


# NLP Solutions and Scheme for IAT-1

145

USN ICR22AI093

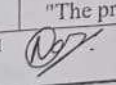
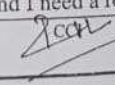
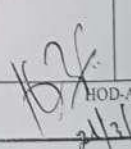


Internal Assessment Test 1 – March 2025

Sub	Natural Language Processing	Sub Code:	BAI601	Branch:	AIML
Date:	24/03/2025	Duration:	90 mins	Max Marks:	50
		Sem / Sec:	VI / A & B		OBE

Answer Any of 5 Questions

		MARKS	CO	RET																																																							
1a)	Explain various levels of Natural Language Processing with suitable examples.	[10]	CO1	L2																																																							
b)	Describe Paninian framework for Indian languages.																																																										
c)	Explain Transformational Grammar with Examples																																																										
2 (a)	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>	[10]	CO1	L3																																																							
b)	List the problems associated with the n-gram model. Explain how these problems are handled.																																																										
3	Calculate the Minimum Edit Distance Algorithm by given strings "ELEPHANT", "RELEVANT" and write the Algorithmic steps.	[10]	CO2	L3																																																							
4 (a)	Explain the following with suitable example 1) Xbar-theory 2) Theta Theory	[06]	CO1	L2																																																							
(b)	Define morphology. Explain Stem and Affix classes of morphemes with examples.	[04]	CO2	L2																																																							
5.	What is POS tagging? List and explain different taggers with Suitable Examples.	[10]	CO2	L2																																																							
6 a)	Explain Naive Bayes classifier and types. Classify Vehicle based on features following Stolen=yes,colour=yellow,Type=SUV,origin =domestic	[5]	CO3	L3																																																							
	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>Example No.</th> <th>Color</th> <th>Type</th> <th>Origin</th> <th>Stolen?</th> </tr> </thead> <tbody> <tr><td>1</td><td>Red</td><td>Sports</td><td>Domestic</td><td>Yes</td></tr> <tr><td>2</td><td>Red</td><td>Sports</td><td>Domestic</td><td>No</td></tr> <tr><td>3</td><td>Red</td><td>Sports</td><td>Domestic</td><td>Yes</td></tr> <tr><td>4</td><td>Yellow</td><td>Sports</td><td>Domestic</td><td>No</td></tr> <tr><td>5</td><td>Yellow</td><td>Sports</td><td>Imported</td><td>Yes</td></tr> <tr><td>6</td><td>Yellow</td><td>SUV</td><td>Imported</td><td>No</td></tr> <tr><td>7</td><td>Yellow</td><td>SUV</td><td>Imported</td><td>Yes</td></tr> <tr><td>8</td><td>Yellow</td><td>SUV</td><td>Domestic</td><td>No</td></tr> <tr><td>9</td><td>Red</td><td>SUV</td><td>Imported</td><td>No</td></tr> <tr><td>10</td><td>Red</td><td>Sports</td><td>Imported</td><td>Yes</td></tr> </tbody> </table>	Example No.	Color	Type	Origin	Stolen?	1	Red	Sports	Domestic	Yes	2	Red	Sports	Domestic	No	3	Red	Sports	Domestic	Yes	4	Yellow	Sports	Domestic	No	5	Yellow	Sports	Imported	Yes	6	Yellow	SUV	Imported	No	7	Yellow	SUV	Imported	Yes	8	Yellow	SUV	Domestic	No	9	Red	SUV	Imported	No	10	Red	Sports	Imported	Yes			
Example No.	Color	Type	Origin	Stolen?																																																							
1	Red	Sports	Domestic	Yes																																																							
2	Red	Sports	Domestic	No																																																							
3	Red	Sports	Domestic	Yes																																																							
4	Yellow	Sports	Domestic	No																																																							
5	Yellow	Sports	Imported	Yes																																																							
6	Yellow	SUV	Imported	No																																																							
7	Yellow	SUV	Imported	Yes																																																							
8	Yellow	SUV	Domestic	No																																																							
9	Red	SUV	Imported	No																																																							
10	Red	Sports	Imported	Yes																																																							
b)	Write steps to ensure that the model accurately captures and interprets the emotions and opinions expressed in text data of the Customer Support dataset. messages = [ "I am extremely unhappy with the service I received.", "Thank you for the excellent support!", "The product arrived damaged and I need a replacement."]	[5]	CO3	L2																																																							

