Anaphora Resolution**
- Unlike the coreference resolution, pronominal anaphora resolution may be defined as the task of finding the antecedent for a single pronoun. For example, the pronoun is his and the task of pronominal anaphora resolution is to find the word Ram because Ram is the antecedent.

Anaphora Resolution

- **Anaphora Resolution == the problem of resolving what a pronoun, or a noun phrase refers to.**
- In the following example, 1) and 2) are utterances; and together, they form a discourse.
  1) John helped Mary.
  2) He was kind.
- As human, readers and listeners can quickly and unconsciously work out that the pronoun "he" in utterance 2) refers to "John" in 1). The underlying process of how this is done is yet unclear... especially when we encounter more complex sentences:
  An example involving Noun phrases (Webber 93)

1a) John traveled around France twice.
1b) They were both wonderful. ??

2a) John took two trips around France.
2b) They were both wonderful.

Anaphora resolution is the process of interpreting the link between the anaphor (i.e., the repeated reference) and its antecedent (i.e., the previous mention of the entity).

Hobbs Algorithm — Pronoun Resolution

- What is Pronoun resolution?
- To whom the pronoun **‘his’** refers to ??

Jack and Jill went up the hill

to fetch a pail of water.

Jack fell down and broke his crown

and Jill came tumbling after.

we as a human can easily relate that the word ‘his’ refers to Jack and not to the Jill, hill or the crown. But do you think is this task easy for computers as well-NO

- *The task of locating all expressions that are coreferential with any of the entities identified in the text is known as **coreference resolution**, and it occurs when two or more expressions in the text relate to the same person or object. As a result, pronouns and other referring expressions must be resolved in order to infer the correct understanding of the text.*

- Hobbs algorithm is one of the several approaches for pronoun resolution. The algorithm is mainly based on the syntactic parse tree of the sentences. To make the idea more clear let's consider the previous example of Jack and Jill and understand how we humans try to resolve the pronoun 'his'.
- As shown, the possible candidates for resolving pronoun 'his' were Jack, Jill, hill, water and crown.
- But then why we didn't even thought of crown as a possible solution? Maybe because the noun 'crown' came after the pronoun 'his'. This is the first assumption in the Hobbs algorithm, where the search for the referent is always restricted to the left of the target and hence crown is eliminated.
- Then can Jill, water or hill be the possible referents?
- But we know that 'his' may not refer to Jill because Jill is a girl. Generally animate objects are referred to either by male pronouns like- he, his; or female pronouns like- she, her, etc. and inanimate objects take neutral gender like- it.. This property is known as gender agreement which eliminates the possibilities of Jill, hill and water.
- Pronouns can only go a few sentences back, and entities closer to the referring phrase are more important than those further away… which finally leaves us with the only possible solution i.e. Jack. This property is known as Recency property.

Hobbs 1978

- Works on parse trees of sentence containing pronoun and of all previous sentences.
- Approximates binding theory, recency, and grammatical role preferences.
- Uses info on gender, person, and number constraints as a final check.

Algorithm

1.Begin at NP immediately dominating the pronoun

2. Go up tree to first NP or S node encountered. Call it X and path to it p.

3.Traverse all branches below X to left of path p in a left-to-right, breadth-first fashion. Propose as antecedent any NP node encountered which has an NP or S node between it and X.

4.If X is highest S node in sentence, traverse parse trees of previous sentences in order of recency, each in a left-to-right, breadth-first manner, and when an NP is encountered, propose as antecedent. If X not highest, go to 5.

5. From X go up to first NP or S. Call new node X and path to it p.

6.If X is NP and p did not pass through Nominal that X immediately dominates, propose X as antecedent.

7.Traverse all branches below X to left of p in left-toright, breadth-first manner, but do not go below any NP or S encountered.

8.If X is S node, traverse all branches of X to right of p in left-to-right, breadth-first manner, but do not go below any NP or S node encountered. Propose any NP encountered as antecedent.

9. Go to step 4.

Centering Algorithm

- Claim: There is single entity being "centered" on at any point in the discourse.
- Let U_n, U_{n+1} be 2 consecutive utterances.
- Backward looking center of U_n, written $C_b(U_n)$, represents focus after U_n interpreted.
- Forward looking centers of U_n, written $C_f(U_n)$, forms ordered list of entities in U_n that can serve as $C_b(U_{n+1})$.
- $C_b(U_{n+1})$ is highest ranking elt of $C_f(U_n)$ mentioned in U_{n+1}.

- Order of entities in $C_f(U_n)$:
  - subject > existential predicate nominal > object > indirect object > demarcated adverbial PP
- Let $C_p(U_{n+1})$ be highest ranked forward looking center

State-Based Transitions

	$C_b(U_{n+1}) = C_b(U_n)$ or undefined $C_b(U_n)$	$C_b(U_{n+1}) \neq C_b(U_n)$
$C_b(U_{n+1}) = C_p(U_{n+1})$	Continue	Smooth-Shift
$C_b(U_{n+1}) \neq C_p(U_{n+1})$	Retain	Rough-Shift

- Rule 1: If any elt of Cf(Un) is realized by a pronoun in Un+1 then Cb(Un+1) must be realized as a pronoun also.
- Rule 2: Transition states are ordered. Continue > Retain > Smooth-Shift > Rough-Shift.

Centering Algorithm

- Generate possible C_b - C_f combinations for each possible set of reference assignments.
- Filter by constraints (syntactic coreference constraints, selectional, centering rules and constraints).
- Rank by transition orderings
- Assign referents based on Rule 2, if Rule 1 and other constraints not violated.