TRACKING MENTAL HEALTH USING SOCIAL MEDIA

by

MANISH RANJAN

(Under the guidance of Shannon Quinn)

Recently, “Big Data” techniques have been successfully used to solve challenging problems in healthcare. Such techniques have given rise to the development of “biosurveillance” frameworks. These frameworks are an application of big data processing paradigms which addresses the problem of identifying and predicting threats to public health. However, existing biosurveillance platforms are limited in their applicability to task such as detecting seasonal outbreaks of flu or specific mental disorder conditions like schizophrenia. We present a biosurveillance framework that not only anticipates public health threats, but can identify at risk individuals from social media for non contagious diseases in the realm of mental health disorders. In our proposed framework, we combine topic modelling with sentiment analysis to provide an estimate of toxic or abusive behavior, identifying a pool of potentially at-risk users via their content on social media.  This framework can be tuned on incoming data incrementally over a period of time, which ensures better result over time on unobserved data. In a more mature phase our framework could be used by medical professionals to monitor and study users for their mental health illness/disorders more closely and accurately.

INDEX WORDS:     Mental Health, Behavioral health, health-surveillance, Twitter, Data Mining, Scalable Machine Learning, wellness, Topic modelling, NLP techniques.

TRACKING MENTAL HEALTH USING SOCIAL MEDIA

by

MANISH RANJAN

B.TECH., SASTRA UNIVERSITY, INDIA, 2008

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2016

© 2016

Manish Ranjan

All Rights ReservedTRACKING MENTAL HEALTH USING SOCIAL MEDIA

by

MANISH RANJAN

Major Professor: Shannon Quinn

Committee: Lakshmish Ramaswamy

Timothy G. Heckman

Electronic Version Approved:

Suzanne Barbour

Dean of the Graduate School

The University of Georgia

December 2015

DEDICATION

This is for you, xxxx.

ACKNOWLEDGEMENTS

I would like to thank Dr. Shannon Quinn for his guidance, not only in this

project, but also in my overall graduate academic career. I would also like to thank Dr.

Lakshmish Ramaswamy from whom I learned a lot during my two fruitful years at

University of Georgia. I would like to thank Dr. Timothy Heckman for accepting to

provide me guidance. I would also like to thank Al Aila, BahaaEddin, a MS in Artificial Intelligence student at the University of Georgia for all those white board discussions. Lastly, I would like to thank all my professors from whom I have learnt a lot throughout my academic career.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS v

LIST OF TABLES viii

LIST OF FIGURES ix

CHAPTER

1 Introduction 1

1.1 Introduction 1

1.2 Challenges 4

1.3 Major Contribution 4

2 Background 7

2.1 Twitter and its characteristics 7

2.2 Use of Machine Learning in Twitter 9

2.3 CyanoHABs and Meghdoot/Cyanotracker Project 10

3 Related Work 14

3.1 Disaster Management using social media 14

3.2 Disease Monitoring using social media 21

4 System Architecture and Overview 23

4.1 Challenges 23

4.2 System Architecture 25

4.3 Data Collection 26

5 Keyword Analysis 28

6 Machine Learning Analysis 38

7 Conclusion 54

REFERENCES 56

LIST OF TABLES

Page

Table 1: Number of Tweets 26

Table 2: Rate and Percentage of Relevant and Irrelevant Tweets 30

Table 3: Unique, Reliable and Active Users for each hashtag 31

Table 4: Correctly Classified Instance for two algorithms and average for each hashtag 45

Table 5: F-Measure for two algorithms and average for each hashtag 45

Table 6: Recall for two algorithms and average for each hashtag 46

LIST OF FIGURES

Page

Figure 1: CyanoTracker Architecture 12

Figure 2: System Architecture and Overview 25

Figure 3: Number of Relevant and Irrelevant Tweets 31

Figure 4: Top-20 frequent words in the Description of Relevant Users (Red Tide) 34

Figure 5: Top-20 frequent words in the Description of Relevant Users (Cyanobacteria) 34

Figure 6: Top-20 frequent words in the Description of Relevant Users (Blue Green Algae) 35

Figure 7: Top-20 frequent words in the Description of Relevant Users (Algae Bloom)...35

Figure 8: Same month Experimental Setup for Machine Learning Characteristics …….40

Figure 9: Different month Experimental Setup for Machine Learning Characteristics …40

Figure 10: Average F-Measure Vs. Difference in Months ……………………………...47

Figure 11: Average Correctly Classified Instances Vs. Difference in Months ………….47

Figure 12: Average Recall Vs. Difference in Months …………………………………...48

Figure 13: Jaccard’s Coefficient Vs. Difference in Months …………………………….49

Figure 14: Precision improvement after using retraining model (Cyanobacteria) ……...51

Figure 15: Precision improvement after using retraining model (Algae Bloom) ……….51

Figure 16: Precision improvement after using retraining model (Red Tide)……………52

CHAPTER 1

INTRODUCTION

**Come back to this** “In this chapter, we have briefly talked about the growing popularity of online social media, and also discussed how microblogs like Twitter, are used as data gathering platforms in citizen science activities. We have also described major challenges associated with social media for data analysis and our contributions for the same.

**1.1 Introduction**

Being mentally healthy is defined by World Health Organization (WHO) as state of well being in which (1) every individual realizes her or his own potential, (2) can cope with the normal stresses of life, (3) can work productively and fruitfully, and (4) is able to make a contribution to her or his community. However, having mental illness is a rather more serious problem. National Institute of Mental Illness (NAMI’s) report suggest that there are 43.8 million adults experiencing mental illness per year. NAMI’s latest reports also indicate that,

* 1 in every 5 adults in America experiences a mental illness.
* Nearly one in 25 adults in America live with a serious mental illness.
* One half of all the chronic illnesses begin by the age of 14 and three quarter by age of 30.

Above data suggests mental disorders are common in the United States. However, mental

disorder can be categorized as a disease only if a person experiences disability due to

serious mental illness (SMI). The criterion to define SMI is as follows as per National

Survey on Drug Use and Health (NSDUH).

* A mental, behavioral, or emotional disorder (excluding developmental and substance use disorders)
* Diagnosable currently or within the past year;
* Of sufficient duration to meet diagnostic criteria specified within the 4th edition of the Diagnostic and Statistical Manual of Mental Disorders(DSM-IV); and,
* Resulting in serious functional impairment, which substantially interferes with or limits one or more major life activities.

Given the widespread presence of variations of mental disorder, tracking it from its early onset with a generic approach is very important.

