UNIT-3

PAGE NO.:
DATE: / /

Syntactic Analysis.

- Context free grammar, Probabilistic CFGs, Lexicalized
- Grammars Rules for english
- Tree Bank
- Normal forms
- Dependency Grammars
- Parsing (Syntactic, Dynamic Programming, Evaluation)
- Ambiguity.
- Probabilistic Chomskey form (CYK) (Locke-kasami Younger)

* Context Free Grammar:-

* Grammars contains the set of Rules to construct the Sentences in a language.

CFG → (V, T, P, S)

V → Finite set of variables
T → finite set of terminals
P → Production Rules
S → Start variable

* Natural language Processing is the fascinating and a Rapidly evolving field that intersects computer science, artificial intelligence and linguistics.
* It focuses on the interaction b/w the computer and human language, enabling machines to understand, generate and predict human language in a way that is both meaningful and useful.

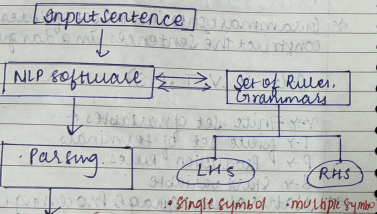
E-TIAU

PAGE NO.:
DATE: / /

⇒ Natural language processing is the capability of the computer software to understand the natural language.

⇒ There are variety of languages in the world. Each language has its own structure, and the structure is known as Grammar.

⇒ Grammar contains the set of Rules to construct sentences in a language



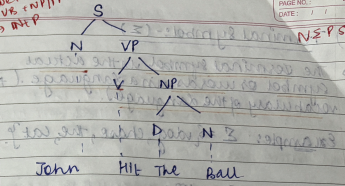
Eg: John hit the Ball → Input

S → NP + VP
NP → NN + DT
VB → PP + VB
PP → NN + PP

LHS RHS

RL: A → aB or A → a
LL: A → Ba or A → a

S → NP + VP
NP → Det + N
VB → VB + NP | PP
PP → PP + P



PAGE NO.:
DATE: / /

N ≥ PS

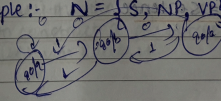
⇒ The context free grammar is a formal system used to describe the syntax of languages. It is widely used in natural language processing, parsing, to model the structure of sentences, and generate valid sequences of symbols or words in a language.

⇒ A CFG consists of 4 components, represented as $G = (N, \Sigma, P, S)$

(1) Non Terminal Symbol (N):-

⇒ It is a Non Terminal or abstract symbol that is used to represent grammatical constructs such as Noun, ~~Pronoun~~, Phase, verb phase, or sentences (S).

• Example:- $N = \{S, NP, VP\}$



(ii) Terminal Symbol:- (Σ)

⇒ The terminal symbol is the actual symbol or words in a language. (the vocabulary of the language)

- Example: Σ = {dog, chase, the, cat}.

(iii) Production Rule (P):-

⇒ The Rules that describes how the non-terminal symbols can be expanded into sequence of non-terminals & terminals

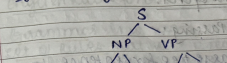
Eg:
S → NPVP
NP → det Noun
VP → verb NP

(iv) Start Symbol (S):-

⇒ The start symbol is the non-terminal symbol that Represents the Root of the grammar & the Entire sentence Structure.

Eg: S = Sentence

Eg: "The dog chases the cat"



the dog chases the cat

- Parsing in CFG:-

⇒ Parsing is the process of analyzing a sentence to determine if it can be generated by a given CFG and finding its structure.

Types of Parsers:-

(i) Top-down Parsing:-

⇒ Starts with symbol S and tries to derive the sentences using the Production Rules.

(ii) Bottom up Parsing:-

⇒ Starts with the input sentence and tries to construct a parse tree by applying the Rules in Reverse.

* Applications of CFGs:-

(i) Syntactic Parsing:

⇒ Building Parse Tree for sentence to understand their Grammatical structure.

(ii) Grammar checking:-

⇒ validating whether a sentence follows the grammatical Rules of a language

(iii) Natural Language understanding:-

→ understanding the structure of a sentence to extract meaning

* Advantages:-

(i) Simplicity

(ii) Parse Tree Generation

(iii) Flexible

* disadvantage:-

(i) Ambiguity

(ii) complexity in Real Languages

* English Rules for Grammar:-

⇒ Grammar is a word that contains the set of Rules that are used to construct sentences in languages.

