1 (a) Explain the components of transformational grammar.

Transformational Grammar has 3 Components

1. Phrase Structure Grammar
2. Transformational Rules
3. Morphophonemic rules

All three components consist of a set of rules.

Phrase structure grammar:

Phrase structure grammar consists of rules that generate natural language sentences and assign a structural description to them. Following are the set of rules:

S - sentence
VP - verb phrase
NP - noun phrase
V - verb
Det - determiner
Aux - Auxiliary verb

For example, Transformational Grammar for the sentence - I ate apple

S -> Np + VP
NP -> I
VP -> V+NP
V -> ate
NP -> apple

Transformational grammar for the sentence - Pooja plays veena

S -> NP+VP
Np -> pooja
VP -> V+Np
V -> plays
NP -> veena

2. Transformational rules:

The second component of transformational grammar is a set of **transformational rules**, which transform one phrase-maker into another phrase-maker. These rules are applied to the terminal string generated by phrase structure rules. Transformational rules are heterogeneous and may have more than one symbol on their left-hand side. These rules are used to transform one surface representation into another, for example, an active sentence into a passive sentence. The rule relating to active and passive sentences is -

NP1- Aux – V - NP2 □ NP2 – Aux + be + en – V – by - NP1

Transformational Rules can be obligatory or optional. An obligatory transformation is one that ensures agreement in number of subject and verb. An optional transformation is one that modifies the structure of a sentence while preserving its meaning.

Example for transformational rules:

Consider the active sentence - *The police will catch the snatcher.* - *(1)*

The application of phrase structure rules will assign the structure as shown below

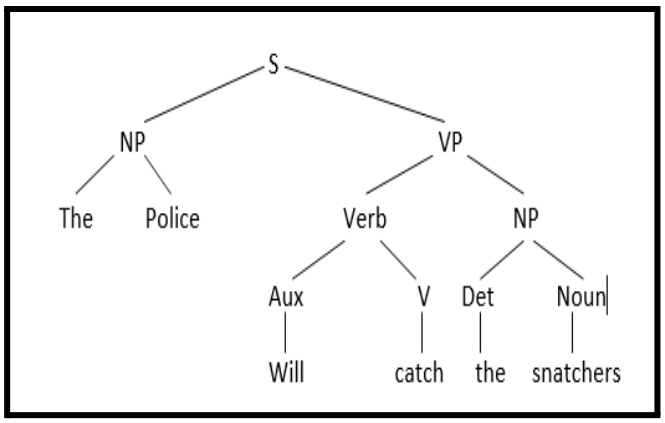


Fig: Parse structure of the sentence - *'The police will catch the snatcher'*

The passive transformation rules will convert the sentence into
The + snatcher + will + be + en + catch + by + the + police

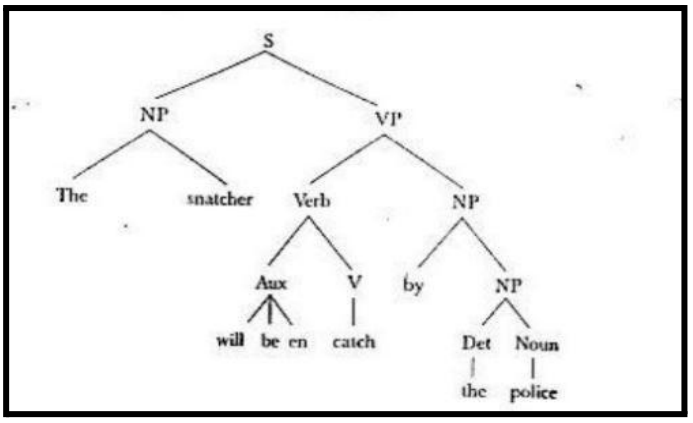


Fig: Structure of sentence (1) after applying passive transformation

The Police will catch the snatcher. NP1-Aux-V-NP2
The snatcher will be caught by the police NP2-Aux-be+en-V+by+NP1
The rule relating to the conversion of active to passive sentence -
NP1- Aux – V - NP2 NP2 – Aux + be + en – V – by - NP1

3. Morphophonemic rules:
Morphophonemic rules match each sentence representation to a string of phonemes. Morpho phonemics involves an investigation of the phonological variations within morphemes, usually marking different grammatical functions; *e.g.,* the vowel changes in "sleep" and "slept," "bind" and "bound," "vain" and "vanity," and the consonant alternations in "knife" and "knives," "loaf" and "loaves."

