Project Mood Ring

September 20, 2021

Authors:

Gretchyn Drown (quasar), Patrick Murphy (pbmurphy), and Brian Seko (bseko)

Intent Summary: We intend to perform sentiment analysis on tweet responses to 5 historical newsworthy events, comparing community moods to these events with the purpose of building a system that will, in future iterations, be able to predict a particular area's response to a categorized event.

Motivation: Inspired by the prolonged protests and rioting in Portland, Oregon over the shooting death of George Floyd and in support of the Black Lives Matter movement, we wondered whether it would be possible to predict a community's reaction to a current event [1]. Twitter is a major source of community sentiment, and as such is useful for examining this possibility.

DATA SOURCE 1
Twitter API v2
with Academic Research Track developer access

Description: tweets collected via API calls/ HTTP GET requests; aggregated in server hosted MySQL database

Format: JSON

Size: 12,000 tweets

Variables: tweet data, associated place information, and user data

Time periods:

- December 14–15, 2012 (Sandy Hook)
- March 23–24, 2014 (Ebola)
- June 26–27, 2015 (Same Sex Marriage)
- November 8–9, 2016 (Trump Election)
- October 31–November 1, 2019 (Trump Impeachment)

DATA SOURCE 2
EmoLex [2]

Description: maps words to emotions/sentiments

Format: TXT

Size: 2600 kb

Variables: 140,000 words, 8 emotions, 2 sentiments

DATA SOURCE 3
EmojiLib [3]

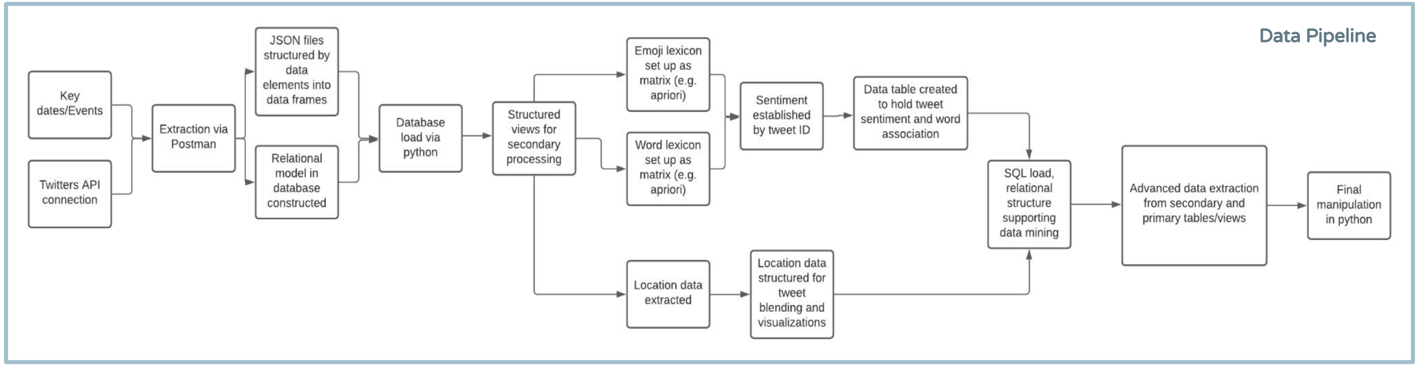
Description: maps emojis to emotions

Format: JSON

Size: 161 kb

Variables: 1,800 emojis, 6800 word associations

Source code, database design, data extracts, and replication instructions located on GitHub



Data Pipeline

Key dates/Events

Twitters API connection

Extraction via Postman

JSON files structured by data elements into data frames

Relational model in database constructed

Database load via python

Structured views for secondary processing

Emoji lexicon set up as matrix (e.g. apriori)

Word lexicon set up as matrix (e.g. apriori)

Location data extracted

Sentiment established by tweet ID

Location data structured for tweet blending and visualizations

Data table created to hold tweet sentiment and word association

SQL load, relational structure supporting data mining

Advanced data extraction from secondary and primary tables/views

Final manipulation in python

Twitter Data Pipeline

Data Pipeline Principles

Data pipelines are necessary for the continuous cleaning of incoming data to be staged in a usable format, reducing step replication. Raw data extracted through APIs require business rules to guide final use, including how cardinality will affect queries, data formats, and usable records [4].