HOD-AIML  
24/3/2025

## Scheme

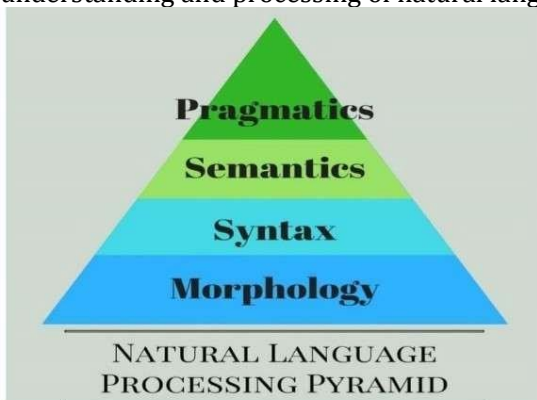
1a)	Explain various levels of Natural Language Processing with suitable examples. Descriptions-2 Examples-2	10	CO1	L2
b)	Describe Paninian framework for Indian languages. Descriptions-1 Examples-1 Drawings-1			
c)	Explain Transformational Grammar with Examples Descriptions-1 Examples-1 Drawings-1			
2 (a)	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> Descriptions-3 Steps- 2	10	CO1	L3
b)	List the problems associated with the n-gram model. Explain how these problems are handled. Descriptions-2 Steps- 2 Examples-1			
3	Calculate the Minimum Edit Distance Algorithm by given strings “ELEPHANT” , “RELEVANT” and write the Algorithmic steps. Algorithm- 4 Steps-4 Examples-2	[10]	CO2	L3
4 (a)	Explain the following with suitable example 1.Xbar -theory 2) Theta Theory 1-Descriptions-3 2-Descriptions-3	[06]	CO1	L2
(b)	Define morphology. Explain Stem and Affix classes of morphemes with examples. 1-Descriptions-2 2-Descriptions-2	[04]	CO2	L2

5.	What is POS tagging? List and explain different taggers with Suitable Examples. 1-Descriptions-2 Types-3 2-Descriptions-3 Examples-2	[10]	CO2	L2																																																							
6 a)	Explain Naive Bayes classifier and types. Classify Vehicle based on features following Stolen=yes,colour=yellow,Type=SUV,origin =domestic  <table border="1"> <thead> <tr> <th>Example No.</th><th>Color</th><th>Type</th><th>Origin</th><th>Stolen?</th></tr> </thead> <tbody> <tr><td>1</td><td>Red</td><td>Sports</td><td>Domestic</td><td>Yes</td></tr> <tr><td>2</td><td>Red</td><td>Sports</td><td>Domestic</td><td>No</td></tr> <tr><td>3</td><td>Red</td><td>Sports</td><td>Domestic</td><td>Yes</td></tr> <tr><td>4</td><td>Yellow</td><td>Sports</td><td>Domestic</td><td>No</td></tr> <tr><td>5</td><td>Yellow</td><td>Sports</td><td>Imported</td><td>Yes</td></tr> <tr><td>6</td><td>Yellow</td><td>SUV</td><td>Imported</td><td>No</td></tr> <tr><td>7</td><td>Yellow</td><td>SUV</td><td>Imported</td><td>Yes</td></tr> <tr><td>8</td><td>Yellow</td><td>SUV</td><td>Domestic</td><td>No</td></tr> <tr><td>9</td><td>Red</td><td>SUV</td><td>Imported</td><td>No</td></tr> <tr><td>10</td><td>Red</td><td>Sports</td><td>Imported</td><td>Yes</td></tr> </tbody> </table> 1.Descriptions- 3 Calculation-2	Example No.	Color	Type	Origin	Stolen?	1	Red	Sports	Domestic	Yes	2	Red	Sports	Domestic	No	3	Red	Sports	Domestic	Yes	4	Yellow	Sports	Domestic	No	5	Yellow	Sports	Imported	Yes	6	Yellow	SUV	Imported	No	7	Yellow	SUV	Imported	Yes	8	Yellow	SUV	Domestic	No	9	Red	SUV	Imported	No	10	Red	Sports	Imported	Yes	[5]	CO3	L3
Example No.	Color	Type	Origin	Stolen?																																																							
1	Red	Sports	Domestic	Yes																																																							
2	Red	Sports	Domestic	No																																																							
3	Red	Sports	Domestic	Yes																																																							
4	Yellow	Sports	Domestic	No																																																							
5	Yellow	Sports	Imported	Yes																																																							
6	Yellow	SUV	Imported	No																																																							
7	Yellow	SUV	Imported	Yes																																																							
8	Yellow	SUV	Domestic	No																																																							
9	Red	SUV	Imported	No																																																							
10	Red	Sports	Imported	Yes																																																							
b)	Write steps to ensure that the model accurately captures and interprets the emotions and opinions expressed in text data of the Customer Support dataset. messages = [ "I am extremely unhappy with the service I received.", "Thank you for the excellent support!", "The product arrived damaged and I need a replacement."] 1.Descriptions-3 Calculation-2	[5]	CO3	L2																																																							

