

# Foundation of AI and ML (4351601) - Summer 2024 Solution

Milav Dabgar

May 16, 2024

## Question 1(a) [3 marks]

What do you mean by Narrow AI or Weak AI?

### Solution

**Narrow AI** or **Weak AI** refers to artificial intelligence systems designed to perform specific, limited tasks within a narrow domain.

**Table 1.** Narrow AI Characteristics

Aspect	Description
Scope	Limited to specific tasks
Intelligence	Task-specific expertise
Examples	Siri, chess programs, recommendation systems
Learning	Pattern recognition within domain

### Mnemonic

“Narrow = Specific Tasks Only”

## Question 1(b) [4 marks]

Define: Classification, Regression, Clustering, Association Analysis.

### Solution

**Table 2.** Machine Learning Techniques

Technique	Definition	Type	Example
Classification	Predicts discrete categories/classes	Supervised	Email spam detection
Regression	Predicts continuous numerical values	Supervised	House price prediction
Clustering	Groups similar data points	Unsupervised	Customer segmentation
Association Analysis	Finds relationships between variables	Unsupervised	Market basket analysis

### Mnemonic

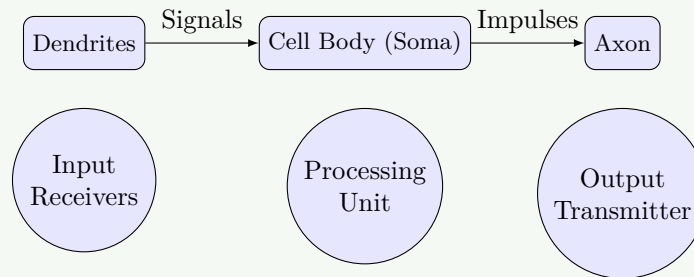
“CRCA - Categories, Real-numbers, Clusters, Associations”

## Question 1(c) [7 marks]

**Illuminate the three main components of neuron.**

### Solution

The three main components of a biological neuron that inspire artificial neural networks are:



**Figure 1.** Neuron Components

**Table 3.** Neuron Components

Component	Function	AI Equivalent
Dendrites	Receive input signals from other neurons	Input layer/weights
Cell Body (Soma)	Processes and integrates signals	Activation function
Axon	Transmits output signals to other neurons	Output connections

#### Key Points:

- **Dendrites:** Act as input receivers with varying connection strengths.
- **Cell Body:** Sums inputs and applies threshold function.
- **Axon:** Carries processed signal to next neurons.

### Mnemonic

“DCA - Dendrites Collect, Cell-body Calculates, Axon Announces”

## Question 1(c) OR [7 marks]

**Explicate back propagation method in Artificial Neural Network.**

### Solution

**Back Propagation** is a supervised learning algorithm used to train multi-layer neural networks by minimizing error through gradient descent.

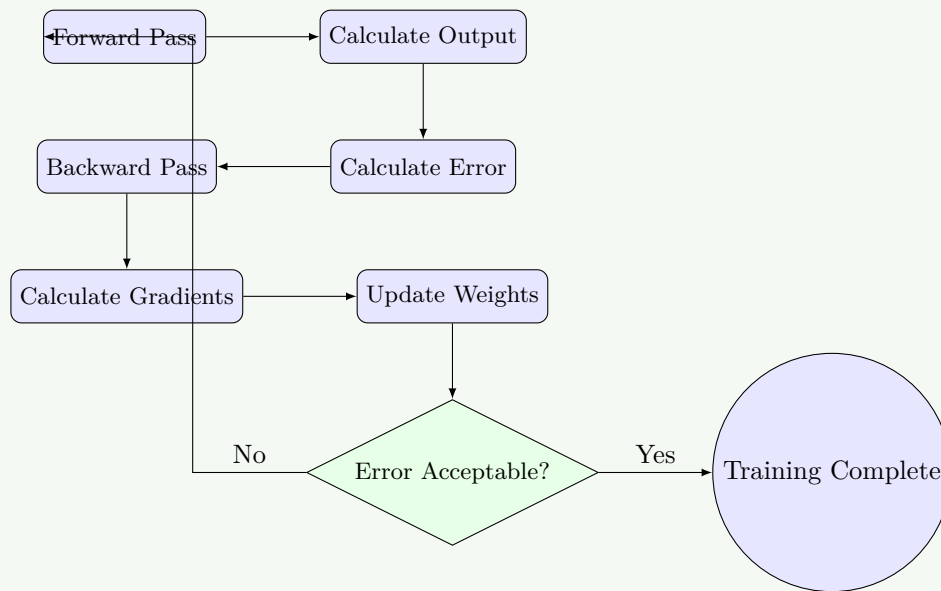


Figure 2. Back Propagation Flow

Table 4. Back Propagation Steps

Step	Process	Formula
Forward Pass	Calculate outputs layer by layer	$y = f(\sum(w_i x_i + b))$
Error Calculation	Compute loss function	$E = \frac{1}{2}(target - output)^2$
Backward Pass	Calculate error gradients	$\delta = \partial E / \partial w$
Weight Update	Adjust weights using learning rate	$w_{new} = w_{old} - \eta \cdot \delta$

