21/11/23

# UNIT-II

## Syntactic Analysis

**Parsing:**
parsing is the process of finding the structure of the sentence to know the meaning of the sentence

It has 3 types:

(i) Parts of speech tagging

(ii) syntactical parsing

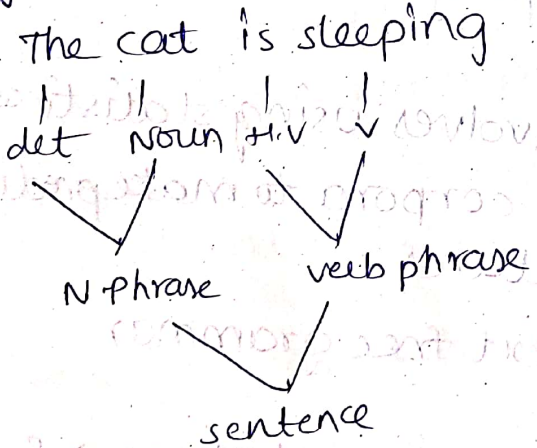
(iii) semantical parsing

POS

Ex: The cat is sleeping = sentence

det Noun H.V Verb = tags

(ii) syntactical parsing




(iii) Semantic paesing

The cat is sleeping

Agent: The cat

Action: sleeping

Methods of Parsing:

(i) Rulebased parsing
(ii) statistical Parsing
(iii) chart parsing

(i) Rule based parsing:

Rule based parsing utilizes grammatical rules to analyze the syntactical structure of a sentence

Ex: context free grammar and phrase-structure rules.

2*(3+5) the grammar provides a set of production rules that guide the generation of valid sentences in the language

(ii) statistical parsing:

statistical parsing involves using statistical models trained on large corpora to make predictions about the syntactic structures

Ex: Probabilistic, context free grammar

<sentence> → <Nounphrase> <verb phrase> [1.0]

<Nounphrase> → <Determinant> <Noun> [0.6]
<Noun phrase> <preposition phrase> [0.4]

<Det> → 'the' [0.8] | 'a' [0.2]

<Noun> → 'cat' [0.5] | 'dog' [0.5]

<VP> → <V> [0.7 | <PP> <VP> [0.3]

<V> → 'chased' [0.4] | 'ate' [0.4]

<S> – sentence
<NP> – Noun phrase
<VP> – verb phrase
<PP> – prepositional phrase
<V> – verb
<N> – Noun

IN probabs
The numbers in square brackets represents the probabilities associated with each production rule

③ Dep chart Parsing:
Parsing strategy that bills a chart datastructure to efficiently store and retrieve parsing results

④ Machine learning based parsing:
It utilizes machine learning algorithms trained on annotated data to predict syntactic structure

Ex:- support vector machines (SVM)
conditional random fields (CRF)
(or)
Neutral Network based models

Neutral Network based models are used to capture complex language patterns for parsing.

* SVM is used to classify emails based on word frequencies

REPRESENTATION OF SYNTACTIC STRUCTURE

* It is the representation of ~~syntactical~~ grammatical structure of sentence is captured and represented
* There are several ways to represent syntactic structure

(i) Syntax analysis using dependency graph
(ii) Syntax analysis using phrase structure tree

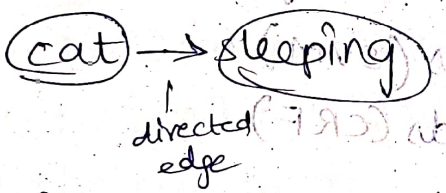
## (i) Syntax analysis using dependency graph

* It is the common approach in NLP
* It represents grammatical structure of a sentence as a directed graph
* words are represented by nodes
* Relationship between words are represented by directed edges.

