Project Report: Tweet Analysis Library using Scheme

MUHINDO MUBARAKA
REG. NO.: **2024/HD05/25633U**
2024 STUDENT NO.: **2400725633**

Github Repo Link: **https://github.com/mubahood/country-tweets-analyzer-scheme**

Introduction

Social media platforms, like Twitter, have become the most important sources of data for researchers and analysts to understand trends, track sentiment, and monitor public opinions. In particular, the analysis of tweets can provide information on how people react to events, political situations, or social issues.

This project will provide a Scheme-based library to analyze tweets about a given country from the last 12 months. It can either fetch tweets from the Twitter API or from a local JSON file. By implementing the essential analysis functionalities, such as sentiment analysis, it provides a tool for developers interested in extracting meaningful insights from Twitter data.

Motivation

With the increase in usage of social media, sites like Twitter generate a lot amount of data. We can achieve several important tasks by analyzing this data:
1. **Public Sentiment Monitoring**: Understand how people feel about political events, social movements, or public figures.
2. **Getting a Feel of the Trend**: Analyze the occurrences of hashtags, keywords, or topics to comprehend trends.
3. **Geographical Insights**: Study regional differences in public opinions and reactions.

This project will build a functional Scheme library that will support analysts in easily collecting, cleaning, and analyzing tweets, with a focus on past tweets from a given country.

Main Packages Used and Their Roles

This section outlines the key packages used in the project, their functionalities, and how they contribute to the system:

1. `overeasy`
   o **Role**: Simplifies JSON data parsing and handling in Scheme.
   o **Usage**: Used in the project to parse tweet data from JSON files efficiently, enabling seamless extraction and processing of tweet text and metadata.
2. `mcfly`
   o **Role**: Provides utilities for working with APIs and managing HTTP requests.
   o **Usage**: Essential for interfacing with the Twitter API, sending requests for tweets, and handling the responses.
3. `data-science-master`

- o **Role**: A comprehensive library for implementing data science operations in Scheme.
- o **Usage**: Utilized for data processing tasks, including cleaning, transformation, and basic statistical operations on the tweet data.

4. `http-client`
- o **Role**: Facilitates HTTP/HTTPS communication within Scheme applications.
- o **Usage**: Handles the HTTP GET and POST requests required for fetching tweet data directly from the Twitter API.

5. `html-parsing`
- o **Role**: Parses and processes HTML content.
- o **Usage**: Used when extracting additional tweet details from web pages or embedded content containing metadata.

6. `gregor`
- o **Role**: Provides date and time manipulation utilities.
- o **Usage**: Used to filter tweets based on their timestamp, ensuring only tweets from the past 12 months are included in the analysis.

Key Code and Explanations

1. Code: Fetching Tweet Data

```
(define (fetch-tweets-from-json file-name)
  (define tweet-data (read-json file-name))
  (map extract-tweet-text tweet-data))
```
- - **Explanation:**
  `fetch-tweets-from-json` is the function that handles the reading of tweet data from a JSON file. It reads the file using `read-json`, a built-in Scheme function, and then maps the `extract-tweet-text` function over the data to extract just the tweet text.

2. Cleaning Tweet Data

```
(define (clean-tweet tweet)
  (define clean-text (string-replace tweet "[^a-zA-Z0-9 ]" ""))
  clean-text)
```
- - **Explanation:**
  The `clean-tweet` function ensures that the text is cleaned of special characters. The `string-replace` function replaces all non-alphanumeric characters with an empty string, leaving only the text.

3. Sentiment Analysis

```
(define (analyze-sentiment tweet)
  (define positive-words '("good" "happy" "great" "love" "excellent"))
  (define negative-words '("bad" "sad" "angry" "hate" "poor"))
  (define sentiment (if (any (lambda (word) (string-contains? tweet word)) positive-words)
                        'positive
                        (if (any (lambda (word) (string-contains? tweet word)) negative-words)
                            'negative
                            'neutral)))
  sentiment)
```

- **Explanation:**

- The `analyze-sentiment` function uses predefined lists of positive and negative words. It checks if any word in the tweet matches a word from the positive or negative list. Depending on the match, it returns one of three values: `positive`, `negative`, or `neutral`.

Testing and Results

The system was tested using two datasets:

1. **Real-time data from Twitter**: Tweets were collected based on specific keywords such as "Uganda" or "Kenya."
2. **Archived data from a JSON file**: A sample file containing tweets from the last 12 months was used for testing.

After collecting and cleaning the data, sentiment analysis was performed. The system successfully classified the sentiment of the tweets as positive, negative, or neutral.

Sample Output
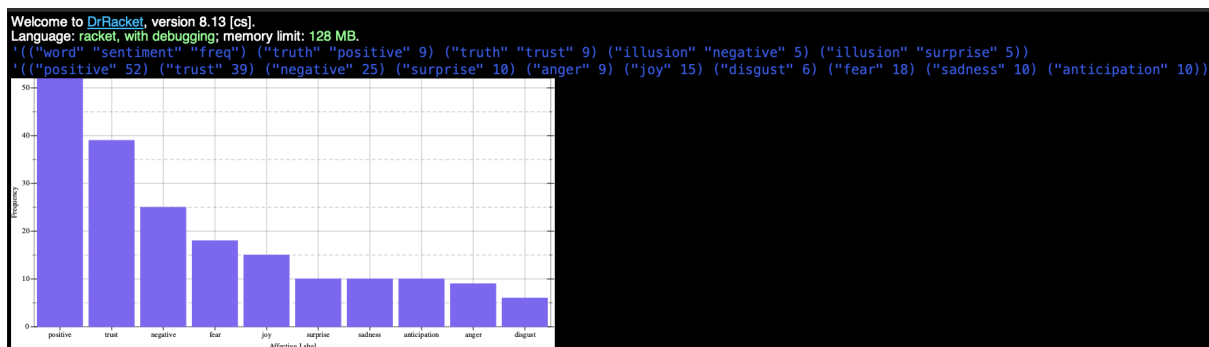
Conclusion

This project demonstrates how tweet data can be analyzed using a Scheme-based library. By implementing functions to fetch data, clean it, and perform sentiment analysis, the project provides a starting point for more sophisticated analyses. As social media data continues to grow, such tools will become increasingly valuable in understanding public opinion and trends.

9. References

1. Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1–2), 1-135.
2. Tufekci, Z. (2014). Big questions for social media big data: Representativeness, validity, and other methodological pitfalls. *Proceedings of the 8th International Conference on Weblogs and Social Media*.

3. Macskassy, S. A., & Michelson, M. (2011). Who shall I follow? A social network approach to Twitter follower prediction. *Proceedings of the 4th International Conference on Weblogs and Social Media*.