Document Management and Analysis Project

Overview

This project focuses on developing a Python-based text analysis tool that leverages MongoDB for efficient data storage and retrieval.

The tool's core functionality includes processing documents and empowering users to perform complex queries. These queries can be used to locate specific terms within multiple documents, calculate term frequency, identify the most common words, and evaluate similarity and dissimilarity metrics.

Team Members

Name	ID
أحمد محمد محمود محمود محمدى	22010038
عبدالله عاطف خميس عبد الونيس	22010140
احمد عماد عبد الفتاح عبد الغني	22010027
الحسين ياسر إبراهيم السيد	22010056
عمر ايهاب محمد جنيدي	22010162

Key Features

- **Document Management:** The system will allow users to upload various text-based documents or paragraphs, such as PDF, DOCX, or TXT files, as input. Additionally, it will provide functionality to delete documents, enabling efficient management of stored files.
- **Metadata Extraction and Storage:** Essential metadata about each document (e.g., no. of pages, type, modify date) will be extracted and stored alongside the document content in a MongoDB database.
- **Regex-Powered Queries:** Users will be able to construct and execute sophisticated queries using regular expressions to extract valuable insights from the text data.

- **Searching**: Users can search within the document database for specific document contents or metadata.
- **Common Word Identification:** Users can identify the most frequently occurring words across the specified document.
- **Term Frequency Analysis:** The tool will provide functionality to calculate the frequency of specific terms or phrases within the documents.
- **Similarity Analysis**: Users can compare two documents to evaluate their similarity using distance measures such as cosine similarity and Jaccard similarity. The comparison is based on numerical representations of words, such as Bag of Words vectors.
- **Dissimilarity Calculation**: The system also calculates the dissimilarity between the two documents using the Euclidean distance between their numerical representations.

Technical Components

- **Python:** The primary programming language for developing the application logic and text processing functionalities.
- **MongoDB:** A NoSQL database for storing and managing the document collection and metadata.
- **Regex Library:** The Python `re` module for powerful pattern matching and text manipulation using regular expressions.
- **Math Library:** The Python `Math` module was necessary in implementing the various similarity and dissimilarity calculations.
- **User Interface:** A command-line interface for user interaction and query execution.

Project Workflow

1. **Document Input:** Users will provide text documents for processing and storage.
2. **Metadata Extraction:** The system will automatically extract relevant metadata from the input documents.
3. **Data Storage:** Both the document content and metadata will be stored in a structured format within the MongoDB database.
4. **Query Interface:** Users will interact with the system to construct and execute regex or math based queries.
5. **Result Processing:** The system will process the queries and retrieve the relevant information from the database.

Potential Applications

- **Academic Research**: Enables researchers to analyze large volumes of academic papers, identify patterns, and compare textual data.
- **Corporate Document Management**: Assists businesses in managing, searching, and analyzing corporate documents and reports effectively.
- **Legal Analysis**: Supports legal professionals in comparing contracts, analyzing legal documents, and extracting relevant clauses.
- **Market Analysis**: Facilitates marketing teams in analyzing customer reviews, product descriptions, and other market-related textual data.

Project Timeline

- **Phase 1:** Requirements gathering, system design, and database setup.
- **Phase 2:** Implementation of core database operations, text processing, text comparison and query functionalities.
- **Phase 3:** Integration and assimilation of the disconnected functionalities into one connected component.
- **Phase 4:** Development of a command-line-interface and testing.
- **Phase 5:** Deployment and documentation.

Conclusion

The Document Management and Analysis Project combines the versatility of Python with the efficiency of MongoDB to provide a comprehensive solution for managing and analyzing textual data. With its robust features, including document management, metadata extraction, regex-powered queries, and advanced similarity metrics, the tool empowers users to handle complex text analysis tasks efficiently, facilitating the process of managing and querying documents.