Ex: subject-verb relationship is represented by edge from subject-word to verb-word

The cat is sleeping

subject → verb




Ex: object-verb relationship is represented by an edge from object-word to verb-word

Uses of dependency graph:

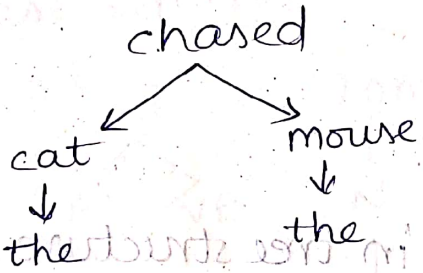
* Syntax analysis task
* Name entity Recognition
* sentiment Analysis

## Goal of Dependency graph:

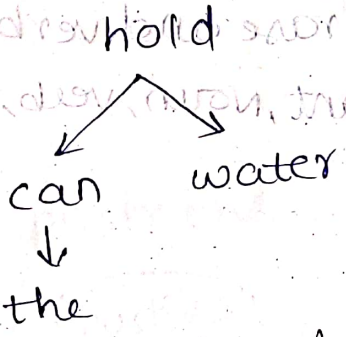
Goal is to automatically generate dependency graph for a given sentence

* Algorithm used for dependency graph or transition based algorithm or graph based algorithm.

Ex: The cat chased the mouse




Ex: The can can hold water



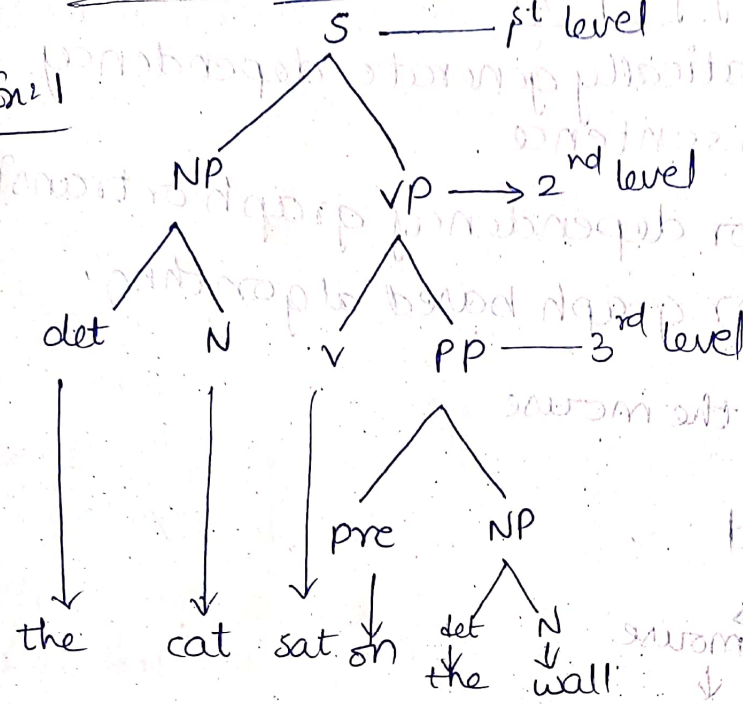

24/11/23
Friday

## (ii) Syntax analysis using phrase structure tree:

* It represents the grammatical structure of a sentence as a tree structure
* The sentence is broken down into phrases. each phrase is represented as a node in the tree
* The words in the sentence are then assigned to appropriate nodes based on their syntactic rules

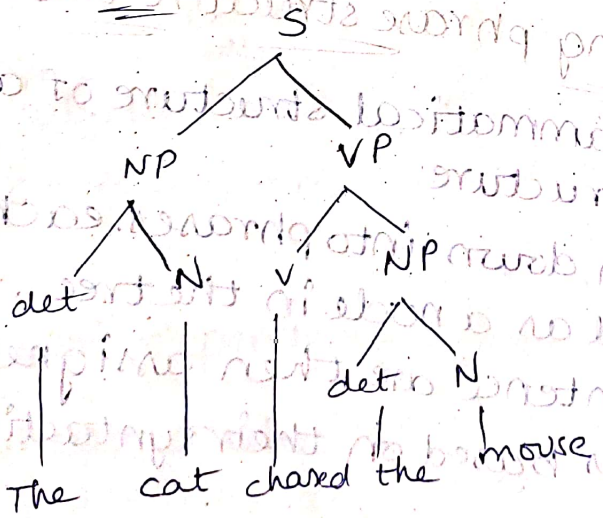
Phrase structure tree :-

Ex:1




* The sentence is represented in tree structure in the root node (1st level).
* 2nd level nodes are Noun phrase and verb phrase
* 3rd level nodes are determinant, Noun, verb, preposition phrase.