**Key Features:**

- **Gradient Descent:** Uses calculus to find minimum error.
- **Chain Rule:** Propagates error backward through layers.
- **Learning Rate:** Controls speed of weight updates.

**Mnemonic**

“FEBU - Forward, Error, Backward, Update”

**Question 2(a) [3 marks]**

List out any five popular algorithms used in Machine Learning.

**Solution**

Table 5. Popular ML Algorithms

Algorithm	Type	Application
Linear Regression	Supervised	Prediction of continuous values
Decision Tree	Supervised	Classification and regression
K-Means Clustering	Unsupervised	Data grouping
Support Vector Machine	Supervised	Classification with margins
Random Forest	Supervised	Ensemble learning

**Mnemonic**

“LDKSR - Learn Data, Keep Samples, Run”

**Question 2(b) [4 marks]**

What is Expert System? List out its limitations and applications.

**Solution**

**Expert System** is an AI program that mimics human expert knowledge to solve complex problems in specific domains.

**Table 6.** Expert System Overview

Aspect	Details
<b>Definition</b>	AI system with domain-specific expertise
<b>Components</b>	Knowledge base, inference engine, user interface

**Applications:**

- **Medical Diagnosis:** Disease identification systems.
- **Financial Planning:** Investment advisory systems.
- **Fault Diagnosis:** Equipment troubleshooting.

**Limitations:**

- **Limited Domain:** Works only in specific areas.
- **Knowledge Acquisition:** Difficult to extract expert knowledge.
- **Maintenance:** Hard to update and modify rules.

**Mnemonic**

“EXPERT - Explains Problems, Executes Rules, Tests”

**Question 2(c) [7 marks]**

What is tokenization? Explain with suitable example.

**Solution**

**Tokenization** is the process of breaking down text into smaller units called tokens (words, phrases, symbols) for NLP processing.

**Table 7.** Tokenization Types

Type	Description	Example
<b>Word Tokenization</b>	Split by words	“Hello world” → [“Hello”, “world”]
<b>Sentence Tokenization</b>	Split by sentences	“Hi. How are you?” → [“Hi.”, “How are you?”]
<b>Subword Tokenization</b>	Split into subwords	“unhappy” → [“un”, “happy”]

**Code Example:**

```
1 import nltk
2 text = "Natural Language Processing is amazing!"
3 tokens = nltk.word_tokenize(text)
4 # Output: ['Natural', 'Language', 'Processing', 'is', 'amazing', '!']
```

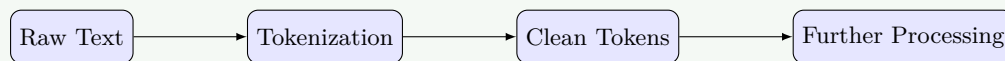


Figure 3. Tokenization Process

**Key Benefits:**

- **Standardization:** Converts text to uniform format.
- **Analysis Ready:** Prepares text for ML algorithms.
- **Feature Extraction:** Enables statistical analysis.

**Mnemonic**

“TOKEN - Text Operations Keep Everything Normalized”

**Question 2(a) OR [3 marks]**

Compare Supervised and Unsupervised Learning.

**Solution**

Table 8. Supervised vs Unsupervised Learning

Aspect	Supervised Learning	Unsupervised Learning
<b>Training Data</b>	Labeled data with target outputs	Unlabeled data without targets
<b>Goal</b>	Predict specific outcomes	Discover hidden patterns
<b>Examples</b>	Classification, Regression	Clustering, Association rules
<b>Evaluation</b>	Accuracy, precision, recall	Silhouette score, elbow method
<b>Applications</b>	Email spam, price prediction	Customer segmentation, anomaly detection

**Mnemonic**

“SU - Supervised Uses labels, Unsupervised Uncovers patterns”

**Question 2(b) OR [4 marks]**

Explain all about AI applications in Healthcare, Finance and Manufacturing.

