CS 7311 Project Report
Texas State University
Fall 2019

Sirichi Bobby Srisan
sbs98
sbs98@txstatedu

Sharmila Kanthaiya Srinivasan
s_k309
s_k309@txstate.edu

Abstract

An improvement is proposed to an original pipeline to include data cleaning, network analysis, and a method for community detection. It integrates domain expert feedback on relevant phrases to create n-grams to include in the community model. The community model create groupings based on collections of topic profiles. A post processing module identifies influencers in respective communities as well as topics of those communities. Visualizations are provided to assess baseline communities.

1. Introduction

Social networks is a rich data source for many areas of research. Twitter is a social networking service on wherein the core feature is user posts and interactions. The speed, volume, and density of Twitter data has been shown useful for different topics of analysis in cultural trends, sociology, disaster response, politics, and beyond. The nature of social network data also presents challenges from a data-driven research perspective. The data research pipeline has common steps regardless of the research or decision goal. These steps are also applicable to social networks other than Twitter. It begins with obtaining data either from the source or an aggregation service. Then data must be brought a structured and consistent form for downstream processing. This includes data cleaning, structuring collections, and forming coherent data elements. Exploratory analysis is used to obtain a cursory view of the points of interest, which are then augmented with domain knowledge. Mathematic modeling methods can then be used to gain further insight and move onto predictive models.

This study proposes a flexible pipeline to analyze community trends on social media. Given a main topic, the pipeline processes user data and user content and clusters data sets into communities. The goal is to find community and phrases representing distinct ideologies to inform sociology research on data consumption behavior. Three different algorithmic modeling methods are used for topic detection are studied. The topics are annotated by domain experts and fed into a machine learning model to find communities.

The case study for this data research pipeline will use Twitter data to model people links and events around the #MeToo conversation within the past years. Additionally, while this case study uses MongoDB and select machine learning Python libraries, this pipeline can be applied to comparable NoSQL databases and software libraries.

This paper is presented in five sections. The first section rationalizes database design and software library usage. Some data munging processes are reviewed. The next section reviews machine learning models for topic detection. The third section presents case study with data from the #MeToo dataset and incorporates insight from interested sociologists. The fourth section evaluates results from using Louvain modularity for community detection. Finally, an assessment of the pipeline and possible improvements to the pipeline are considered.

2. Problem Statement

Social networks contain dense and complex information. Conversations that occur in social media contain informative signals, but they are mixed in with irrelevant and redundant noise[**?**]. Content is also generated at very large a volume and rate. For context, over 300 million people use Twitter every month, and event topics occur instantly. Consequently, it is difficult to manually glean actionable insight directly from streaming data. A manual content review on a single Facebook page can take well over a month. Aggregation tools have been developed to manage social feeds, but they lack quantitative, data-backed modeling, and they are still prone to noise.

The problem narrows down to how to extract usable information from massive and heterogeneous data in an acceptable time frame. Included in this problem is how data is accessed and stored so that it is readily processed. Additionally, many database and data modeling tools are available, but they are not specialized for social media exploration on a software development level.

2.1. Related Work

Past work efforts have sought to ingest information relevant for immediate, informed action, such as first responders from tweets generated during disasters[**?**].

The basis of this study is based on a prior: an automated and flexible pipeline to perform analysis on Twitter data with focus on natural language processing (NLP) research [**?**]. The study is extended in this paper to include feedback by domain experts, additional visualizations, and modeling and exploration into baseline communities.

3. Methods

This overall goal of this research is to use social media data to discover information about users and implicit communities of interaction. The pipeline developed in the course of this research is used to analyze communities in social networks using available user content and account data. Twitter #MeToo is used to populate the data in the pipeline. The data set amounts to over 25GB of text over a period of two years, so we had to consider ways to would run through this data quickly and produce reliable scores.

3.1. Data Wrangling Considerations

Social network data is often given in the form of structured data. For Twitter, tweets are represented as JSON documents, which includes metadata such as user id, geolocation, date, language, retweets, replies, and much more. This raw data must be prepared so that redundant or irrelevant information is stripped. Building on the basis work, the input to the modeling algorithms will integrate feedback on relevant phrases to create bi-grams and tri-grams in topic forming. Moreover, it is observed that initial topic labels contain words that are specific to Twitter, and they must be treated as stop words. For example, "rt" is an abbreviation for retweet. Since retweet status is already handled by document metadata, it does not need to be an input topic detection.

