Paraphrase generator - Overview

Script Overview

The Python script is designed to accept a user-provided text paragraph, validate its length, and then generate multiple paraphrases using both a custom-trained Pegasus model and OpenAI's GPT-3 model. It further evaluates these paraphrases using BLEU scores for quality and computes semantic similarity using a Sentence Transformer model. The script outputs a series of paraphrases along with their quality metrics and processing latencies.

Key Components

1. **Text Preprocessing and Validation**:
    - **Preprocess Text**: Cleans the text by removing special characters and excessive spaces.
    - **Validate Input**: Checks if the processed text meets the word count requirement (200-400 words).
2. **Paraphrase Generation**:
    - **Pegasus Paraphrase Generator**: A class that loads and operates the Pegasus model specifically fine-tuned for paraphrase tasks. It generates multiple paraphrase outputs(5 outputs in this case).
    - **OpenAI Paraphrase Function**: Uses OpenAI's GPT-3 model to generate a paraphrase through a structured chat prompt.
3. **Quality and Performance Metrics**:
    - **BLEU Score Calculation**: Assesses the quality of the paraphrases relative to the GPT-generated text.
    - **Semantic Similarity**: Uses Sentence Transformer model embeddings to calculate cosine similarities, indicating the semantic closeness between the original GPT-3 paraphrase and the Pegasus-generated paraphrases.
4. **Latency Measurement**:
    - Captures the time taken to generate paraphrases with each method, providing insights into the performance efficiency of each approach.

Operation Flow

1. **User Input**: The script prompts the user to enter a text paragraph.
2. **Text Processing**: Text is preprocessed and validated for word count.
3. **Paraphrase Generation**:
    - Multiple paraphrases are generated using the Pegasus model.
    - A single paraphrase is generated using OpenAI's GPT-3.
4. **Evaluation**:
    - Each Pegasus-generated paraphrase is evaluated against the GPT-3 paraphrase using BLEU scores and semantic similarity metrics.

5. **Output**:
    - The script outputs detailed information about each paraphrase, including the text, latency for generation, BLEU scores, and semantic similarity scores.

      Sample output structure,

      ```
      {

        "cpg_output": ["list of paraphrased texts"],

        "cpg_latency": "processing time for generating paraphrases with the Pegasus model",

       "openai_paraphrase": "paraphrased text generated by OpenAI's GPT-3 model",

        "openai_latency": "processing time for generating a paraphrase with OpenAI's GPT-3",

        "bleu_scores": ["list of BLEU scores comparing each Pegasus paraphrase with the GPT-3 paraphrase"],

        "similarity_scores": ["list of semantic similarity scores comparing each Pegasus paraphrase with the GPT-3 paraphrase"]

      }
      ```

Files & instructions

1. Python script file
   Install requirements.txt
   Run - python paraphrase.py
2. FastAPI project
   Install requirements.txt
   Run - fastapi dev main.py
   Once its running goto Swagger using http://localhost:8000/docs#