**Solution**

Table 9. AI Applications by Industry

Industry	Applications	Benefits
<b>Healthcare</b>	Medical imaging, drug discovery, diagnosis	Improved accuracy, faster treatment
<b>Finance</b>	Fraud detection, algorithmic trading, credit scoring	Risk reduction, automated decisions
<b>Manufacturing</b>	Quality control, predictive maintenance, robotics	Efficiency, cost reduction

**Healthcare Examples:**

- **Medical Imaging:** AI detects cancer in X-rays and MRIs.
- **Drug Discovery:** AI accelerates new medicine development.

**Finance Examples:**

- **Fraud Detection:** Real-time transaction monitoring.
- **Robo-advisors:** Automated investment management.

**Manufacturing Examples:**

- **Quality Control:** Automated defect detection.
- **Predictive Maintenance:** Equipment failure prediction.

### Mnemonic

“HFM - Health, Finance, Manufacturing benefit from AI”

## Question 2(c) OR [7 marks]

What is syntactic analysis and how it is differ from lexical analysis?

### Solution

**Syntactic Analysis** examines the grammatical structure of sentences, while **Lexical Analysis** breaks text into meaningful tokens.

Table 10. Lexical vs Syntactic Analysis

Aspect	Lexical Analysis	Syntactic Analysis
<b>Purpose</b>	Tokenize text into words	Parse grammatical structure
<b>Input</b>	Raw text	Tokens from lexical analysis
<b>Output</b>	Tokens, part-of-speech tags	Parse trees, grammar rules
<b>Focus</b>	Individual words	Sentence structure
<b>Example</b>	"The cat runs" → [The, cat, runs]	Creates parse tree showing noun-verb relationship

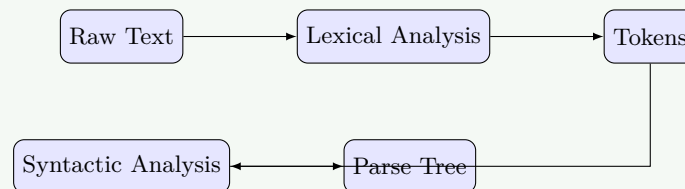


Figure 4. Analysis Process

### Example:

- **Lexical:** "She reads books" → ["She", "reads", "books"]
- **Syntactic:** Identifies "She" as subject, "reads" as verb, "books" as object.

### Key Differences:

- **Scope:** Lexical works on words, Syntactic on sentence structure.
- **Complexity:** Syntactic analysis is more complex than lexical.
- **Dependencies:** Syntactic analysis depends on lexical analysis.

### Mnemonic

“LEX-SYN: LEXical extracts, SYNtactic structures”

## Question 3(a) [3 marks]

List out various characteristics of Reactive machines.

## Solution

Table 11. Reactive Machines Characteristics

Characteristic	Description
No Memory	Cannot store past experiences
Present-focused	Responds only to current input
Deterministic	Same input produces same output
Task-specific	Designed for particular functions
No Learning	Cannot improve from experience

## Examples:

- **Deep Blue:** IBM's chess computer.
- **Game AI:** Tic-tac-toe programs.

## Mnemonic

"REACT - Responds Exactly, Always Consistent Tasks"

## Question 3(b) [4 marks]

Differentiate: Positive Reinforcement v/s Negative Reinforcement.

## Solution

Table 12. Positive vs Negative Reinforcement

Aspect	Positive Reinforcement	Negative Reinforcement
Definition	Adding reward for good behavior	Removing penalty for good behavior
Action	Give something desirable	Take away something undesirable
Goal	Increase desired behavior	Increase desired behavior
Example	Give treat for correct answer	Remove extra work for good performance

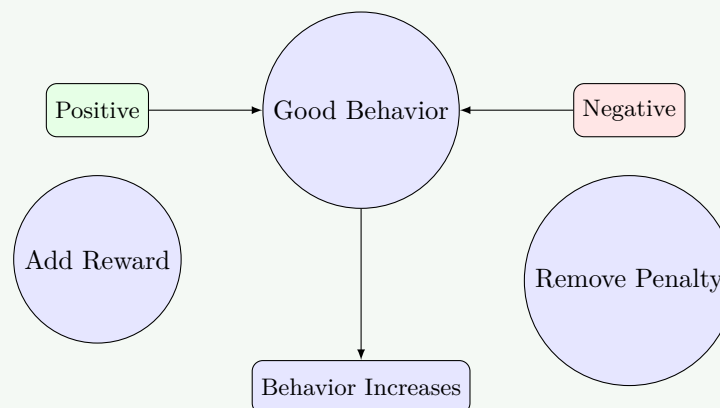


Figure 5. Reinforcement Types

## Key Points:

- **Both increase behavior** but through different mechanisms.
- **Positive adds** something pleasant.
- **Negative removes** something unpleasant.

