Lab 5 Part 1 - Oil Wells Data Extraction

Important Files and Folders

1. `.env`: File with environment variables.
2. `test-5681a-202ae82505a3.json`: Service Account Key for Authenticating Google Drive API.
3. `requirements.txt`: File with dependencies to be installed for the project.
4. `docs/`: Documentation including meeting minutes and README in PDF format.
5. `notebooks/`: Experimental usage and testing of concepts.

Setup

1. Navigate to the `oil-wells-data-scraper` directory:

   ```
   cd oil-wells-data-scraper
   ```

2. Create and activate a virtual environment:

   ```
   virtualenv venv
   source venv/bin/activate  # or "venv\Scripts\activate" on Windows
   ```

3. Install the necessary libraries:

   ```
   pip install -r requirements.txt
   ```

Running the Script

1. Ensure that:

   - All the important files and folders listed above are present at the correct location.
   - The virtual environment is created and activated.

2. To run the script:

   - Move into the `/src` directory:

     ```
     cd src
     ```

   - Execute the `<file_name>.py` script:

     ```
     python <file_name>.py
     ```

Execution Flow

1. **PDF Extraction and Conversion**:
    - Run `pdf2txt.py` to download PDF files from Google Drive, convert them into small-sized text files, and store them locally for further processing.
2. **Data Processing and Database Ingestion**:
    - Run `ingest.py` to process raw text files, extract relevant information, back it up in the database, filter out usable columns, extract additional data from URLs present in raw data, validate the data, and push it into the database.

About Scripts

File Name	Purpose
`crud.py`	Inserts scraped post information in bulk into the database, discarding posts that already exist.
`database.py`	Establishes a connection between Python and the SQL server.
`ingest.py`	Ingests raw and clean data into the database.
`extract.py`	Cleans and preprocesses text from scraped Reddit posts and linked websites, and extracts top keywords.
`pdf2txt.py`	Abstraction to download PDF documents from Drive, convert them to text files, and store them locally.
`model.py`	Defines schema for data storage in the MySQL server. Creates tables if they don't exist.
`schema.py`	Creates Pydantic models to validate data format before storing it in the database.
`settings.py`	Sets up access to certificates and environment credentials required for connecting to the MySQL database.