The Transformational rule will reorder 'en + catch' to 'catch + en' and subsequently one of the morphophonemic rules will convert 'catch + en' to 'caught'.

'en + catch' ---> 'Catch + en' ---> Caught

(b) Write the transformational grammar for the sentence "The boy hit the girl."
Transformational Grammar for the sentence - A boy hit the girl

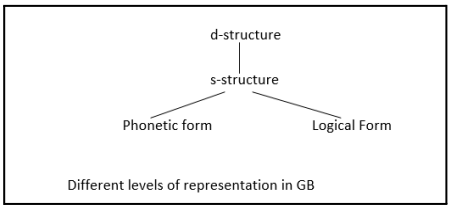
S -> NP + VP
NP -> Det + Noun
VP -> V + NP
V -> Aux + V
Det -> A, the
Noun -> boy, girl
Verb -> hit

2 What is Government and Binding theory? Explain with an example.

Chomsky was the first to put forward the Government and Binding (GB) theory. GB theory has four levels

1. s-level (surface level of transformational grammar is renamed as s-level)
2. d-level (deep root level of transformational grammar is renamed as d-level)
3. phonetic form
4. logical form

phonetic form and logical form are parallel to each other.



Different levels of representation in GB

Language is the representation of some **'meaning'** in a **'sound'** form.

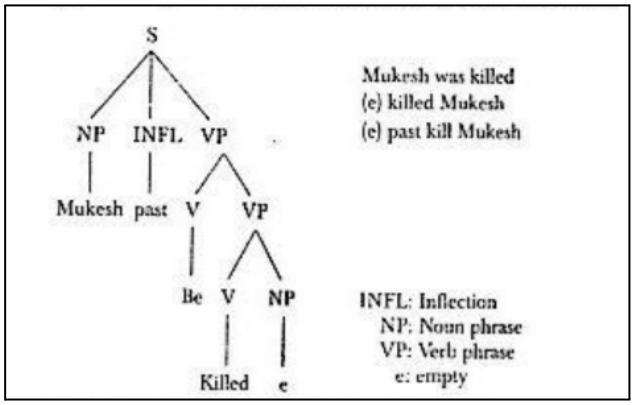
'meaning' logical form (LF) and
'sound' phonetic form (PF)

GB is concerned with the Logical Form rather than the Phonetic Form.

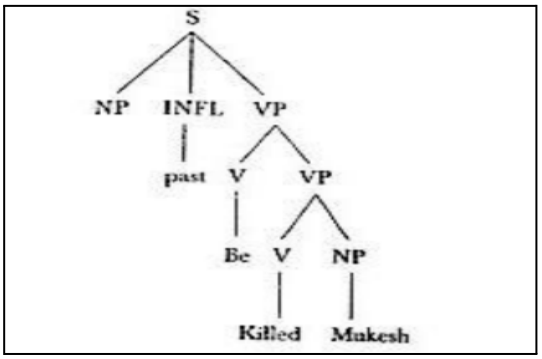
The vision of GB is, if we define rules for **structural units at the deep level**, it will be possible to generate any language with fewer rules. These deep-level structures are common to all languages. **The existence of deep-level, language-independent, abstract structures, and the expression of these in surface-level, language-specific structures with the help of simple rules is the main concern of GB theory.**

Example to explain d-structure and s-structure:

Consider the sentence: Mukesh was killed



s-structure of the sentence - Mukesh was killed



d-structure of the sentence - Mukesh was killed.

Components of GB:

GB comprises a set of theories that map the structures from d-structure to s-structure and to logical form. The phonetic form is not considered. A general transformational rule called "**Move ∝**" is applied to the d-structure level and s-structure level. This can move the constituents at any place if it does not violate the constraints put by several theories and principles.

GB is represented as shown in fig.

