Unit 3: **Syntax Parsing**

i) **Constituency Grammars:** Context free grammer, grammar rules for English, treebanks, grammar equivalence and normal forms, lexicalized grammer

1. **Syntactic Constituency**:
    - Syntactic constituency refers to the idea that groups of words can behave as single units or **constituents**.
    - When developing a grammar, we build an inventory of these constituents in the language.
    - For instance, a **noun phrase (NP)** can be composed of either a **Proper Noun** or other elements.
2. **Context-Free Grammars (CFG)**:
    - Context-free grammars are widely used formal systems for modeling constituent structure in English and other natural languages.
    - They are also known as **Phrase-Structure Grammars** and are equivalent to **Backus-Naur Form (BNF)**.
    - A CFG consists of a set of rules (productions) that express how symbols of the language can be grouped and ordered together.
    - Additionally, it includes a lexicon of words and symbols.
3. **Formal Definitions and Normal Forms**:
    - We introduce more formal definitions of context-free grammars and discuss grammar normal forms.
    - These normal forms help standardize the representation of grammatical rules.
4. **Treebanks**:
    - Treebanks are corpora that have been annotated with syntactic structure.
    - They provide labeled parse trees for sentences, allowing us to study the grammatical relationships between words.
5. **Example: Grammar of English**:
    - English has a rich grammar, and we can illustrate its structure using treebanks.
    - One such domain is the **Air Traffic Information System (ATIS)**, where relatively simple sentences like “I’d like to fly to Atlanta” are analyzed

ii) **Constituency Parsing:** Ambiguity, CKY parsing, span based neural constituency parsing, evaluation parsers, partial parsing, CCG parsing

1. **Ambiguity and Structural Ambiguity**:
    - **Ambiguity** arises when a sentence can be assigned more than one parse by a grammar.
    - **Structural ambiguity** specifically refers to cases where a grammar can produce multiple valid parses for a single sentence.

    - For example, consider Groucho Marx's famous line: "One morning I shot an elephant in my pajamas. How he got into my pajamas I don't know." The phrase "in my pajamas" can be part of either the NP or the VP, leading to structural ambiguity[1].
2. **Cocke-Kasami-Younger (CKY) Algorithm**:
    - The CKY algorithm is a dynamic programming approach for syntactic parsing.
    - It efficiently generates a set of parse trees for a sentence but doesn't determine which parse tree is correct.
    - Augmenting CKY with scores for each possible constituent helps identify the most likely parse tree.
    - Neural span-based parsers, which incorporate scores, enhance parsing accuracy[1].
3. **Neural Constituency Parsing**:
    - Recent large language models (LLMs) have shown remarkable performance across various tasks.
    - These models can be leveraged for constituency parsing.
    - Researchers have explored neural span-based parsers and their compatibility with classical dynamic programming techniques[2].
4. **Evaluation and Metrics**:
    - To assess parser accuracy, we use standard evaluation metrics.
    - These include precision, recall, F1-score, and labeled/unlabeled attachment scores.
    - Evaluating parsers helps improve their performance and reliability[3].
5. **Combinatory Categorial Grammar (CCG) Parsing**:
    - CCG is an alternative to context-free grammars.
    - It combines syntactic and semantic information.
    - Supertagging is used for parsing CCG, and partial parsing methods are employed when superficial syntactic analysis suffices

iii) Dependancy parsing: dependancy relations, dependancy formalism, dependancy treebank, transition and graph based dependency parsing, evaluations

Dependancy Relations

- In dependency parsing, we focus on the connections between linguistic units (usually words) using directed links. These links represent the dependencies between words.
- The central point of structural organization is the **finite verb**, which serves as the core of clause structure.
- All other syntactic units (words) are either directly or indirectly connected to the verb through these dependencies.

2. Dependancy Formalism

- Dependency grammar (DG) provides the formal framework for expressing these linguistic relationships.

- Unlike phrase structure grammar, which emphasizes phrasal nodes, DG primarily focuses on the relations between individual words.
- DG structures are flatter and well-suited for languages with free word order, such as Czech or Warlpiri.

3. Dependancy Treebanks

- A **dependency treebank** is a collection of annotated sentences where each word is linked to its head (usually the governing word) through labeled dependencies.
- These treebanks allow us to study and analyze the syntactic structures across different languages.
- One notable initiative is **Universal Dependencies (UD)**, which provides consistent annotation guidelines for over 200 treebanks in more than 100 languages. UD contributors have produced high-quality data for research and NLP applications.

4. Transition and Graph-Based Dependency Parsing

- Dependency parsing can be approached using two main paradigms:
    - **Transition-based parsing**: Models transitions from one parser state to the next, based on the parse history. It greedily selects the highest-scoring transition until a complete dependency graph is derived.
    - **Graph-based parsing**: Scores possible dependency graphs for a given sentence. It factors the graphs into individual arcs and searches for the highest-scoring structure.
- Researchers have explored both approaches, leading to significant advancements in parsing accuracy.

5. Evaluations

- Evaluating dependency parsers involves assessing their performance on annotated treebanks.
- Metrics include labeled attachment score (LAS), unlabeled attachment score (UAS), and other language-specific measures.
- Researchers continually refine and enhance dependency parsing models to achieve better results.