USN | | | | | | | | | |

Internal Assessment Test 2 – May 2025

Sub:	NATURAL LANGUAGE PROCESSING				Sub Code:	BAI601	Branch:	AIML
Date:	**24/05/25**	Duration:	90 min	Max Marks: 50	Sem/Sec:	IV /A, B		OBE

	Answer any FIVE FULL Questions	**MARKS**	**CO**	**RBT**
1	Review / Positive(1) / Negative(0) "I love this phone" / 1 / 0 " This Phone is amazing" / 1 / 0 "I hate this Phone" / 0 / 1 "This phone is bad" / 0 / 1 compute the probabilities and classify using Naive bayes of the given sentence "This phone is amazing."?	10	CO3	L3

Review	Positive(1)	Negative(0)
"I love this phone"	1	0
" This Phone is amazing"	1	0
"I hate this Phone"	0	1
"This phone is bad"	0	1

1) Extract Data :

+ve Review

I love this phone → {I, love, this, phone}

This phone is amazing → {this, phone, is, amazing}

−ve Review

I hate this phone → {I, hate, this, phone}

This phone is bad → {this, phone, is, bad}.

Vocabulary = {I, love, this, phone, is, amazing, hate, bad}.

Word Counts : +ve

I − 1, love − 1, this − 2, phone − 2, is − 1, amazing − 1, hate − 0, bad − 0.

Total word in positive = 1 + 1 + 2 + 2 + 1 + 1 = 8.

word count. −ve.

I − 1, hate − 1, this − 2, phone − 2, is − 1, bad − 1, love − 0, amazing − 0.

Total words in Negative = 1+1+2+2+1+1 = 8

P(positive) = 2/4 = 0.5 P(negative) = 2/4 = 0.5

Naive Bayes with Laplace Smoothing.

$$P(w|c) = \frac{\text{count}(w, c) + 1}{\text{total words in } C + \text{Vocabulary size}}$$

V = 8

total words = 8.

P(I|+ve) = 0.125

P(love|+ve) = 0.125

P(this|+ve) = 0.1875 ....

p(I | -ve) = 0.125

p(hate | -ve) = 0.125 . . .

Classify sentence = This phone is amazing.

p(this | positive) = 0.1875

p(phone | positive) = 0.1875

p(is | positive) = 0.125

p(amazing | positive) = 0.125

Likelihood = 0.1875 x 0.1875 x 0.125 x 0.125 = 0.000054913

Score = p(+ve) x p(sentence | +ve) = 0.5 x 0.00054913

= 0.00002746

Negative

Likehood = 0.187 x 0.1875 x 0.125 x 0.0625 = 0.000027465

Score = p(-ve) x p(sentence | -ve) = 0.5 x 0.00002746

= 0.000013729

Total score = 0.00002746 + 0.000013729 = 0.00004198

p(positive | sentence) = 0.000027465 / 0.0000041198

≈ 0.667

p(negative | sentence) = 0.0000137329 / 0.000041198

≈ 0.3333

Sentence is classified as positive

---

2 a — With a suitable example explain cluster-based IR modeling. — CO4 — L2

Cluster-Based Information Retrieval (IR) Modeling is a method used to improve the performance of search systems by grouping similar documents together (clustering), and then using those clusters to retrieve relevant documents more efficiently and accurately.

Clustering is the process of grouping a set of documents (or data points) such that documents in the same group (called a *cluster*) are more similar to each other than to those in other groups.

Goal in IR: Improve retrieval by organizing documents into meaningful groups before searching, instead of searching the entire document collection blindly.

Types of Clustering

There are two main types:

- **Hierarchical Clustering**
  - Builds a hierarchy (tree) of clusters.
  - Two approaches:
    - **Agglomerative (bottom-up)**: Each document starts as its own cluster. Pairs of clusters are merged as one moves up.
    - **Divisive (top-down)**: All documents start in one cluster, and splits are performed recursively.
  - Output: Dendrogram (a tree diagram).

- Suitable when the number of clusters is unknown.

- **Partitioning Clustering (e.g., K-Means)**
  - Documents are divided into K clusters, where K is specified beforehand.
  - Common algorithm: **K-Means**
    - Randomly initialize K centroids.
    - Assign each document to the nearest centroid.
    - Recalculate centroids.
    - Repeat until convergence.