Ex:-2

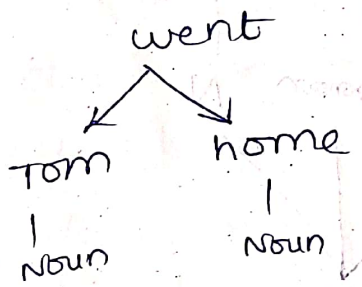



Uses :-

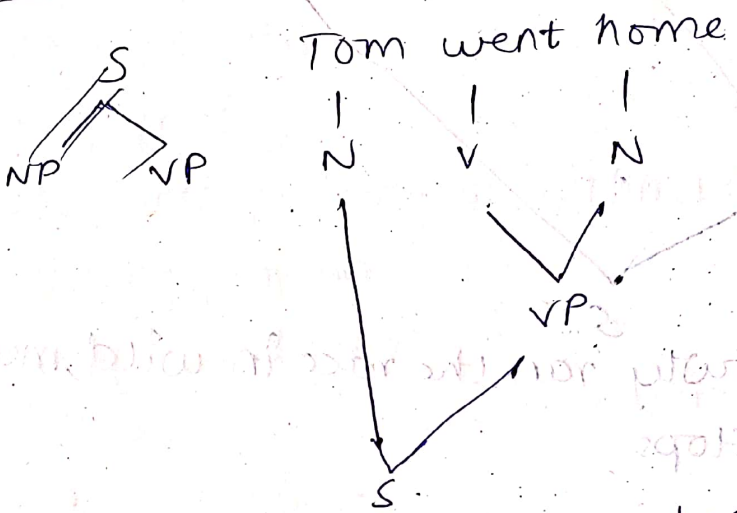
* Parsing
* Translation
* Text to speech synthesis (converting text into speech)

