Natural Language Processing

Q.No.1:- You are building a customer support chatbot for an e-commerce platform. The chatbot needs to handle a variety of user queries such as order status, product recommendations, and returns. The chatbot has trouble understanding ambiguous queries like:

1. "I want to return my order."
2. "Can you suggest a product for me?"

Questions:

1. **Syntax**: How would you analyze the syntactic structure of the sentence "I want to return my order"? Why might the chatbot have trouble with complex sentences that contain multiple clauses?
2. **Semantics**: If the chatbot mistakenly identifies "return" as a payment action instead of a product return, what semantic error could be causing this? How can semantic analysis be improved?
3. **Pragmatics**: A user types, "Can I get a refund?" The chatbot, however, replies with, "Please enter your order number." What pragmatic error is occurring, and how can the chatbot better handle user intent?
4. **Discourse**: After answering a customer's query about a refund, the chatbot doesn't refer back to the user’s previous query about order status. How does this failure to use **discourse** affect the conversation flow and user experience?

Q.No.2:- You are working in a research lab developing an NLP-based system to understand medical records for automatic diagnosis suggestions. The technology is currently in its early stages.

Questions:

1. Describe how NLP's evolutionary curves have changed over the past decade. How has the shift from rule-based systems to deep learning models like transformers impacted medical NLP systems?
2. Given that NLP models are still struggling with domain-specific vocabulary, such as medical terminology, what future directions do you foresee for improving NLP in specialized fields like healthcare and law?

3. Imagine the NLP system is still unable to accurately understand medical jargon. What current challenges would you face, and how might you overcome them in the next 3–5 years?

Q.No.3:- You are tasked with building a machine learning model for sentiment analysis on social media posts. The raw data includes various text elements like hashtags, mentions, emojis, and slang.

Questions:

1. You need to preprocess the text data before feeding it into the machine learning model. How would you use NLTK or SpaCy to:
    - Tokenize the text into words and sentences?
    - Remove unnecessary parts of the text (e.g., hashtags, mentions)?
    - Handle special characters like emojis and URLs?
2. You decide to use SpaCy for this task. Write a small code snippet that performs tokenization, removes stopwords, and converts the text to lowercase. Explain why these steps are necessary in the context of sentiment analysis.

Q.No.4:- You are working on building a recommendation system for a news aggregation website. The system recommends articles based on keyword matches from the articles' content. However, you're noticing that irrelevant words like "a", "the", "in", etc., and punctuation marks are cluttering up the data, affecting the system's accuracy.

Questions:

1. Why is it important to remove stopwords (like "a", "the", "is") from the text before processing? How does their presence negatively impact the performance of NLP models for tasks like keyword matching?
2. You are also instructed to remove punctuation from the dataset. What are the potential pros and cons of removing punctuation in text preprocessing, especially when working with tasks like keyword extraction or sentiment analysis?
3. After preprocessing, you notice that some meaningful words are still removed, like "not" in the sentence "This is not good." How would you handle negation in a preprocessing pipeline?

Q.No.5:- Imagine you're building an automatic email categorizer that classifies emails into "spam" and "ham" categories. The raw emails contain various issues like misspellings, unusual symbols (e.g., "!!!"), and concatenated words (e.g., "greatjob").

Questions:

1. Which preprocessing steps using NLTK or SpaCy would you perform to deal with:
   - Misspelled words?
   - Unusual symbols or noise?
   - Concatenated words like "greatjob"?
2. How would stemming or lemmatization help improve the categorization of emails?
3. Would you use word embeddings (Word2Vec or GloVe) for this email categorization task? Why or why not?