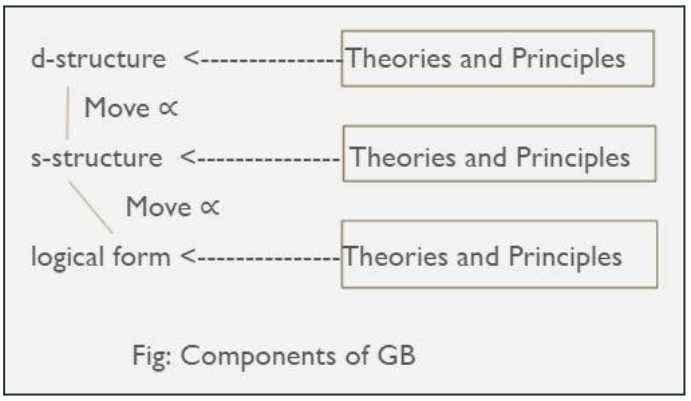


Fig: Components of GB

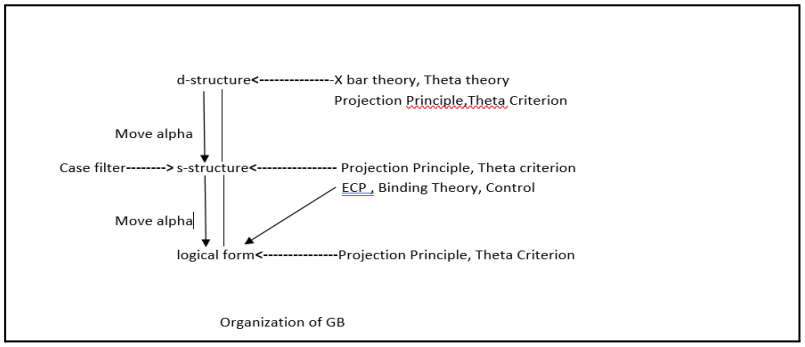
Move **∝:**

Chomsky reduced all the long-distance movements to one rule called move **∝.** In the rule 'Move **∝',** alpha stands for any constituent. The rule does not specify where **∝** can be moved from, or where it should be moved to: all this is left 'free'. In effect, this rule says, 'Move anything from anywhere to anywhere.

GB consists of, ' a series of modules that contain constraints and principles' applied at various levels of its representation and the transformation rule, Move **∝.**
The modules include

- X-bar theory
- Projection Principle
- θ -Theory
- θ - Criterion
- C-Command and Government
- Case Study or Theory
- Empty Category Principle (ECP)
- Binding Theory

General Characteristics of GB:



Organization of GB

X-Bar Theory:

The x bar theory is one of the central concepts in GB. Instead of defining several **phrase structures and the sentence structure** with separate sets of rules, X bar theory defines them both as **maximal projections of some head.** Hence the entities defined become language-independent.
Noun phrase (NP) is the maximal projection of Noun (N)
verb phrase (VP) is the maximal projection of Verb (V)
adjective phrase (AP) is the maximal projection of Adjective (A)
preposition phrase (PP) is the maximal projection of Preposition (P)
They can be represented as head X of their corresponding phrase (where X = {N, V, A, P}
Even the sentence structure S' - which is a projection of sentence can be regarded as the maximal projection of inflection (INFL).
GB envisages projection at 2 levels:

1. The projection of the head at semi-phrase level, denoted by X bar

2. The maximal projection at the phrase level, denoted by the X double bar

For sentences -

the first level projection is denoted as S

and the second level maximal projection is defined as S'.

General Phrase and Sentence Structure:

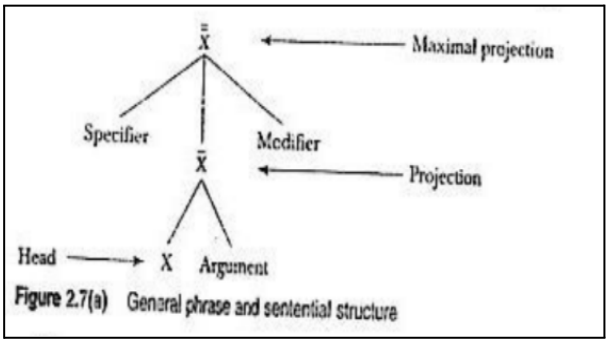
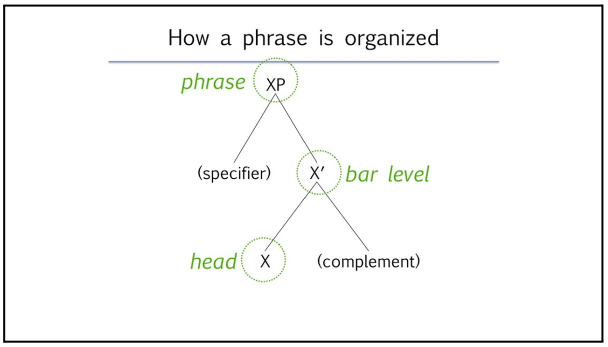


Figure 2.7(a) General phrase and sentential structure



Projection principle

Projection principle places a constraint on the three syntactic representations and their mapping from one to the other. The projection principle states that representations at all syntactic levels (i.e., d-level, s-level, LF-level) are projection from the lexicon (head). Thus, lexical properties of categorical structure (sub-categorization) must be observed at each level.