## Solutions

1a. Explain various levels of Natural Language Processing with Examples.

Natural Language Processing (NLP) is a field within artificial intelligence that allows computers to comprehend, analyse, and interact with human language effectively. The process of NLP can be divided into **five distinct phases: Lexical Analysis, Syntactic Analysis, Semantic Analysis, Discourse Integration, and Pragmatic Analysis**. Each phase plays a crucial role in the overall understanding and processing of natural language.



### 1. Lexical and Morphological Analysis

- Involves breaking text into words or tokens (**Tokenization**).
- Removes unnecessary words (**Stop-word Removal**), converts words to root form (**Lemmatization**), and corrects spellings.
- **Morphological Analysis** studies the structure of words, distinguishing root words and suffixes (e.g., "running" → "run").

## 2. Syntactic Analysis (Parsing)

- Examines the grammatical structure of sentences.
- Uses **Parsing** and **Part-of-Speech (POS) tagging** to identify noun, verb, adjective, etc.
- Example:
  - "She is playing football." (Correct Syntax)
  - "Playing is she football." (Incorrect Syntax)

## 3. Semantic Analysis

- Extracts the **meaning** of words and sentences.
- Resolves **Word Sense Disambiguation (WSD)** (e.g., "bank" as a financial institution vs. riverbank).
- Identifies **Named Entities** like people, places, and organizations (**Named Entity Recognition - NER**).

## 4. Discourse Integration

- Ensures **contextual understanding** across multiple sentences.
- Example: "John bought a car. He is excited." → "He" refers to John.

## 5. Pragmatic Analysis

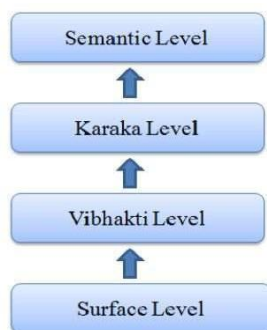
- Analyses **implied meaning** beyond literal words.
- Example: "Can you open the door?" → A request, not a yes/no question.

1b. Describe Paninian framework for Indian Languages

### Paninian Framework for Indian Languages

The **Paninian framework** is a linguistic model based on **Panini's Ashtadhyayi**, an ancient Sanskrit grammar system. It provides a structured method for analyzing Indian languages, emphasizing **morphological, syntactic, and semantic relationships**. This framework is highly relevant in **Natural Language Processing (NLP)** due to its rule-based approach to

word formation and sentence structure.



### Layered Representation of Paninian Grammar

Paninian grammar follows a **layered approach**, organizing linguistic information at different levels:

1. **Phonological Layer:** Deals with sound patterns and pronunciation.
2. **Morphological Layer:** Focuses on word formation rules, such as suffixes and prefixes.
3. **Syntactic Layer:** Defines sentence structure using verb-centered rules.
4. **Semantic Layer:** Establishes meaning relations between words, primarily through **Karaka theory**.

1c. Explain Transformational Grammar with Examples.

### Transformational Grammar

Transformational Grammar (TG), proposed by **Noam Chomsky**, is a theory that explains how sentences in a language are generated and transformed. It describes the relationship between deep structure (basic sentence form) and surface structure (final sentence form after transformations).

