DSCI-560: Data Science Practicum Laboratory Assignment 5
Instructor: Young Cho, Ph.D.

This assignment focuses on web scraping, data preprocessing, data analysis, clustering algorithms, and real-time data processing. You will better understand how to collect and organize data from online forums and create a system for clustering related documents. When you review the document, you will find that many of its contents are irrelevant, misleading, and/or undesirable. You will need to use reasonable methods to clean the dataset before storing it in a database.

1) Initial Setup

a) Tools / Libraries

You may use requests/selenium, beautifulsoup4, and a MySQL database for this assignment. The installations are like the ones used earlier, and you should already have them set up. If you prefer a different system, you may use them in their place if you can prove that the tools will give you experience that will benefit your career.

Linux OS(Ubuntu)/Python script should be used for the assignment. **(Make sure to document any setup steps/requirements for running your scripts in the document you submit)**
Do not spend much time on the installation and setup; instead, invest your time in exploring the concepts and improvising your submission.

b) Resource

You may use **BeautifulSoup** or the **Praw** API to understand how to scrape Reddit data using. Both methods have their merits, and your team may choose one based on how you want to structure your web scraper. If you found a better API for this, you may use it. However, please document why the alternatives are better than the suggested interfaces.

Scrape Reddit on Python: Scrape Reddit
https://brightdata.com/blog/web-data/how-to-scrape-reddit-python

Praw API: Scraping Reddit using Python Reddit API Wrapper (PRAW)
https://medium.com/analytics-vidhya/scraping-reddit-using-python-reddit-api-wrapper-praw-5c275e34a8f4

BeautifulSoup4 Guide: Scraping Reddit with Python and BeautifulSoup 4
https://www.datacamp.com/tutorial/scraping-reddit-python-scrapy

2) Data Collection / Storage

Select a topic on the Reddit website. The following are a couple of examples:

https://www.reddit.com/r/tech/
https://www.reddit.com/r/cybersecurity/

Your script must take the number of posts to fetch as an input, fetch them, and store them in the database after preprocessing.

API has a threshold of 1000 posts or a timeout of 60 seconds. Make sure your code can handle requests of size 5000 or requests that take 400 secs to fetch all results by calling the API multiple times without letting it run out of bounds, ensuring all results are fetched correctly, and the request doesn't fail. Read about the timeout and max limit of API requests, and modify your code so that it can handle large requests

3) Data Preprocessing

Preprocess the data by removing HTML tags, special characters, and irrelevant content such as promoted messages and advertisements.

Transform the data into a suitable format for analysis, such as converting timestamps and masking the usernames to maintain data privacy. Identify keywords and topics from messages and store them as additional fields in the database, along with the actual messages.

Some messages may include images. Image recognition tools like Pytesseract (https://pypi.org/project/pytesseract/) and other OCR readers can extract text from these images and store them as additional fields in the database. Consider this additional text while identifying keywords and topics to which the messages belong.

4) Forum Analysis & Clustering Algorithms

You must implement a clustering algorithm to group similar messages and display messages closest to the centroid of each cluster. Additionally, you will automate the data collection, processing, and storage process to run at fixed intervals.

a) Message Content Abstraction

The first analysis step is to convert messages into vector values of fixed dimensions that represent the meaning of the messages. This vector can be stored in the database along with the cleaned message. This process is commonly referred to as embedding.

One readily available open-source tool is called doc2vec. This tool employs an artificial neural network model to produce an embedding for an input document.

https://www.geeksforgeeks.org/doc2vec-in-nlp/

There are other ways to obtain document embedding, and you are free to explore and use a method that you deem sufficient for this lab as long as you can give convincing reasons behind your choice.

b) Clustering Messages

Cluster the document based on its content. If you are unfamiliar with clustering, please refer to the following tools.

- Scikit-Learn:Scikit-Learn Library Clustering
- NLTK Python:NLTK (Natural Language Toolkit)
- TextBlob: Documentation

Implement an algorithm to cluster the messages based on the embeddings from the text. Identify keywords that are associated with all messages in each cluster. You may choose the libraries and toolkits or use other tools to cluster the vectors representing the meaning of the messages.

Use any visualization tool to display the results, including K clusters (as in K-means clustering) of messages and their keywords. Verify that the contents of each cluster are similar by displaying and comparing message contents.

5) **Automation**

Write an encompassing script that will call the web scraping, pre-processing, and storage periodically to keep the database updated in real time.

This script should accept parameters for a specific interval (in minutes) from the user as a command line argument (i.e. “python filename.py 5”) to run the scripts and update the database at a specified time interval (in the above example: 5 mins). The script should provide proper messages such as fetching data, processing data, database updates, and appropriate error messages in case any operations fail.

When the script is not updating the database in the background, the command line prompt should take keywords or a message as input and find the cluster that matches closest to the input. The messages from a selected cluster should be displayed with a graphical representation.

6) **Team Discussions**
Your team is expected to meet in person / virtually each day of the week and discuss the assignment progress & next steps. Document and compile minutes of all meetings in a separate file called **'meeting_notes_<team_name>.pdf'**

7) **Submission**
Make one submission per team. Each team must submit all the code files for the working solution, a readme document containing information for running the code in PDF format, and a document that outlines the minutes of all team meetings in PDF format.

You must include a detailed GitHub history with descriptions of what each member of the team submitted.

Provide a video per team that demonstrates the entire working solution and explains how the data tables were loaded, demonstrates query results, and talks about the design decisions made along with reasoning for the same. Also, include details about how your team preprocessed the data. Please include the team's name and the names of all members in the video.

There will be a 50% penalty for all late submissions.