Why a Pipeline is Required

This project is intended as a snapshot-in-time of an ongoing analysis of Twitter data. Large extractions are necessary to generate enough data to obtain reliable results.

A single set of files to clean, retain, and structure data for analysis in not feasible: the data must be obtained in regular intervals in such amounts that even in binary format, storage would severely limit querying abilities based around memory heap sizes.

Audit Records

An audit table was created and updated for each load type into the database. Failure records were extracted and saved as CSVs to ensure transparency of the ETL integrity.

Database Design

This dataset was delivered in JSON format, and the database was constructed to act as a data warehouse to retain raw input, a data mart to deliver structured clean data in a Bus format for incremental data loads, as well as a hybrid relational database following the Kimball/Star-Schema models [4]. Surrogate keys were created for additional efficiency, utilizing b-tree indexes. This mapping allowed automatic rejection of records that did not meet the initial load of authors and raw tweets lacking a foreign key. This process further maximized storage to allow for scalability. Data was stored on a shared server utilizing MySQL hosted by BlueHost.

The database was designed to maximize space for tweet text, which comprised most of the storage. Initial load tables link to a central fact table that limits subsequent record insertion. Any data that is replicated in metadata or search criteria is set as a dimensional link. This reduces the overall storage needed to retain a dataset requiring millions of records. This design represents a hybrid of several database types, though closely resembles a cross between Kimball and the Star-Schema approach [5], as seen in the entity relational diagram (ERD) on the following page.

Built for scalability, this data pipeline model can be fully automated; the GET request can be completely replicated using the Python library requests and can be moderated with Windows Scheduler or crontab. Using a .bat file or bash script, each Python script can be called in succession to load data, eliminating the need for manual oversight.

