Scenario NLP SET 1

1. Scenario: Spelling Correction in a Text Editor

Question: You're creating a simple text editor. How would you correct spelling mistakes in a sentence?

1. **Break the sentence into words**
   → Tokenization
2. **Check each word in a dictionary**
   → Compare with a list of valid words
3. **Find closest matching correct word**
   → Using Levenshtein distance / edit distance
4. **Replace incorrect word with best match**
   → e.g., "speling" → "spelling"
5. **Reconstruct corrected sentence**
   → Join all corrected words back

2. Scenario: Counting Words in Customer Reviews

Question: You want to analyze customer reviews. How will you count how many times each word is used?

1. **Convert text to lowercase**
   → “Good” and “good” become same
2. **Remove punctuation**
   → Clean text
3. **Split text into individual words**
   → Tokenization
4. **Use a dictionary or Counter to store counts**
   → {"good": 3, "bad": 1, ...}
5. **Display most frequent words**
   → Helps analyze customer sentiment

3. Scenario: Identifying Questions in Chat Messages

Question: In a chatbot, how do you know if the user typed a question or not?

1. **Check if the text ends with a question mark (?)**
2. **Look for question words**
   → who, what, why, how, when
3. **Check sentence structure**
   → "Can you...?" / "Do you...?"
4. **Use simple rules (rule-based classifier)**
5. **Label message as question or statement**

4. Scenario: Language Detection for Multilingual Inputs

Question: A user enters text on your website, and you need to detect the language. How would you do that?

1. **Take user input text**
2. **Compare patterns with known language models**
   → character frequency, word patterns
3. **Use a pre-trained language detection library**
   → langdetect, fastText
4. **Model predicts probability for each language**
   → {"en": 0.92, "fr": 0.05, ...}
5. **Pick language with highest probability**

5. Scenario: Creating Tags from Blog Articles

Question: You want to create automatic tags for blog articles. How can you generate useful keywords?

1. **Clean text**
   → remove punctuation, stopwords
2. **Tokenize and split words**
3. **Find frequently occurring important words**
   → TF-IDF or frequency count
4. **Pick top keywords**
   → "AI", "machine learning", "healthcare"
5. **Use these as automatic tags**

6. Scenario: Convert Text to Lowercase for Uniformity

Question: Why is converting all text to lowercase useful in NLP?

1. **Makes text consistent**
   → "Apple", "apple", "APPLE" become same
2. **Reduces vocabulary size**
3. **Improves search matching**
4. **Helps models learn better patterns**
5. **Simplifies NLP preprocessing**