REPORT

By Abhiram Vadrevu

1. Dataset Loading:
   * The code starts by loading the "cnn\_dailymail" dataset using the **load\_dataset** function from the Hugging Face library. This dataset is commonly used for text summarization tasks and contains articles from the CNN and Daily Mail news sources.
2. Dataset Exploration:
   * The code prints the available features in the dataset to provide an overview of the data structure. It helps in understanding the columns and their names, such as "article" and "highlights," which are relevant for the summarization task.
   * An example article and its corresponding summary are displayed to provide a visual understanding of the data. This sample serves as a reference for evaluating the generated summaries later.
3. Baseline Summary:
   * The code defines a function **baseline\_summary\_three\_sent** that generates a baseline summary by extracting the first three sentences from the article. This simple approach serves as a reference baseline for the summarization task.
   * The function is applied to the sample article, and the generated baseline summary, consisting of the first three sentences, is stored in the **summaries** dictionary.
4. GPT-2 Summary:
   * The code utilizes the GPT-2 model from the Transformers library to generate a summary for the sample article. GPT-2 is a powerful language model capable of generating coherent text.
   * The GPT-2 model is loaded through the **pipeline** function, specifying the "text-generation" task.
   * The GPT-2 model is fed the sample article as input, with an additional prompt "TL;DR:" to guide the model to generate a summary.
   * The generated summary from GPT-2 is extracted from the model's output and stored in the **summaries** dictionary.
5. T5 Summary:
   * The code uses the T5 model from Transformers to generate a summary for the sample article. T5 is a transformer-based model known for its strong performance in text generation tasks.
   * Similar to GPT-2, the T5 model is loaded through the **pipeline** function, specifying the "summarization" task.
   * The T5 model is provided with the sample article as input, and it generates a summary using its pre-trained knowledge.
   * The generated summary from T5 is extracted and stored in the **summaries** dictionary.
6. BART Summary:
   * The code employs the BART model from Transformers to generate a summary for the sample article. BART is a transformer-based model specifically designed for sequence-to-sequence tasks like text summarization.
   * The BART model is loaded through the **pipeline** function, specifying the "summarization" task.
   * The sample article is passed to the BART model, and it generates a summary based on its understanding of the input text.
   * The generated summary from BART is extracted and stored in the **summaries** dictionary.
7. Pegasus Summary:
   * The code utilizes the Pegasus model from Transformers to generate a summary for the sample article. Pegasus is a state-of-the-art transformer-based model known for its exceptional performance in abstractive summarization tasks.
   * The Pegasus model is loaded through the **pipeline** function, specifying the "summarization" task.
   * The sample article is fed to the Pegasus model, which generates a summary by leveraging its large-scale pre-training and fine-tuning on summarization tasks.
   * The generated summary from Pegasus is extracted and stored in the **summaries** dictionary.
8. Ground Truth:
   * The code prints the ground truth summary for the sample article. This is the actual summary provided in the dataset, serving as a reference for evaluating the generated summaries.
9. BLEU Metric:
   * The code loads the BLEU metric from the sacrebleu library, which is a popular metric for evaluating the quality of machine-generated summaries by comparing them to reference summaries.
   * The BLEU score is calculated between the Pegasus-generated summary and the ground truth summary. BLEU measures the precision of the generated summary by comparing n-gram overlap with the reference summary.
   * The precision values are rounded and stored in the **bleu\_df** DataFrame for further analysis and comparison.
10. ROUGE Metric:
    * The code initializes the ROUGE metric from the datasets library, which is another widely used evaluation metric for text summarization tasks. ROUGE measures the overlap of n-grams, word sequences, and skip-bigrams between the generated summary and the reference summary.
    * The ROUGE scores are calculated for all the generated summaries (including the baseline) using the ROUGE metric.
    * The ROUGE scores for each model (GPT-2, T5, BART, Pegasus) are stored in a DataFrame for further analysis and comparison.
11. Report:
    * The code prints the generated summaries for each model (including the baseline) along with the respective model names. This provides an overview of the summarization performance of each model.
    * It displays the BLEU score in the **bleu\_df** DataFrame, providing a quantitative measure of the precision of the Pegasus-generated summary compared to the ground truth summary.
    * Finally, it shows the ROUGE scores for each model in the form of a DataFrame, offering a detailed evaluation of the summaries based on different ROUGE metrics (ROUGE-N, ROUGE-L, ROUGE-SU).

The code provides a comprehensive evaluation of various state-of-the-art models (GPT-2, T5, BART, Pegasus) in generating summaries for the given dataset. It uses both the BLEU and ROUGE metrics to assess the quality of the generated summaries, providing valuable insights into the performance of each model. These metrics help in comparing the summarization capabilities of different models and understanding their strengths and weaknesses in capturing important information from the source text.