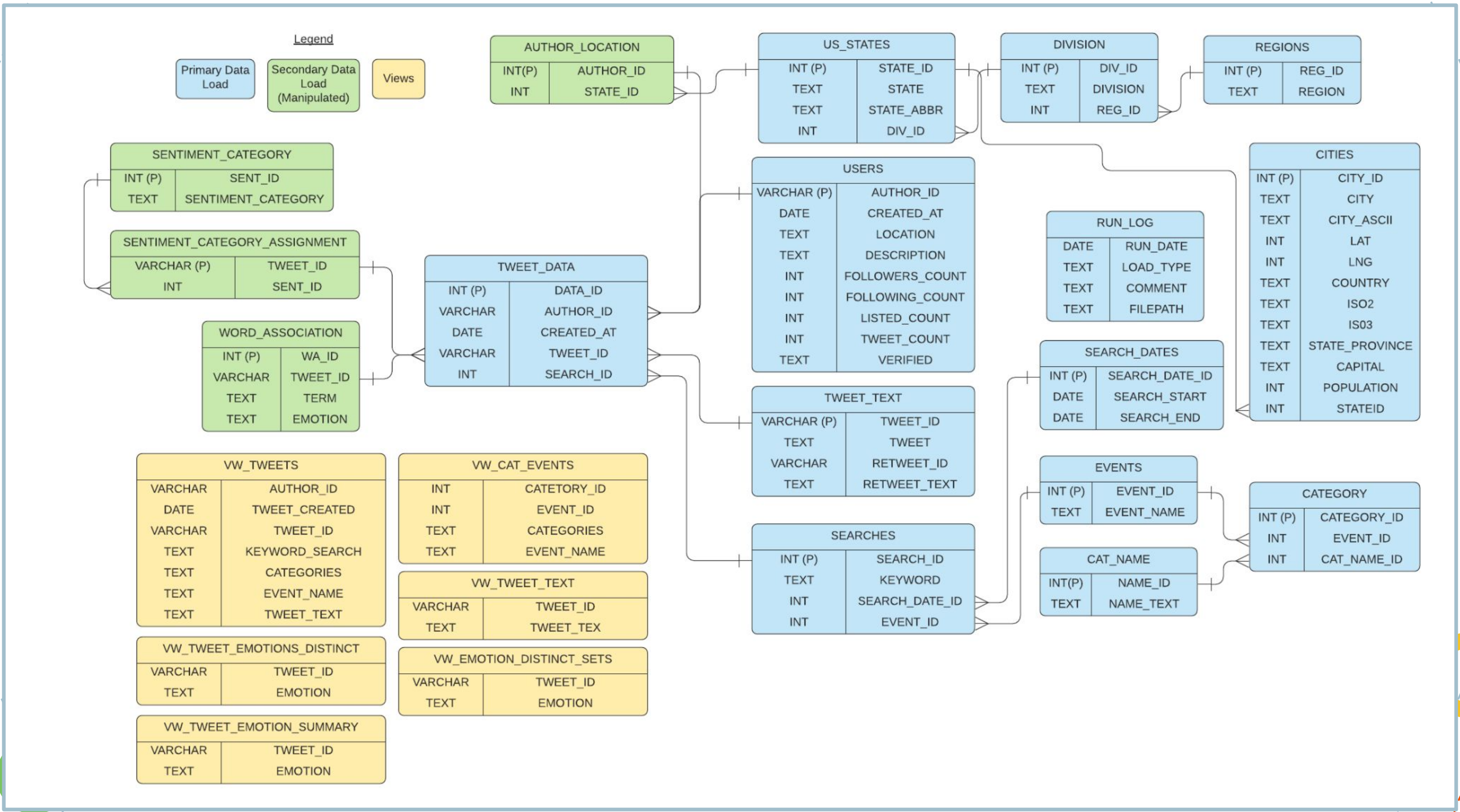
ETL (Extract, Transform, Load) Process

Initial Load: Data abstraction starts with a Postman GET request utilizing the Twitter API environment. The specific GET request used in this process can be imported for replication. JSON files were named in a manner to support metadata loading. Since the goal of this pipeline is to repeat the ETL as frequently as possible, minimal losses were accepted to favor a fast-loading process.

Secondary Load: These were created to conform manipulated data to existing facts and effectively capture the relationship of the data with existing metadata structures [6]. Scripting this process and distributing to end-users as a data mart guarantees consistent evaluation and minimized hand-coding of data [6].

Atomic Load Structure: Python files were used for data loads, making use of the MySQL library. This connection type was selected due to compatibility with the Linux server version and C++ connector. This library also allowed use of atomic commits that perform individual insert statements so that the database can reject records that do not meet field and key definitions.

Twitter Pipeline Entity Relationship Diagram



Legend

Primary Data Load

Secondary Data Load (Manipulated)

Views

AUTHOR_LOCATION	
INT(P)	AUTHOR_ID
INT	STATE_ID

US_STATES	
INT (P)	STATE_ID
TEXT	STATE
TEXT	STATE_ABBR
INT	DIV_ID

DIVISION	
INT (P)	DIV_ID
TEXT	DIVISION
INT	REG_ID

REGIONS	
INT (P)	REG_ID
TEXT	REGION

SENTIMENT_CATEGORY	
INT (P)	SENT_ID
TEXT	SENTIMENT_CATEGORY

SENTIMENT_CATEGORY_ASSIGNMENT	
VARCHAR (P)	TWEET_ID
INT	SENT_ID

USERS	
VARCHAR (P)	AUTHOR_ID
DATE	CREATED_AT
TEXT	LOCATION
TEXT	DESCRIPTION
INT	FOLLOWERS_COUNT
INT	FOLLOWING_COUNT
INT	LISTED_COUNT
INT	TWEET_COUNT
TEXT	VERIFIED

CITIES	
INT (P)	CITY_ID
TEXT	CITY
TEXT	CITY_ASCII
INT	LAT
INT	LNG
TEXT	COUNTRY
TEXT	ISO2
TEXT	IS03
TEXT	STATE_PROVINCE
TEXT	CAPITAL
INT	POPULATION
INT	STATEID

