Social Media Data Analysis

Stephen BOUCHARDON

May 23, 2025

Contents

List of Tables

1 Introduction

This project consists on extracting and analyzing the data from a social media. For accessibility reasons, the chosen provider is the open-source social media, **Bluesky**. With its API, we collect the users posts and analyse the data with Python tools.

2 Task 1

2.1 Tools

Tool	Use
Bluesky API (via ATProto)	Fetch posts, users, and followers
Python and modules	Primary language for data analysis
Jupyter Notebook	Development environment
GitHub	Version control
LaTeX	Report writting
JSON	Testing API responses

Table 1: Table of tools

2.2 Algorithm

1. **Register to Bluesky**: In order to connect to the API, I will first setup an account with an username and a password. In a `.env` file, will be stored those credentials to access the API through the module `python-dotenv`. In the Jupyter Notebook, I will use the ATProto API module and connect to the client. Once connected, the client can fetch posts different feeds : global, custom, or specific user.

2. **Data Storage**: After retrieving the raw post data, I will filter its content to save only posts containing the specified tag (such as #football). Each posts contain the following structure :

   - `view` – the complete post metadata object
   - `creator` – the user and creator of the post
   - `display_name` – the displayed name of the user
   - `avatar` – the user's avatar URL
   - `like_count` – the number of likes of the post
   - `text` – the content of the post
   - `created_at` – the date of publication of the post

   Within the `text`, you can search for the tag and use as a filter a regular expression to only find exact matches. I will then store the matches in a Pandas DataFrame for analysis of : the sentiment of the posts, the top tags and users, and the followers of a specific user.

3. **Sentiment Analysis**: To analyze the sentiment of each post, I will use a sentiment analysis module such as `TextBlob` or `NLTK`. These libraries provide sentiment polarity scores we can then classify by positivity, negativity or neutral with `Numpy`. For each post, this task will be done on the `text` row. Also, we can count likes and views to add another layer to this interpretation.

4. **Top Tags and Users Extraction**: I will use a frequency counter with `Panda` or `Numpy` to identify the top 10 most used tags and the most active users based on the number of posts they made. Moreover, those results can be visualized using `Matplotlib`.

5. **Followers and Network Analysis**: For a given user, I will call with the API `app.bsky.graph.getFollowers` to retrieve their followers. For each follower, I will obtain their profile metadata `app.bsky.actor.getProfile`. This data is stored in a `DataFrame` for numerical data analysis with `NumPy`.

2.3 Modules

- `dotenv` - Secured importation of the credentials
- `pandas` – Structured data manipulation
- `TextBlob, NLTK` – Sentiment analysis of post content
- `matplotlib` – Data visualization, such as comparing the top tags or user activity
- `numpy` – Counting tag and user frequencies, use of statistical methods like standard deviation and mean value for comparison
- `json` – Testing the API responses after requests

2.4 Data Structures

- `list` – Collection of posts, followers, or any sub elements from a DataFrame
- `dictionary` - Key-value pairs, such as the username as key and the number of followers as the value
- `numpy array` – Numerical data analysis like counting likes
- `DataFrame` – Two dimensional tabular representation

3 References

1. Bluesky API Documentation : https://docs.bsky.app/docs/

2. Applied Data Science with Python - Mr. Lengyel - IN,IT - SoSe 25