**Mnemonic**

“PN - Positive adds Nice things, Negative removes Nasty things”

**Question 3(c) [7 marks]**

Explain all about Term-Frequency-Inverse Document Frequency(TF-IDF) word embedding technique.

**Solution**

**TF-IDF** is a numerical statistic that reflects how important a word is to a document in a collection of documents.

**Formula:**

$$\text{TF-IDF} = \text{TF}(t, d) \times \text{IDF}(t)$$

Where:

- $\text{TF}(t, d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total terms in document } d}$
- $\text{IDF}(t) = \log \left( \frac{\text{Total documents}}{\text{Documents containing term } t} \right)$

**Table 13.** TF-IDF Components

Component	Formula	Purpose
<b>Term Frequency (TF)</b>	$tf(t, d) = \text{count}(t, d) /  d $	Measures word frequency in document
<b>Inverse Document Frequency (IDF)</b>	$idf(t) = \log(N / df(t))$	Measures word importance across corpus
<b>TF-IDF Score</b>	$tf-idf(t, d) = tf(t, d) \times idf(t)$	Final word importance score

**Example Calculation:**

- Document: "cat sat on mat"
- Term: "cat"
- $\text{TF} = 1/4 = 0.25$
- If "cat" appears in 2 out of 10 documents:  $\text{IDF} = \log(10/2) = 0.699$
- $\text{TF-IDF} = 0.25 \times 0.699 = 0.175$

**Applications:**

- **Information Retrieval:** Search engines.
- **Text Mining:** Document similarity.
- **Feature Extraction:** ML preprocessing.

**Advantages:**

- **Common words get low scores** (the, and, is).
- **Rare but important words get high scores.**
- **Simple and effective** for text analysis.

**Mnemonic**

“TF-IDF - Term Frequency x Inverse Document Frequency”

**Question 3(a) OR [3 marks]**

Define Fuzzy Logic Systems. Discuss its key components.



**Solution**

**Fuzzy Logic Systems** handle uncertainty and partial truth, allowing values between completely true and completely false.

**Table 14.** Fuzzy Logic Components

Component	Function	Example
<b>Fuzzifier</b>	Converts crisp inputs to fuzzy sets	Temperature 75°F → "Warm" (0.7)
<b>Rule Base</b>	Contains if-then fuzzy rules	IF temp is warm THEN fan is medium
<b>Inference Engine</b>	Applies fuzzy rules to inputs	Combines multiple rules
<b>Defuzzifier</b>	Converts fuzzy output to crisp value	"Medium speed" → 60% fan speed

**Key Features:**

- **Membership Functions:** Degree of belonging (0 to 1).
- **Linguistic Variables:** Human-like terms (hot, cold, warm).
- **Fuzzy Rules:** IF-THEN statements with fuzzy conditions.

**Mnemonic**

"FRID - Fuzzifier, Rules, Inference, Defuzzifier"

**Question 3(b) OR [4 marks]**

Explain elements of reinforcement learning: Policy, Reward Signal, Value Function, Model

**Solution****Table 15.** Reinforcement Learning Elements

Element	Definition	Purpose
<b>Policy</b>	Strategy for selecting actions	Defines agent's behavior
<b>Reward Signal</b>	Feedback from environment	Indicates good/bad actions
<b>Value Function</b>	Expected future rewards	Estimates long-term benefit
<b>Model</b>	Agent's representation of environment	Predicts next state and reward

**Detailed Explanation:**

- **Policy ( $\pi$ ):** Defines behavior, can be deterministic or stochastic.
- **Reward Signal (**R**):** Immediate feedback (positive/negative).
- **Value Function (**V**):** Expected long-term return from state/action.
- **Model:** Predicts environment dynamics (transitions and rewards).

**Mnemonic**

"PRVM - Policy chooses, Reward judges, Value estimates, Model predicts"

**Question 3(c) OR [7 marks]**

Differentiate: frequency-based v/s prediction-based word embedding techniques.