RUN_LOG	
DATE	RUN_DATE
TEXT	LOAD_TYPE
TEXT	COMMENT
TEXT	FILEPATH

TWEET_DATA	
INT (P)	DATA_ID
VARCHAR	AUTHOR_ID
DATE	CREATED_AT
VARCHAR	TWEET_ID
INT	SEARCH_ID

WORD_ASSOCIATION	
INT (P)	WA_ID
VARCHAR	TWEET_ID
TEXT	TERM
TEXT	EMOTION

SEARCH_DATES	
INT (P)	SEARCH_DATE_ID
DATE	SEARCH_START
DATE	SEARCH_END

TWEET_TEXT	
VARCHAR (P)	TWEET_ID
TEXT	TWEET
VARCHAR	RETWEET_ID
TEXT	RETWEET_TEXT

EVENTS	
INT (P)	EVENT_ID
TEXT	EVENT_NAME

CATEGORY	
INT (P)	CATEGORY_ID
INT	EVENT_ID
INT	CAT_NAME_ID

VW_TWEETS	
VARCHAR	AUTHOR_ID
DATE	TWEET_CREATED
VARCHAR	TWEET_ID
TEXT	KEYWORD_SEARCH
TEXT	CATEGORIES
TEXT	EVENT_NAME
TEXT	TWEET_TEXT

VW_CAT_EVENTS	
INT	CATETORY_ID
INT	EVENT_ID
TEXT	CATEGORIES
TEXT	EVENT_NAME

SEARCHES	
INT (P)	SEARCH_ID
TEXT	KEYWORD
INT	SEARCH_DATE_ID
INT	EVENT_ID

CAT_NAME	
INT(P)	NAME_ID
TEXT	NAME_TEXT

VW_TWEET_TEXT	
VARCHAR	TWEET_ID
TEXT	TWEET_TEX

VW_TWEET_EMOTIONS_DISTINCT	
VARCHAR	TWEET_ID
TEXT	EMOTION

VW_EMOTION_DISTINCT_SETS	
VARCHAR	TWEET_ID
TEXT	EMOTION

VW_TWEET_EMOTION_SUMMARY	
VARCHAR	TWEET_ID
TEXT	EMOTION

Data Manipulation: Datasets

EmoLex [2]

- The EmoLex data was originally in tidy data format via text file.
- Tidy data, a standard for applications that use structured data, was developed by H. Wickham; it comprises a single table (an observational unit) in which columns and rows follow a specific structure: each variable is a column and each observation is a row [7] [8].
- This format provides ease of use when pivoting is needed. To use the EmoLex data, we first changed it from .txt to .csv format.
- We found this conversion sufficient to apply to our data due to its tidy format, and the data needed no extra manipulation.

Twitter API v2

Location Filtering

- Original idea: use the geo-tag attribute of tweet metadata to limit tweet gathering to certain geographical areas. Emotional content of tweet responses to events could then be compared between those locations.
- Twitter estimates that only 1-2% of tweets are tagged with the location where the user was at the time of posting, yet approximately 30-40% of Twitter user accounts have set a home location [9].
- The API v1 developer access we were granted allowed us to perform location searching, but we were limited to the prior 7 days of tweets, leaving us unable to measure emotional response to historical events. We next applied for, and were granted, Academic Research Track developer access, which is part of the newly developed API v2.
- While location information could be returned as part of the tweet payload, searching by location was not yet enabled in API v2.

Event Filtering

- For each event, we chose a number of relevant keywords to use in finding tweets about that event. We attempted to choose keywords that would provide both positive and negative tweets about the event to reduce selection bias.
- Using the event keywords, tweets were collected for a 24-hour timespan around each event.
- We had to interpret the user-entered location field, which was not constrained to any fixed format, or even required to be a real place. We matched locations at a US state-level resolution, and discarded tweets from user locations that didn't match to one of the 50 states or District of Columbia.
- This process retained 20% of the tweet authors, which is similar to [9].

EmojiLib [3]

- The EmojiLib data was originally in JSON format.
- We converted the JSON object to a Pandas Dataframe using json.loads().
- The dataframe required some manipulation, including renaming the columns (they were named as integers, which interfered with some later code) and adding a "demoji" column using [10], an emoji package that aligns with the Unicode Consortium [11].
- Another cleaning step involved removing any emoji with missing demoji in order to prevent these "false" emojis from cluttering the dataframe and skewing any summary statistics.