Transformational Grammar has 3 Components

- Phrase Structure Grammar
- Transformational Rules
- Morphophonemic rules

### 1. Phrase Structure Grammar

Phrase Structure Grammar provides the **basic structure** of sentences using a set of rules that define how words and phrases are combined. It helps break down a sentence into **constituents** (phrases and sub-phrases).

**Example of Phrase Structure Rules:**

- **S** → **NP VP** (*A sentence consists of a Noun Phrase and a Verb Phrase.*)
- **NP** → **Det N** (*A noun phrase consists of a determiner and a noun.*)
- **VP** → **V NP** (*A verb phrase consists of a verb and a noun phrase.*)

**Example Sentence Breakdown:**

For the sentence "**The boy eats an apple**", the structure would be:

S → NP VP

NP → Det N (The boy)

VP → V NP (eats an apple)

This represents the **deep structure** of a sentence.

### 2. Transformational Rules

These rules modify the deep structure to generate different sentence forms like **questions, negations, passives, and complex sentences**.

These transformations change the structure of a sentence **without altering its fundamental meaning**.

### 3. Morphophonemic Rules

Morphophonemic rules deal with **word formation and pronunciation changes** that occur when morphemes (smallest units of meaning) combine. These rules help in **correctly forming and pronouncing words**.

**Examples:**

#### 1. Plural Formation:

- o cat + -s → cats
- o dog + -s → dogs

**2a. Explain the statistical language model and find the probability of the test sentence**

**P(They play in the 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 a beautiful garden. </S>**

$P(\text{They play in the garden})$

Convert the given sentence into unigram and bigram.  
and find the respective frequencies of each in the given corpus.

Sentence =  $\langle s \rangle + \text{root} + \langle /s \rangle$   
 $= \langle s \rangle \text{ they play in the garden } \langle /s \rangle$

unigrams and frequencies	Bigrams and their frequencies
$\langle s \rangle = 3$	$(\langle s \rangle, \text{they}) = 1$
$\text{they} = 1$	$(\text{they}, \text{play}) = 1$
$\text{play} = 1$	$(\text{play}, \text{in}) = 1$
$\text{in} = 1$	$(\text{in}, \text{the}) = 1$
$\text{the} = 1$	$(\text{the}, \text{big}) = 1$
$\text{big} = 1$	$(\text{big}, \text{garden}) = 0$ (smoothing required)
$\text{garden} = 3$	$(\text{garden}, \langle /s \rangle) = 3$
$\langle /s \rangle = 3$	

length of vocabulary =  $|V| = 14$

$P(\text{Sentence given}) = P(\text{they} | \langle s \rangle) \times P(\text{play} | \text{they}) \times P(\text{in} | \text{play})$   
 $\times P(\text{the} | \text{in}) \times P(\text{big} | \text{the}) \times P(\text{garden} | \text{big})$   
 $\times P(\langle /s \rangle | \text{garden})$

we know  $P(\text{class} | w) = \frac{\text{count}(w, \text{class}) + 1}{\text{count}(w) + |V|}$  (smoothing)

$$= \left( \frac{1+1}{3+14} \right) \left( \frac{1+1}{1+14} \right) \left( \frac{1+1}{1+14} \right) \left( \frac{1+1}{1+14} \right) \left( \frac{0+1}{1+14} \right) \left( \frac{3+1}{3+14} \right)$$

$$= 5.467 \times 10^{-7}$$

2b. List the problems associated with the n-gram models. Explain how these problems are handled.

n-gram models are widely used in natural language processing (NLP) for predicting the next word in a sequence. However, they suffer from several limitations:

### 1. Data Sparsity

- As  $n$  increases, the number of possible  $n$ -grams grows exponentially, leading to many unseen sequences in the training data.
- Example: If a bigram ("deep learning") exists in training but ("deep study") does not, the model struggles to generate it.

### 2. Context Limitation

- $n$ -gram models consider only a fixed-length context, missing long-range dependencies in language.
- Example: In a sentence like "The cat that the dog chased ran away," a trigram model would not capture dependencies between "cat" and "ran."