**1.2 Why tracking through social media makes sense**

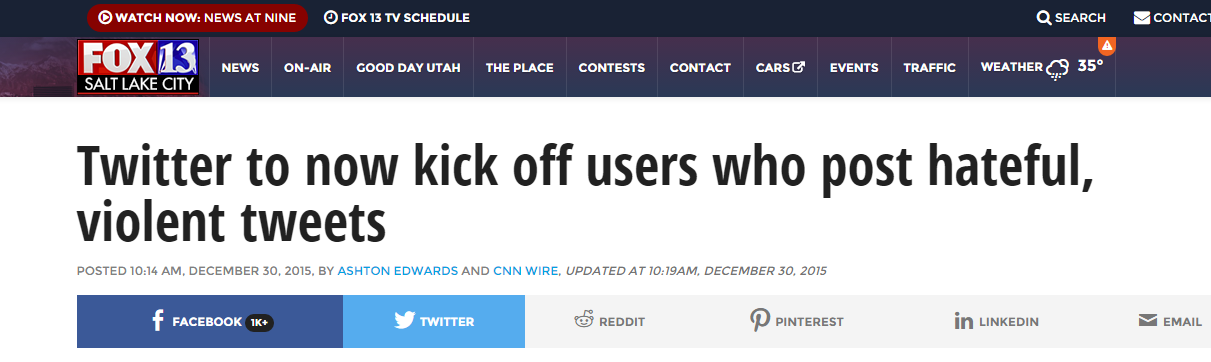
NAMI’s statistics suggests that mental illness is a problem which affects younger demographics more than adult ones. However, mental illness does not manifest the same way as more traditional disease like influenza; often there are no physical symptoms until it is too late.

The spectrum of disorder consists of: anxiety, mood, psychotic, eating, addiction, personality, obsessive compulsive disorders (OCD), Post-traumatic Stress disorders (PTSD) etc. [1]. Here, anxiety disorder is chronic mental condition characterized by an excessive and persistent sense of apprehension. Mood disorder is another psychological disorder characterized by the elevation or lowering of a person's mood, such as depression or bipolar disorder, to define a few.

Many states have already started the process to set up “Bio Surveillance” systems to detect epidemics, finding new health topics and trends using social media as input to harvest upon the social data derived knowledge power. US Department of Health & Human Services (<http://nowtrending.hhs.gov/>) is one such example.

However, being able to set up a broader criterion like “negativity over all” or “sharing toxic content” is not yet solved in a meaningful way. There has been some work like “Tracking Mental Disorders Across Twitter Users” [2]. It is a quality work but it restricts itself to very specific mental health issues like depression. Further, approach is naive, non-scalable and relies on crowdsourcing to verify accuracy. A potential problem with crowdsourcing with respect to tagging mental illness is, it is a very subtle problem in early phases. Detection by just looking at text is difficult even for very experienced psychiatrist. Crowdsourcing such detection hence is definitely not an accurate way to go forward. “Economic Costs of Alcohol and Drug Abuse and Mental Illness” 1985 is another sound theoretical work from an economist’s perspective, but suffers from the lack of readily available implementation strategy.

**1.3 Problem recognized for its seriousness by ‘Tech Giants’**

****

On the same line, Apple has released an app named “HealthKit” for tracking mental

health of individuals in last quarter and has shared plans on roadmap of this app as well.

**1.4 Challenges in Social Media**

Tracking mental health behavior in an online age could be attempted by tracking social media posts. Online social data has in general has two famous outlets, blog posts and Twitter/Facebook. Blog posts are   very difficult to sample content from a large number of unique users. One possible way to use blog posts is to connect with 4k sampled unique user’s blog individually and download their content. This is a non scalable approach since it involves detection of user of interest, finding their blog posts and then downloading content one by one.

Twitter on the other hand makes it very easy with its open interface architecture. We can plugin the API to get access to samples from a public stream.

Despite twitter’s ease of access to data, it has few unique challenges. Firstly, Twitter provides a very conducive platform for the rapid development of online rhetoric. These rhetorics neither appear in dictionary nor are formally recognized as a part of language. But it carries semantic meaning to large number of users and it is therefore critical to identify user’s intent from the rhetorics. Secondly, 140 characters’ limitation on tweets forces users to place many words together with incomplete and incorrect spellings. This makes data processing and building models in NLP domain a challenge. This is partially unsupervised problem in some sense, as we do not have a priori knowledge of all possible terms that are indicative of mental well-being. Furthermore, these terms are changing constantly, as new terms appear on the fly in response to changing circumstances.

**1.5 Major Contribution**

We wanted to study users who were filtered under broader criterion like “negativity over

all” or “sharing toxic content”. Further we wanted to study in detail if sharing of negative

and toxic content on social media could be a potential proxy for underlying mental

illness.

We focus on such filtered users based on not only toxic or negative content they

share, whom they follow, who follows them and what signals they choose to amplify.

Along with our own sentiment analysis algorithm, by adding network features, we seek to improve detection accuracy.

For detection of such user at high scale, we use generative statistical model Latent Dirichlet Allocation (LDA) from Natural Language Processing (NLP) domain, which allows set of observations to be explained by unobserved groups. Our goal is to design a scalable pipeline which could, use twitter data in holistic way, using scalable machine learning technique. Our pipeline should also tune its learning by training on incoming data incrementally over a period of time. Incremental learning would ensure better results with time on unseen test data.

We wanted to avoid using crowdsourcing as validation mechanism. Mental illness in early stages is really subtle and has little or no physical symptoms. Hence, even experienced psychiatrists find it difficult to diagnose it accurately. Hence we could not have left it to internet to decide how good the model was. Another issue with crowdsourcing is, it’s not a scalable approach. We consciously wanted to pick only scalable components for this pipeline, because we wanted the pipeline to work at scale of Twitter [13].

**1.6 Broader Impact**

One such pipeline could not only filter negative users at a very early stage but also unearth hidden pattern associated with negative content on social media.

One such pipeline idea is noble which uses NLP’s generative statistical model technique to filter and create unsupervised cluster of users, is capable of classify unseen users based on content of tweets, could self learn with time to improve accuracy and fall backs on an improved sentiment analysis algorithm for validation. [one potential work could be including feedback]

The rest of the thesis is organized as follows: CHAPTER 2 describes characteristics of Twitter, use of machine learning algorithms on Twitter data and also describes the architecture overview of CyanoTracker project. CHAPTER 3 describes related work, CHAPTER 4 gives an overview on system architecture followed by Keyword Analysis and Machine Learning Analysis in CHAPTER 5 and CHAPTER 6. CHAPTER 7 gives the conclusion of the analysis we have performed. [To be changed]

CHAPTER 2

BACKGROUND

In this chapter, we have briefly described different characteristics of Twitter, which play an important role in social media analysis. Also, we have described the importance of machine learning algorithms on Twitter Data. We have also given an overview of the CyanoTracker project initiated by the researchers at the University of Georgia. [**to be changed**]

**2.1 Social Media and its characteristics**

Tracking mental health behavior in an online age could be attempted by tracking social media posts. Online social data has in general has two famous outlets, blog posts and Twitter/Facebook. Blog posts are   very difficult to sample content from a large number of unique users. One possible way to use blog posts is to connect with 4k sampled unique user’s blog individually and download their content. This is a non scalable approach since it involves detection of user of interest, finding their blog posts and then downloading content one by one.

Twitter on the other hand makes it very easy with its open interface architecture. We can

plugin the API to get access to samples from a public stream.

Last 4-5 years we have seen a huge surge in effort to design good distributed systems

[11], [14] which can handle data at large scale. These systems are also designed to

consume huge volume of data to train machine learning algorithm which are specifically

designed to work in distributed setup. This has created opportunity for researchers and

scientist to create bio-surveillance systems which are predictive in nature and consume

social media as input. Some noticeable works are predicting PTSD [10], Postpartum

changes [15] using social media as input to name a few. While there has been lot of

quality work using twitter as a medium to detect and predict specific mental health

disorders, we will use this section to mention few of the most correlated with our problem

definition.