Cosine Similarity in Clustering

Cosine Similarity measures the angle between two document vectors in a multi-dimensional space. It is commonly used in text clustering because it measures orientation, not magnitude.

$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{||A||\,||B||}$$

Where:

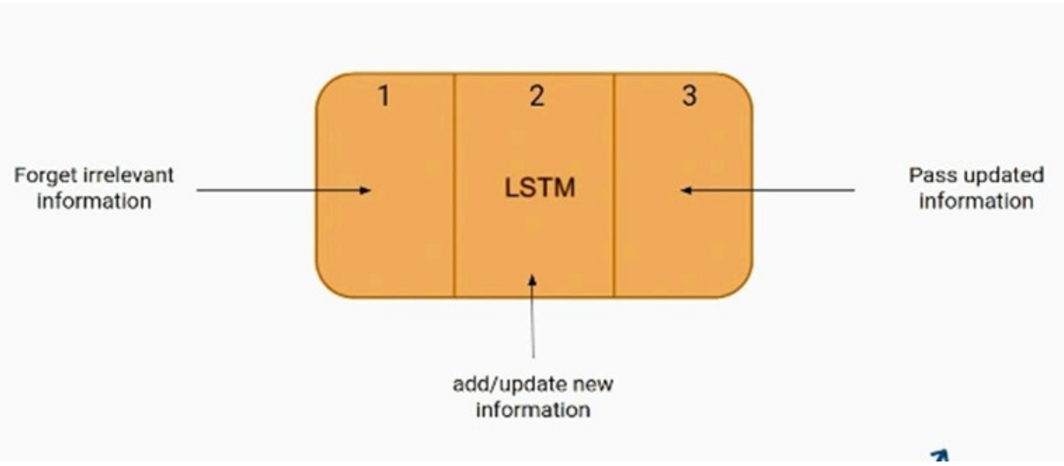
- $A \cdot B$ is the dot product of vectors A and B.
- $||A||\,||B||$ are the magnitudes (lengths) of vectors A and B.

Range: $[0, 1]$ for non-negative documents.

- **1** means documents are identical in terms of direction.
- **0** means documents are orthogonal (no similarity).

2 b	Explain the LSTM Model and architecture	5	CO4	L2

LSTM is a type of Recurrent Neural Network (RNN) designed to handle sequential data and overcome the vanishing gradient problem typically faced by traditional RNNs. It is widely used in applications such as text generation, speech recognition, and time series prediction.



LSTM Architecture:

LSTM consists of a series of memory cells, each of which has three main components (gates):

1. Forget Gate:

Decides what information from the previous cell state should be forgotten.

2. Input Gate:

Decides what new information should be added to the cell state.

3. Output Gate:

Decides what information from the current cell state should be output to the next hidden state.

• Cell State (C_t):

Acts as long-term memory that carries important information across many time steps.

• Hidden State (H_t):

Acts as short-term memory, used at each step for prediction.

Working

1. Forget Irrelevant Info:

Forget gate checks which part of the past information is no longer relevant.

2. Update with New Info:

Input gate determines what new data to add.

3. Pass On Important Info:

Output gate sends updated data to the next time step.

Equations of Gates:

In order to do their respective tasks the gates of the **LSTM** model are using certain equations in order to do the calculations and operations. They are:

1. Forget Gate:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

2. Input Gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

3. Cell State Update:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

4. Output Gate:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \odot \tanh(C_t)$$

Terminologies of the gate are as follows:

	x_t: Input at time step t h_{t-1}: Hidden state from the previous time step C_{t-1}: Cell state from the previous time step σ: Sigmoid activation function $\tanh$: Hyperbolic tangent activation $\odot$: Element-wise multiplication			
3	Given two binary word vectors w1 and w2 as follows: w1= [1010011010] w2= [0011111100] Compute the cosine similarity between them.	10	CO4	L3

$w_1 = [1010011010]$ $w_2 = [0011111100]$

i) Cosine similarity $= \dfrac{(w_1 \times w_2)}{(|w_1|\,|w_2|)}$

$w_1 \cdot w_2 = (1*0) + (0*0) + (1*1) + (0*1) +$
$(0*1) + (1*1) + (1*1) + (0*1) + (1*0) + (0*0)$