⇒ Grammar Rules for English in NLP includes defining the structure of sentences, used to create parser, validate syntax,

⇒ These Rules can be formalized using context Free Grammar or Probabilistic Context Free Grammar

(i) Sentence (S):-

• a sentence consists of a Noun Phrase and a verb Phrase

• Rule S → NP, VP

* application
- Parsing
- Text Generation
- Machine Translation
- Speech Recognition

(ii) Noun Phrase: (NP)

⇒ A Noun Phrase can be a determiner followed by the noun (N) or a pronoun (PN)

• Rule NP → Det N
NP → Pronoun

(iii) Verb Phrase (VP):-

⇒ A verb Phrase consist of verb & ~~wrong~~ may include additional NP, Prepositions & adverbs

Corpus → collection of written / spoken text
Linguistic ⇒ study of language

* Tree Bank:-

⇒ Tree bank is a corpus of text annotated with syntactic or semantic structures typically in the form of parse trees.

⇒ Tree banks are used in natural language Processing for training & evaluating parsers and other models that analyze the structure of language.

⇒ Tree bank is a collection of sentences that have been syntactically parsed and annotated with their corresponding tree structures.

⇒ These tree structures represent the grammatical structure of the sentence, such as parts of speech and hierarchical relationships

- Sentences:- A tree bank consists of large no. of sentences from a specific language. Each sentence is carefully chosen to cover grammatical constructions
- Parse Tree:- Each sentence is annotated with parse tree, which visually represents the syntactic structure. A parse tree shows how a sentence can be broken down into its constituent parts.

- Annotations:

⇒ These trees are often manually created and include annotations for parts of speech, syntactic categories and grammatical functions.

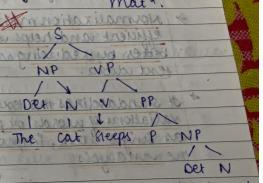
* Applications:-

(i) machine Translation

(ii) semantic analysis

(iii) Speech Recognition

* Eg: "The cat sleeps on the mat".



The cat sleeps on the mat

* Types of Tree Banks:-

(i) Constituency Tree banks

* Represents the syntactic structure using Phrase structure tree

(ii) Dependency Tree banks:

⇒ Represent syntactic structure based on word dependencies

eg: saw

The

* Challenges:

- Scalability
- Ambiguity

* Normal Form:- + CFG

→ The Normal form in Natural language Processing is a way to normalize or standardize the text data.
It's like tidying up a messy Room.

→ Normalization makes text processing more efficient and helps algorithm perform better by reducing variability in the text data.

→ It standardizes the text data so that different variations of words or phrases are treated as single item, improving the quality of the analysis

→ These forms ensure consistency in Representation and ease the complexity of tasks such as parsing, Machine Translation, Speech Recognition and Semantic Analysis.

* Below are some Primary types of Normal forms :-

(i) Chomsky Normal form :-

→ Chomsky Normal form is a standardized form for context free grammars, where every production Rule satisfies one of the criteria:

- A → BC :- A Non terminal Produces exactly two non-terminals
- A → a :- A Non terminal produces a single terminal
- S → ε : The start symbol produces an empty string.

→ It simplifies parsing algorithm (CYK)
→ It ensures a uniform structure for Grammar Rules.

(ii) Greibach Normal form :-

→ The Greibach Normal Form ensures that every production Rule of a CFG has :

- A → aα :- A Non-terminal produces a single terminal followed by zero or more non-terminals.

→ It is used for constructing top-down parsers.
→ It guarantees that parsing can always start with a Terminal.

Original form	GNF
S → AB	S → aB
A → a	B → b
B → b	

(iii) Lemmatized Form

(iv) Stemmed Form

* Application of Normal form:-

(1) Parsing and Syntax analysis

(2) Text Normalization

(3) Machine Translation

(4) Speech Recognition

* Advantages:-

(1) Reducing ambiguity

(2) Simplification of Processing

(3) Improved Performance

(4) Enhanced consistency

* Dependency Grammar:- CFG +

→ Dependency Grammar is a framework for syntactic analysis that represents the grammatical structure of a sentence based on the relationship between words. Unlike constituency like grammar, which uses phrase structure rule, dependency grammar focuses on word to word relationships, where one word (the Head) governs another (the dependent)

* (1) Head and dependent:-

→ In dependency grammar, the word are connected through a directed relationship.

→ The Head word governs the dependent word and the dependent word is reliant on the Head for its syntactic role.

Eg: "The cat chased the mouse"

- 'Chased' (verb) is the Head
- 'cat' (subject) and "mouse" (object) are dependents

(2) Root:-

→ Every dependency tree has a single Root, usually the main verb of the sentence

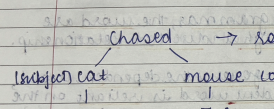
(iii) Tree structure:-

→ Dependency grammar represents sentences as trees, with arcs connecting Heads to their dependents.

→ Each word depends on one & only one Head.

* Dependency Tree Example:-

"The cat chased the mouse"



* Advantages:-

(1) Represents the direct syntactic & semantic structure for words b/w words.

(2) Simplicity.

(3) Flexibility.

*Challenges:-

(1) Ambiguity

(2) Inconsistent annotations.

