Section 1: Multiple Choice Questions

1. Which of the following NLP tasks is typically modeled as an unsupervised learning problem?

- A) Document Classification
- B) Sentiment Analysis
- C) Document Retrieval
- D) Topic Detection (using methods like LDA)

2. The process of breaking a stream of text into words, phrases, symbols, or other meaningful elements is called:

- A) Lemmatization
- B) Vectorization
- C) Tokenization
- D) Stemming

3. In a machine learning context, Document Classification is fundamentally a type of:

- A) Regression problem
- B) Supervised learning classification problem
- C) Unsupervised clustering problem
- D) Reinforcement learning problem

4. Which text preprocessing step involves removing common, low-information words like "the", "is", and "in"?

- A) Lowercasing
- B) Stop word removal
- C) Punctuation removal
- D) Stemming

5. Which text vectorization technique assigns a weight to a word based on its frequency within a specific document relative to its frequency across the entire corpus?

- A) Bag-of-Words (BoW)
- B) One-hot encoding
- C) Term Frequency-Inverse Document Frequency (TF-IDF)
- D) Word Embeddings

6. The primary goal of both stemming and lemmatization is to:

- A) Convert words to lowercase.
- B) Remove all punctuation from the text.
- C) Reduce different inflected forms of a word to a common base form.
- D) Assign numerical weights to words.

7. Sentiment Analysis is typically framed as a machine learning problem in order to:

- A) Rank documents by relevance to a query.
- B) Assign abstract topics to a document.
- C) Classify the emotional tone or polarity (e.g., positive/negative/neutral) of text.
- D) Convert text into a sequence of tokens.

8. When phrasing Document Retrieval as a machine learning problem, the primary task is often one of:

- A) Feature engineering
- B) Clustering
- C) Ranking
- D) Generation

9. Which of the following is NOT a necessary step in most standard NLP text preprocessing pipelines?

- A) Removing HTML tags/special characters

- B) Tokenization
- C) Training a new neural network model
- D) Vectorization using BoW or TF-IDF

10. Why is text converted into numerical vectors (vectorization) before being fed into a machine learning algorithm?

- A) Because ML algorithms are better at handling string data directly than numerical data.
- B) Because vectorization removes stop words automatically.
- C) Because ML algorithms require numerical input to perform mathematical operations.
- D) Because it simplifies the tokenization process.

Section 2: Short Answer Questions

11. Explain the difference between the Bag-of-Words (BoW) model and the TF-IDF model for text vectorization. How does TF-IDF improve upon the limitations of BoW?

12. Describe how a task like 'Spam Detection' in emails is framed as a machine learning problem. What type of learning is used, and what would the input features likely be after preprocessing?

13. What is the difference between *stemming* and *lemmatization*? Provide a single example word pair to illustrate the difference.

14. Explain the main challenge when framing Document Retrieval as a simple classification problem. Why is a "ranking" approach typically more effective in practice?

15. **List the five primary steps you would use to preprocess a raw news article text file before feeding it into an algorithm for sentiment analysis**.

16. Explain the fundamental difference between traditional static word embeddings (like Word2Vec or GloVe) and modern **contextualized word vectors** (like those generated by models such as BERT or ELMo).

Use the word "bank" in the following two sentences to illustrate how a contextualized approach resolves the limitation of a static approach:

- Sentence 1: "I need to go to the **bank** to deposit money."
- Sentence 2: "We sat by the river **bank** and watched the water flow."

17. Below are several common steps found in a typical Natural Language Processing (NLP) pipeline. Order these steps from the raw text input stage to the final stage just before a machine learning model receives the data.
Steps:

- A) Tokenization
- B) Feature Extraction / Vectorization (e.g., TF-IDF, Embeddings)
- C) Stop Word Removal / Normalization (Lemmatization/Stemming)
- D) Raw Text Input
- E) Punctuation and Special Character Removal

Correct Order (Fill in the blanks):
D) Raw Text Input → → → → → (To ML Model)

Question #	Correct Answer
1	D) Topic Detection (using methods like LDA)
2	C) Tokenization
3	B) Supervised learning classification problem
4	B) Stop word removal
5	C) Term Frequency-Inverse Document Frequency (TF-IDF)
6	C) Reduce different inflected forms of a word to a common base form.
7	C) Classify the emotional tone or polarity (e.g., positive/negative/neutral) of text.
8	C) Ranking
9	C) Training a new neural network model
10	C) Because ML algorithms require numerical input to perform mathematical operations.

Section 2: Short Answer Solutions

11. Explain the difference between the Bag-of-Words (BoW) model and the TF-IDF model for text vectorization. How does TF-IDF improve upon the limitations of BoW?