Data Manipulation: Tweet Content

Tweet Text

- We removed URLs, @tags, and any tweets composed solely of a notification that the user posted a photo or video.
- We first attempted to use the NLTK package [12] to parse words from the tweets, but experienced significant data loss (91%).
- As an alternative, we created a dictionary of the tweets, then used split() to parse the words, matching them to the lexicon one at a time and optimizing loss at 35%.
- This parsed data was entered into a matrix and merged with the EmoLex dataset, associating terms with a sentiment/emotion.
- We used our judgment to label each emotion as either positive or negative, recognizing that some terms may be considered both depending on the term associated with them.
- We then calculated apriori using k=3 to ensure an odd number of emotions so that a value of either positive or negative could be applied by majority.

News Stories

- We originally chose 4 major American cities in which to measure Twitter emotional reaction: Chicago, Houston, Los Angeles, and New York City.
- Twenty-eight news events, most with a connection to one of the cities, were chosen in a variety of categories, such as sports, politics, civil rights, and global.
- After the API location limitation was discovered, we narrowed our focus to 5 events and chose not to concentrate on specific locations.

Tweet Emojis

- Originally, we intended to analyze both text and emojis.
- Only about 4% of tweets displayed emojis, which was insufficient for robust analysis.
- Nevertheless, we extracted unique emojis and created a matrix of their counts, adding a 'demoji' column as a key to avoid unicode errors.
- Only terms associated with a particular emotion were included. That is, neutral language was excluded.
- We then merged the extracted emojis with EmojiLib to find corresponding associated terms, and with EmoLex to find the terms' corresponding emotions.
- The use of this process was validated by [13], which follows the same steps to associate emojis with emotions. However, as the authors of [13] suggest, the frequency of certain emojis is too low to use for analysis of such a large dataset because it could result in misleading sentiment scores. We found this to be true of emojis in general over our entire dataset in addition to less frequently used emojis, hence our decision to exclude emojis in this iteration.

Events/Key Search Words

Ebola: Africa, ebola, epidemic, outbreak, WHO

Sandy Hook: conspiracy, gun law, promise, Sandy Hook, victim

Same Sex Marriage: equality, love, same sex marriage, sin, supreme court

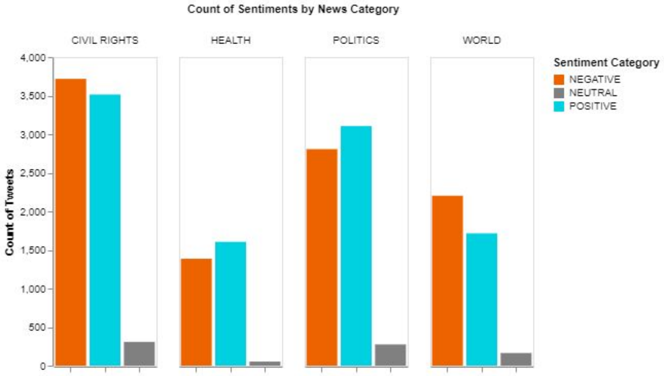
Election: Donald Trump, election, Hillary Clinton, popular vote, why did

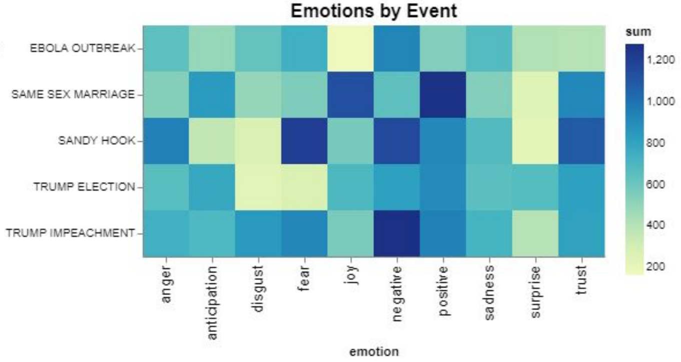
Impeachment: collusion, fake news, impeach, Trump, Zelensky

Analysis & Visualizations

This Altair grouped bar graph...

- shows major trends by news category and general sentiment
- allows monitoring of shifting emotions
- provides public interest levels





This Altair heatmap...

- checks whether we were accurately extracting emotions
- matches our general expectations (e.g. Sandy Hook was most associated with fear)
- validates our extraction methods
- gives an idea of the general quantity of reactions per event

Analyzing data by news category shows both the public interest and overall sentiment in event types. Tweets centered around Civil Rights were prominent in the dataset and expressed more negative sentiment. Political events were also an area of focus and prompted mostly positive responses. Monitoring these trends can reveal when sentiment shifts.