Ex:- Tom went home

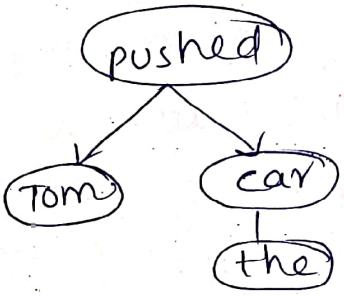
Dependency graph

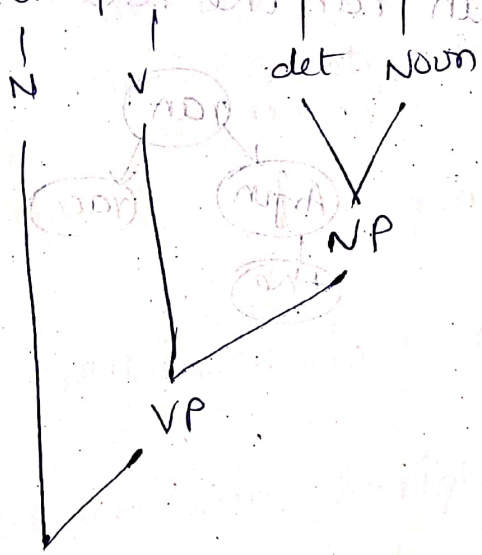



phrase structure tree -

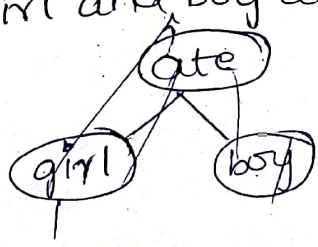


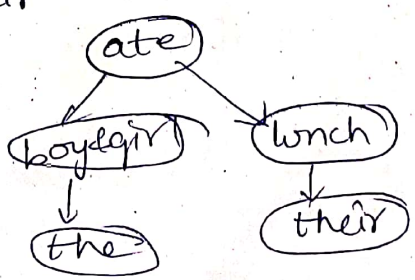

Ex: Tom pushed the car



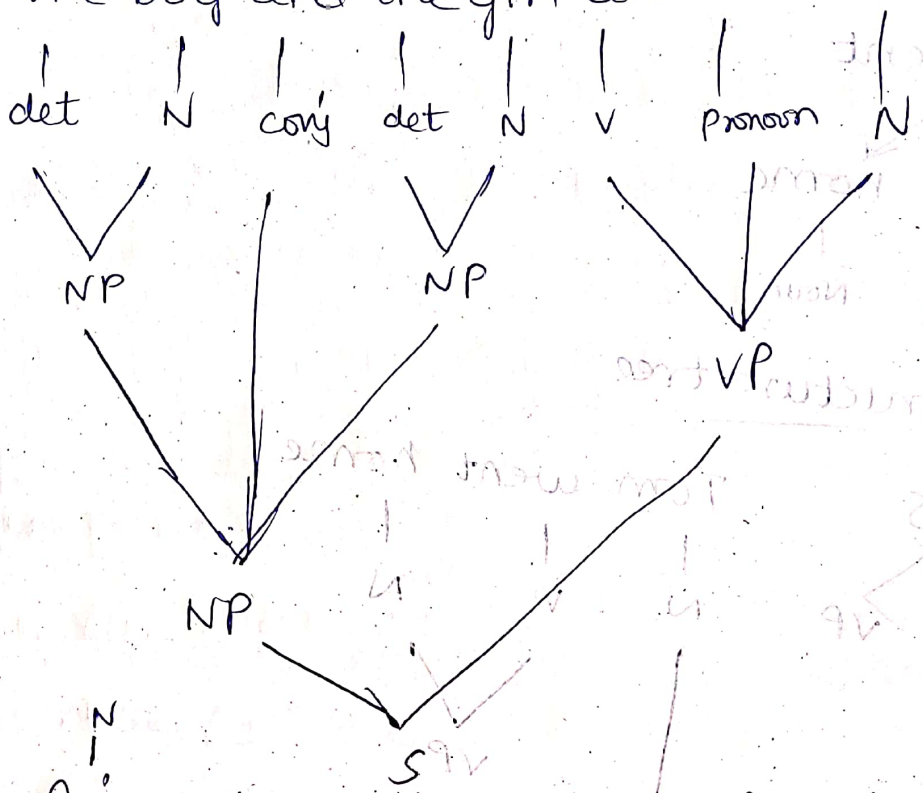




Ex: The girl and the boy ate their lunch



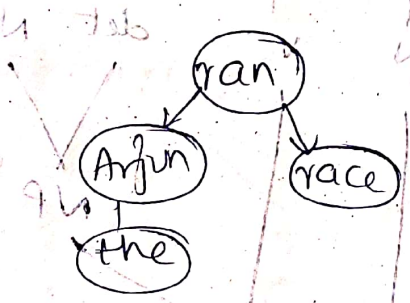



## Phrase tree structure.




Ex:- Arjun slowly ran the race in wild, multi-colored flipflops

Arjun ran the race - simple sentence



## Parsing algorithms:

* Parsing algorithms are used in computer science to analyze the structure of the string of symbols in a particular formal languages or computer language or data structure
* Typically, represented as context free grammar

These are two types

1. Top-down parsing
2. Bottom-up parsing

* Commonly used parsing algorithms are
  1. shift reduce parsing
  2. Hypergraph and chart parsing
  3. Minimum spanning tree and dependency parsing

### 1. shift reduce parsing:

shift reduce parser is the bottom up parsing technique commonly used in compiler construction.

It involves in two main actions.

(i) shifting
(ii) Reducing

(i) shifting: shifting move the input symbols into the stack

(ii) Reducing: If the top of the stack forms the right hand side of the production replace symbols with the corresponding non-terminals.

Algorithm steps:

step-1: It aims to find the words and phrases sequence that corresponds to the right hand side of the grammar production and replaces them with the left hand side of the production

Step-2: It tries to find the word sequence that continues until the whole sentence is reduced.