- **Difference:** BoW simply counts the frequency of each word in a document. It treats every word as equally important. TF-IDF calculates a weighted score for each word; it considers not only how often a word appears in a specific document (Term Frequency, TF) but also how rare the word is across the entire collection of documents (Inverse Document Frequency, IDF).
- **Improvement:** TF-IDF improves on BoW by down-weighting highly frequent but less informative words (like "said" or "people") that appear in almost every document, while giving higher weight to rare, meaningful keywords that help distinguish one document from another.

12. Describe how a task like 'Spam Detection' in emails is framed as a machine learning problem. What type of learning is used, and what would the input features likely be after preprocessing?

- **Framing:** Spam detection is framed as a **supervised binary classification** problem. The model is trained on a large dataset of emails already labeled as "Spam" or "Not Spam" (Ham).
- **Type of Learning:** Supervised Learning.
- **Input Features:** After preprocessing (tokenization, lowercasing, stop word removal), the emails would be converted into numerical features, most commonly using TF-IDF vectors or Bag-of-Words vectors, which represent the presence and frequency of specific words in each email.

13. What is the difference between *stemming* and *lemmatization*? Provide a single example word pair to illustrate the difference.

- **Stemming:** A heuristic process that chops off the end of words to reach a "stem" or root form, which may not be a real dictionary word. It is generally faster but less accurate.
- **Lemmatization:** A more sophisticated, dictionary-based process that uses vocabulary analysis and morphology to return the base or canonical form of a word (the lemma). It is generally more accurate but slower.
- **Example:**
    - *Input Word:* **"running"**
    - *Stemming Result:* **"run"** (A real word)
    - *Input Word:* **"am"** (as in the verb 'to be')
    - *Stemming Result:* **"am"** or **"a"** (varies by algorithm)
    - *Lemmatization Result:* **"be"** (The actual base form/lemma)

14. Explain the main challenge when framing Document Retrieval as a simple classification problem. Why is a "ranking" approach typically more effective in practice?

- **Main Challenge:** In a real-world scenario (e.g., searching the web), there are billions of documents. It's not efficient or practical to train a simple classifier to decide "relevant" or "not relevant" for every single document against every

possible user query. A classification model struggles with the scale and the sheer volume of "negative" (irrelevant) examples.

- **Ranking Approach:** A ranking approach (Learning-to-Rank) is more effective because it focuses on *relative* relevance. It learns a scoring function that sorts the documents based on their likelihood of matching the query, presenting the *best* options at the top. The model only needs to worry about the order of a manageable subset of documents, rather than an absolute binary classification of the entire collection.

15. List the five primary steps you would use to preprocess a raw news article text file before feeding it into an algorithm for sentiment analysis.

1. **Tokenization:** Split the continuous text into individual words or phrases.
2. **Lowercasing:** Convert all tokens to lowercase to standardize the input.
3. **Punctuation/Special Character Removal:** Remove symbols like commas, periods, HTML tags, etc., that don't convey sentiment.
4. **Stop Word Removal:** Eliminate common words (e.g., "the", "is", "a") that lack sentimental value.
5. **Vectorization (e.g., TF-IDF or Word Embeddings):** Convert the cleaned, tokenized text into a numerical format that the machine learning algorithm can process.

15. Explanation:
The fundamental difference lies in how a word's meaning is represented numerically.

- **Static Word Embeddings** (Word2Vec, GloVe): These models assign a *single, fixed vector* representation to a word regardless of the sentence it appears in. The word "bank" has the exact same vector in every context. This fails to capture polysemy (words having multiple meanings).
- **Contextualized Word Vectors** (BERT, ELMo): These models use deep learning architectures (usually transformers or LSTMs) to dynamically generate the word vector based on the *surrounding words* in that specific sentence. The representation changes depending on the context.

Illustration with "bank":

Sentence	Static Embedding	Contextualized Embedding
"I need to go to the **bank** to deposit money."	The vector for "bank" is the same as below.	The vector is numerically closer to words like "money", "ATM", and "finance".

"We sat by the river **bank** and watched the water flow."	The vector for "bank" is the same as above.	The vector is numerically closer to words like "river", "water", and "shore".

The contextualized model produces two distinct vectors for the word "bank", accurately reflecting its specific meaning in each sentence.

16. NLP Pipelines (Ordering)
Topic: The typical flow and steps within a Natural Language Processing (NLP) pipeline.
Steps:

- A) Tokenization
- B) Feature Extraction / Vectorization (e.g., TF-IDF, Embeddings)
- C) Stop Word Removal / Normalization (Lemmatization/Stemming)
- D) Raw Text Input
- E) Punctuation and Special Character Removal

Correct Order:
The correct sequential flow of a typical NLP preprocessing pipeline is:
D) Raw Text Input

A) Tokenization

E) Punctuation and Special Character Removal

C) Stop Word Removal / Normalization (Lemmatization/Stemming)

B) Feature Extraction / Vectorization (e.g., TF-IDF, Embeddings)

(To ML Model)