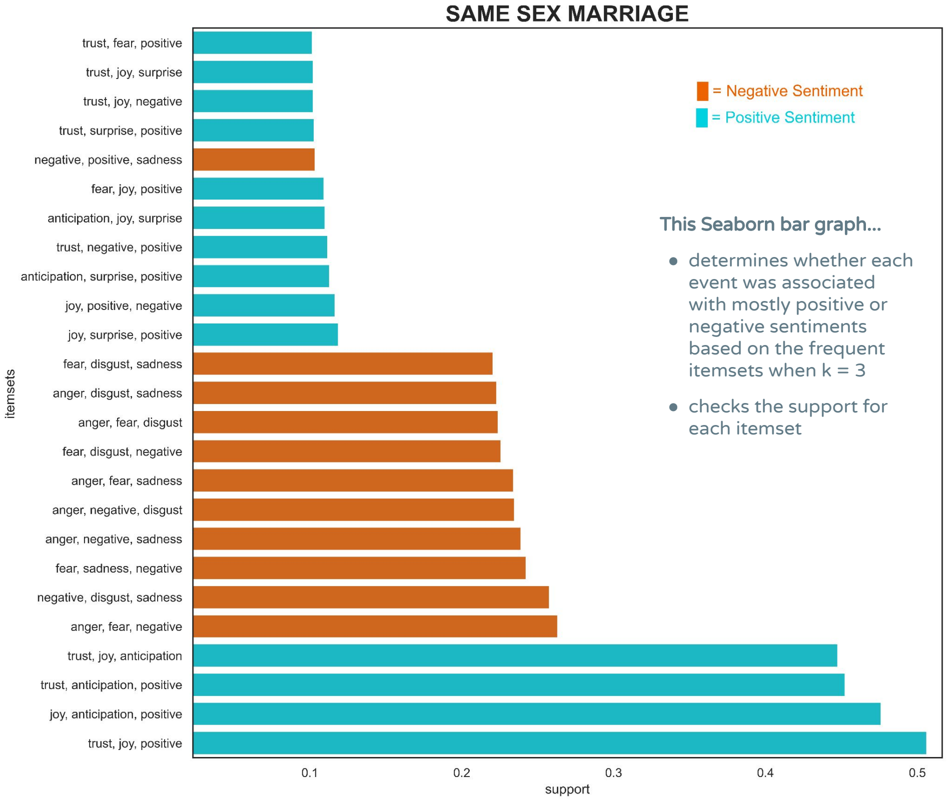
A growing discontent toward civil rights, for example, may be utilized as warning signs for more extreme behavior by tracking rapid sentiment shifts. This can then be expanded to see what specific emotions are becoming more prominent for a target group, location, or topic through exploration of the heatmap to the right.

To fully utilize this information as explained above, future iterations will need to include time series data.

One aspect of this emotion by event analysis that stood out to us is how sentiment can change over time. The Trump election shows a lack of emotions such as disgust, fear, anger, and negative. However, we see these emotions more prominently as sentiment toward his presidency shifted by the time of his impeachment. Additionally, the expression of emotions seems to have increased during this time, indicating a less indifferent attitude towards Trump's presidency.

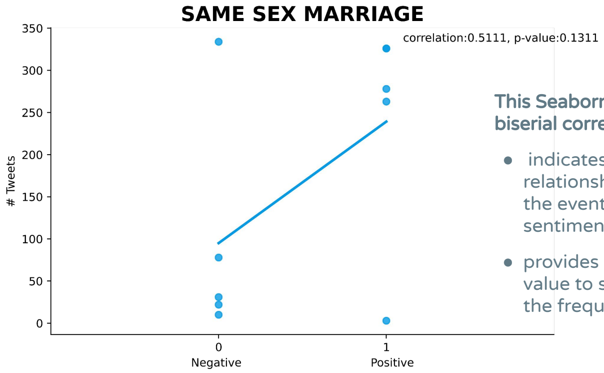
Categories such as political events can be tracked over time, revealing the dynamic emotional state of the public. This tracking could be utilized to predict emotions from previous trends, allowing local governments to plan for potential violent behavior or provide mental health support for their communities.