### 3. High Computational Cost

- Requires large storage for  $n$ -gram counts and probabilities.
- Increasing  $n$  leads to exponential growth in memory requirements.

### 4. Generalization Issues

- Cannot handle new words or phrases effectively.
- Works well only on words seen during training.

## How Transformers Address These Problems

Transformers, such as **BERT** and **GPT**, overcome these limitations using attention mechanisms and deep learning techniques.

### 1. Handling Data Sparsity

- Transformers use embeddings to represent words in a continuous vector space, making it easier to generalize across similar words.
- Example: The word "study" can be inferred to be similar to "learning" based on vector similarities.

### 2. Capturing Long-Range Dependencies

- The **self-attention mechanism** allows transformers to consider all words in a sequence rather than just a fixed window.
- Example: In "*The cat that the dog chased ran away,*" the model can correctly associate "cat" with "ran."

### 3. Efficient Computation with Parallelism

- Unlike n-gram models that require sequential processing, transformers use **multi-head attention**, allowing parallel computations.
- This speeds up training and inference.

### 4. Generalization and Handling New Words

- Transformers use **subword tokenization (Byte Pair Encoding - BPE, WordPiece, etc.)**, enabling them to process new or rare words.
- Example: The word "unhappiness" might be split into "un", "happi", and "ness," allowing better generalization.

3. Calculate the minimum edit distance algorithms by given strings "ELEPHANT", "RELEVANT" and write the algorithmic steps.

The **minimum edit distance** between two strings is the minimum number of editing operations required to Insert, delete and update/substitute to convert one string to another string

The algorithm follows **Dynamic Programming (DP)** by filling a matrix  $D[i][j]$ , where:

- $i$  represents the characters in the source word.
- $j$  represents the characters in the target word.

The recurrence relation used:

$$D(i, j) = \min \begin{cases} D(i-1, j) + 1 & \text{(Deletion)} \\ D(i, j-1) + 1 & \text{(Insertion)} \\ D(i-1, j-1) + \text{cost} & \text{(Substitution, cost = 0 if same, else 1)} \end{cases}$$

where  $D(i-1, j)$ ,  $D(i, j-1)$ , and  $D(i-1, j-1)$  represent previous computations.

```

import numpy as np
def min_edit(text1, text2, m, n):
    dp = np.zeros((m + 1, n + 1), dtype=int)

    for i in range(m + 1):
        dp[i][0] = i

    for j in range(n + 1):
        dp[0][j] = j

    for i in range(1, m + 1):
        for j in range(1, n + 1):
            if text1[i - 1] == text2[j - 1]:
                dp[i][j] = dp[i - 1][j - 1]
            else:
                dp[i][j] = min(
                    dp[i][j - 1] + 1,
                    dp[i - 1][j] + 1,
                    dp[i - 1][j - 1] + 1
                )

    print(dp)

    return dp[m][n]

text1 = input("Enter first string: ")
text2 = input("Enter second string: ")
print("Minimum edit distance:", min_edit(text1, text2, len(text1), len(text2)))

```

Given words are

ELEPHANT

RELEVANT

```

Enter first string: ELEPHANT
Enter second string: RELEVANT
[[0 1 2 3 4 5 6 7 8]
 [1 1 1 2 3 4 5 6 7]
 [2 2 2 1 2 3 4 5 6]
 [3 3 2 2 1 2 3 4 5]
 [4 4 3 3 2 2 3 4 5]
 [5 5 4 4 3 3 3 4 5]
 [6 6 5 5 4 4 3 4 5]
 [7 7 6 6 5 5 4 3 4]
 [8 8 7 7 6 6 5 4 3]]
Minimum edit distance: 3

```



4a. Explain the following with suitable example.

- (i) X-bar theory
- (ii) Theta theory

X-bar theory is a principle in **syntactic theory** (a part of **Natural Language Processing (NLP)** and **linguistics**) that describes the structure of phrases in sentences. It is a **universal framework** for phrase structure, ensuring that all languages follow a similar pattern in sentence construction.

#### **Core Idea of X-Bar Theory**

X-bar theory proposes that **all phrases** in human languages follow a hierarchical structure with three main levels:

1. **X (Head)** → The core of the phrase (e.g., a noun in a noun phrase).
2. **X' (X-bar)** → An intermediate level that contains the head and optional modifiers.
3. **XP (Maximal Projection)** → The full phrase that includes all necessary components.