$= 0 + 0 + 1 + 0 + 0 + 1 + 1 + 0 + 0 + 0.$

$= 3$

$|w_1| = \text{sqrt}(1\text{^}2 +)(0\text{^}2) + (1\text{^}2) + (0\text{^}2) +$
$(0\text{^}2) + (1\text{^}2) + (1\text{^}2) + (0\text{^}2) + (1\text{^}2) +$
$(0\text{^}2))$

$= \text{sqrt}(1 + 0 + 1 + 0 + 0 + 1 + 1 + 0 + 1 + 0)$

$= \text{sqrt}(5)$

$|w_2| = \text{sqrt}(6).$

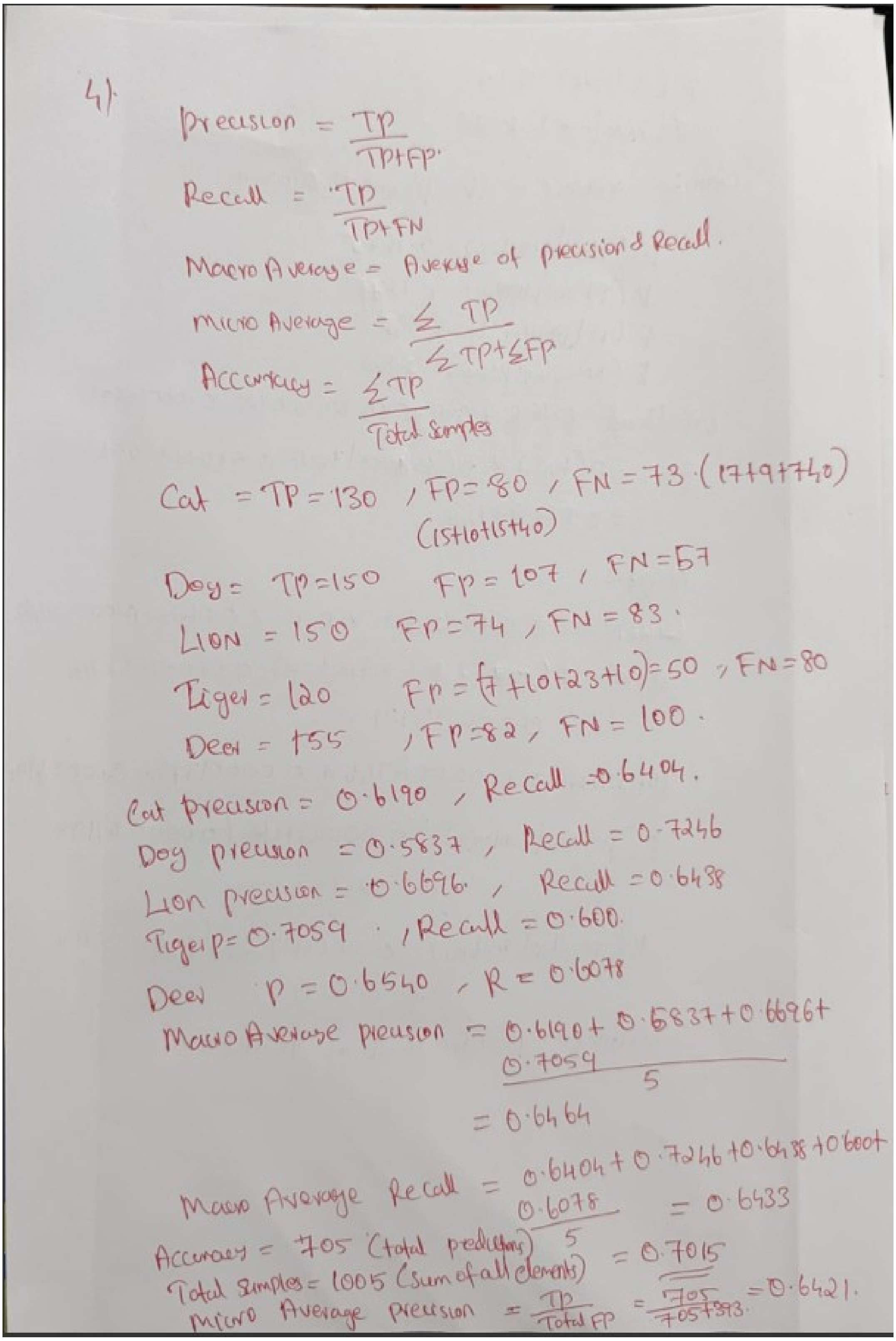
Cosine similarity $= \dfrac{3}{\text{sqrt}(5) \times \text{sqrt}(6)} = \dfrac{3}{\text{sqrt}(30)} = \dfrac{3}{5.48} = 0.547$

4	Suppose you have trained an image classifier with 5 classes-cat, dog, lion, tiger and deer. Consider the confusion matrix shown	10	CO4	L3

	cat	dog	lion	tiger	deer
cat	130	17	9	7	40
dog	15	150	25	10	7
lion	10	45	150	23	5
tiger	15	15	20	120	30
deer	40 30	20	10	155	

What is the macro average precision? What is the macro averaged recall?