Analysis & Visualizations



This Seaborn bar graph…

- determines whether each event was associated with mostly positive or negative sentiments based on the frequent itemsets when k = 3
- checks the support for each itemset



This Seaborn point biserial correlation…

- indicates a relationship between the event and sentiment
- provides an empirical value to supplement the frequent itemset

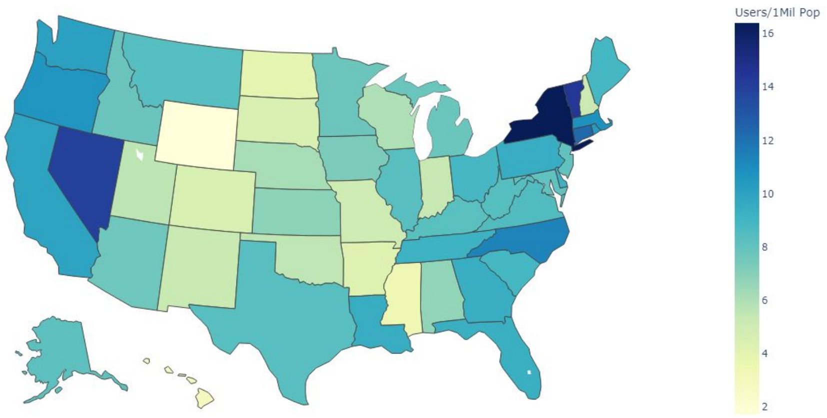
While the bar graph to the left may give the impression that the sentiments are mixed, the visualization above indicates a moderate to strong relationship to positive sentiment about the named event, with the p-value signifying that there is only a 13% chance that this relationship is random. The process by which we sorted emotions and their sentiments, therefore, is validated.

In the bar graph, there is a clear demarcation for most of the frequent itemsets, and support for positively-coded itemsets outweighs the negatively-coded ones. The tweets with mixed sentiment could represent the user expressing emotions about different, but related, topics in the same tweet.

Using “positive, negative, sadness” from the bar graph as an example, one way this could happen is if the user expressed positive thoughts about the current event (legalization of same-sex marriage in the United States) compared to the LGBTQIA+ community’s negative experiences in the past.

Analysis & Visualizations

Number of Twitter Users Who Commented on Analyzed Events Per State



This Plotly choropleth…

- maps the overall user location data
- shows whether users in certain areas of the country were more likely to tweet about the selected events.

The number of Twitter users per million population who posted about one of the selected events is shown in the choropleth above. Washington D.C. had by far the highest response rate, at 225 users per million population, over an order of magnitude higher than New York, the second highest at 16.4. The choropleth's scale omits Washington D.C.'s count in order to increase the visualization's effectiveness. It is not clear if this high user activity is due to being the locality of three of the events (election and impeachment, plus the site of the Supreme Court for same sex marriage), or a result of the user-defined locations. More data would be needed to refine this analysis.

Types of Twitter Location Data

Twitter users must opt-in to location tracking, which is represented in different ways:

1) **Tweet location tags**: Users can choose where to tag their content, creating latitude/longitude coordinates of the location their tweet is referencing. This does not mean the tweet originated from this area.

2) **Location tracking**: Opting-in to this means that the physical location of the user is tracked at the time a tweet is released, even if the tweet is tagged with another location.

3) **User "home" locations**: Setting a "home" location in a user profile can anchor a user to an area. However, this is a free text field that is user-entered. Users are not required to enter real locations, resulting in potential mismatches to true location.

For this analysis we referenced the users' home location when a U.S. state was entered, reasoning that the home location of a user represents the thoughts and feelings from that geographical area.

Twitter notes 30-40% of users enter home location

Summary & Expansion Ideas

Project Summary

Project Mood Ring, created as a Milestone 1 project at the University of Michigan School of Information's Master of Applied Data Science program, represents an application of data science that can monitor, assess, and influence reaction to community expressions of emotion and overall sentiment regarding newsworthy events. This iteration features exploratory analysis and process debugging to aid in future development. Our eventual goal is to present this tool as a user-facing dashboard for real-time tweet analysis.