As a part of literature survey, we investigated different research papers related to mental disorders and use of social media like Twitter and Facebook as user put an increasing amount of personal information on these platforms. Research at Johns Hopkins to use Twitter to track the flu [1] and tweets analysis to provide insight in to metal illness [2] have established methods that can link the content of tweets to disease outbreak as well as specific mental disorders. Another notable work is around discovering co-occurrence Patterns of Asthma and Influenza [3]. There has been lot of work on finding accurate sentence sentiment of given short text using multiple techniques. However, work by Sara Rosenthaon combined study of many popular algorithms to find most accurate sentiment [4] was most comprehensive. The impact of celebrities’ influence on their followers was also studied in detail [5]. Looking at related area research papers, we have concluded to the best of our knowledge, that no direct work has been done to detect toxic content sharing among people. Further, most of the early attempts are very specific to one mental conditions. [8][9] [10]

**2.2 Sentiment Analysis**

Sentiment analysis has been handled as NLP task at many levels. Starting from document level classification task to (Turney, 2002; Pang and Lee, 2004), to at the sentence level (Hu and Liu, 2004; Kim and Hovy, 2004) and most recently at the phrase level Wilson et al., 2005; Agarwal et al., 2009).

However, for a more detailed and summarized study of role of social media in mental health research, we would like to refer users to De Choudhury 2013. De Choudhury has identified many ways in which NLP can be used to identify and predict mental health issues both at individual and population level.

For population-level analysis, surveys such as the Behavioral Risk Factor Surveillance

System (BRFSS) are conducted via telephone (Centers for Disease Control and

Prevention (CDC), 2010). Some of these surveys cover relatively few participants (often

in the thousands), have significant cost, and have long delays between data collection and

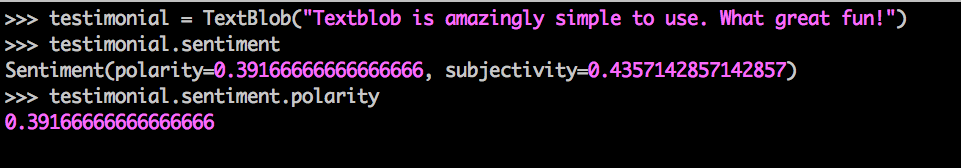
dissemination of the findings.

We were interested in usage of negative and positive words used by users. at the same time, we also wanted to know the distribution of positive and negative score for each individual. This was to get an overall sense of mental health by combining it with user’s network feature like who follows user, user follows whom, what is the impact of celebrity count.

We zeroed down on two approaches to handle sentiment analysis. (1) To use TextBlob package. [26] (2) We wrote a python piece of code which works on bag of word model to assign sentiment to a word based on a published sentiment scores file [27]. We will get in to detail of our implementation later in chapter [x] section [y]. We wanted two different approaches one taking sentence context in to consideration and another just using bag of word model.

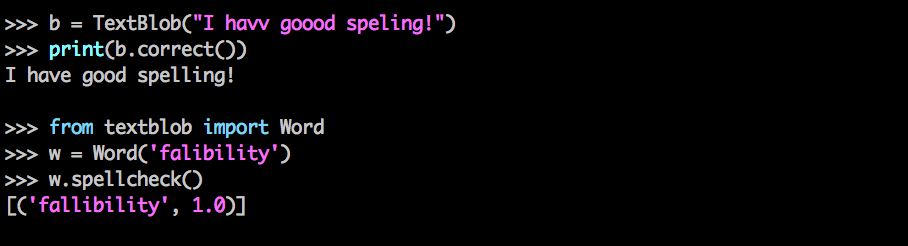
TextBlob’s [26] python package’s sentiment property returns a namedtuple of the form Sentiment (polarity, subjectivity). The polarity score is a float within the range [-1.0, 1.0]. The subjectivity is a float within the range [0.0, 1.0] where 0.0 is very objective and 1.0 is very subjective. It could take a sentence and return the sentence score.

e.g.



Reason behind using TextBlob’s sentiment package was it had really consistent and rich set of Application Programming Interfaces (APIs). Additionally, this decision was also influenced by TextBlob’s API’s capability to correct word spelling and spellcheck.

e.g.



However, while the accuracy of TextBlob was really good, speed was very slow. We believe that it also had something to do with data quality we had. For instance, if we had to look in to a formatted text doc and correct the spelling, the no of occurrences API will have to invoke corrector method will be very less. However, in case of tweets, because its very common to not pronounce word correctly, it was just proving to be overhead. Hence we moved on with our own Dictionary based scorer, which will be discussing in detail in chapter [x] section [y].

**2.3 Latent Dirichlet Allocation (LDA)**

In this section we will briefly topic modelling approach LDA[28] we used. In later section [x] we will introduce the adaptation of this model for twitter data.

LDA is an unsupervised machine learning technique whose primary purpose is to identify latent topics and word probability distribution for those topics from a large document collection. In LDA each document is represented as probability distribution of topics while each topic itself is probability distribution of words. LDA as a model allows us to play around with number of topic we expect the document to produce and also word in each topic given a word corpus as input. This also clearly indicates that LDA is designed on bag of word concept. In context of this problem, each document here is created by preprocessing user’s last accessible tweets. This kind of unsupervised topic modelling was especially suitable for us as we were looking at generic negative content and later how well each user’s tweet (document) was correlated with topics LDA has figured out.

**2.4 Unsupervised Clustering Techniques**

Two unsupervised clustering used in this experiments to cluster users are:

1. K-Mean [30]
2. Expectation Maximization (EM) [31]

The basic rationale behind selection these two algorithms was to find if the users have clear boundary when they are clustered together (K–Mean would work well) or they have fuzzier distribution. (EM should work well). We used “sklearn. cluster” [32] module of python to perform these clustering operation.

K means

1. Hard assign a data point to one particular cluster on convergence.
2. It makes use of the L2 norm when optimizing (Min {Theta} L2 norm point and its

centroid coordinates).

In contrast, EM

1. Soft assigns a point to clusters (so it gives a probability of any point belonging to any centroid).
2. It doesn't depend on the L2 norm, but is based on the Expectation, i.e., the

probability of the point belonging to a particular cluster. This makes K-means

biased towards spherical clusters.

**2.5 t-SNE (t-Distributed Stochastic Neighbor Embedding)**

t-SNE is a machine learning algorithm for dimensionality reduction developed by Laurens van der Maaten and Geoffrey Hinton. It is a nonlinear dimensionality reduction technique that is particularly well suited for embedding high-dimensional data into a space of two or three dimensions, which can then be visualized in a scatter plot. Specifically, it models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points [33].

It was well suited for visualization of our user’s cluster as they were expected to have high dimensional data as each topic is one of the dimension if user’s feature vector. We will get in to finer details of our user feature vector in section [x].

CHAPTER 3

**PIPELINE ARCHITECTURE**

Here is an Architectural overview at a very high level.