#### **Structure of a Phrase in X-Bar Theory**

A phrase (XP) consists of:

- **Head (X)**: The central element that determines the phrase type (e.g., Noun for a **Noun Phrase (NP)**, Verb for a **Verb Phrase (VP)**).
- **Specifier**: An optional element that provides additional information (e.g., determiners like *the, a*).
- **Complement**: A phrase that completes the meaning of the head (e.g., *an apple* in *eat an apple*).
- **Adjunct**: An optional modifier that adds extra information (e.g., *quickly* in *eat quickly*).

#### **Example in English**

##### **1. Noun Phrase (NP)**

o "The big cat"

o Structure:

♣ **Head**: "cat" (N)

♣ **Specifier**: "the"

♣ **Adjunct**: "big"

##### **2. Verb Phrase (VP)**

o "ate an apple"

o Structure:

♣ **Head**: "ate" (V)

♣ **Complement**: "an apple".

## **Theta Theory (θ-Theory) in Syntax and NLP**

Theta Theory (θ-Theory) is a principle in syntax, particularly in Generative Grammar, that deals with the assignment of thematic roles (θ-roles) to different sentence elements. It ensures that verbs assign specific roles to their arguments, helping to explain sentence structure and meaning.

---

### **Core Idea of Theta Theory**

Theta Theory states that verbs (or predicates) impose specific roles on the arguments they take, ensuring a structured relationship between the verb and its dependents. Each argument must receive exactly one theta role, and each theta role must be assigned to one argument.

---

## Key Components of Theta Theory

### 1. Theta Roles ( $\theta$ -Roles)

Theta roles define the semantic relationship between the predicate (usually a verb) and its arguments. Common  $\theta$ -roles include:

- **Agent:** The doer of the action (e.g., *John* in *John kicked the ball*).
- **Theme (Patient):** The entity undergoing the action (e.g., *the ball* in *John kicked the ball*).
- **Experiencer:** The entity experiencing a state or emotion (e.g., *Mary* in *Mary felt happy*).
- **Goal:** The target or destination of an action (e.g., *the park* in *He went to the park*).
- **Source:** The starting point of an action (e.g., *New York* in *She flew from New York*).
- **Instrument:** The means by which an action is performed (e.g., *a knife* in *He cut the paper with a knife*).

### 2. Theta Criterion

The Theta Criterion states that:

- Each argument in a sentence must receive exactly one  $\theta$ -role.
- Each  $\theta$ -role must be assigned to only one argument.

### 3. Verb Argument Structure

Different verbs require different numbers of arguments (Valency):

- **Intransitive verbs** (e.g., *sleep*): Require only one argument (*John sleeps* – Agent).
- **Transitive verbs** (e.g., *kick*): Require two arguments (*John kicked the ball* – Agent, Theme).
- **Ditransitive verbs** (e.g., *give*): Require three arguments (*She gave him a book* – Agent, Goal, Theme).

---

## Example Analysis

### 1. Simple Sentence

- Sentence: *John gave Mary a gift.*
- Verb: *gave* (ditransitive verb)
- Theta Roles:
  - *John* → **Agent** (who performs the action)
  - *Mary* → **Goal** (who receives the gift)
  - *a gift* → **Theme** (the object being given)

### 2. Intransitive Verb Example

- Sentence: *Tom sleeps.*
- Verb: *sleeps* (intransitive)
- Theta Role:
  - *Tom* → **Experiencer**

4b. Define morphology. Explain stem and affix classes of morphemes with examples

Morphology is the branch of linguistics that studies the structure and formation of words. It examines how words are formed using **morphemes**, the smallest units of meaning in a language. Morphology plays a vital role in **Natural Language Processing (NLP)** for tasks like **text analysis, machine translation, and speech recognition**.

---

## Morphemes: The Building Blocks of Words

A **morpheme** is the smallest unit of meaning in a word. Morphemes can be classified into **stems** and **affixes** based on their function in word formation.

---

### 1. Stem

The **stem** (also called the **root**) is the core part of a word that carries its basic meaning. It is the fundamental unit to which **affixes** can be added.