Step-3: The parser starts with the input symbol and aims to construct the parser tree upto the end symbol. At shift action, the current symbol in the input string pushed into the stack

25/11/23
Saturday

Grammar rules

$S \leftarrow S+S$

$S \leftarrow S-S$

$S \leftarrow (S)$

$S \leftarrow a$

Input string

$a_1-(a_2+a_3)$

Parsing table

| input string | Input string | Action |
|---|---|---|
| $ | $a_1-(a_2+a_3)$$ | shift $a_1$ |
| $a_1 | $-(a_2+a_3)$$ | Reduce $a_1$ |
| $S | $-(a_2+a_3)$$ | shift - |
| $S- | $(a_2+a_3)$$ | shift ( |
| $S-( | $a_2+a_3)$$ | shift $a_2$ |
| $S-(a_2 | $+a_3)$$ | Reduce $a_2$ |
| $S-(S | $+a_3)$$ | shift + |
| $S-(S+ | $a_3)$$ | shift $a_3$ |

| | | |
|---|---|---|
| $S-(S+a3 | )$ | Reduce a3 |
| $S-(S+S | )$ | Reduce S+S |
| $S-(S | )$ | shift ) |
| $S-(S) | $ | Reduce (S) |
| $S-S | $ | Reduce S-S |
| $S | $ | Accepted |

## Hyper graphs

* Hyper graphs are normal directive graphs
* Hyper graphs is the generalization of a graph that allows an edge connect more than two vertices
* Each edge can connect multiple source nodes to multiple target nodes
* Hyper graphs are useful in semantic role labeling (SRL)
* SRL involves identifying the role of words in a sentence such as subject, object and predicate.
* In Hyper graph representation, each word could be a node and hyper edge that connects the words, that form a semantic role

Ex: John (sub) eats (verb) an apple (object) / sudeshna (sub) studies (verb) NLP

John → eats → apple

sudeshna → studies → NLP

This allows more expressive representation of relationship between words and sentences

Ex: social Hyper graph

push →
pull ←

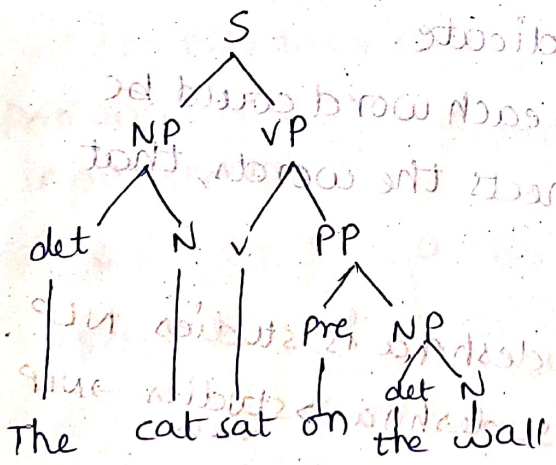
Consider a social network where a group of friends form based on their common interest. Each group is represented as hyperedge connecting multiple individuals. Here vertices are people in the social network and hyperedges are groups of friends with common interest

* Hyper edges connects more than two vertices forming hyper graph.

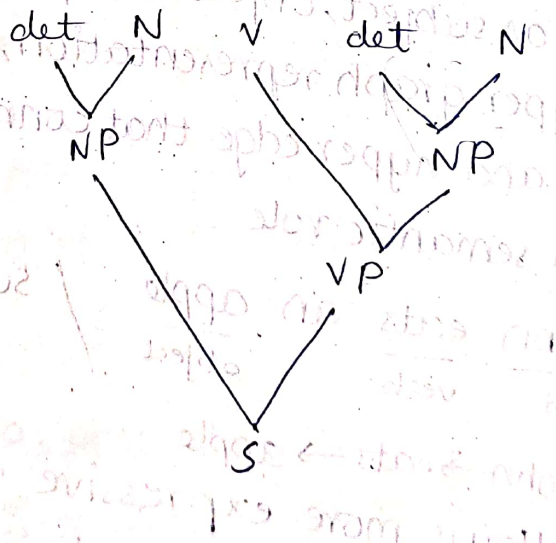
Chart Parsing:

Chart Parsing is a technique used in NLP to generate parse tree for sentences based on given grammar. It is particularly useful for ambiguous sentence where multiple valid parse trees can be generated for a single sentence.

