CSE508 Information Retrieval

Assignment - 4

> Ankit Kumar 2021015

This script is designed for cleaning, preprocessing, and training a summarization model using Amazon reviews. It involves the use of GPT-2 from the Hugging Face library and evaluates the performance of the model using ROUGE metrics.

Prerequisites

- Python environment with packages: `pandas`, `bs4`, `nltk`, `transformers`, `sklearn`, `torch`, and `rouge_score` installed.
- An `Reviews.csv` file containing Amazon reviews with '`Text`' and '`Summary`' columns.

Usage

Step 1: Data Preprocessing This step includes cleaning HTML tags from the text, removing stopwords, tokenizing, and normalizing the text. This is performed using the BeautifulSoup library for HTML cleaning and NLTK for tokenization and stopword removal.

Key Functions:

- `clean_html(text)`: Removes HTML tags using BeautifulSoup.
- `tokenize_and_remove_stopwords(text)`: Tokenizes the text and removes stopwords.
- `preprocess_text(text)`: Integrates all preprocessing steps into one function.
- `SummaryDataset`: A class to store the processed text and summary.
- `compute_rouge_scores(model, dataset, tokenizer, dataframe)`: function is designed to evaluate the quality of text summaries generated by a model against reference summaries. It uses the rouge_score library, which provides an implementation of the ROUGE

Step 2: Model Initialization and Data Preparation

- Initialize a GPT-2 model and tokenizer.
- Split the dataset into training (75%) and testing (25%) sets.
- Implement a custom dataset class for use in training.

Step 3: Model Training

- Set up training arguments and start the training process using the Hugging Face Trainer API.
- The model is fine-tuned on the preprocessed Amazon reviews.

Step 4: ROUGE Evaluation

- Summary Generation: For each entry in the dataset, the function generates a summary using the provided model and tokenizer.

- Score Computation: It then computes the ROUGE scores by comparing each generated summary to its corresponding reference summary. The scores include precision, recall, and F1-score for three types of ROUGE metrics:

  - ROUGE-1: Measures the overlap of unigrams between the generated and reference summaries.
  - ROUGE-2: Measures the overlap of bigrams, providing insight into the sequential word agreement.
  - ROUGE-L: Measures the longest common subsequence, focusing on the longest co-occurring sequence of words in the summaries.

- Data Aggregation: The computed scores for each summary are collected into a list of dictionaries.

- DataFrame Compilation: This list is converted into a DataFrame and concatenated with the original DataFrame, enhancing it with detailed ROUGE scores for further analysis.

- Return Value: Returns an enhanced DataFrame that includes the original data along with detailed ROUGE scores for each record.

Output

I've run the script twice on different datasets and different setting and the output is saved with various naming conventions. The output includes the following files:

- `final_rouge_scores.csv`
- `final_rouge_scores_2nd_Setting.csv`

There are two different script notebooks, one for each setting. The first setting uses the default GPT-2 model, while the second setting uses a custom GPT-2 model with a smaller size. epoch is also reduced to 1 for faster training of larger dataset.

- code.ipynb
- code1.ipynb

Sample Output is as follows code.ipynb

Id	ProductId	UserId	ProfileName	HelpfulnessNumera+E1:Z4tor	HelpfulnessDenominator	Score	Time	Summary	Text

Id	ProductId	UserId	ProfileName	HelpfulnessNumera+E1:Z4tor	HelpfulnessDenominator	Score	Time	Summary	Text
0	927	B000ER6YO0	A2F0WNTW3QQZYS	Ruth	0	0	5	1298505600	One of our favorites
1	631	B000G6RYNE	A1IVFBJA9KAI1M	Shane Martin	2	3	4	1191369600	Tasty!
247	110	B001REEG6C	AY12DBB0U420B	Gary Peterson	0	0	5	1316390400	My Idea of a Good Diet Food.

code1.ipynb

Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text	If you don't mind the inevitabl increase can...
0	476269	B000SQLQ0Y	A31RSJTGLVV3TR	T. Wayne	5	8	1	1304812800	Made in China - With CANCER	While I did not a any time imagine that I was...
1	288473	B000ENUC3S	A2QN7FECIWB7D2	Pm Rodgers "pmiker"	0	1	5	1312070400	Real Cherry Flavor	

I achived the best model performance with the following hyperparameters:

```
training_args = TrainingArguments(
    output_dir='./results',
    num_train_epochs=1,  # Reduced for faster testing cycles
    per_device_train_batch_size=2,  # Reduced to ensure it fits into CPU memory
    per_device_eval_batch_size=2,
    warmup_steps=100,  # Reduced warmup steps
    weight_decay=0.01,
    logging_dir='./logs',
    logging_steps=10,
    evaluation_strategy="epoch",  # Evaluate at the end of each epoch to save time during training
    save_strategy="no",
    load_best_model_at_end=False,
)
```

Thankyou

- Ankit
- 2020115
- 17.04.24