Automating Question Generation for Documents with GPT-3.5-Turbo by Jacamein

Description

The Importance of Semantic Search

In large databases like the "Nerd Brain" we're building, information retrieval can become increasingly challenging. Semantic search improves this by understanding the context and intent behind a query, offering more accurate and relevant results.

Utility of the Code

This Python code automates a crucial aspect of enhancing semantic search: the generation of questions for each document based on its content. By placing these questions at the beginning of each document, it becomes easier for the search algorithm to understand the context and content of the document, thus improving the quality of search results.

This simple yet powerful code can save you hours of manual work. It uses OpenAI's GPT-3.5-Turbo to automatically generate and insert questions into your documents, making them more searchable and accessible.

Complete Python Code

```
from dotenv import load_dotenv
import os
import openai
import time

# Load environment variables
load_dotenv()

# Set up OpenAI API key
openai.api_key = os.environ.get("OPENAI_API_KEY")

# Directory where the documents are located
dir_path = "/path/to/your/directory"

start_time = time.time()

# Loop through each file in the directory
for idx, filename in enumerate(os.listdir(dir_path)):
    if filename.endswith(".md"):
```

```
        filepath = os.path.join(dir_path, filename)

        # Read the file content
        with open(filepath, 'r') as f:
            content = f.read()

        # Take a snippet from the content to generate more specific
questions
        snippet = content[:100]

        # Generate the prompt
        prompt = f"Based on the following snippet of the document,
generate a list of hypothetical questions that someone might ask to find
this content:\n\n{snippet}"

        # Generate questions using GPT-3.5-Turbo
        response = openai.ChatCompletion.create(
            model="gpt-3.5-turbo-16k",
            messages=[
                {"role": "system", "content": "You are assisting in
generating questions."},
                {"role": "user", "content": prompt}
            ],
            max_tokens=500
        )

        questions = response['choices'][0]['message']
['content'].strip().split("\n")[:10]

        # Add the questions to the beginning of the file content
        new_content = "\n".join(questions) + "\n\n" + content

        # Save the new content in the file
        with open(filepath, 'w') as f:
            f.write(new_content)

        # Time logging
        elapsed_time = time.time() - start_time
        print(f"Document {idx+1} processed. Elapsed time:
{elapsed_time:.2f} seconds")
```

Steps for Installation from Scratch

1. **Install Python**: If you don't already have Python installed, you can download it from here.
2. **Install Pip**: Pip usually comes with Python. If you don't have it, you can install it following these instructions.
3. **Install OpenAI Library**: Open your terminal and run `pip install openai`.

4. **Install `python-dotenv`**: This package is used for handling environment variables. Install it with `pip install python-dotenv`.
5. **Create a `.env` File**: Create a file in the same directory as your script and name it `.env`. Inside this file, add your OpenAI API key as follows: `OPENAI_API_KEY=your_key_here`.
6. **Run the Script**: Navigate to the directory where your script is located and run `python script_name.py`.