IR Assignment - 4

Saksham Pandey - 2021486

1. Processing Steps

Downloading NLTK Resources:

- punkt: A tokenizer for splitting text into a list of sentences.
- stopwords: Commonly used words ('the', 'is', etc.) that usually have little lexical content.
- wordnet: A lexical database for the English language which helps in lemmatization.

Loading the Data:

- The dataset Reviews.csv is loaded from the ./Dataset/ directory. A random sample of 0.5% (frac=0.005) of the data is taken to simplify the processing load.

Dropping Missing Values:

- Any reviews with missing 'Text' or 'Summary' fields are dropped to ensure data integrity.

Text Cleaning Function:

- clean_text: Defines a function to clean the text fields.
- Removes any character that is not alphanumeric (using a regular expression).
- Converts the text to lowercase and splits it into words (tokenization).
- Removes common English stopwords.
- Applies lemmatization to each word to reduce it to its base or root form.

Applying the Cleaning Function:

- The clean_text function is applied to both the 'Text' and 'Summary' columns of the dataset.

2. Saving the Preprocessed Files

The preprocessed fraction of the dataset was stored in a separate csv file for further processing and model training

3. GPT-2 Fine-Tuning for Text Summarization

Tokenization:

- The GPT2Tokenizer is loaded from the pre-trained 'gpt2' model.
- This tokenizer converts text to a format suitable for input to the GPT-2 model.

Dataset Splitting:

- The dataset is split into training and testing sets using train_test_split with 25% of the data reserved as the test set.

Custom Dataset Class

- ReviewSummaryDataset:

    - A custom PyTorch Dataset class to handle the tokenization and preparation of data for the model.
    - The class takes lists of texts and summaries and a tokenizer, tokenizing and encoding them into the required format for GPT-2.
    - The dataset returns input_ids and attention_mask for each item.

Model Initialization and Training

- Model Initialization:
    - The GPT2LMHeadModel is instantiated from the pre-trained 'gpt2' configuration.
- DataLoader Setup:
    - A DataLoader is created to handle batching and shuffling of the data during training.
- Training Loop:
    - The model is trained for several epochs using the AdamW optimizer.
    - Loss is calculated and optimized for each batch.

4. GPT-2 Summarization Evaluation

Model and Tokenizer Initialization:

- The GPT-2 model and tokenizer are loaded from the pre-trained 'gpt2' configuration. The model is set to evaluation mode to disable training-specific operations like dropout.

Data Preparation:

- The ReviewSummaryDataset is instantiated with test data, and a DataLoader is created to manage batching.

Summary Generation Function:

- A function to generate summaries from the test dataset using the model. The function handles model inference, generating outputs for given input data, and decoding these outputs back into text.

ROUGE Score Calculation:

- After generating summaries, their quality is assessed using the ROUGE metric which compares them to actual summaries. This function calculates and returns ROUGE-1, ROUGE-2, and ROUGE-L scores.