## Solution

Table 16. Frequency-based vs Prediction-based Word Embeddings

Aspect	Frequency-based	Prediction-based
<b>Approach</b>	Count-based statistics	Neural network prediction
<b>Examples</b>	TF-IDF, Co-occurrence Matrix	Word2Vec, GloVe
<b>Computation</b>	Matrix factorization	Gradient descent
<b>Context</b>	Global statistics	Local context windows
<b>Scalability</b>	Limited by matrix size	Scales with vocabulary
<b>Quality</b>	Basic semantic relationships	Rich semantic relationships

**Frequency-based Methods:**

- **TF-IDF:** Term frequency  $\times$  Inverse document frequency.
- **Co-occurrence Matrix:** Word pair frequency counts.

**Prediction-based Methods:**

- **Word2Vec:** Skip-gram and CBOW models.
- **GloVe:** Global Vectors for Word Representation.

**Advantages:**

- **Frequency-based:** Simple, fast for small data, good for basic similarity.
- **Prediction-based:** Dense vectors, better semantics, scalable.

## Mnemonic

“FP - Frequency counts, Prediction learns”

## Question 4(a) [3 marks]

List out the key characteristics of reactive machine.

## Solution

Table 17. Reactive Machine Key Characteristics

Characteristic	Description
<b>Stateless</b>	No memory of past interactions
<b>Reactive</b>	Responds only to current inputs
<b>Deterministic</b>	Consistent outputs for same inputs
<b>Specialized</b>	Designed for specific tasks
<b>Real-time</b>	Immediate response to stimuli

**Examples:**

- **Deep Blue:** Chess-playing computer.
- **Google AlphaGo:** Go-playing system (early version).

## Mnemonic

“SRDSR - Stateless, Reactive, Deterministic, Specialized, Real-time”

## Question 4(b) [4 marks]

List out various pre-processing techniques. Explain any one of them with python code.

## Solution

Table 18. Text Pre-processing Techniques

Technique	Purpose	Example
Tokenization	Split text into words	"Hello world" → ["Hello", "world"]
Stop Word Removal	Remove common words	Remove "the", "and", "is"
Stemming	Reduce words to root form	"running" → "run"
Lemmatization	Convert to dictionary form	"better" → "good"

**Stemming Explanation:** Stemming reduces words to their root form by removing suffixes.

**Python Code for Stemming:**

```

1 import nltk
2 from nltk.stem import PorterStemmer
3
4 # Initialize stemmer
5 stemmer = PorterStemmer()
6
7 # Example words
8 words = ["running", "flies", "dogs", "churches", "studying"]
9
10 # Apply stemming
11 stemmed_words = [stemmer.stem(word) for word in words]
12 print(stemmed_words)
13 # Output: ['run', 'fli', 'dog', 'church', 'studi']

```

**Benefits of Stemming:**

- Reduces vocabulary size for ML models.
- Groups related words together.
- Improves text analysis efficiency.

## Mnemonic

"TSSL - Tokenize, Stop-words, Stem, Lemmatize"

## Question 4(c) [7 marks]

Illuminate the Word2vec technique in detail.

## Solution

**Word2Vec** is a neural network-based technique that learns dense vector representations of words by predicting context.

Table 19. Word2Vec Architectures

Architecture	Approach	Input	Output
Skip-gram	Predict context from center word	Center word	Context words
CBOW	Predict center word from context	Context words	Center word

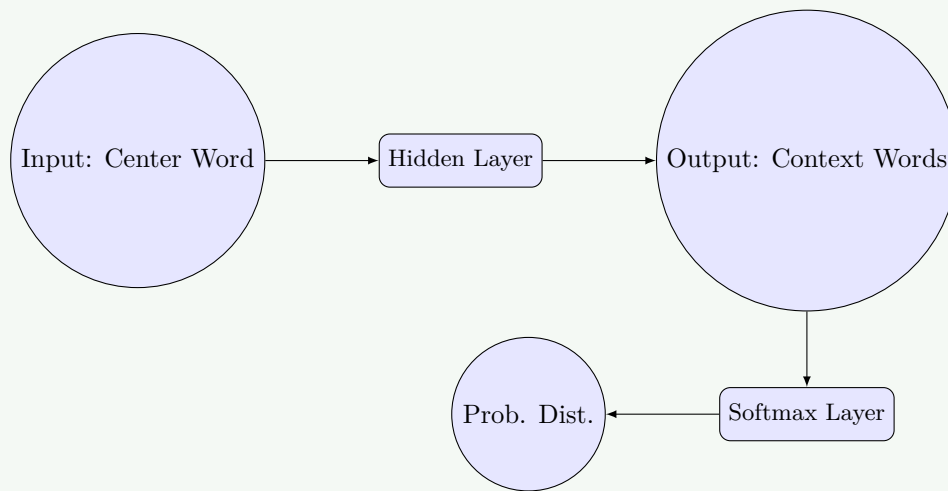


Figure 6. Skip-gram Model

**Training Process:**

- **Sliding Window:** Move window across text.
- **Word Pairs:** Create (center, context) pairs.
- **Neural Network:** Train to predict context.
- **Weight Matrix:** Extract word vectors.