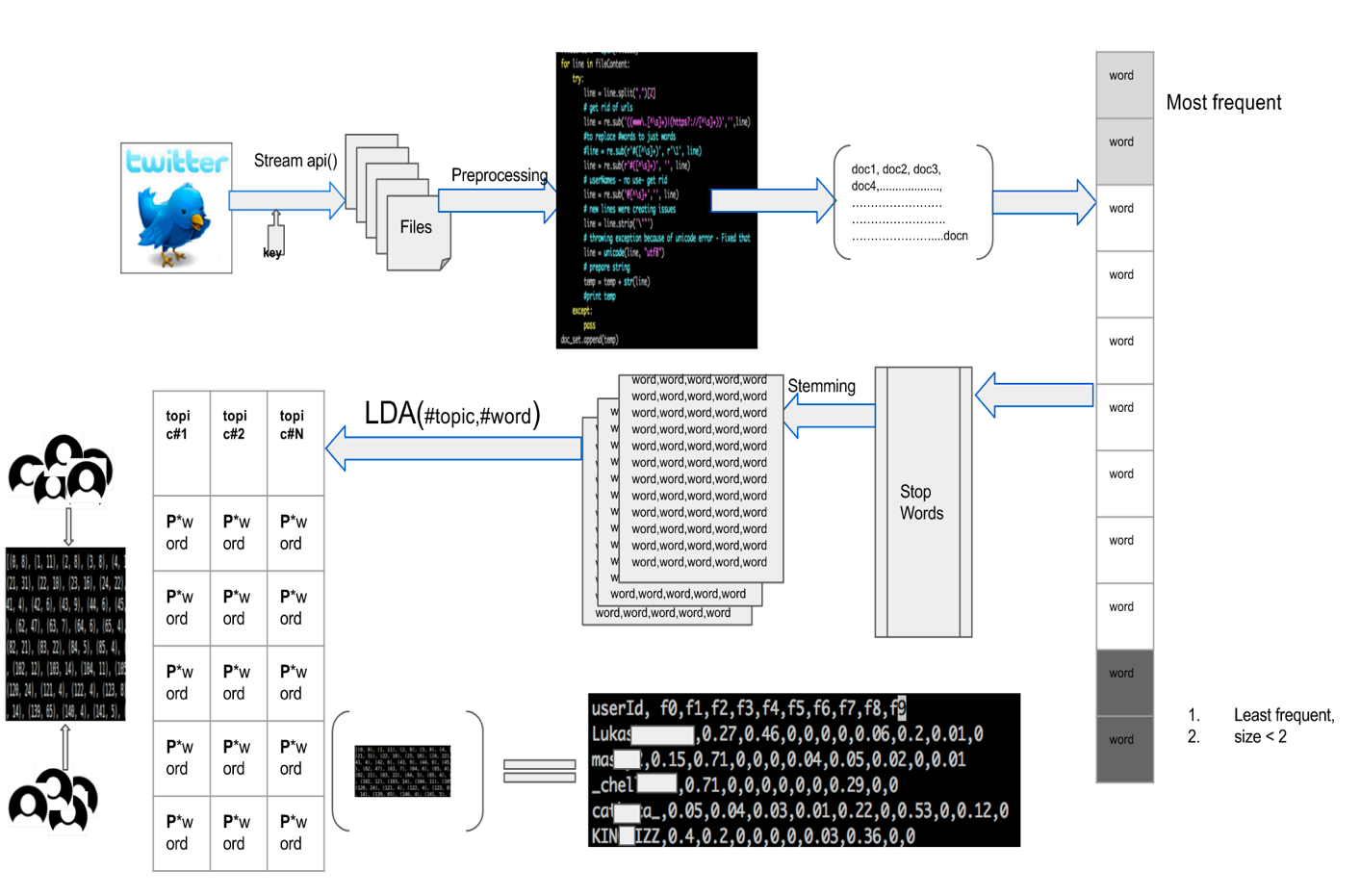


Figure High Level System Architecture for generating user feature vector

This pipeline illustrates how do we calculate user vector based on correlation with topic generated by LDA model.

High level of how Sentiment Analysis and other network feature works:



Figure High level system architecture for generating a feature vector in terms of user twitter data properties

**3.1 Data Collection**

All the data we obtain is public, and posted between 2014 and 2015 and was made available from Twitter via their APIs. At any point in time we did **not** made any attempt to obtain data which was either marked private or shared via direct message.

We used cloud virtual machine, Droplets by Digital Ocean [34] to download most of the twitter data. Some time when we needed more machine in parallel to work, we used Amazon’s EC2 [35] as well. Twitter provides two ways to access data. Because we wanted the data not to be biased, we went with stream option where we get access to 1% of public stream data without any filter.

Data downloading was mostly divided in 4 phases. Step 1 was achieved using stream API access where as 2,3 and 4 was achieved using REST APIs.

1. Connect to Twitter stream to download tweets from 1% sample access provided by twitter for free. We did this between Nov-Dec 2015. We then filtered user based on their language preference set as English (en).
2. At this stage we had more than 400,000 users. We just selected 10,000 of them randomly and downloaded their last available tweets. Twitter allows developers to have access to last 3200 tweets for a given userid (userid is identity of user on twitter and it has nothing of do with user’s name or any demographic info, it’s just a name which user have chooses to represent her/him on twitter network), using their APIs. We had then data for almost 9000 users as for some of the filtered users had private data policy set.
3. Out of 9000 users, we ran a script to remove users who had less than 100 tweets. The number 100 was selected based on purely experimental basis. We were left with close to 8000 users after applying this filter.
4. For these 8000 users, we also wanted to look at their friend network. Hence we used 6 EC2 machines to download their friends. This was very time consuming as users tend to have many friends. We later restricted the number to 5000 maximum friends for a given userid.

**3.2 Pre-processing**

Extracted tweets were in Java Script Object Notation (JSON) format. We extracted information we needed from this JSON object.

