NLP UNIT V – Discourse Analysis and Lexical Resources

This unit delves into two advanced areas: understanding language beyond the sentence level (Discourse Analysis) and exploring the foundational datasets and tools that make modern NLP possible (Lexical Resources).

Part A: Discourse Analysis

Standard sentence-level analysis can't tell us how "He loves it" relates to the previous sentence, "John just bought a new car." Discourse analysis is the study of these inter-sentence connections that create a meaningful text.

1. Discourse Segmentation

- **Elaborated Concept: Discourse Segmentation is the task of automatically identifying the topic structure of a text. Imagine reading a long news article that first discusses the political background of an event, then shifts to the economic impact, and finally discusses public reaction. Segmentation is the process of drawing a line between these sections. These coherent sections are often called "discourse segments."**
- **Why it's Crucial: Without segmentation, a text is just a long, undifferentiated string of words. For a summarization system to work, it needs to identify the main points of each topic segment. For a question-answering system, it can first find the segment most relevant to the question and then search for the answer within that smaller, focused area.**
- **Example in Action:**
    - **Text:**

The upcoming mayoral election is heating up. Candidate Smith has focused her campaign on improving public transport, promising new bus lines. Candidate Jones, on the other hand, is campaigning on a platform of fiscal responsibility and lower taxes. Polls show the race is currently too close to call.

In related news, the city's economy has shown signs of a slowdown. The latest jobs report indicates a rise in unemployment, particularly in the manufacturing sector. Economists are concerned about the potential for a recession if these trends continue.

    - **Segmentation Output: An algorithm would place a boundary between the two paragraphs. It would do this by noticing a lexical shift: the first paragraph uses words like election, campaign, candidate, polls, while the second paragraph shifts to economy, jobs, unemployment, recession. This change in vocabulary is a strong signal of a topic change.**

2. Coherence

- **Elaborated Concept: Coherence is what makes a text make sense. It's the underlying logical connection between ideas. It is supported by cohesion, which are the explicit linguistic "glue" words we use. A text can have cohesive markers but still lack coherence.**
- **Clear Distinction with Examples:**
    - **Example of Cohesion without Coherence:**

"I bought a new car. The car is painted blue, which is my favorite color. Colors can be described by their wavelength. Wavelength is a concept in physics, a subject I never liked in college."

- **Analysis: This text is *cohesive*. Each sentence links to the previous one ("car" -> "car", "color" -> "colors", "college"). However, it is not *coherent*. It's a nonsensical ramble with no logical flow or central idea.**
    - **Example of a Coherent Text:**

"I bought a new car. It is much more fuel-efficient than my old one. I expect to save a lot on gas during my daily commute."

- **Analysis: This text is *coherent*. Each sentence logically builds on the previous one to develop a central idea (the benefits of the new car). It uses cohesion (e.g., the pronoun "It") to support this coherence.**

3. Anaphora Resolution Algorithms

Anaphora resolution is the task of finding the antecedent for a pronoun. Here are two famous algorithms explained in more detail.

Hobbs' Naive Algorithm

- **Elaborated Idea: This algorithm is "naive" because it uses no real-world knowledge or discourse context; it relies purely on the syntactic structure of the sentences. It performs a simple, orderly search through the parse tree.**
- **A Step-by-Step Walkthrough:**
    - **Sentence: "John has a new car. He polishes it."**
    - **Goal: Resolve "He" in the second sentence.**
    - **Simplified Parse Tree for the first sentence: (S (NP John) (VP (V has) (NP a new car)))**
    - **Hobbs' Search Process:**

- **Start at the pronoun: "He".**

- **Search the current tree: (Let's assume we are parsing "He polishes it."). The algorithm finds no antecedent in the current clause.**
- **Move to the previous sentence's parse tree: (S (NP John) ...)**
- **Begin search: Traverse the tree from the root S node, left-to-right.**
- **The first NP encountered is (NP John).**
- **Check for agreement: Does "John" agree with "He"?**
    - **Number: John is singular, He is singular. (Match)**
    - **Gender: John is masculine, He is masculine. (Match)**
- **Resolution: Since a match is found, the algorithm resolves "He" to "John" and stops.**

