CPSC 4830

FINAL

Aug 12, 2024

Start: 6.30PM **Due: 9.45 PM**

Using GPT API for SQL Query Generation and PDF Word Similarity Analysis

---

Task 1: Text-to-SQL Query Generation (25 Marks)

Description:

Develop a Python-based application within a Jupyter Notebook that uses the GPT API to convert natural language queries into SQL queries. These SQL queries should be executed on a database containing credit card fraud data, and the results should be displayed as a DataFrame.

Dataset: You will be using the Credit Card Fraud dataset available at: Credit Card Fraud Data.

Steps and Requirements:

1. **Set Up the Database:**
   - Create a **Jupyter Notebook** using Python and read `Credit_Card_Fraud.csv` file.
   - Load the csv file into a database (SQLite).
   - Ensure the schema is correctly defined with appropriate data types during the import process.
2. **Text-to-SQL Functionality:**
   - Write a Python script in the notebook using the GPT API to convert user-provided natural language questions into corresponding SQL queries.
   - Execute the generated SQL queries on the database and display the filtered or aggregated data as a pandas DataFrame in the notebook. For example, if the user write a question “filter the fraud records in the year 2019”, it should first display the SQL code → “SELECT * FROM FRAUD_TABLE WHERE IS_FRAUD = 1” then it should the filtered/aggregated data as pandas Dataframe.
3. **SQL Query Display:**
   - Along with the data results, the notebook should also display the SQL query that was used to retrieve the data.
4. **Documentation and Explanation:**
   - Document your code and provide explanations within the notebook for each step.
   - Explain how the GPT API is used for text-to-SQL conversion.

Task 2: Word Similarity Extraction from PDFs and Word Cloud Visualization (25 Marks)

Description: Develop a Python script within a Jupyter Notebook to process and analyze text from three provided PDFs (in D2L). The PDFs are:

1. "Mapping-Cost-of-Balanced-Diet-December-2014.pdf" - A food and travel blog.
2. "Air_Canada_Booking_Confirmation.pdf" - An airline booking receipt.
3. "Receipt_22Jun2022.pdf" - A food delivery receipt.

The objective is to extract text from these PDFs, identify words similar to specified keywords, and visualize them.

Steps and Requirements:

1. **PDF Extraction:**
   - Use a suitable library (such as PyPDF2, pdfplumber or any pdf library of your choice) to extract text from the PDFs.
   - Reference (PyPDF2): https://pypi.org/project/PyPDF2/
   - Reference (pdfplumber): https://pypi.org/project/pdfplumber/
   - Ensure the text from all PDFs is clean and well-structured.
2. **Finding Similar Words in the Blog PDF:**
   - Utilize the Gensim library with the GloVe model (`glove.6B.50d.txt`) to analyze the text from the "Mapping-Cost-of-Balanced-Diet-December-2014.pdf" blog. **Reference:** https://machinelearningmastery.com/develop-word-embeddings-python-gensim/
   - Identify words similar to the keywords "FOOD" and "RESERVATION".
   - Store these similar words in two separate Bag of Words (BoW) models.
3. **Word Cloud Visualization:**
   - Create a word cloud visualization from the words in the BoW models for the blog PDF.
   - Ensure the word cloud clearly highlights the most frequent words.
4. **Analyzing Additional PDFs with GPT API:**
   - Use the GPT API to find words similar to "FOOD" and "RESERVATION" in the "Air_Canada_Booking_Confirmation.pdf" and "Receipt_22Jun2022.pdf".
   - Display the list of similar words extracted from these additional PDFs.
5. **Documentation and Explanation:**
   - Document your code with explanations in the notebook for each step.
   - Describe how Gensim was used to find word similarities and how the GPT API was applied to analyze the additional PDFs.

Deliverables:

- Jupyter Notebook File (.ipynb): Containing the application code and all steps outlined above.

- Extracted Text: Well-structured text from all PDFs displayed in the notebook.
- Word Cloud Visualization: An image showcasing the similar words identified from the blog PDF.
- List of New Similar Words: A list of new similar words found using the GPT API from the additional PDFs.
- Detailed Explanations: Markdown cells explaining each code block and its functionality.

Submission:

- Submit your .ipynb file with all code, markdown explanations, and results.
- Ensure the notebook is well-documented and easy to follow.