README
DSCI 560 – Lab 5
Reddit Data Collection and Clustering System

1. Project Overview

This project implements a complete Reddit data collection and clustering system. The system automatically collects posts from a selected subreddit, preprocesses the data, stores structured records into a SQLite database, generates text embeddings, and applies clustering to group similar posts.
The system supports:
· Pagination for large requests (e.g., 5000 posts)
·Text preprocessing and keyword extraction
·Embedding storage in database
·K-Means clustering
·Cluster ID persistence
·Periodic automation
·Interactive query

2. System Requirements

Python 3.10 or higher is recommended.
Required packages:
requests, scikit-learn, beautifulsoup4
If not installed, run:
pip install requests scikit-learn beautifulsoup4

3. File Structure

automation.py # Main automation and clustering pipeline
collect_store.py # Reddit data collection and database storage
preprocess.py # Text cleaning and preprocessing logic
cluster_analysis.py # Clustering analysis and reporting
reddit.db # SQLite database (generated after running)

4. How to Run the Project

Step 1: Run the automation script:

python automation.py <interval_minutes> --limit <number_of_posts>
Example:
python automation.py 1 --limit 200
This will:
1. Fetch Reddit posts
2. Preprocess text

3. Store data in SQLite
4. Generate embeddings
5. Perform clustering
6. Save embedding and cluster_id to database

Step 2: Verify Database

Open SQLite:
sqlite3 reddit.db
Verify number of records:
SELECT COUNT(*) FROM reddit_posts;
Verify embeddings stored:
SELECT embedding IS NOT NULL FROM reddit_posts LIMIT 5;
Verify cluster IDs:
SELECT cluster_id FROM reddit_posts LIMIT 10;

5. Large Request Handling (5000 Posts)

The system supports large request sizes (e.g., 5000 posts) through pagination.

It uses:
· 'after' parameter for Reddit API pagination
·Iterative batch fetching (up to 100 posts per request)
·Looping logic until requested total is reached

Note: Reddit's public API typically limits accessible posts to approximately the most recent ~1000 posts per subreddit. However, the implementation correctly supports arbitrary request sizes as required by the assignment.

Example:
python automation.py 1 --limit 5000

6. Database Design

All processed posts are stored in the 'reddit_posts' table.
Each record contains:
·Post ID
·Subreddit
·Title
·Clean text
·Keywords (JSON format)
·Topic label
·Embedding (stored as JSON vector)
·Cluster ID

·Metadata (score, comments, URL, etc.)
Embeddings are stored persistently to ensure reproducibility.

7. Clustering Method

Text embedding is generated using TF-IDF vectorization.
Clustering is performed using K-Means.
Cluster results include:
·Cluster ID
·Top keywords per cluster
·Representative posts
·Silhouette score evaluation

8. Automation

The script runs continuously using:python automation.py <interval_minutes>
The system will:
·Update data periodically
·Rebuild clustering model
·Store updated embeddings
·Allow interactive cluster queries

9. Design Decisions

We chose TF-IDF for embedding because:
·It is computationally efficient
·It works well for topic-based clustering
·It integrates easily with scikit-learn
We chose K-Means because:
·It is simple and effective for document clustering
·It scales well with medium-sized datasets
·It provides interpretable cluster centroids
Embeddings are stored in the database to maintain persistence and avoid recomputation.

10. Team Members

Boming Xi | USC ID: 5879982708
Tianyi Ge | USC ID: 5804514679
YuanYuan Xue | USC ID: 7945306882