◆ **Example:**

- **Happy** (stem) → **unhappy** (prefix + stem)
- **Play** (stem) → **replaying** (prefix + stem + suffix)

The stem remains unchanged in meaning even when affixes are added.

---

### 2. Affix

An **affix** is a morpheme attached to a stem to modify its meaning or grammatical function. Affixes are divided into four major types:

#### a) Prefix (Added before the stem)

A **prefix** is an affix that appears at the **beginning** of a word.

◆ **Example:**

- **Un-** + *happy* → **Unhappy** (*not happy*)
  - **Re-** + *play* → **Replay** (*play again*)
- 

#### b) Suffix (Added after the stem)

A **suffix** is an affix added at the **end** of a word to modify its meaning or grammatical role.

◆ **Example:**

- *Joy* + **-ful** → **Joyful** (*full of joy*)
  - *Read* + **-ing** → **Reading** (*present participle form*)
- 

#### c) Infix (Inserted within a stem)

An **infix** is an affix placed inside a word. In English, infixes are rare but exist in some informal expressions.

◆ **Example:**

- "fan**bloody**tastic" (infix **bloody** added for emphasis)

In other languages like Tagalog:

- **Sulat** (write) → **Sumulat** (past tense of write, "wrote")

---

**d) Circumfix (Attached on both sides of the stem)**

A **circumfix** consists of two parts—one before and one after the root. English has few circumfixes, but some languages, like German, use them frequently.

◆ **Example (German):**

- *Geben* (to give) → **ge+geb+en** (past participle **gegeben**)

5a. What is POS tagging? List and explain different taggers with suitable examples.

**Part-of-Speech Tagging**

- **Part-of-speech tagging** is the process of assigning a part-of-speech to each word in a sentence.
- The **input** to a tagging algorithm is the sequence of words and specified tag sets.
- The **output** is a single best part-of-speech tag for each word.

**Rule-Based Tagger**

- Most rule-based taggers follow a **2-stage architecture**:

**1st Stage: Dictionary Look-up**

o Returns:

- ♣ A set of **potential tags**
- ♣ Appropriate **syntactic features** for each word

**2nd Stage: Hand-Coded Rules**

o Used to:

- ♣ Discard **contextually illegitimate** tags
- ♣ Assign a **single part-of-speech** to each word

**Example 1**

**Sentence:** *The show must go on.*

- **Potential tags for "show"** → {VB (Verb), NN (Noun)}

• **Disambiguation Rule:**

o *IF preceding word is a determiner (e.g., "The"), THEN eliminate the VB tag.*

- **Result:** "show" is tagged as **NN (Noun)** in the given sentence.

**Example 2**

**Morphological Rule:**

- *IF a word ends in "-ing" and the preceding word is a verb, THEN label it as a verb (VB).*

**Advantages of Rule-Based Taggers**

- ✓ **Speed** – Rule-based taggers are fast.
- ✓ **Deterministic** – They provide consistent outputs.

**Limitations of Rule-Based Taggers**

- ✗ Requires **significant skill and effort** to write disambiguation rules.

**X Time-consuming** – Writing a complete rule-set takes time.

**X Language-Specific** – Rules must be created separately for each language.

6a. Explain Naïve Bayes classifier and types.

Classify vehicle based on features following

Stolen=yes, colour=yellow,type=SUV,origin =domestic

Example No.	Color	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

**Naive Bayes classifiers** are a powerful and commonly used technique in Natural Language Processing (NLP), particularly for tasks like text classification, spam detection, sentiment analysis, and more. They're based on Bayes' Theorem and the assumption that features (in this case, words) are conditionally independent given the class label.