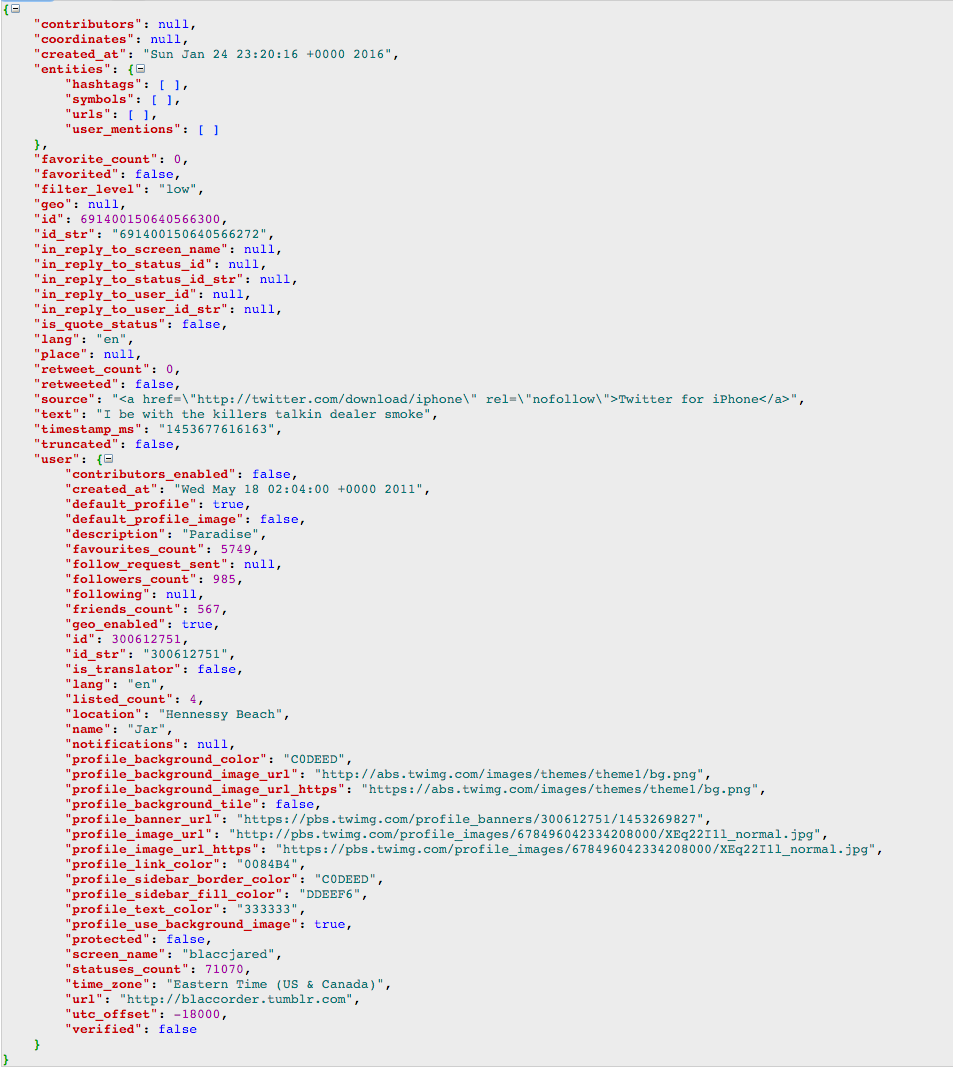


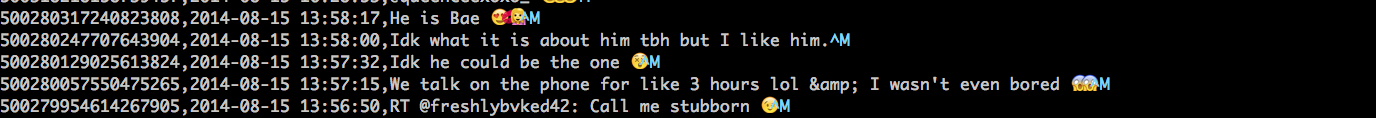
Figure Typical JSON format downloaded from twitter

As we can see that a typical tweet has lot of metadata associated with it. However, we were mostly interested in few very basic ones. At first just to get the user who had set English (en) as their language.

1. **def** extractTweet(inputdata):
2. #count = 0
3. **for** line **in** inputdata:
4. resultDict = json.loads(line)
5. **try**:
6. lan = resultDict["user”]["lang"]
7. **if** lan == "en":
8. var = resultDict["user”]["screen\_name"]
9. var = make\_unicode(var)
10. **print** (var)
11. **except** KeyError:
12. **pass**
13. **except** ValueError:
14. **pass**
15. **except**:
16. **pass**
18. **def** make\_unicode(input):
19. **if** type(input) != unicode:
20. input =  input.decode('utf-8')
21. **return** input
22. **else**:
23. **return** input

Once we had list of these users, we started downloading their last available tweets. For that we used tweepy [36]. We just downloaded in format {id, created\_at, text} format

e.g.



once we had a file for each user with this data, we were all set to perform sentiment analysis. These were the basic preprocessing steps we ran on these texts.

1. Remove the part of tweet’s text which had universal resource locators (urls) i.e. https or https
2. Remove hashtags (#), userids (@string), RT, RT”
3. Remove unicode characters
4. Remove the stopwords
5. Convert to lowercase

**3.3 Finding User’s Features using Twitter’s traditional data properties**

After preprocessing, we used TextBlob API to get sentiment score. TextBlob APIs call

produces one sentiment score for a given text. Modifying the source code for it to work on each word as text was overhead and was making program very slow.  Additionally, TextBlob uses inbuilt dictionary for scoring, hence it was not giving any weightage to slangs, shortcuts, and misspelled words which are very common in context of Twitter. We tried to fix this problem by trying to correct the misspelled words so we could use sentiment scores provided by TextBlob. However, experiment on sample data showed that, TextBlob.correct module was extremely slow. It took on an average two and half minute to process one document (one document is one user’s entire tweet text corpus). Keeping that number in mind we would have needed 2.30 minute each for 4000 users, which is approximately 10000 minutes, which is approximately 150 hours. Even with 3 machine working in parallel it was taking 2 days for one run. And we wanted to run it for all the possible topic counts like, 10, 20, 30, 40, and 50.  Hence we went ahead with our own sentiment analysis code. We used a simple algorithm which iterates over each tweet of a given user, after doing the preprocessing step, it tokenizes the words and then looks into to dictionary item for the score associated with the found word. It then based on sign of number (positive integer for positive word, negative integers for negative words) add the numbers to positive score or negative score.  This is a bag of word based model to compute the positive and negative score for each user. For creating a rich set of word and sentiment score we merged sentiment scores from two different sources to create a single word to score file. The negative word and files were:

1. AFINN-111.txt [27]
2. Negative-word.txt [36]

When a word was not found in dictionary, we gave it a zero (neutral) score. We cross validated it with TextBlob’s sentiment module. We were getting 13 out of 20 users same in the top negative list. We plotted these top 20% users to get a sense of trend.