Ex: Suppose the 'object' is not present at the d-level, then another NP cannot take this position at the s-level.

Mukesh was killed OBJECT

Projection principle, in conjunction with the possibility of the presence of an empty category and other theories (Binding theory) ensures correct movement and well-formed structure.

Theta Theory (θ-theory) or the Theory of Thematic Relations

GB puts restrictions on lexical heads through which it **assigns certain roles to its arguments.** These roles are pre-assigned and cannot be violated at any syntactic level as per the projection principle. These role assignments are called **theta roles** and are related to 'semantic selection.

Theta-role and theta-criterion

There are certain thematic roles from which a head can select. These are called theta-roles and they are mentioned in the lexicon.

Ex: The verb 'eat' can take arguments with θ-roles '(Agent-Theme)'.

Agent is a special type of role which can be assigned by a head to outside arguments (external arguments) whereas other roles are assigned within its domain (internal arguments).

C-command and government

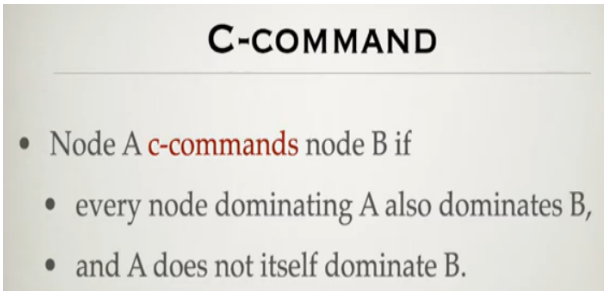
C-Command

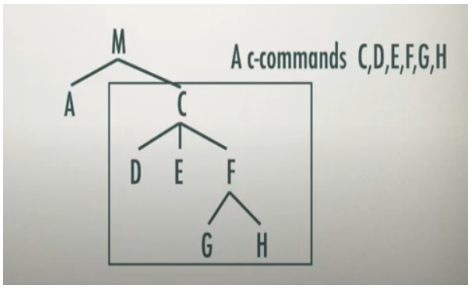
C-command define the scope of maximal projection. It is a basic mechanism through which many constraints are defined on move α. If any word or phrase (say ∝ or β) falls within the scope of and is

determined by a maximal projection, then it is dominated by maximal projection.

The Necessary and sufficient condition for C-command:

- If there are two structures α and β related in such a way that '**every maximal projection dominating α dominates β',** we say that **α C-commands β**.
- The definition of C-command does not include all maximal projections dominating β, but only those dominating α.



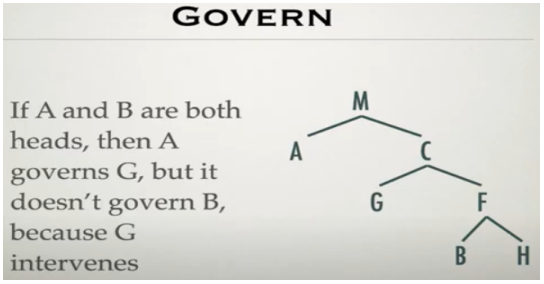


Government:

Government is a special case of C-command.

- **α governs β iff:**
- **α C-commands β**

α is an X (head, eg: noun, verb, preposition, adjective, and inflection), and every maximal projection dominating β dominates α. No maximal projection can intervene between the governor and governee.



Binding theory:

Binding theory applied at A-positions is given as -

An anaphor (+a) is bound in its governing category.
A pronominal (+p) is free in its governing category
An R-expression (-a, -p) is free.

Example:

A: Mukesh$_i$ knows himself$_i$
B: Mukesh$_i$ believes that Amrita knows him$_i$
C: Mukesh believes that Amrita$_j$ knows Nupur$_k$

Empty category principle (ECP)

'Proper government'

α properly governs β iff
α governs β and α is lexical (i.e, N, V, A or P) or
α locally A binds β.