Expansion Ideas

Compare sentiment of tweets over time

Our event analysis focused on tweets that were generated in the immediate aftermath of our chosen events. We imagine that there may be some useful insights that arise from viewing reactions over a longer time frame as more information and clarity about each event becomes known. Looking at this longer time frame will help community leaders plan their response to the entire arc of the event and reaction.

City-level location matching

Due to the time constraints of this project, we limited the resolution of our location matching to the level of the state. We recognize that there can be large differences in how people in different areas of a state react to the same event, which would limit the ability of a community leader to use our results for their individual jurisdiction. With more programming and processing time, we could generate reports at the level of Metropolitan Statistical Areas or individual cities. This would require a far larger tweet volume to generate a meaningful training set for each city.

Additional keywords

Adding more keywords for each event could provide a broader picture of the Twitter response. These words could be determined by picking words frequently mentioned in the tweets that also contain our chosen keywords.

Emoji supplementation

Although insufficient emojis were present in the tweets to include a separate analysis, we could supplement the tweet text emotion count with emoji emotion count, possibly yielding a more complete picture of the associated emotions per tweet.

ETL Automation

Scripting the extraction and ETL process using .bat files or bash allows for higher volume tweet analysis, enabling daily extraction of 10K tweets or more to stage, clean, and prepare for analysis or machine learning training.

Statement of Work & Sources

Statement of Work

ALL: Report writing

Gretchyn Drown: EmoLex and Emojilib manipulation, NLP for emojis and text, heatmap visualization, report design/editing, citation management

Patrick Murphy: Twitter JSON parsing, event research, tweet location matching, choropleth visualization

Brian Seko: Twitter API development, Twitter database pipeline, project management, bar graph and biserial correlation visualizations, ERD and data pipeline diagrams

Visit our GitHub for all source code, database design, data extracts, and replication instructions.

Sources

[1] A. Joshi et al., “Automated monitoring of tweets for early detection of the 2014 Ebola epidemic,” PLOS ONE, vol. 15, no. 3, p. e0230322, Mar. 2020, doi: 10.1371/journal.pone.0230322. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0230322

[2] “NRC Emotion Lexicon.” https://www.saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm (accessed Aug. 02, 2021).

[3] muan, *emojilib*. 2021. https://github.com/muan/emojilib. (accessed: Aug. 12, 2021).

[4] R. Kimball and M. Ross, *The Data Warehouse Toolkit*, Second Edition. United States of America: John Wiley and Sons, Inc., 2002.

[5] L. Corr and J. Stagnitto, *Agile Data Warehouse Design 201*, February 2014. DecisionOne Press, 2014.

[6] Kimball, Ralph and J. Caserta, *The Data Warehouse ETL Toolkit*. United States of America: Wiley Publishing, Inc., 2004.

[7] “2 Sentiment analysis with tidy data | Text Mining with R.” https://www.tidytextmining.com/sentiment.html (accessed Aug. 02, 2021).

[8] H. Wickham, “Tidy Data,” *J. Stat. Softw.*, vol. 59, no. 1, Art. no. 1, Sep. 2014, doi: 10.18637/jss.v059.i10. Available: https://www.jstatsoft.org/index.php/jss/article/view/v059i10.

[9] “Advanced filtering for geo data.” https://developer.twitter.com/en/docs/tutorials/advanced-filtering-for-geo-data (accessed Sep. 08, 2021).

[10] B. Solomon, *demoji: Accurately remove and replace emojis in text strings*. https://github.com/bsolomon1124/demoji. (Accessed: Sep. 14, 2021).

[11] “Full Emoji List, v13.1.” https://unicode.org/emoji/charts/full-emoji-list.html. (accessed Sep. 11, 2021).

[12] “NP_chunking_with_nltk/NP_chunking_with_the_NLTK.ipynb at master · umsi-data-science/NP_chunking_with_nltk,” *GitHub*. https://github.com/umsi-data-science/NP_chunking_with_nltk. (accessed Aug. 02, 2021).

[13] A. A. M. Shoeb, S. Raji, and G. de Melo, “EmoTag – Towards an Emotion-Based Analysis of Emojis,” in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, Varna, Bulgaria, Sep. 2019, pp. 1094–1103. doi: 10.26615/978-954-452-056-4_126. Available: https://aclanthology.org/R19-1126.