What is the accuracy of your classifier? What is the micro averaged precision?



4).

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

Macro Average = Average of precision & Recall.

$$\text{micro Average} = \frac{\sum TP}{\sum TP + \sum FP}$$

$$\text{Accuracy} = \frac{\sum TP}{\text{Total Samples}}$$

Cat = TP = 130, FP = 80 (15+10+15+40), FN = 73 (17+9+7+40)

Dog = TP = 150 FP = 107, FN = 57

Lion = 150 FP = 74, FN = 83.

Tiger = 120 FP = (7+10+23+10) = 50, FN = 80

Deer = 155, FP = 82, FN = 100.

Cat precision = 0.6190, Recall = 0.6404.

Dog precision = 0.5837, Recall = 0.7246

Lion precision = 0.6696, Recall = 0.6438

Tiger P = 0.7059, Recall = 0.600.

Deer P = 0.6540, R = 0.6078

$$\text{Macro Average precision} = \frac{0.6190 + 0.5837 + 0.6696 + 0.7059}{5} = 0.6464$$

$$\text{Macro Average Recall} = \frac{0.6404 + 0.7246 + 0.6438 + 0.600 + 0.6078}{5} = 0.6433$$

Accuracy = 705 (total predictions) = 0.7015

Total Samples = 1005 (Sum of all elements)

$$\text{micro Average precision} = \frac{TP}{\text{Total FP}} = \frac{705}{705+393} = 0.6421.$$

5a Write a brief note on bias and word embedding models | 5 | CO 5 | L 2

Word Embedding Models (like Word2Vec, GloVe, or fastText) represent words as dense vectors in a high-dimensional space. These models capture semantic relationships by learning from large text corpora.

Bias in NLP refers to systematic favoritism or prejudice in language models. It often reflects societal stereotypes, such as associating certain professions with a specific gender or race.

Word Embedding Models & Bias

Word embedding models (e.g., **Word2Vec**, **GloVe**, **FastText**) represent words as vectors based on their surrounding context.

These embeddings may capture and even amplify biases from the training corpus.

Approach: Corpus-Based Bias Detection

1. **Corpus Creation**:
   - Create two gender-specific corpora:
     - **Male Corpus**: Texts with male references (e.g., *he, man, king, father, John*).
     - **Female Corpus**: Texts with female references (e.g., *she, woman, queen, mother, Mary*).
2. **Embedding Generation**:
   - Train or use existing embeddings (e.g., Word2Vec).
   - Generate vectors for key words like *doctor*, *nurse*, etc.
3. **Bias Detection for Given Sentence**:
   - **Input**: *"The doctor treated his patient."*
   - Extract keywords (e.g., *doctor*).
   - Compute **average similarity** of *doctor* with male corpus vs. female corpus:

$$\text{sim}_{\text{male}} = \frac{1}{n}\sum_{i=1}^{n}\cos(\vec{doctor}, \vec{male_i})$$

$$\text{sim}_{\text{female}} = \frac{1}{m}\sum_{i=1}^{m}\cos(\vec{doctor}, \vec{female_i})$$

Threshold-Based Bias Detection:

- Define a threshold (e.g., **0.1** difference).
- If:
  - $$Y = \begin{cases} 1 & if\ f(x) \geq \theta \\ 0 & if\ f(x) < \theta \end{cases}$$

Q.No	Question	Marks	CO	L
5 b	Discuss the various fairness measures for evaluating bias in NLP and apply the Gender Bias for the given sentence "The doctor treated his patient".	5	CO 5	L 2

Fairness in NLP

Fairness in NLP aims to ensure that language models do not promote stereotypes or unfair associations, such as linking certain professions with specific genders, races, or identities.

