Team 7:
SE20UCSE202
SE20UCSE119
SE20UCSE163
SE20UCSE014
SE20UCSE178
SE20UCSE014

NLP TEXT SUMMARIZATION PIPELINE

About the code:

1) Data Loading and Preprocessing:

- The code begins by importing necessary libraries such as pandas, nltk, scikit-learn, and others.
- It loads a dataset from a CSV file named "data.csv" and creates a dataframe called 'df' to hold the data.
- The 'abstracts' and 'titles' columns from the dataframe are extracted and stored as lists.
- Text preprocessing is performed using NLTK library. The 'preprocess_text' function converts text to lowercase, tokenizes it into sentences and words, removes stopwords and punctuation.
- Abstracts and titles are processed using the 'preprocess_text' function, resulting in 'processed_abstracts' and 'processed_titles' lists.

2) TF-IDF Feature Extraction:

- The code uses the TfidfVectorizer from scikit-learn to convert the preprocessed abstracts into a TF-IDF matrix called 'tfidf_matrix'.
- A helper function 'calculate_tfidf_scores' is defined to calculate the TF-IDF scores for each sentence in the abstracts.
- The TF-IDF scores are averaged across all other sentences to get a sentence score, indicating the importance of each sentence.
- The 'select_top_sentences' function selects the top 'n' sentences based on their TF-IDF scores.
- The top sentences are stored in the 'selected_sentences' list.
- Sentence Scoring and Summarization:

- The code calculates the TF-IDF scores for each sentence in the abstracts using the 'calculate_tfidf_scores' function.
- The 'select_top_sentences' function is called to select the top 3 sentences based on their TF-IDF scores.
- The selected sentences are stored in the 'selected_sentences' list.

3) Evaluation:

- The code evaluates the performance of the generated summaries by calculating the beLU (best Lower and Upper bounds) measure.
- The 'calculate_belue_measure' function calculates the beLU measure for each generated summary and its corresponding title.
- The beLU scores are stored in the 'belue_scores' list.

- The average beLU score is calculated by summing up all the beLU scores and dividing by the total number of scores.

4) Final Output:

The code prints the average beLU score, which is the performance measure for the generated summaries.

The code also prints the number of generated summaries and the number of processed titles, indicating the size of the dataset used (10,000 rows).

In conclusion, the code performs text preprocessing, calculates TF-IDF scores for each sentence in the abstracts, selects the top sentences based on their scores, evaluates the performance using the beLU measure, and prints the average beLU score (0.395) and the number of rows taken (10,000).