**3.3.1 Users who fell in top 20% category of sharing negative content**

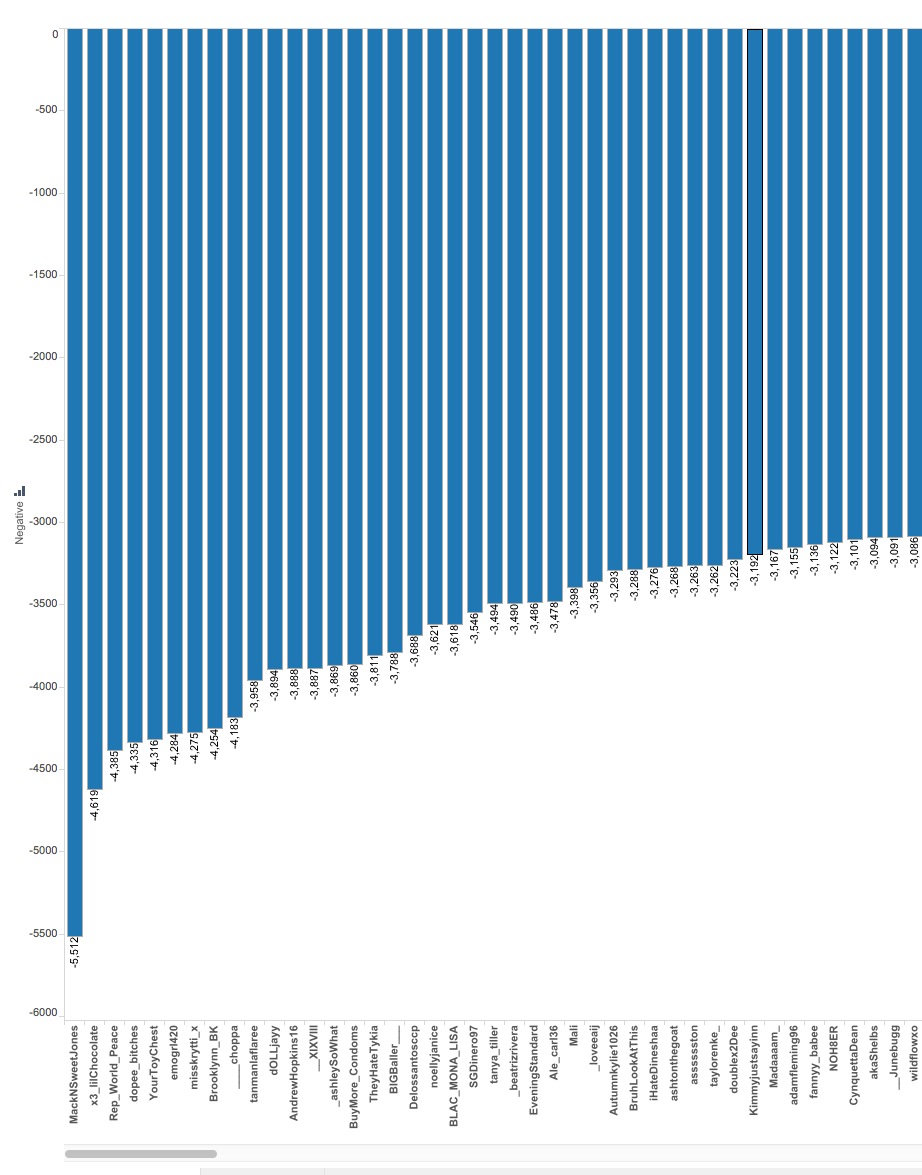


Figure User Sorted on Positive score

We also plotted users to whose negative contribution to social media was more than positive ones, i.e. the graph of user sorted on total score (positive + negative) .

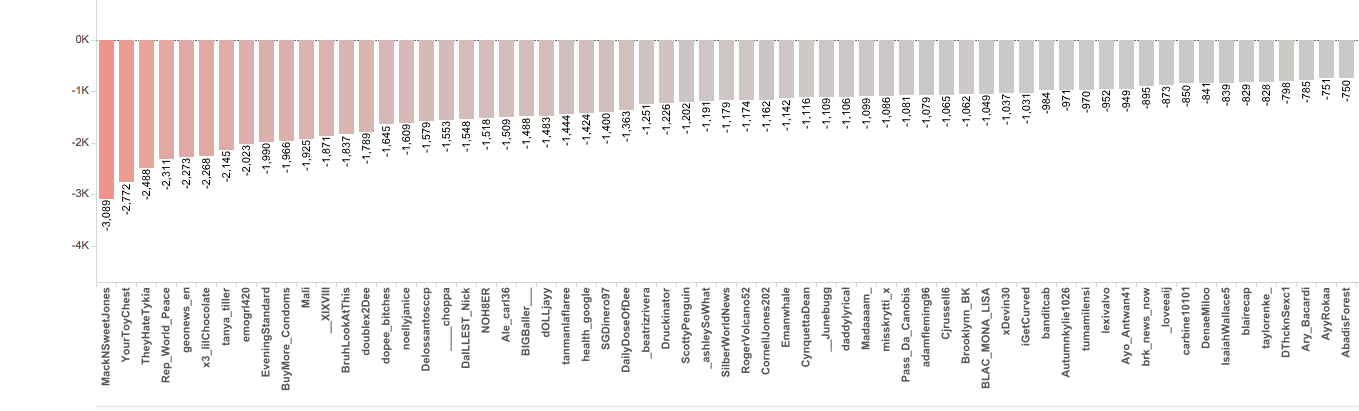


Figure User sorted on total score( negative + positive)

**3.3.2 Positive to Negative sentiment correlation**

We also looked at the distribution of positive to negative score across users to figure out how the distribution.

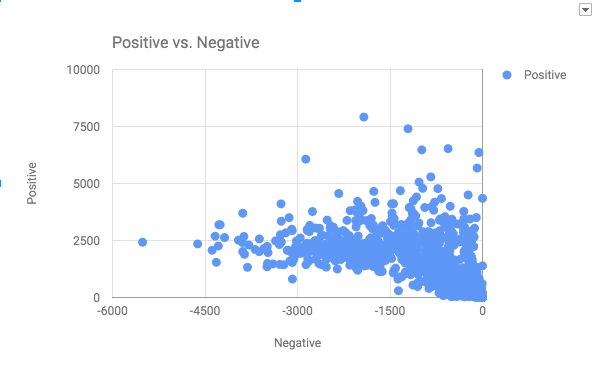


Figure Positive and Negative distribution of scores across 1k sampled users

Distribution was showing a clear trend that some users were polarized to share negative content [between -4500 to -3000] predominantly, but also had positive sentiment score of 1000-2500. However, users with high positive sentiment (greater than 5000) were sharing very less negative content.

**3.3.3 Sentiment correlation with “reply user does” and “gets”**

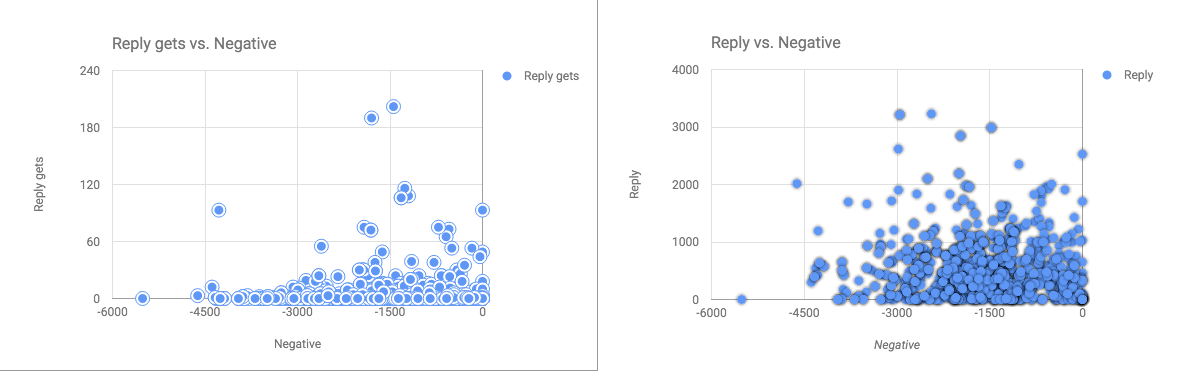


Figure Negative to Reply and Negative to reply gets

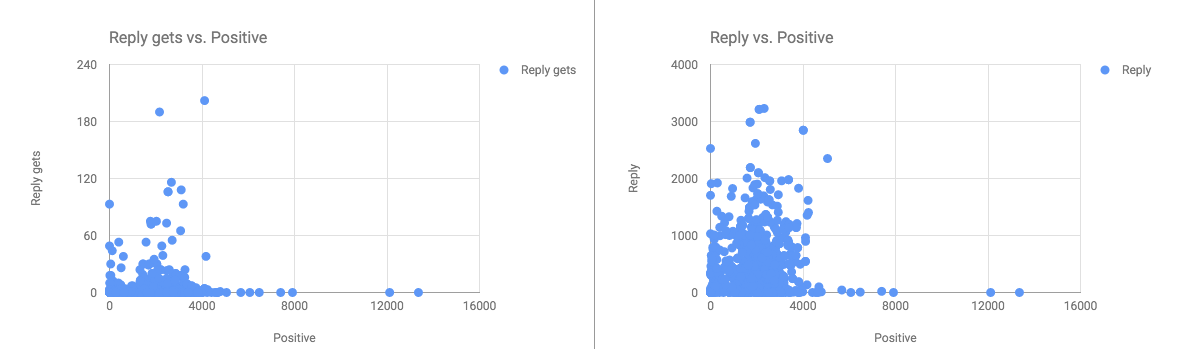


Figure Positive to reply and reply gets

Above graphs suggest that top 1% of negative user tend to reply lot more than top 1% positive users. This is a good signal to utilize as it seems user who have high negative score, get involve in communication with network users.