**Mathematical Concept:**

$$\text{Objective} = \max \sum \log P(\text{context}|\text{center})$$

$$P(\text{context}|\text{center}) = \frac{\exp(v_{\text{context}} \cdot v_{\text{center}})}{\sum \exp(v_w \cdot v_{\text{center}})}$$

**Applications:**

- **Similarity:** Find similar words.
- **Analogies:** King - Man + Woman = Queen.
- **Feature Engineering:** ML input features.

**Mnemonic**

“W2V - Words to Vectors via neural networks”

**Question 4(a) OR [3 marks]**

List out any four applications of Natural Language Processing. Explain spam detection in detail.

**Solution**

Table 20. NLP Applications

Application	Description
Spam Detection	Identify unwanted emails
Sentiment Analysis	Determine emotional tone
Machine Translation	Translate between languages
Chatbots	Automated conversation systems

**Spam Detection Details:**

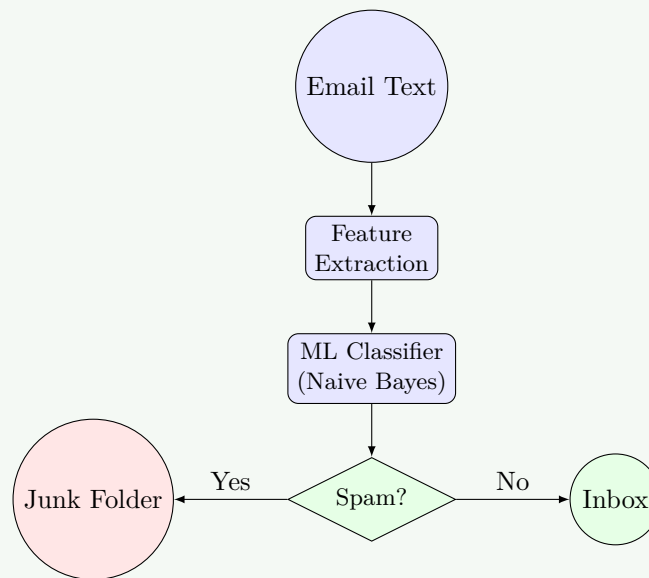


Figure 7. Spam Detection

- **Feature Extraction:** Word frequency, headers, URL analysis.
- **Classification:** Using algorithms like Naive Bayes.
- **Decision:** Classify as Spam or Legitimate.

**Mnemonic**

“SMTP - Spam, Machine Translation, Sentiment, Phishing detection”

**Question 4(b) OR [4 marks]**

Explain about discourse integration and pragmatic analysis.

**Solution**

Table 21. Discourse Integration vs Pragmatic Analysis

Aspect	Discourse Integration	Pragmatic Analysis
<b>Focus</b>	Text coherence and structure	Context and intention
<b>Scope</b>	Multiple sentences/paragraphs	Speaker's intended meaning
<b>Elements</b>	Anaphora, cataphora, connectives	Implicature, speech acts
<b>Goal</b>	Understand text flow	Understand real meaning

**Examples:**

- **Discourse:** "Mary owns a car. The vehicle is red." ("vehicle" refers to "car").
- **Pragmatic:** "Can you pass the salt?" (Request, not question of ability).

**Mnemonic**

“DP - Discourse connects, Pragmatics interprets context”

**Question 4(c) OR [7 marks]**

Discuss about the Bag of Words word embedding technique in detail.

**Solution**

**Bag of Words (BoW)** is a simple text representation method that treats documents as unordered collections of words.

**Table 22.** BoW Process

Step	Description	Example
<b>Vocabulary</b>	Collect unique words	["cat", "sat", "mat"]
<b>Vector</b>	Count occurrences	[1, 1, 1] for "cat sat mat"
<b>Representation</b>	Document → Vector	Vectors form Matrix

**Example:** Documents: 1. "The cat sat on the mat" 2. "The dog ran in the park" Vocab: [the, cat, sat, on, mat, dog, ran, in, park] Doc1 Vector: [2, 1, 1, 1, 1, 0, 0, 0, 0]

**Advantages:**

- **Simplicity:** Easy to understand and implement.
- **Effectiveness:** Works well for many tasks like spam detection.