Ex: I saw the man with telescope




the cat chased the mouse



## Minimum spanning tree and Dependency graph:

In NLP, sentences are represented as graph, where words are nodes and relationships are edges. While a Minimum spanning tree in graph theory aims to connect all the nodes with the minimum possible edge weight

* In NLP, a similar goal might be achieved by extracting the essential components of a sentence.

Ex: The quick brown fox jumped over the lazy dog.

28/11/23
Tuesday

## Models for ambiguity resolution in parsing:

① Probabilistic context free grammar
② Generative model for parsing
③ Discriminative model for parsing

### ① Probabilistic context free grammar:

(statistical parsing)

### ② Generative model for parsing:-

Generative model for parsing aims to generate sentences or structures according to the specified grammar. Commonly used in parsing are

* probabilistic context free grammar (PCFG)
* Hidden Markov's model (HMM)

### Hidden Markov's model:

HMM consists of two basic steps

(i) Hidden states (POS tags)
(ii) Observable states (words)

* HMM can be used to model the generation of sequence of observable states (words) given an underlying sequence of syntactic structures.
* Each syntactic structure associated with probability distribution over possible words.

HMM in POS tagging;

⇒ which is a form of syntactic parsing. In this hidden states represents pos tags and observable states represents words.

Hidden POS tags :- Noun, verb, adjective, preposition, adverb, determinant etc

observable state :-

cat, dog, chased

Context free grammar : generative process

$S \rightarrow VP \quad NP\ [0.6]$

$NP \rightarrow det \quad N\ [0.4]$

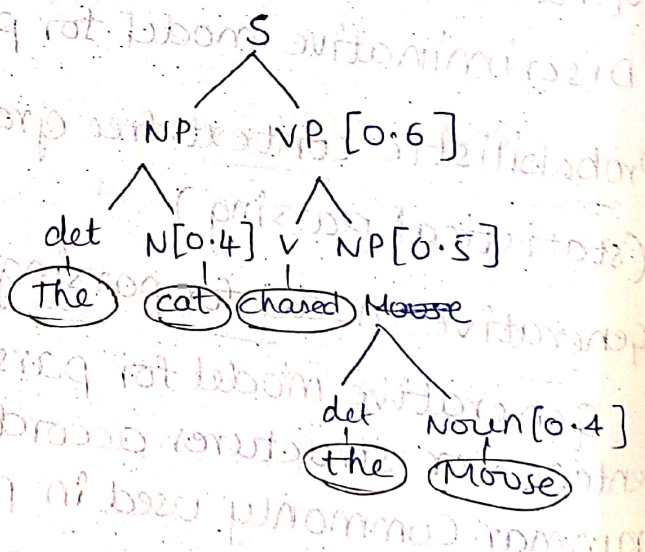
$VP \rightarrow V \quad NP\ [0.5]$

$det \rightarrow$ "The" $[0.7]$

$N \rightarrow$ "cat" $[0.5]$

$V \rightarrow$ "chased" $[0.8]$

$N \rightarrow$ "Mouse" $[0.5]$



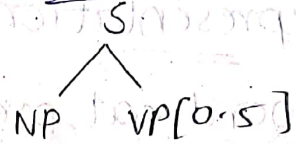

The generative sequence or sentence is

The cat chased the mouse

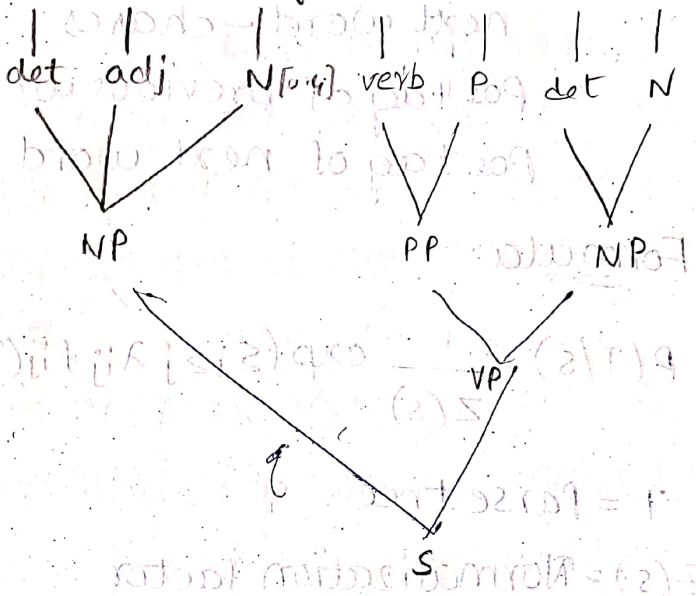
Context free grammar

$S \rightarrow NP\ VP[0.5]$
$NP \rightarrow Det\ N[0.6]$
$NP \rightarrow Det\ Adj\ N[0.4]$
$VP \rightarrow V[0.7]$
$VP \rightarrow V\ NP[0.3]$
$Det \rightarrow$ 'The' $[0.4]$
$Det \rightarrow$ 'A' $[0.6]$
$N \rightarrow$ 'cat' $[0.5]$
$N \rightarrow$ 'dog' $[0.5]$
$adj \rightarrow brown[0.8]$
$V \rightarrow jumps[0.6]$
$V \rightarrow runs[0.4]$
$P \rightarrow$ "over" $[0.6]$

Generative process




The brown dog jumps over the cat




05/12/23
Tuesday

Discriminative model for parsing