3.2. Database Design Considerations

Document driven NoSql databases such as MongoDB store data as collections of documents, which in this case includes JSON tweets. The schema should be open to adaptation and scalable in data storage and movement as new models or topics necessitate. There should be structure to allow for query management logic and data auditing. For example, it may hold records for past queries. The initial data set was stored in MongoDB, hence we are continuing data analysis using this database software. The following are collections previously established and are being used again in the pipeline:

dbSettings: This collection contains the basic configuration desired for analysis. Fields are not included if they are deemed not important for the analysis.

loadedFiles: This collection keeps record of directory and file names that have already been loaded.

loadStatus: This collection is used for data recovery management.

twitterSearches: This collection keeps record of the searches requested to Twitter API and is only used when the tweets are saved directly from the Twitter API.

tweet: This is a collection of complete tweet documents, and it contains a sequence number used in the tweet recovery process.

focusedTweet: This collection contains reduced detail tweets with fields in dbSettings.

tweetWords: This collection holds separate words and metadata from every tweet.

htTopics: This collection holds topic information for each hashtag using the three different unsupervised ML models.

hashTagCountAgg : This collection holds frequency of all hashtags in the dataset.

tweetCountByFileAgg: This collection aggregates the count of all tweets in the dataset by files loaded.

tweetCountByPeriodAgg: This collection aggregates the count of all tweets in the dataset by period specified in the dbSettings.

tweetCountByLanguageAgg: This collection aggregates the count of all tweets in the dataset by language.

tweetCountByUserAgg: This collection aggregates the count of all tweets in the dataset by user id. It includes details about the user as specified in dbSettings.

3.3. Infrastructure Considerations

While performing data analysis on personal computing machines can establish initial of concepts, working with big data requires access to more computing and storage resources. This initial setup is often overlooked.

The total storage size of the raw collections in the study is nearly 40 gigabytes of data, with most data residing in the collection of tweet documents and user data. Remote hosting should be considered for the benefit of access and storage capacity. There exists database-as-a-service such as MongoDB Atlas, which minimizes database administration in exchange for remote server usage costs.

Rendering optimal layout of graph visualizations of over 100k nodes and edges also requires heavy compute resources. This computation requirement will be discussed in the conclusion section.

3.4. Data Modeling Considerations

We look to the following algorithmic machine learning to model topics and communities.

Topic modeling methods need to provide right level of abstraction of collection of documents). It need to tells us what topics are present in any given document by observing all the words in it and producing a profile. The profile should be quantifiable features so that the topics can be ranked and ordered by similarity.

The community detection method should allow the discovery to overlapping groups. Generally, as the topics of discussion occur at different levels, the method should also allow scoping in and out of different granularities.

4. Infrastructure Tools

The choices in pipeline infrastructure leverages Python and public Python library contributions to machine learning algorithms. MongoDB is a cross-platform, document-oriented database. As a NoSQL database program, MongoDB is flexible, scalable and readily handles Twitter's JSON documents.

5. Data Processing Tools

5.1. Data Cleaning

Python's NLTK is used in the first stage of data preprocessing to remove stop words, normalize, and tokenize documents. The Gensim library also has functions to refine topic classification and creating the n-grams. From an original list of lemmatized words extracted from TF-IDF filtered words, we built a list of common words, which were deemed non-relevant in experts' research goals. We decided to include these words in the final n-grams, but marked as common terms so that are not weighted in the n-gram algorithm. The common terms thus have no statistical weight in n-gram detection, but the result is a longer n-gram that is more human readable, for example, including the non-weighted word "getting" in "conspiracy$_w$omen$_g$etting$_b$rave" $gives the reader more context than$ "conspiracy$_w$omen$_b$rave".

The pipeline also includes complete removal of non-relevant words from topic analysis. These words are considered artifacts. They originate from hashtag terms, the fact that Twitter users use shorthand for terms, and that word preprocessing of hyperlinks (i.e."https"). Any of these terms that are not grouped into n-gram common words are removed from the documents completely, before feeding them into the topic modeling algorithm.

5.2. Machine Learning Tools

The selected machine learning Python libraries are:

- python-louvain - Python module implements community detection
- Scikit-Learn - Data analysis and topic extraction toolkit
- Gensim - Topic modeling toolkit

Gensim was chosen to perform Latent Dirilect Allocation (LDA) and Latent Semantic Indexing (LSI). LSA learns latent topics by performing a matrix decomposition (SVD) on the term-document matrix. LDA is a generative probabilistic model, that assumes a prior over the latent topics. In practice, LSI is much faster to train than LDA, but has lower accuracy. We use topic modeling to identify general phrases that occur in communities of users.