**Disadvantages:**

- **No Word Order:** "cat sat mat" = "mat sat cat".
- **Sparse Vectors:** Many zeros in large vocabularies.
- **No Semantics:** No understanding of word meanings.

**Mnemonic**

"BOW - Bag Of Words counts occurrences"

**Question 5(a) [3 marks]**

What is the role of activation functions in Neural Network?

**Solution****Table 23.** Activation Function Roles

Role	Description
<b>Non-linearity</b>	Enables learning complex patterns
<b>Output Control</b>	Determines neuron firing threshold
<b>Gradient Flow</b>	Affects backpropagation efficiency
<b>Range Limiting</b>	Bounds output values

**Common Activation Functions:**

- **ReLU:**  $f(x) = \max(0, x)$  - Simple and efficient.
- **Sigmoid:**  $f(x) = 1/(1 + e^{-x})$  - Smooth probability output.
- **Tanh:** Zero-centered.

**Mnemonic**

"NOGL - Non-linearity, Output control, Gradient flow, Limiting range"

**Question 5(b) [4 marks]**

Describe architecture of Neural Network in detail.

## Solution

Table 24. Neural Network Architecture Components

Component	Function	Example
Input Layer	Receives input data	Features/pixels
Hidden Layers	Process information	Pattern recognition
Output Layer	Produces final result	Classification/prediction
Connections	Link neurons between layers	Weighted edges

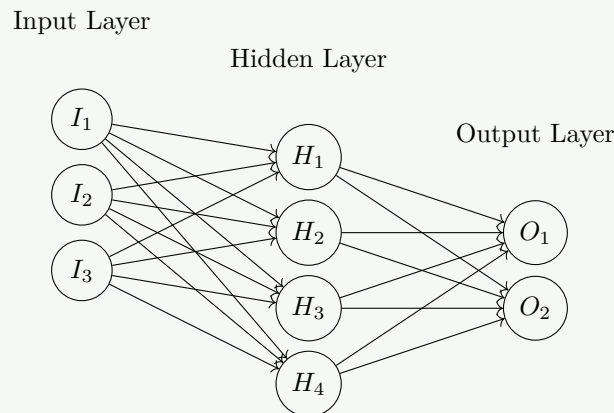


Figure 8. Neural Network Architecture

## Information Flow:

- **Forward Pass:** Input  $\rightarrow$  Hidden  $\rightarrow$  Output.
- **Weighted Sum:**  $\Sigma(w_i \times x_i + bias)$ .
- **Activation:** Apply activation function.

## Mnemonic

“THOC - Input, Hidden, Output, Connections”

## Question 5(c) [7 marks]

List out and explain types of ambiguities in Natural Language Processing.

## Solution

**Ambiguity** in NLP occurs when text has multiple possible interpretations.

Table 25. Types of NLP Ambiguities

Type	Definition	Resolution
<b>Lexical</b>	Word has multiple meanings	Context analysis
<b>Syntactic</b>	Multiple parse structures	Grammar rules
<b>Semantic</b>	Multiple sentence meanings	Semantic analysis
<b>Pragmatic</b>	Context-dependent meaning	Intent recognition
<b>Referential</b>	Unclear pronoun reference	Anaphora resolution

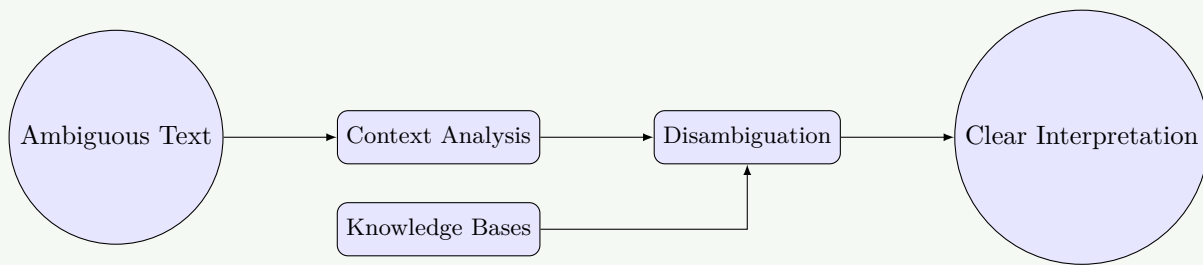


Figure 9. Resolution Process

**Examples:**

- **Lexical:** "Bank" (river vs financial).
- **Syntactic:** "I saw her duck" (bird vs action).
- **Referential:** "John told Bill he was late" (Who is late?).