**3.3.4 Sentiment correlation with “Retweet Count”**

Next we plotted Below average negative to retweet count and above average positive to retweet count as well to see if there was something obviously standing out as trend.

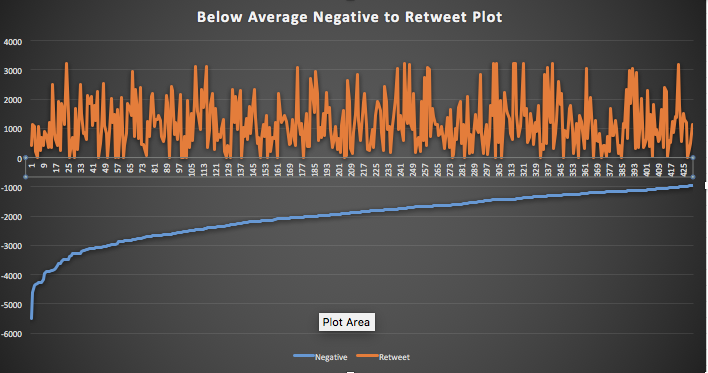


Figure Below Average Negative Sentiment to Retweet Count

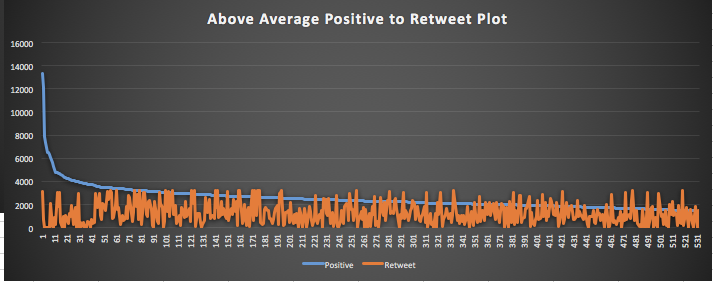


Figure Above average Positive Sentiment to Retweet Count

The plotted graph suggested that there wasn’t any pattern in the data we had for experiment. So we stopped pursuing this for feature selection.

**3.3.5 Sentiment correlation with “correctness (spelling) of tweets”**

However, there were few interesting patterns about sentiment to sentence correctness factor. We found it quite fascinating that there was obvious link between sentence correctness to negative sentiment.

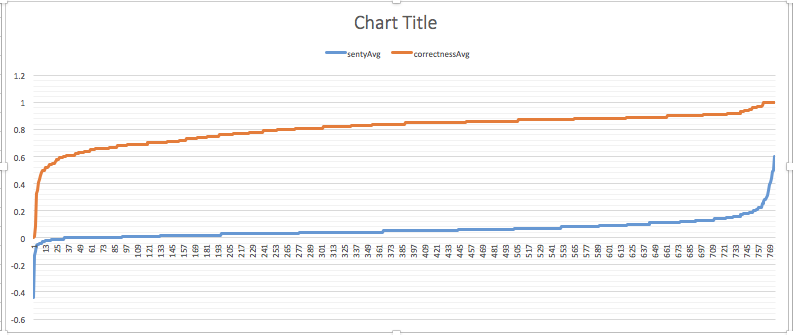


Figure Sentiment to Correctness Correlation

The graph seems to be suggestive of sentiment score directly proportional to correctness of the tweet. That suggest that most positive tweets are the well formatted ones and have very less spelling mistake. However, the most negative ones are also the least correct ones.

**3.3.6 Sentiment correlation with “Choice of words”**

We also investigated the idea if “choice of word” was different for people who were mostly tweeting negative in comparison to random people. For this after doing preprocessing steps on tweets, we ran a simple wordcount program to look at the frequency of words.



Figure WordCloud for “Word Choice” of 20 most negative users



Figure WordCloud for “Word Choice” of 20 most positive users

**3.3.7 User’s network structure**

We also looked at the user’s network structure to catch some strong differentiating network substructure between negative and positive subcategory. We dominantly used Gephi for that purpose. However, we quickly realized that, most of the users whose negative sentiment was dominant, were mostly either “comedy host shows” or “sports club” ids. We downloaded all the friends for userids and created a adjacency matrix representation for it.

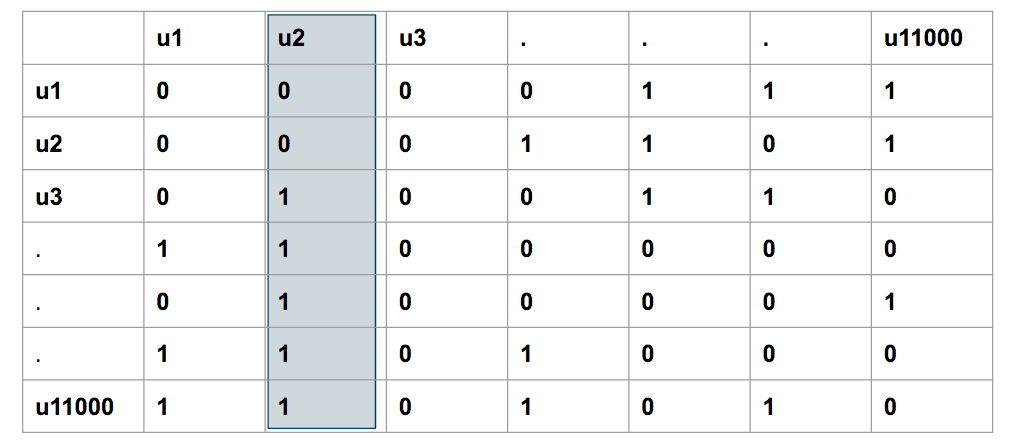


Figure Adjacency matrix representation of user and their friend network

To understand this matrix, a value of 1 at u3,2 represents that u3 has u2 in her/his friend’s list.The idea behind this graph was to find if there were “in-network” celebrities and if their content is affecting the network negativity or positivity over all.

When we plotted this matrix using tool Gephi [37], we got a graph like pasted below. Here, every user is represented by a node and size of the node is proportional to indegree of that node. Also, every edge represents a direct relationship between two nodes.

