

3-Mark Questions

1. Define **NLU, NLP, and NLG** with one real-world example each.
2. Differentiate between **morpheme, grapheme, and phoneme** with examples.
3. Write a Python regular expression to **extract all email IDs** from a given text.
4. Explain the difference between **stemming and lemmatization** with examples.
5. Tokenize the sentence "AI helps machines understand human language." into words and punctuation.
6. What is the role of **stop words** in NLP? Give two examples of stop words in English.
7. Write the **bigram model representation** of the sentence "I love NLP".
8. What is **Bag of Words (BoW)**? Represent the sentences "NLP is fun" and "NLP is powerful" in BoW format.
9. Explain the concept of **syntax and semantics** in linguistics with a short example.
10. Define **Markov Model** with an example of a probability transition matrix.

5-Mark Questions

1. Explain with examples the differences between **Morphology, Syntax, Semantics, and Pragmatics** in linguistics.
2. Write a Python program to **remove stop words** from the given sentence:
"Natural Language Processing makes language easy for machines."
3. Construct a **trigram probability** for the sentence "I love learning NLP" assuming add-one smoothing.
4. Apply **PoS tagging** on the sentence "Time flies like an arrow." and explain ambiguity in tagging.
5. Describe the **Viterbi Algorithm** with a worked-out example for finding the most probable PoS tag sequence.
6. Using **Hidden Markov Model (HMM)**, explain how Parts-of-Speech tagging works with state and observation sequences.
7. Compare **Classical Approaches of NLP** with modern machine learning approaches, giving two examples of each.
8. Scenario: You are building a **chatbot for a banking system**. Explain how you would use
 - Tokenization
 - Stop word removal
 - Lemmatization

- PoS tagging
to preprocess user queries.
9. Write a Python program using **regular expressions** to extract all words starting with a capital letter from a paragraph.
 10. Explain with an example how **n-gram models** help in predicting the next word in a sentence.
-
1. Differentiate between **NLU** and **NLG** with examples.
 2. Write a Python code snippet to **perform word tokenization** using NLTK.
 3. Define **Regular Expressions**. Write a regex to match all numbers in a given string.
 4. What is the difference between **keywords** and **stopwords** in NLP? Give examples.
 5. Construct the **unigram probabilities** for the sentence "NLP is amazing".
 6. Explain **pragmatics** in linguistics with a real-life dialogue example.
 7. Write the **PoS tags** for "She sells sea shells on the seashore."
 8. What is the importance of **normalization** in text preprocessing?
 9. Give one example each of **morphology** and **syntax errors** in English sentences.
 10. Write a Python snippet to perform **stemming** on the words: ["running", "flies", "studies"].

5-Mark Questions

1. Explain **grapheme**, **phoneme**, **morpheme** with two examples each in English.
2. Write a Python program to convert the sentence "Dogs are running faster than cats." into **its lemmatized form**.
3. Construct a **bigram probability table** for the text "I enjoy learning NLP". Assume add-one smoothing.
4. Discuss how **Hidden Markov Models** are used in **speech recognition** with a worked-out state diagram.
5. Scenario: You are analyzing customer reviews. Explain how you would use:
 - Tokenization
 - Stopword removal
 - Stemming
 - Bag of Words
to build a sentiment classification pipeline.

6. Explain with example how **regular expressions** can be used to validate email addresses and phone numbers.
7. Given the transition probabilities:
 - $P(\text{Noun} \rightarrow \text{Verb}) = 0.4$, $P(\text{Noun} \rightarrow \text{Noun}) = 0.6$
 - $P(\text{Verb} \rightarrow \text{Noun}) = 0.7$, $P(\text{Verb} \rightarrow \text{Verb}) = 0.3$Find the probability of the sequence "Noun \rightarrow Verb \rightarrow Noun".
8. Describe the **Viterbi Algorithm steps** with an example of decoding "Janet will back the bill" in PoS tagging.
9. Explain the differences between **Bag of Words** and **N-gram models**. Which one captures context better and why?
10. Write a Python program to generate **n-grams (bigrams and trigrams)** for the sentence "Natural Language Processing is powerful".