Centering Algorithm

- **Elaborated Idea: This algorithm models the "focus of attention" in a discourse. It assumes that there's a central entity being discussed (the "center") and that pronouns are most likely to refer to this entity. It's more sophisticated than Hobbs because it uses discourse-level information.**
- **A Step-by-Step Walkthrough:**
- **Sentence 1 (S1): "John took the book from Mary."**
    - **The algorithm identifies the entities mentioned and ranks them by grammatical prominence to create the Forward-looking Centers (Cf) list.**
    - **Cf(S1): { John (Subject), the book (Object), Mary (Indirect Object) }**
- **Sentence 2 (S2): "He gave it to her."**
    - **The algorithm now tries to resolve the pronouns in S2.**
    - **It first determines the Backward-looking Center (Cb). This is the highest-ranked entity from Cf(S1) that is also mentioned in S2. Since "He" is the subject of S2, it's the most likely link to the highest-ranked entity of S1, which is "John".**
    - **Therefore, Cb(S2) = John.**
    - **Resolution:**
        - **The pronoun "He" is resolved to the Backward-looking Center, John.**
        - **The pronoun "it" is resolved to the next most prominent entity in Cf(S1), which is the book.**
        - **The pronoun "her" is resolved to the last remaining entity, Mary.**

4. Coreference Resolution

- **Elaborated Concept: Coreference resolution is the superset of anaphora resolution. It's the task of finding *every mention* of a specific entity in a text and linking them together in a "coreference chain." This includes pronouns, names, and descriptive noun phrases.**
- **Building a Coreference Chain (Example):**
    - **Text: "[Margaret Thatcher], the former Prime Minister of the UK, was a controversial figure. [The Iron Lady] was known for her firm policies. When [she] was in power, [the British leader] privatized many state-owned industries."**
    - **Coreference Chain 1:**
        1. **Margaret Thatcher**
        2. **The Iron Lady**
        3. **she**
        4. **the British leader**
    - **An NLP system identifies that all four of these expressions point to the same person and groups them. This is crucial for deep understanding.**

Part B: Lexical Resources (Elaborated)

These are the foundational tools and datasets that have enabled progress in NLP.

1. Stemmer vs. Lemmatizer: A Detailed Comparison

This is a classic comparison in NLP. Both aim to reduce words to a root form, but they do it very differently.

Feature	Porter Stemmer	Lemmatizer
Goal	To chop off suffixes using simple rules.	To find the true dictionary form (the "lemma").
Method	Algorithmic suffix-stripping.	Uses a dictionary (like WordNet) and considers the word's Part-of-Speech.
Output	Can be a non-word.	Is always a real dictionary word.

Feature	Porter Stemmer	Lemmatizer
Example: "studies"	studi	study
Example: "better"	better (no rule applies)	good (knows this is the comparative of "good")
Example: "meeting"	meet	meeting (if used as a noun) or meet (if used as a verb)
Complexity	Very fast and simple.	Slower and more complex (requires a dictionary).

Conclusion: Use a stemmer when speed is critical and you can tolerate some errors. Use a lemmatizer when you need grammatically correct root words and accuracy is more important.

2. Penn Treebank

- **Elaborated Description: The Penn Treebank (PTB) is arguably one of the most influential resources in NLP. It's not just a collection of text; it's a corpus where each sentence has been meticulously annotated by linguists with two layers of information:**
    1. **Part-of-Speech Tags: Every word is assigned a tag like NN (noun), VBD (verb, past tense), JJ (adjective).**
    2. **Syntactic Structure: The full grammatical phrase structure is marked up using brackets.**
- **Concrete Example: The sentence "The cat sat." is stored as:**
- **(S**
- **(NP (DT The) (NN cat))**
- **(VP (VBD sat))**
- **(. .))**
- **Impact: Before the PTB, NLP models were mostly rule-based. The PTB provided a large-scale, high-quality dataset that enabled the development of the first successful statistical parsers and taggers, which now dominate the field.**