Key Fairness Measures

1. **Direct Bias**
   Checks if neutral words (e.g., *doctor*, *engineer*) are more similar to gendered words (like *he* or *she*).

2. **Indirect Bias**
   Evaluates whether words associated with a neutral term reflect bias, even if the word itself seems unbiased.

3. **Word Embedding Association Test (WEAT)**
   Statistically analyzes how strongly groups of words (e.g., professions) are linked to gendered terms.

4. **Projection Test**
   Measures whether a neutral word leans more toward male or female concepts within the model.

5. **Counterfactual Fairness**
   Tests if changing gendered terms in a sentence (e.g., *his* to *her*) leads to different outputs or interpretations.

6. **Sentence-Level Testing**

Examines full sentences to see if models treat them differently based on gendered language.

Approach: Corpus-Based Bias Detection

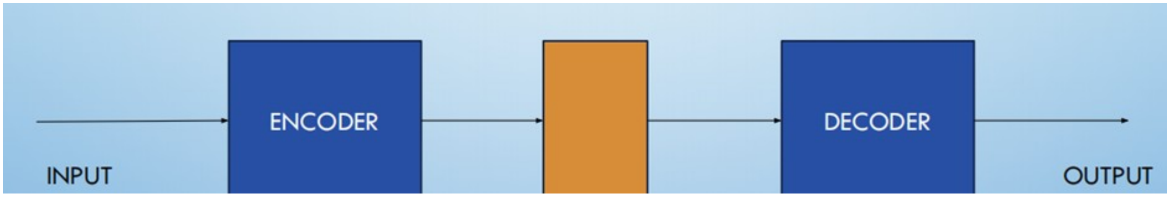
- **Step 1**: Create two small corpora — one containing male-associated words and another with female-associated words.
- **Step 2**: For the given sentence “The doctor treated his patient,” identify keywords like *doctor*.
- **Step 3**: Compare how closely the word *doctor* relates to the male and female corpora.
- **Step 4**: If *doctor* is significantly closer to the male corpus, the model shows **male bias**.

Application to Example Sentence

Sentence: *“The doctor treated his patient.”*

- The use of “his” and the association of *doctor* more strongly with male terms indicates **gender bias**.
- The model may be assuming *doctor* is male, which reflects a **lack of fairness**.

6	Describe the Simple encoder-decoder model and identify the attention mechanism of the given sentence “The girl liked the pink frock “.	10	CO5	L3



(note: middle layer represents the hidden layer)

1. Encoder-Decoder Architecture The **Encoder-Decoder model** is used in sequence-to-sequence tasks like **machine translation**, **summarization**, and **text generation**.

- **Encoder**:
  - Takes the input sentence (e.g., “The girl liked the pink frock”).
  - Converts each word into a hidden state.
  - Final hidden state summarizes the full sentence (context vector).
- **Decoder**:
  - Starts with the context vector from the encoder.
  - Generates the output sequence word-by-word (e.g., in translation).

Limitation: All input information is compressed into a single fixed-length vector, which may lose details for long sentences.

---

2. Attention Mechanism

To overcome this, **Attention** was introduced. Instead of relying only on the final encoder state, the decoder looks at **all encoder hidden states** and decides **which words to focus on** at each decoding step.

---

3. Applying Attention to the Sentence

Sentence: *“The girl liked the pink frock.”*

When processing this sentence using an encoder-decoder model **with attention** (e.g., for translating it to another language):

- The **encoder** generates hidden states for each word:
  - "The", "girl", "liked", "the", "pink", "frock"
- At each decoding step (say translating "liked" into another language), the **attention mechanism** helps the decoder focus more on:
  - The most **relevant words** — e.g., "girl" and "liked"
- For example, when decoding "frock", the decoder may attend more to "pink" and "frock" from the encoder side.

Basic Equations

Let's say:

- h_i = encoder hidden state for the i-th word
- s_t = decoder hidden state at time t
- α_{ti} = attention weight for h_i at time t

1. Score Calculation (e.g., dot product):

$$score(s_t, h_i) = s_t^\top h_i$$

2. Softmax Attention Weights:

$$\alpha_{ti} = \frac{\exp(score(s_t, h_i))}{\sum_j \exp(score(s_t, h_j))}$$

3. Context Vector:

$$c_t = \sum_i \alpha_{ti} h_i$$

Faculty Signature CCI Signature HOD Signature