It focus on learning the conditional probability of a parse tree given an input sentence

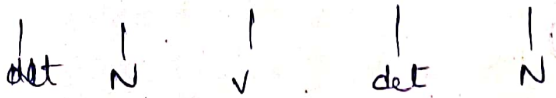
* Discriminative model uses maximum entropy model

Maximum entropy model:

The task is to assign parts of speech tagging to each word in a sentence.

Features:

* Word, previous word, next word, pos tag of previous word and pos tag of next word.
* The cat chases the mouse



* Feature representation :-

For the word cat, previous word feature is 'the'

word - cat

previous word - the

next word - chases

Pos tag of previous word - det

Pos tag of next word - verb

Formula:

$$P(T/s) = \frac{1}{Z(s)} \exp\left(\sum_i \sum_j \lambda_{ij} f_{ij}(T,s)\right)$$

T = Parse tree

Z(s) = Normalization factor

S = Input sentence

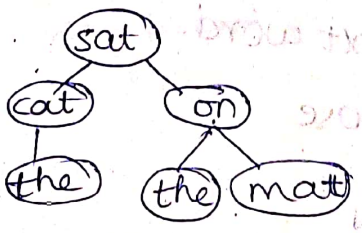
$\lambda_{ij}$ = Model parameters

$f_{ij}(T,s)$ = Feature function that captures relevant information about the parse tree and the sentence

Example :-

Feature function could be indicates whether the certain ~~free~~ phrase structure exist in the parse tree, helping the model to learn relationships b/w the words in the sentence

Ex:- the cat sat on the mat



## Multilingual Issues:-

### Token:-

Token refers to a sequence of characters that represents a single unit

* In multilingual setting, the definition of token can be more complex this is because different languages may use different writing systems, character encodings or word segmentation conventions which can effect how tokens are defined and processed
* In multilingual, natural pro language processing it is important to carefully define and standardize the tokenization process in order to ensure that input text is processed consistently and accurately across different languages and scripts
* This may involve developing language specific tokenization rules or machine learning techniques to automatically segment text into tokens.

Ex:- I love Beijing Tiananmen
我爱北京天安门。

The sentence consists of six characters which could be considered tokens in chinese language processing pipeline. However, the sentence could be segmented into 4 words

### Case:-

Case refers the capitalization of a word in a piece of text. It is useful to convert all words to lower case inorder to reduce the number of distinct tokens and simplify subsequent analysis.

## Encoding:

Encoding is a process of converting text into numerical representation that can be processed by machine learning algorithms.

Ex:- The cat sat on the mat

6 tokens

Parse matrix (Identity matrix)

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$