BUAN6342 / Individual project

Project Title: Automatic Text Summarization System

Scenario:

As a committee member responsible for summarizing daily news for morning meetings, you need to summarize the day's news for the committee and highlight the ten most important stories that you think the committee should focus on. With limited time to read every news article in detail, you want to use an automatic text summarization system to quickly generate a summary of the news headlines. This is where the automatic text summarization system developed in this project comes in handy. You can use the system to quickly generate a summary of the news headlines and identify the ten most important stories based on the summary. This will save you time and ensure that you are well-informed before the committee meeting.

Project Requirements:

Part One: NLTK

1. Import the required libraries such as Pandas, NLTK, and Scikit-learn.
2. Load the "headlines.csv" file into a Pandas data frame.
3. Perform text preprocessing on the headlines using NLTK functions such as removing stop words, tokenization, stemming, and lemmatization.
4. Perform Part-of-Speech (POS) tagging on the preprocessed headlines using NLTK's POS tagger.
5. Perform Named Entity Recognition (NER) on the preprocessed headlines using NLTK's NER tagger.
6. Use CountVectorizer and TF-IDF vectorizer for classification based on content.
7. Generate a summary of the news headlines using the TextRank algorithm or other summarization techniques. The summary will consist of the top 10 most important sentences in the article.
8. Visualize the summary using matplotlib or other visualization tools.

Part Two (spaCy and scikit-learn):

1. Utilize spaCy for Part Two.
2. Use the same "headlines.csv" dataset as the input file.
3. Preprocess the headlines using spaCy functions such as removing stop words, tokenization, stemming, and lemmatization.
4. Perform Part-of-Speech (POS) tagging on the preprocessed headlines using spaCy's POS tagger.
5. Perform Named Entity Recognition (NER) on the preprocessed headlines using spaCy's NER tagger.
6. Utilize spaCy's pipeline object to efficiently perform text processing tasks.
7. Use scikit-learn's TfidfVectorizer and LinearSVC in a pipeline for text vectorization and classification.
8. Generate a summary of the news headlines using the TextRank algorithm or other summarization techniques. The summary will consist of the top 10 most important sentences in the article.
9. Visualize the summary using matplotlib or other visualization tools.

Conclusion

After comparing the output from NLTK and spaCy visually, conclude one of the following:

a) Both libraries (spaCy & NLTK) are effective for text summarization, with NLTK being slightly more efficient.

b) Both are effective, but spaCy outperforms NLTK.

c) Both libraries successfully generate summaries that meet the committee's needs, and the summaries highlight the top 10 most important stories of the day's news. Therefore, depending on the specific project needs, either NLTK or spaCy could be chosen for text summarization.

Delivery:

1. Provide a complete Python program for each part within the same Jupyter notebook. Use a heading and step numbering to separate and identify each step and its associated code (`# 2.4 Load the "headlines.csv" file into a Pandas dataframe.`)
2. At the end of the Jupyter notebook, display two grids side by side to present the 10-line summaries. The left grid should show the results for NLTK, while the right grid should show the results for spaCy.
3. Power Point presentation

   Prepare a PowerPoint presentation that includes the following slides:

   a. Slide 1: Highlighting the importance and role of preprocessing steps like POS and NER. Explain why these steps are crucial and how they contribute to the summarization process.
   b. Slide 2: Providing flow charts for each method (boxes and arrows), illustrating the workflow of NLTK and spaCy summarization processes.
   c. Slide 3: Displaying side-by-side output comparisons between NLTK and spaCy summarization results.
   d. Slide 4: Presenting the conclusion, specifying which library is more efficient and practical for the text summarization task.

   Special Note:

   After grading, a selection of five students will be invited to present their analysis and insights, showcasing their technical and analytical expertise.

Grading:

Delivery #1 (30 points)

Delivery #2 (30 points)

Delivery #3 (40 points)