Text Summarization Using Sentence Similarity and PageRank

Introduction

The project focuses on automatic text summarization, which is the process of shortening a set of data computationally to create a summary that retains the most important information. This project utilizes a combination of natural language processing techniques, sentence similarity calculation, and the PageRank algorithm to generate summaries of articles.

Data Description

- **Dataset**: The dataset contains articles with multiple sentences. Each article is represented in a column named `article_text`.
- **Features**:
  - **article_text**: Contains the full text of the articles.

Steps and Methodology

1. **Importing Libraries**:
   - Imported necessary libraries such as numpy, pandas, nltk for text processing, and networkx for implementing PageRank.
2. **Loading the Dataset**:
   - Loaded the dataset using pandas.
   - Displayed the first article to understand the structure and content of the dataset.
3. **Sentence Tokenization**:
   - Tokenized the articles into individual sentences using `nltk.sent_tokenize`.
4. **Text Cleaning**:
   - Removed non-alphabetical characters and converted sentences to lowercase.
   - Removed stopwords using NLTK's predefined list of English stopwords.
   - Defined a function `remove_stopwords` to filter out stopwords from the sentences.
5. **Sentence Vectorization**:
   - Converted each sentence into vectors using word embeddings.
   - Calculated the average vector for each sentence by summing the vectors of words in the sentence and dividing by the number of words.
6. **Sentence Similarity Matrix**:
   - Created a similarity matrix using cosine similarity to measure the similarity between sentence vectors.
   - Filled the matrix where each element `(i, j)` represents the cosine similarity score between sentence `i` and sentence `j`.
7. **PageRank Algorithm**:
   - Constructed a graph using NetworkX where each node represents a sentence.
   - Applied the PageRank algorithm to score sentences based on their importance within the graph.
8. **Generating Summary**:

    - Sorted sentences by their PageRank scores in descending order.
    - Selected the top-ranked sentences to form the summary.

Results

1. **Example Article and Summary**:

    - For the first article:

Conclusion

- The text summarization model successfully identified and extracted the most important sentences from articles using sentence similarity and the PageRank algorithm.
- The summaries generated were coherent and captured the key points of the articles effectively.

Future Work

- **Model Improvement**: Incorporate advanced word embeddings like BERT or GPT to improve the quality of sentence vectors.
- **Summarization Techniques**: Explore abstractive summarization techniques in addition to extractive summarization for more natural summaries.
- **Dataset Expansion**: Test the model on larger and more diverse datasets to evaluate its robustness and scalability.
- **Real-Time Implementation**: Develop a web or mobile application to provide real-time text summarization for user-provided articles or documents.
- **User Feedback Loop**: Implement a feedback mechanism to refine and improve the summarization model based on user feedback.

Prepared Responses for Interview

1. **Introduction**: "This project aims to automatically summarize articles using natural language processing techniques, sentence similarity calculation, and the PageRank algorithm. The goal is to extract the most important sentences to form a concise summary."
2. **Data Description**: "The dataset consists of articles with multiple sentences, and each article is represented in a column named 'article_text'. We tokenize, clean, and preprocess these sentences for summarization."
3. **Steps and Methodology**: "We tokenized the articles into sentences, cleaned the text, removed stopwords, and converted sentences into vectors. We then calculated sentence similarity using cosine similarity, constructed a graph, and applied the PageRank algorithm to rank sentences based on their importance. The top-ranked sentences were selected to form the summary."
4. **Results**: "The summarization model effectively identified and extracted key sentences from the articles. The generated summaries were coherent and captured the main points of the articles."

5. **Conclusion**: "The project successfully implemented an extractive summarization technique using sentence similarity and PageRank. The model provided concise and relevant summaries of the articles."
6. **Future Work**: "Future improvements include using advanced word embeddings, exploring abstractive summarization techniques, expanding the dataset, developing real-time applications, and incorporating a user feedback loop for continuous improvement."