The Louvain method achieves modularities comparably less time than alternative algorithms, so it enables the study of much larger networks. It also generally reveals a hierarchy of communities at different scales. This hierarchical perspective can be useful for understanding the global function of a network. There are pitfalls to interpreting the community structure uncovered by the Louvain Method; these difficulties are actually shared by all modularity optimization algorithms.

A future improvement may exist for using the BERT (Bidirectional Encoder Representations from Transformers) natural language processing framework for named entity recognition and classification of topic-relevant tweets not filterable by #MeToo and outside the initial 3M tweet dataset provided.

5.3. Visualization Tools

- matplotlib - For creating graphs and plots
- seaborn - For enhancing the style of matplotlib plots.

6. Experiments

Setup We restored the first stage data into a MongoDB server. The main datapoints for analysis are derived from the aformentioned focusedTweets collection as it contains tweet documents, user information, as well as information needed to extract user interactions in graph creation. The goal is to identify influence among groups of users based interactions.

Identifying User Interactions The focusedTweet collection of documents have attributes identifying if a user tweet is in response to another user. We build an intermediate database collection of pairs called "user_reply". User replies are queried as any "in_reply_to_screen_name" to the focusedTweet collection.

The second type of interaction is when a user retweets a tweet post. Since the retweet attribute in the focusedTweet collection does not show user name, we instead seek another method for retweeting. Users may also retweet by replying to a post with "rt @mention ...", where the mention is a user name. While this style of retweeting is older, it is observed that many users use this style either out of habit or unawareness that a dedicated retweet feature is available. Additionally, user may mention another user to address a more generalized topic that is not in response to a specific post content. In the data exploration section, the difference of adding more interactions to the graph will be discussed.

Building Graph from User Interactions With all user interactions identified, each unique user is interpreted as a graph node, and the interaction is an edge between respective users. The edge weights are the number of times an interaction of any type has occurred. In these following analysis the graph is interpreted as undirected, but higher granularity with directed graphs can be built in a future iteration. In a directed graph, the node edges would be based on in-degree and out-degree. Outgoing edges identifies a user is doing an action (for example, the user has retweeted), and incoming edges to a user node occurs when the users is incidentally involved by another user's actions (for example, the user has been retweeted).

6.1. Exploratory Data Analysis

Once the graph of user nodes connected interactions is created. We run the Louvain community detection using modularity. Modularity is an overall graph metric used to measure relative density in a network when nodes are assigned to some communities. Modularity measure how much more densely connected nodes are versus how much they would be on average in a random network. The Louvain method has two repeating steps: 1) do a greedy assignment to nodes of a community to favor local optimizations of modularity, and 2) define a coarse network in terms of communities found in step 1. These steps are repeated until "no further modularity-increasing reassignment of communities are possible"

In our initial graph building, we used only user replies (weighted by frequency of replies) as edge connections. This graph in 1 and communities turned out to be sparse. The intuition behind this observation is that users reply less often that they retweet a post. A retweet is an easier action on the user's part, and users may often not have any explicit content to add or reply.

As we go on to build a more connected graph with more edges representing additional mention interactions, the community populations and degrees increased to a satisfactory level to move onto user insights. The higher connected graph is seen in 2 The most populous communities are heavily skewed toward a few number of top communities. As seen, 99% of community populations have less than 12 users.

With the degrees of nodes representing user connections, we also observe that some lesser know profile have a high degree of connections. For example, user account "ajain31" is not an account of a celebrity, news entity, or political figure, but this is a user and some others that have high interactions in the respective community.

Community Topics The communities are further filtered such that the communities with more than 50 users are used to retrieve tweets for phrase detection via topic modeling. The reason for this filtering is that topic modeling is most useful for finding phrases in a very large collection of documents where manual reading is not time efficient.

7. Conclusion

We have created a module that is used as a pipeline in social data analysis. The example in in the figures and notebook show that insightful observations can be made, and the derived results can be used to inform other domain experts' research.

Provided Jupyter Notebooks and complete result data sets are included in `https://git.txstate.edu/DataLab/twitterAnalysis`.