* Parsing:-

→ Parsing in Natural language Processing is the process of analyzing the syntactic structure of a sentence to determine it's grammatical arrangement.

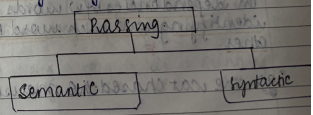
→ It involves identifying relationships between words and phrases and organizing them into a structured representation, such as parse tree or dependency graph.

→ Syntactic Structure or semantic structure:-

→ The Syntactic structure is essential in the Natural language Processing for understanding the structure of a sentence and assessing it's grammatical correctness.

→ The semantic structure of the Natural language Processing is concerned with the literal and contextual meaning of the word, phrase and sentence.

→ The passing of a sentence can be done in two ways; [Types]



(1) Syntactic Parsing :-

→ The syntactic Parsing is a process of analyzing the grammatical structure of the sentence. It helps identify the sentence components such as subjects, predicate, objects, phrases and clauses.

→ The parsing assigns the parts of speech Tagging to the words and phrases, categorizing them to Noun, determinants, adjectives, verbs and pronoun phrase.

→ There are different approaches to Syntactic Parsing, including :-

(i) Constituency Parsing :- This method breaks down the sentence into sub-phrases or constituents, which are hierarchically structured.

→ It's based on phrase structure Rule.

(ii) Dependency Parsing :- This method focuses on dependencies b/w words in a sentence, identifying how each word Relates to other.

Eg: The cat chased the mouse.

(*) Shallow Parsing :-

→ Shallow Parsing, also known as chunking is a Natural Language Processing used to identify and extract chunks or phrases from a sentence, without going into its complete syntactic structure.

→ unlike deep parsing, which analyzes the entire structure of a sentence (eg: full parse tree), a shallow parsing parses only on the base of syntactic unit, such as Noun, Phrase, verb phrase and preposition phrase.

* Working :-

(i) Tokenization :-

→ The Big dog is Running.

The	Big	dog	is	Running

→ Breaking the text in to individual words.

(ii) Part of speech Tagging :-

→ Assigning the Part of speech that is grammatical category to each word.

Eg: The Big dog is Running
DT Adj NN VP VP

(iii) Chunking :-

* This helps in getting the insights of meaning of the text.

→ Grouping adjacent words into chunks.

Eg: The Big dog is Running
DT Adj NN VP VP
NP VP

* Applications of Shallow Parsing.

(1) Information extracting.
(a) Machine Translation
(b) Question Answering
(4) Search Engines

* Application of Parsing :-

(i) Grammar checking
(ii) Machine Translation
(iii) Sentiment Analysis
(iv) Question answering System.

* Advantages :-

(1) Semantic Analysis
(2) Syntactic Analysis
(3) Structured
(4) Improved NLP model

Programming :-

* Dynamic Parsing :- Parsing

→ Dynamic Programming Parsing is a technique in Natural Language Processing used to analyze the and generate a syntactic structure of a sentence by breaking down parsing into smaller, manageable subproblems and solving them optimally.

→ It is commonly used in context free Grammar parsing to reduce the complexity computational.

• Key Features :

(i) Optimal Structure :-

→ The overall problem can be solved optimally by combining solutions to subproblems

(ii) Overlapping Subproblems :-

→ Reuses previously calculated Result for the same subproblem instead of Recalculating them.

(iii) Parsing Table :-

* A tabular structure stores the intermediate Parsing Result, to avoid Redundant computation

(iv) Bottom up or Top down approach:-

⇒ The dynamic programming parsing can be used in both the paradigm.

* Common Dynamic Programming Parsing Algorithm:-

(i) Cocke Younger Kasami Algorithm:-

⇒ The cocke younger kasami algorithm is a dynamic programming parsing algorithm used to parse the sentence and determine whether they belong to a given context free grammar.

⇒ It works with grammars in chomskey Normal form and is widely used in Natural language Processing for Syntactic Analysis.

• Rules:-

(i) The grammar should be in chomskey Normal form i.e

⇒ A → BC : A non terminal produces exactly two non terminals

⇒ A → a: A non terminal produces a single terminal.

⇒ If the grammar is not in CNF, it must be converted.

(ii) construct a triangular parsing table of (nxn) matrix, where n is the length of input sentence

(iii) Bottom up parsing:-

⇒ start with the given sentence & construct the tree by applying the Rules in Reverse.

* Advantages of CYK:-

(1) Efficiency:-

Runs in $O(n^3 \cdot |G|)$ where n is the input sentence & G is the size of the grammar.

(2) Clear structure

(3) Handles ambiguity.

* Advantages of Dynamic Programming:-

(i) versatility:- adapts various grammars types (CYK, CNF, CFG, etc)

(ii) Flexibility:-

(iii) Efficiency:-