Introduction

January 18, 2025

```
[ ]: '''
        Natural Language Processing -->

        Natural Language Processing (NLP) is a branch of artificial intelligence
        that focuses on enabling computers to understand, interpret, and respond
        to human language in a valuable way. It combines linguistics, computer
        science, and machine learning to process and analyze large amounts of
        natural language data.
     '''
```

```
[ ]: '''
        Core Concepts in NLP -->

        Tokenization: Splitting text into words, phrases, or sentences.
        Stemming and Lemmatization: Reducing words to their root forms.
        Part-of-Speech (POS) Tagging: Assigning grammatical tags to words.
        Parsing: Analyzing the grammatical structure of a sentence.
        Word Embeddings: Representing words in a vector space
        (e.g., Word2Vec, GloVe).
        Transformer Models: Deep learning models like BERT, GPT, and T5
        for advanced NLP tasks.

        Python Libraries -->

        NLTK: A classic library for basic NLP tasks.
        spaCy: For industrial-strength NLP processing.
        TextBlob: Simplified NLP processing.
        Transformers (Hugging Face): For state-of-the-art models like
        BERT and GPT.
     '''
```

```
[ ]: '''
        Key Applications of NLP -->

        Text Classification:
        Categorizing text into predefined categories (e.g., spam detection,
        sentiment analysis).
```

```
    Machine Translation:
    Translating text from one language to another (e.g., Google Translate).

    Sentiment Analysis:
    Determining the sentiment behind text (e.g., positive, negative, neutral).

    Named Entity Recognition (NER):
    Identifying entities like names, dates, and locations in text.

    Speech Recognition:
    Converting spoken language into text.

    Question Answering:
    Building systems that can answer questions posed in natural language.

    Text Summarization:
    Automatically creating a summary of a larger text document.

    Chatbots and Virtual Assistants:
    Understanding user input and providing intelligent responses.

    Spell Checking and Grammar Correction:
    Detecting and suggesting corrections for text errors
'''
```

```
[ ]: '''
    Bag Of Words -->

    The Bag of Words (BoW) model is one of the simplest and most widely
    used techniques in Natural Language Processing for text representation.
    It is often used for tasks like text classification, sentiment analysis,
    and information retrieval.

    What is Bag of Words ?

    The Bag of Words model represents text as a collection of words,
    disregarding grammar and word order but keeping track of the frequency
    of each word in the text. The resulting representation is a sparse vector,
    where each dimension corresponds to a unique word in the corpus.
'''
```

```
[ ]: '''
    How Bag of Words Works -->

    Text Preprocessing:

    Convert text to lowercase (to handle case insensitivity).
    Remove punctuation, special characters, and numbers.
    Tokenize the text into individual words.
    Optionally remove stop words (common words like "the," "is," etc.).
    Apply stemming or lemmatization to standardize word forms.

    Vocabulary Creation:

    Compile a list of all unique words (the vocabulary) across the entire
    corpus (collection of documents).

    Vector Representation:

    For each document, count the frequency of each word in the vocabulary.
    Represent the document as a vector where each dimension corresponds to
    a word in the vocabulary, and the value is the word's count in the document.
'''
```

[ ]:
```
'''
    Bag Of Words Example -->

    Input Text:
    Document 1: "The cat sat on the mat."
    Document 2: "The dog barked at the cat."

    Preprocessing:
    Remove stop words: "cat sat mat" and "dog barked cat"

    Tokenize: ["cat", "sat", "mat"] and ["dog", "barked", "cat"]

    Vocabulary:
    ["barked", "cat", "dog", "mat", "sat"]

    Vector Representation:
    Doc1 : [0, 1, 0, 1, 1]
    Doc2 : [1, 1, 1, 0, 0]
'''
```

[1]:
```
#   Bag Of Words -->

from sklearn.feature_extraction.text import CountVectorizer

# Sample text corpus
documents = [
    "The cat sat on the mat.",
    "The dog barked at the cat."
]
```

```
# Initialize CountVectorizer
vectorizer = CountVectorizer(stop_words='english')

# Fit and transform the documents
X = vectorizer.fit_transform(documents)
```

```
[8]: # Display the vocabulary
     print("Vocabulary --> ", vectorizer.get_feature_names_out())
```

```
Vocabulary -->  ['barked' 'cat' 'dog' 'mat' 'sat']
```

```
[7]: # Display the vector representation
     print("BoW Representation -->\n\n", X.toarray())
```

```
BoW Representation -->

 [[0 1 0 1 1]
 [1 1 1 0 0]]
```

```
[ ]: '''
         Bag of Words is often replaced by more advanced techniques like
         Word Embeddings (Word2Vec, GloVe) and Transformers (BERT, GPT)
         for modern NLP tasks, as these models capture context and semantic
         meaning better. However, BoW remains a valuable tool for basic NLP
         tasks and as a baseline method.
     '''
```