Bounding and control theory:

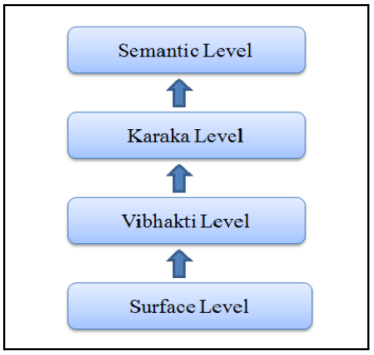
- In English, the long distance movement for complement clause can be explained by bounding theory if NP and S are taken to be bounding nodes.
- The theory says that the application of move α may not cross more than one bounding node.
- The theory of control involves syntax, semantics and pragmatics.

Case Theory and Case Filter

Case Theory:

In GB, case theory deals with the distribution of NPs and mentions that each NP must be assigned a case. In English, we have **Nominative, Objective, Genitive, etc.,** cases which are assigned to NPs at particular positions.

Case Filter:

	An NP is ungrammatical if it has phonetic content or if it is an argument and is not case marked. **Case filters** restrict the movement of NP at a position which has no case assignment.
3 (a)	Describe the Paninian framework for Indian languages. Explain the layered representation of Paninian Grammar and Karaka theory. Paninian grammar (PG) was written by Panini in 500 BC in Sanskrit. PG framework can be used for other Indian languages and even for some Asian languages. Unlike English, Asian languages are SOV (Subject-Object-Verb) ordered and inflectionally rich. The inflections provide important syntactic and semantic for language analysis and understanding. The classical Paninian Grammar facilitates the task of obtaining the semantics through syntactical framework. In PG, an extensive and perfect interpretation of Phonology, Morphology, Syntax, and Semantics is available. **Layered representation in panini grammar**: Paninian Grammar (PG) framework is said to be **syntactico–semantic that is one can go from the surface layer to deep semantics by passing through immediate layers.** PG works on various levels of language analysis to achieve the meaning of the sentence from the hearer's perspective. To achieve the desired meaning, the grammar analysis is divided itself internally into various levels as shown in the figure below.  fig: Layered Representation in PG **Semantic Level:** • Represents the speaker's actual intention, that is, his real thought for the sentence. **Surface Level:** • Surface level is the actual string or the sentence. It captures the written or the spoken sentences as it is. **Vibhakti Level:** • Vibhakti is the word suffix, which helps to find out the participants, gender as well as form of the word. • Vibhakti level is purely syntactic. At this level, the case endings are used to form the local groups of the words. At the Vibhakti level, a noun is formed containing a noun, which contains the instances of noun or pronoun, etc. • Vibhakti for verbs includes the verb form and the auxiliary verbs. • Vibhakti gives Tense Aspect Modality details of the word which is popularly known as TAM. **Karaka Level:** • At the Karaka level, the relation of the participant noun, in the action, to the verb is determined. • Karaka relations are Syntactico-semantic. • These relations are established in between the verb and other constituent nouns that are present in the sentences. Through, these relations, the Karakas try to capture the information from the semantics of the texts. • Kakara level processes the semantics of the language but represents it at the syntactic level. Hence it acts as a bridge between semantic and syntactic analysis of a language. **Karaka Theory:** • The etymological meaning of the word Karaka is 'one who does something', i.e. one who performs an action. • The Karaka and the Kriya, i.e. the cases and verb are bounded with the sense of mutual requirement. • The one who performs an action, accepts an action, or otherwise helps to perform an action is known as a Karaka. • There is a mutual expectancy in between the action i.e. Kriya and the adjuncts i.e. Karaka. • The presence of one calls for the existence of the other. In other words Kirya and Karaka are

mutually exclusive.

Various Karakas -

1. Karta - subject
2. Karma - object
3. Karana - instrument
4. Sampradhana - beneficiary
5. Apadana - separation
6. Adhara or Adhikarana - locus
7. Sambandh - relation
8. Tadarthya - purpose

The Karta, Karma, and Karana are considered the foremost Karakas while Sampradana, Apadana, and Adhikarana Karakas are known as the influenced Karakas.

Karta Karaka:

The Karta Karaka is the premier one according to action and it is used to perform an action independently of its own. An action indicated in a sentence is entirely dependent upon the Karta-Karaka. Activity either resides in or rises from the Karta only.

Eg. Tiger killed the goat

Tiger - karta.

Karma Karaka:

Karma is the Aashraya (locus) of the result. The most desirable nominative by the Karta Karaka is the Karma Karaka. When the Karta carries out any activity, the result of that activity rests in the Karma. As the Karma (object) is the basis of outcome of the primary action, it is one of the important prominent Karaka.

Eg. Tiger killed the goat.

goat is karma.

