Reddit Data Processing and Clustering System (DSCI560_lab5)

This project implements an end-to-end Reddit data collection and clustering pipeline. It integrates with the Reddit API via **PRAW**, performs comprehensive text and image preprocessing (including OCR with Tesseract), stores data in an **SQLite database**, and applies machine learning algorithms (TF-IDF, Doc2Vec, KMeans) for clustering and analysis. The system supports both single-run and automated periodic processing.

---

1. Environment Setup

Requirements

- **Python**: 3.9 – 3.13 (recommended 3.11+)
- **SQLite3** (default on macOS/Linux, required for Windows users)
- **Tesseract OCR** (for image text extraction)

Install Tesseract OCR

On macOS (Homebrew):
```bash
brew install tesseract
```

On Ubuntu/Debian:
```bash
sudo apt-get install tesseract-ocr
```

Clone Repository and Virtual Environment

```bash
git clone
cd DSCI560-lab5

python3 -m venv .venv
source .venv/bin/activate
```

Install Dependencies

```bash
pip install --upgrade pip setuptools wheel
pip install -r requirements.txt
```

```

### *Download NLP Corpora*

Required for **TextBlob** and **NLTK**:
```bash
python -m textblob.download_corpora
# or

python -m nltk.downloader punkt brown
```

---

2. Data Processing Workflow

2.1 Single Run

Execute the entire pipeline for one batch of Reddit posts:
```bash
python direct_iphone_processing.py
```

Steps performed:
1. Fetch Reddit posts (via PRAW API).
2. Preprocess data (clean HTML/Markdown, mask usernames, timestamp conversion).
3. Extract text from images using **pytesseract** OCR.
4. Generate embeddings (TF-IDF, Doc2Vec).
5. Train and apply **KMeans clustering**.
6. Store results in SQLite database and output JSON/PKL files.

2.2 Automated Periodic Processing

Schedule multiple runs with custom interval and post count:
```bash
python automated_processor.py --subreddit iphone --posts 200 --interval 30 --runs 3
```
Parameters:
- `--subreddit`: Target subreddit (e.g., `iphone`)
- `--posts`: Number of posts per run (supports >1000 with pagination)
- `--interval`: Interval between runs (minutes)
- `--runs`: Total number of runs

---

3. Output Structure
```

All results are written to `reddit_data/`:

- **embeddings/**
- `*_clustering.pkl`: KMeans clustering model
- `tfidf_vectorizer.pkl`: TF-IDF vectorizer
- `doc2vec_model.pkl`: Trained Doc2Vec model

- **metadata/**
- `*_clusters.json`: Cluster assignments (post_id → cluster_id)
- `summary.json`: Cluster summary (representative posts & keywords)

- **reddit_posts.db**
- SQLite database storing posts and cluster assignments

■■ This directory is excluded via `.gitignore` and must not be committed.

---

4. Database Schema

`posts` Table

```sql
CREATE TABLE posts (
id TEXT PRIMARY KEY,
session_id TEXT,
subreddit TEXT,
title TEXT,
content TEXT,
cleaned_content TEXT,
image_text TEXT,
keywords TEXT,
topics TEXT,
extracted_urls TEXT,
extracted_mentions TEXT,
extracted_hashtags TEXT,
features TEXT,
embedding TEXT,
created_datetime TEXT,
processed_timestamp TEXT
);
```

`clusters` Table

```sql
CREATE TABLE clusters (
post_id TEXT,
session_id TEXT,
cluster_id INTEGER,
distance REAL,
```

PRIMARY KEY(post_id, session_id)
);
```

---

## 5. Scripts Overview

### `direct_iphone_processing.py`

- Runs the full pipeline once.
- Cleans data, generates embeddings, performs clustering.
- Saves results to `reddit_data/`.

### `reddit_data_processor.py`

- Handles Reddit data fetching, OCR, and preprocessing.
- Includes robust API handling with pagination and retry/backoff logic.

### `automated_processor.py`

- Automates repeated runs at fixed intervals.
- Logs execution progress and errors.
- Parameters: `--subreddit`, `--posts`, `--interval`, `--runs`.

### `show_results.py`

- Utility script to inspect results from SQLite.
- Prints:
- Total posts processed
- Cluster distribution
- Representative posts (closest to cluster centers)

---

## 6. Logging

Logs are saved to:
```
logs/lab5.log
```

Recorded information includes:
- Successful and failed OCR operations
- Retry attempts for API requests
```

- Batch and run progress

---

7. Testing and Verification

Verify Database Tables

```bash
sqlite3 reddit_data/reddit_posts.db ".tables"
```

Count Posts

```bash
sqlite3 reddit_data/reddit_posts.db "SELECT COUNT(*) FROM posts;"
```

Cluster Distribution

```bash
sqlite3 reddit_data/reddit_posts.db "SELECT cluster_id, COUNT(*) FROM clusters GROUP BY cluster_id;"
```

Inspect Representative Posts

```bash
python show_results.py
```

---

8. Submission Requirements

Per Lab 5 instructions:
- All code files
- README in PDF format (convert this markdown)
- `meeting_notes_.pdf`
- Demonstration video (team members explain design and results)
- Detailed GitHub commit history with contributions per member

---

■ With this implementation, the system satisfies **all Lab 5 requirements**:
- Web scraping and preprocessing

- OCR integration
- Embedding generation (TF-IDF, Doc2Vec)
- Clustering with KMeans
- Automation with periodic runs
- Storage and query via SQLite
- Logging and error handling