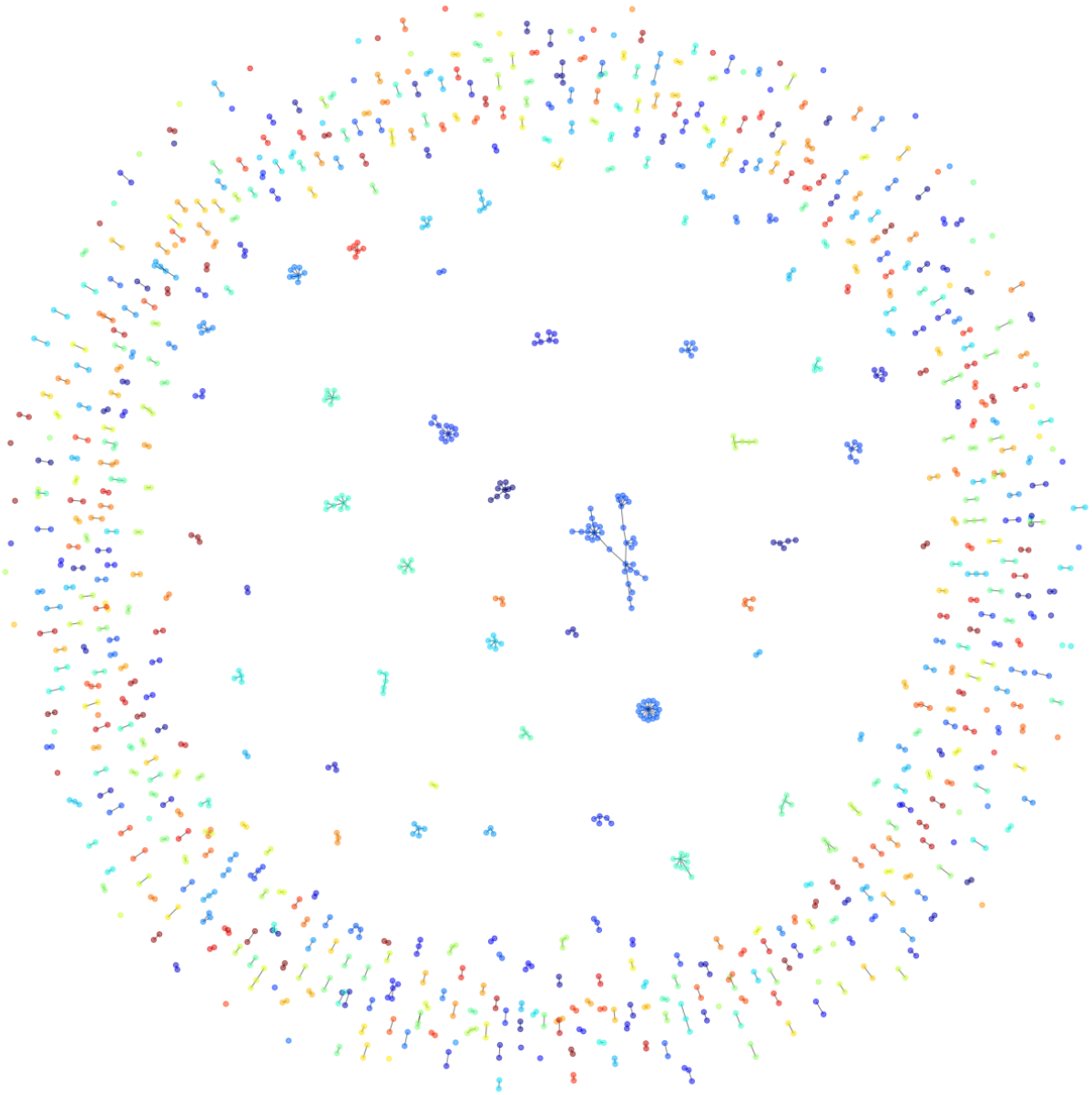

Figure 1. Sparse connected graph of communities. Edges representing replies only.

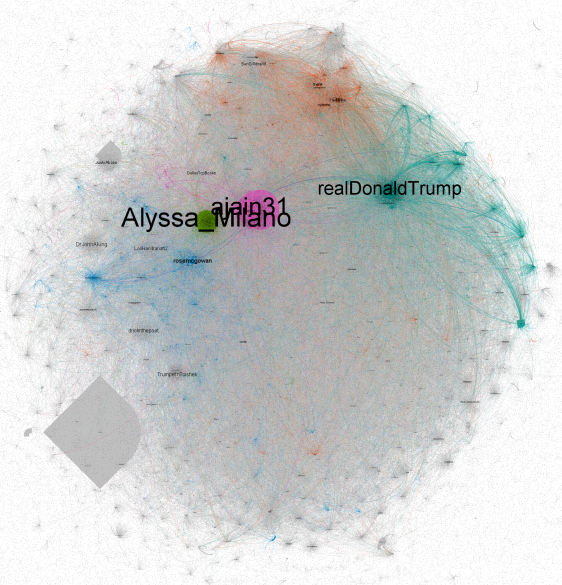

Figure 2. Highly connected graph with replies and mentions as edges. 1192042 nodes and 3075359 edges