Karana Karaka:

Karaka which helps in carrying out the Kriya is known as Karana Karaka. Karana Karaka helps in attaining the desired result ascertained by the Kriya. Karana is the most important tool by means of which the action is achieved. Karana is the only direct participant in the action. Karta and Karma also are directly dependent on the Karana for performing the action. When the Karana Karaka executes its auxiliary actions then only the main action is executed by the Karta Karaka. This is why the Karana is considered as the efficient mean in action accomplishment.

Example:

The man cut the wood **with** an axe

Ram cuts the apple **with** knife.

karana - axe and knife

Sampradhana Karaka:

The word Sampradana can be interpreted as ‘He to whom something is given properly’. Sampradana Karaka receives or gets benefited from the action. It can also be said that, the person/object for which the Karma is intentional, is known as Sampradana. In this regard, the Sampradana is the final destination of the action.

Example:

Dipti gave chocolates to Shambhavi

Shambhavi is sampradhana.

Ram gave me a book.

me is sampradhana

He gave flowers for Shanbhavi

Shambhavi is sampradhana

Apadana Karaka:

About Apadana Karaka Panini stated that, as when separation is affected by a verbal action, the point of separation is called Apadana. During the execution of the action whenever the task of separation from a certain entity is executed then **whatever remains unmoved or constant is known as Apadana.** Thus, an Apadana denotes the starting point of an action of separation. The entity from which something gets separated or is separated out is known as Apadana.

Example:

Shambhavi tore the page from the book with a scissor.

From the book is apadana

Adhikarana Karaka:

'Adhikarana' is the place or thing, which is the location of the action existing in the agent or the object. Adhikarana is assigned to the locus of the action i.e. Kriya. Adhikarana may indicate the place at which the Kriya (the action) is taking place or the time at which the Kriya is carried out. Any action i.e. the Kriya is either bounded by space (place) or by time.

Example:

'Yesterday Shambhavi hit the dog with the stick in front of the shop.'

The Karaka annotation of the above sentence can be given as

hit	: verb (root)
Yesterday	: Kala-Adhikarana (time)
Shambhavi	: Agent i.e. Karta
Dog	: Karma
Stick	: Karana
Shop	: Desh-Adhikarana (location i.e. Place)

(b) Identify different karaka's in the following Hindi sentence "Maan putree ko angan mein haath se roti khilaathi hai"

Karaka role:

khilaathi - Verb (root)
Maan - Karta
roti - Karma
haath - Karana
putree - Sampradhana
angan - Adhikaran

4 Explain Statistical language model and find the probability of the test sentence – P ("They play in a big garden") in the following training set using the bi-gram model

<S>There is a big garden.
Children play in the garden.
They play inside beautiful garden. </S>

A statistical language model is a probability distribution P(s) over all possible word sequences (or any other linguistic unit like words, sentences, paragraphs, documents, or spoken utterances). A number of statistical language models have been proposed in the literature. The dominant approach in modeling is the n-gram model:

n-gram Model:

The goal of a statistical language model is to estimate the probability of a sentence. This is achieved by decomposing sentence probability into a product of conditional probabilities using the chain rule as follows:

$P(s) = P(w_1, w_2, w_3, \ldots, w_n)$

$= P(w_1)\, P(w_2/w_1)\, P(w_3/w_1\, w_2)\, P(w_4/w_1\, w_2\, w_3) \ldots P(w_n/\, w_1\, w_2 \ldots w_{n-1})$

$$= \prod_{i=1}^{n} P(w_i / h_i)$$

where h_i is the history of the word w_i defined as

$w_1\, w_2 \ldots w_{i-1}$

To find the sentence probability, we need to calculate the probability of a word, given the sequence of words preceding it. The n-gram model simplifies the task by approximating the probability of a word given all the previous words by the conditional probability given previous n-1 words only. An n-gram model calculates the probability of a word by modeling language as the Markov model of order n-1, that is, by looking at previous n-1 words only. A model that limits the history to the previous one word only is termed a bi-gram (n = 1) model. A model that conditions the probability of a word to the previous two words, is called a tri-gram (n = 2) model. Using bi-gram and tri-gram estimates, the probability of a sentence can be calculated as

$$P(s) \cong \prod_{i=1}^{n} P(w_i / w_{i-1})$$

and

$$P(s) \cong \prod_{i=1}^{n} P(w_i / w_{i-2}\ w_{i-1})$$

A special word (pseudoword) <s> is introduced to mark the beginning of the sentence in bi-gram estimation. The probability of the first word in the sentence is conditioned on <s>. Similarly, in tri-gram estimation, two pseudo words <s1> <s2> are introduced.