### Step 1: Compute Prior Probabilities

From the dataset:

- Total samples = 10
- Stolen (Yes) = 5
- Not Stolen (No) = 5

$$P(\text{Stolen} = \text{Yes}) = \frac{5}{10} = 0.5$$

$$P(\text{Stolen} = \text{No}) = \frac{5}{10} = 0.5$$

### Step 2: Compute Likelihood Probabilities

For Stolen = Yes

Feature	Value	Count	Probability
Color	Red	3	$\frac{3}{5} = 0.6$
	Yellow	2	$\frac{2}{5} = 0.4$
Type	Sports	4	$\frac{4}{5} = 0.8$
	SUV	1	$\frac{1}{5} = 0.2$
Origin	Domestic	2	$\frac{2}{5} = 0.4$
	Imported	3	$\frac{3}{5} = 0.6$

**For Stolen = No**

Feature	Value	Count	Probability
Color	Red	2	$\frac{2}{5} = 0.4$
	Yellow	3	$\frac{3}{5} = 0.6$
Type	Sports	2	$\frac{2}{5} = 0.4$
	SUV	3	$\frac{3}{5} = 0.6$
Origin	Domestic	3	$\frac{3}{5} = 0.6$
	Imported	2	$\frac{2}{5} = 0.4$

### Step 3: Predict New Example

Let's classify a Yellow SUV (Domestic).

Using Bayes' theorem, we calculate:

**Probability of Stolen = Yes**

$$\begin{aligned}
 P(\text{Stolen} = \text{Yes} | \text{Yellow}, \text{SUV}, \text{Domestic}) &\propto P(\text{Yellow} | \text{Stolen}) \times P(\text{SUV} | \text{Stolen}) \times P(\text{Domestic} | \text{Stolen}) \times P(\text{Stolen}) \\
 &= 0.4 \times 0.2 \times 0.4 \times 0.5 \\
 &= 0.016
 \end{aligned}$$

**Probability of Stolen = No**

$$\begin{aligned}
 P(\text{Stolen} = \text{No} | \text{Yellow}, \text{SUV}, \text{Domestic}) &\propto P(\text{Yellow} | \text{NotStolen}) \times P(\text{SUV} | \text{NotStolen}) \times P(\text{Domestic} | \text{NotStolen}) \times P(\text{NotStolen}) \\
 &= 0.6 \times 0.6 \times 0.6 \times 0.5 \\
 &= 0.108
 \end{aligned}$$

Since  $P(\text{Stolen} = \text{No}) > P(\text{Stolen} = \text{Yes})$ , we classify the Yellow SUV (Domestic) as **Not Stolen**.

6b. Write steps to ensure that the model accurately captures and interprets the emotions and opinions expressed in text data of the Customer Support dataset.

```

messages = [
    "I am extremely unhappy with the service I received.",
    "Thank you for the excellent support!",
    "The product arrived damaged and I need a replacement."
]

```

To ensure that the model accurately captures and interprets emotions and opinions in the **Customer Support dataset**, follow these steps:

## 1. Data Preprocessing

- Convert text to lowercase for consistency.
- Remove special characters, unnecessary punctuation, and stopwords to focus on meaningful words.
- Tokenize and lemmatize words to normalize the text.

## 2. Label Emotions and Opinions

- Manually label a dataset or use sentiment analysis tools to categorize messages as **positive, negative, or neutral**.
- Example labels for given messages:
  - **Negative:** "I am extremely unhappy with the service I received."
  - **Positive:** "Thank you for the excellent support!"
  - **Neutral/Negative:** "The product arrived damaged and I need a replacement."

## 3. Feature Extraction

- Use **TF-IDF, word embeddings (Word2Vec, GloVe, BERT)**, or **transformers** to extract meaningful features from text.
- Consider sentiment-related features like **polarity** and **subjectivity**.

## 4. Train and Fine-Tune a Sentiment Analysis Model

- Use supervised learning with models like **Logistic Regression, Naïve Bayes, Random Forest**, or deep learning models such as **LSTMs, BERT, RoBERTa**.
- Fine-tune pre-trained models (e.g., **BERT-based sentiment classifiers**) on the dataset.

## 5. Evaluate Model Performance

- Use metrics like **accuracy, precision, recall, and F1-score** to assess performance.
- Validate with a diverse test set covering different tones and expressions.

## 6. Handle Context and Ambiguity

- Implement contextual analysis using transformer models (e.g., **BERT, GPT**).
- Detect sarcasm, negations, and implicit sentiments through advanced NLP techniques.

## 7. Continuous Improvement and Feedback

- Regularly update the dataset with new customer interactions.
- Use human feedback to refine sentiment labels and retrain the model periodically.

**THANKS**