3. Brill's Tagger

- **Elaborated Description: This tagger is famous for its elegant learning method, Transformation-Based Learning. It mimics how a human might learn a task: start with a simple strategy and then learn a series of specific correction rules.**
- **Learning Process in Action:**
    1. **Baseline Tagging: First, tag every word with its most frequent tag. This is fast but makes many mistakes. (e.g., "They can/NN fish/NN" - can is often a noun).**
    2. **Error Detection: The system compares its output to the correct tags in a treebank and finds its most frequent error. Let's say the most common error is tagging a word as a noun (NN) when it follows a modal verb (MD) like "can" or "will".**
    3. **Rule Generation: The system generates a transformation rule to fix this specific error: *"Change tag from NN to VB (verb) if the preceding tag is MD."***
    4. **Re-evaluation: This rule is stored, and the process repeats to find the next most common error and generate another rule for it. The final tagger is simply this ranked list of transformation rules.**

4. WordNet

- **Elaborated Description: WordNet is more than a thesaurus; it's a large, machine-readable lexical database. Its core building block is the synset (set of synonyms), which represents a single concept. These synsets are then interconnected in a vast network.**
- **Visualizing the Hierarchy: The most important relation is hyponymy (is-a). A small part of the WordNet hierarchy might look like this:**
    - **animal (Hypernym)**
        - **canine (Hyponym of animal, Hypernym of dog)**
            - **dog (Hyponym of canine)**
                - **poodle, bulldog, beagle (Hyponyms of dog)**
- **Applications: This hierarchical structure allows algorithms to compute semantic similarity. The distance between "poodle" and "bulldog" is very short (they share the immediate hypernym "dog"), so they are very similar. The distance between "poodle" and "cat" is longer, and the distance between "poodle" and "car" is very long.**

5. PropBank vs. FrameNet: Two Views on Semantic Roles

Both resources aim to annotate who did what to whom, but they have different philosophies.

- **PropBank (Proposition Bank):**

    - **Philosophy: Verb-centric. It takes a specific verb and defines a set of numbered, abstract roles for it.**
    - **Example: For the verb buy, it defines:**
        - **ARG0: The buyer (Agent)**
        - **ARG1: The thing bought (Theme)**
        - **ARG2: The seller**
        - **ARG3: The price**
    - **Sentence: "[Mary]ARG0 bought [a book] ARG1 from [John]ARG2 for [$10] ARG3."**
- **FrameNet:**
    - **Philosophy: Frame-centric. It defines a general situation (a "frame") and then identifies the participants. The same frame can be triggered by different words (verbs, nouns, etc.).**
    - **Example: The sentence triggers the general COMMERCE_BUY frame.**
        - **Frame Elements: Buyer, Goods, Seller, Money.**
    - **Sentence: "[Mary]Buyer bought [a book] Goods from [John]Seller for [$10] Money."**
- **Key Difference: PropBank's roles (ARG0, ARG1) are abstract and only defined relative to a specific verb. FrameNet's roles (Buyer, Goods) are more intuitive and consistent across different words related to the same frame (e.g., the COMMERCE_BUY frame could also be triggered by the noun "purchase").**

6. Brown Corpus and British National Corpus (BNC)

- **Elaborated Concept (The Importance of a "Balanced" Corpus): The Brown Corpus was revolutionary because it was balanced. Before Brown, most computer analysis of language was done on whatever text was available. The creators of Brown carefully selected samples from 15 different genres (from news reportage to science fiction) to ensure the corpus was a realistic miniature of written American English. The BNC later did the same for British English on a much larger scale, including spoken language.**
- **Why Balance Matters: An NLP model trained only on news articles will become very good at understanding news but will perform poorly when it encounters fiction, poetry, or spoken dialogue. A balanced corpus ensures that models are trained on a wide variety of language styles, making them more robust and general-purpose.**