Estimation of probabilities is done by training the n-gram model on the training corpus. n-gram parameters are estimated using the Maximum Likelihood Estimation (MLE) technique, that is, using relative frequencies. We count a particular n-gram in the training corpus and divide it by the sum of all n-grams that share the same prefix.

$$P(w_i / w_{i-n+1}, \ldots, w_{i-1}) = \frac{C(w_{i-n+1}, \ldots, w_{i-1}, w_i)}{C(w_{i-n+1}, \ldots, w_{i-1})}$$

Find the probability of the test sentence – P ("They play in a big garden") in the following training set using the bi-gram model

<S>There is a big garden.
Children play in the garden.
They play inside beautiful garden. </S>

<S> There is a big garden. </S>
<S> Children play in the garden. </S>
<S> They play inside a beautiful garden. </S>

Training Set:

<S> There is a big garden. </S>
<S> Children play in the garden. </S>
<S> They play inside a beautiful garden. </S>

Test Sentence:

They play in the big garden

Bi-gram model:

P (They play in the big garden)
= P (They / <S>) *
P (play / they) *
P (in / play) *
P (the / in) *
P (big / the) *
P (garden / big)

= ⅓ * 1/1 * ½ * 1/1 * 0/1 * 1/1
= 0

Applying Add-one Smoothing:

Number of unique words in the training set, V = 12
P (They play in the big garden)
= (1+1) / (3 +12) * (1+1) / (1+12) * (1+1) / (2+12) * (1+1) /(1+12) * (0+1) / (1+12) * (1+1) / (1+12)
= 2/15 * 2/13 * 2/14 * 2/13 * 1/13 * 2/13
= 0.133 * 0.154 * 0.143 * 0.154 * 0.077 * 0.154

	= $5.348 * 10^{-6}$

5	**Design CYK algorithm. Tabulate the sequence of states created by CYK algorithm while parsing the sentence, "A pilot likes flying planes". Consider the following simplified grammar in CNF**
	S -> NP VP **NN -> Pilot** **VBG -> flying**
	NP -> DT NN **NNS -> planes** **JJ -> flying**
	NP -> JJ NNS **VP -> VBG NNS** **Det -> a**
	VP -> VBZ NP **VBZ -> likes**
	
	The CYK Parser:
	The CYK (Cocke-Younger-Kasami) algorithm is a dynamic programming parsing algorithm. It follows a bottom-up approach in parsing. It builds a parse tree incrementally. Each entry in the table is based on previous entries. The process is iterated until the entire sentence is parsed.
	The CYK parsing algorithm assumes the grammar to be in Chomsky normal form (CNF). A CFG is in CNF if all the rules are of only two forms:
	A -> BC
	A -> w, where w is a word.
	
	Let $w = w_1\ w_2\ w_3\ w_i \ldots w_j \ldots w_n$
	and $w_{ij} = w_i \ldots w_{i+j-1}$
	// Initialization step
	for $i := 1$ to n do
	for all rules $A \rightarrow w_i$ do
	chart $[i,1] = \{A\}$
	// Recursive step
	for $j = 2$ to n do
	for $i = 1$ to $n-j+1$ do
	begin
	chart $[i, j] = \phi$
	for $k = 1$ to $j-1$ do
	chart $[i, j] :=$ chart$[i, j] \cup \{A \mid A \rightarrow BC$ is a production and $B \in$ chart$[i, k]$ and $C \in$ chart $[i+k, j-k]\}$
	end
	if $S \in$ chart$[1, n]$ then accept else reject
	