**Mnemonic**

"LSSPR - Lexical, Syntactic, Semantic, Pragmatic, Referential"

**Question 5(a) OR [3 marks]**

List down the names of some popular activation functions used in Neural Network.

**Solution**

Table 26. Popular Activation Functions

Function	Formula	Range	Usage
<b>ReLU</b>	$f(x) = \max(0, x)$	$[0, \infty)$	Hidden layers
<b>Sigmoid</b>	$f(x) = 1/(1 + e^{-x})$	$(0, 1)$	Binary classification
<b>Tanh</b>	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$(-1, 1)$	Hidden layers
<b>Softmax</b>	$f(x_i) = e^{x_i} / \sum e^{x_j}$	$(0, 1)$	Multi-class output

**Mnemonic**

"RSTSL - ReLU, Sigmoid, Tanh, Softmax, Leaky ReLU"

**Question 5(b) OR [4 marks]**

Explain Learning process in artificial Neural Network.

**Solution**

**Learning Process** involves adjusting weights and biases to minimize error through iterative training.

Table 27. Learning Process Steps



Step	Process	Description
<b>Initialize</b>	Random weights	Start with small random values
<b>Forward Pass</b>	Calculate output	Propagate input through network
<b>Calculate Error</b>	Compare with target	Use loss function
<b>Backward Pass</b>	Calculate gradients	Use backpropagation
<b>Update Weights</b>	Adjust parameters	Apply gradient descent

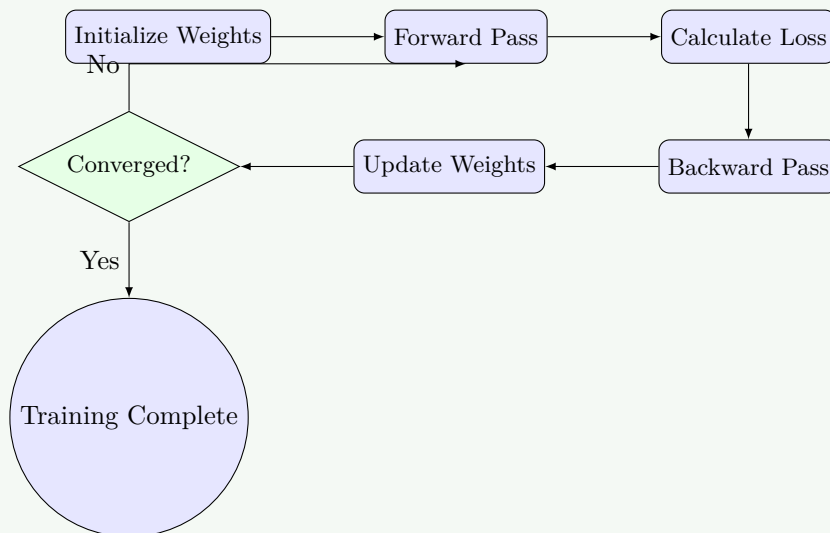


Figure 10. Learning Flow

**Mnemonic**

“IFCBU - Initialize, Forward, Calculate, Backward, Update”

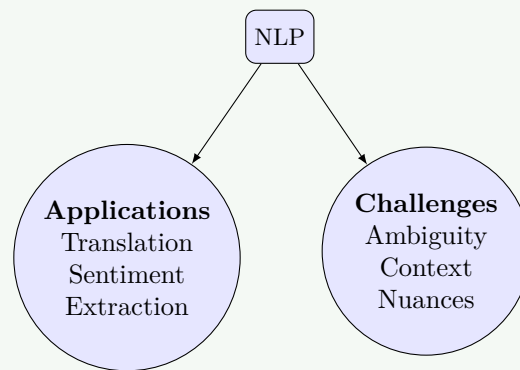
**Question 5(c) OR [7 marks]**

List out various advantages and disadvantages of Natural Language Processing.

**Solution**

Table 28. NLP Advantages and Disadvantages

Advantages	Disadvantages
Automated Text Analysis	Ambiguity Handling
Language Translation	Context Understanding
Human-Computer Interaction	Cultural Nuances
Information Extraction	Computational Complexity
Sentiment Analysis	Data Requirements



**Figure 11.** Applications vs Challenges

**Advantages:**

- **Business:** Customer service chatbots, content analysis.
- **Technical:** Scalability, consistency, speed.

**Disadvantages:**

- **Challenges:** Ambiguity, sarcasm detection, domain specificity.
- **Resources:** Requires large datasets and computational power.

**Mnemonic**

“ALICE vs ACHDR - Automated, Language, Interaction, Content, Extraction vs Ambiguity, Context, Human-nuances”