	Figure 4.12 The CYK algorithm

Sentence: A pilot likes flying planes

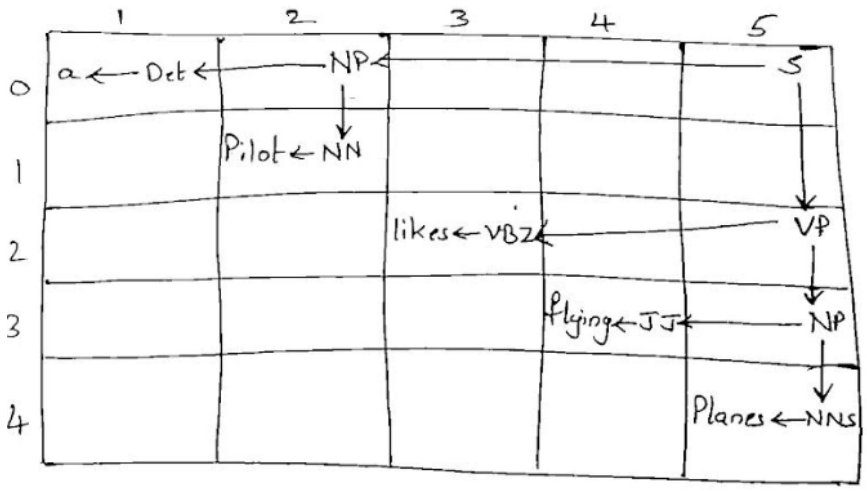
CNF Rules:

S → NP VP
NP → DT NN
NP → JJ NNS
VP → VBZ NP
VP → VBG NNS

NN → Pilot
NNS → Planes
VBZ → likes
VBG → flying
JJ → flying
Det → a

0 A 1 pilot 2 likes 3 flying 4 planes 5

Matrix: 5 × 5



The above table shows the sequence of states created in the chart by the CYK algorithm while parsing the sentence – "A pilot like flying planes".

The table contains entries after a complete scan of the algorithm. The entry in the [0,n]th cell contains a start symbol which indicates that the parse is successful.

6 **Write and explain an algorithm for minimum edit distance correction. Apply the same to find the minimum edit distance between the words 'PEACEFUL' and 'PAECFLU'.**

- The minimum edit distance is the number of insertions, deletions, and substitutions required to change one string into another. For example, the minimum edit distance between the **tutor** and the **tumour** is 2. We substitute 'm' for 't' and insert 'u' before 'r'. No more minor edit sequences can be found for this conversion.

- Edit distance between two strings can be represented as a binary function, **ed**, which maps two strings to their edit distance. **ed is symmetric.** For any two strings s and t, ed (s, t) always equals ed (t, s).

- **Edit distance can be viewed as a string alignment problem**. By aligning two strings, we can measure the degree to which they match. There can be more than one possible alignment between two strings. **The best possible alignment corresponds to the minimum edit distance between the strings.**

- The alignment between Tutor and Tumour has a distance of 2.

  T U **T** O - R

  T U **M** O U R

- A dash in the upper string indicates insertion. A substitution occurs when the two alignment symbols do not match.
- Minimum edit distance has different algorithms namely, the **Levenshtein algorithm, Hamming algorithm, and Longest Common Subsequence algorithm.**
- **Dynamic programming algorithm is used for finding minimum edit distance between two sequences**.
- The dynamic programming algorithm for minimum edit distance is implemented by **creating an edit distance matrix**.
- Edit distance matrix has one row for each symbol in source string and one column for each symbol in the target string.
- The $(i,j)^{th}$ cell in this matrix represents the distance between the first i character of the source and the first j character of the target string.
- The value in each cell is computed in terms of three possible paths.

$$dist[i,j] = \begin{cases} dist[i-1,j] + insert\_cost, \\ dist[i-1,j-1] + subst\_cost[source_i, target_j] \\ dist[i,j-1] + delete\_cost \end{cases}$$

- The substitution will be 0 if the i^{th} character in the source matches with the j^{th} character in the target.
- Minimum edit distance algorithm is shown below –

```
Input: Two strings, X and Y
Output: The minimum edit distance between X and Y
m ← length(X)
n ← length(Y)
for i = 0 to m do
    dist[i,0] ← i
for j = 0 to n do
    dist[0,j] ← j
for i = 0 to m do
    for j = 0 to n do
        dist[i,j] = min{ dist[i–1,j] + insert_cost,
                         dist[i–1,j–1] + subst_cost(Xi, Yj),
                         dist[i,j–1] + delet_cost }
```

Figure 3.13 Minimum edit distance algorithm

	#	p	e	a	c	e	f	u	l
#	0	1	2	3	4	5	6	7	8
p	1	0	1	2	3	4	5	6	7
a	2	1	1	1	2	3	4	5	6
e	3	2	1	2	2	2	3	4	5
c	4	3	2	2	2	3	3	4	5
f	5	4	3	3	3	3	3	4	5
l	6	5	4	4	4	4	4	4	4
u	7	6	5	5	5	5	5	4	5

Minimum Edit Distance between PEACEFUL and PAECFLU = 5