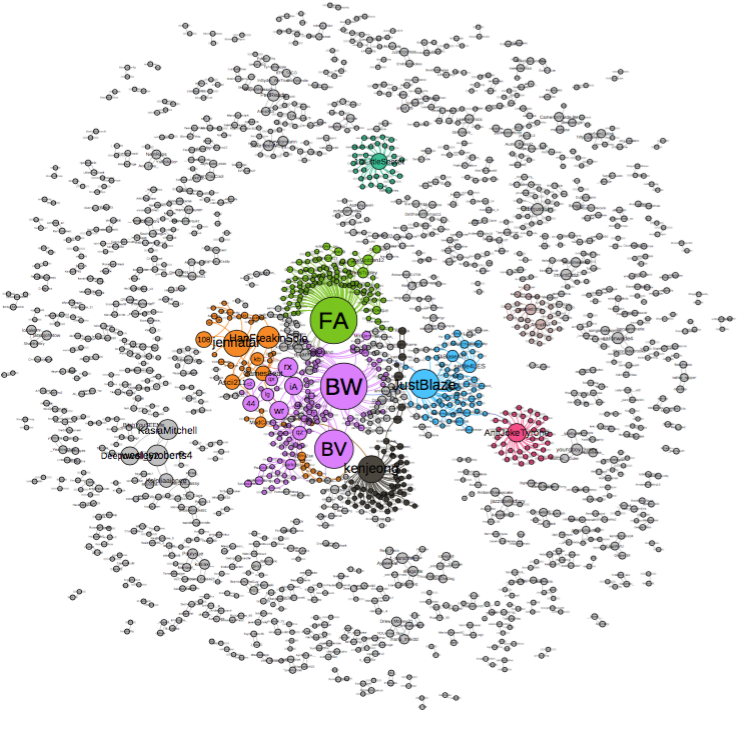


Figure network graph of users and their friends inside network

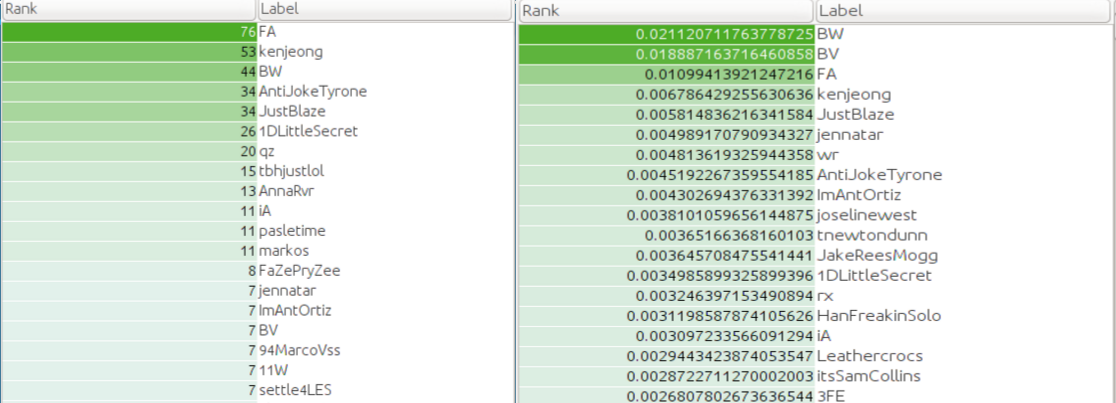


Figure Users in network sorted based on their indegree and pagerank

We did basic manual analysis on users and figured that the users were mostly either comedy shows official page or “one line offensive jokes”. Because most of such ids tweet at a very high frequency, it was very natural for these userids to have high negative sentiment.

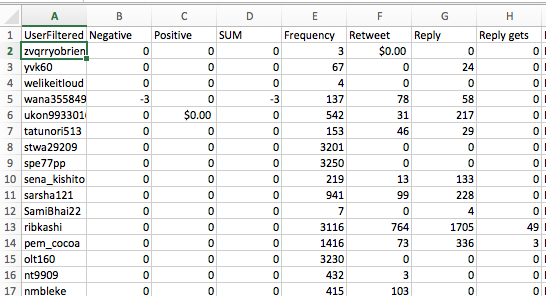


Figure Feature vector of each user based on analysis

**3.3.8 Clustering**

On this feature vector data, we tried to run EM, an unsupervised clustering algorithm.

The expectation was that it should be able to cluster the most negative users in one group.

EM run on this dataset gave us 7 possible clusters.

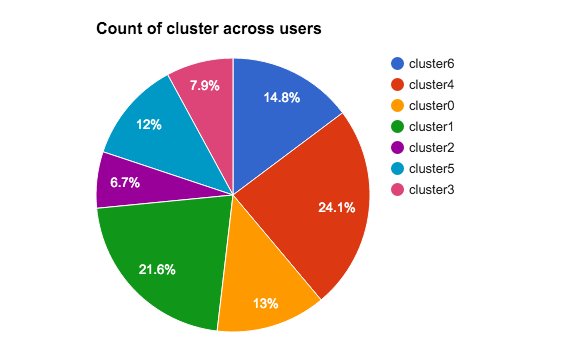


Figure Distribution of Clusters across dataset

However, most of the top 20 most negative users based on negative sentiment score were clustered in single group.

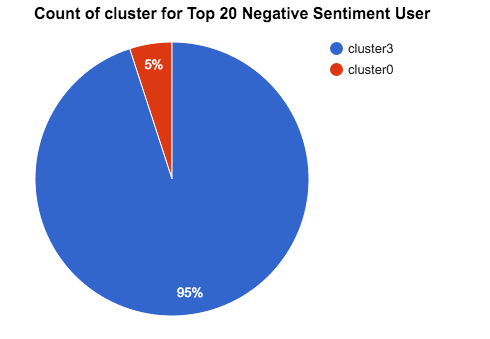


Figure Cluster Distribution for Top 20 Users

This was good result, as EM clustering algorithm was able to find correlation between all negative users. So we ran another experiment by tagging all users who had above average negative sentiment score as negative user and tried to see how the cluster distribution looks like.

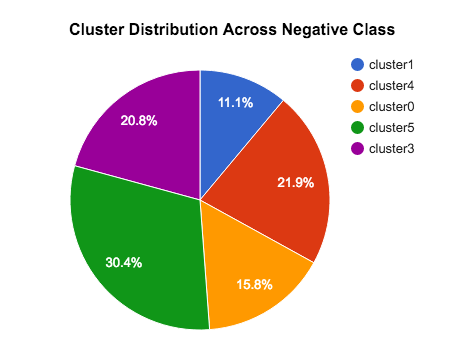


Figure Cluster Distribution of all users marked as Negative

This result was not very encouraging as the negative users were all over the cluster.

So, while the clustering did a really good job on top negative sentiment user, over all the cluster distribution was not okay keeping twitter traditional data properties as feature.

We were also concerned about Sentiment analysis getting too much weightage in the analysis. In previous attempts [8][9] [10] sentiment analysis was already explored as a possibility.

For the validation of results, we had two obvious options in mind. Either manual tagging or crowdsourcing. Both of them highly impractical and non scalable.

**3.4 Finding user’s features using “Topic Modeling” on user’s data**

Hence we decided to try a new pipeline altogether using “Topic modeling” [38] as a technique. Our overall architecture for this, was as described in the following figure.

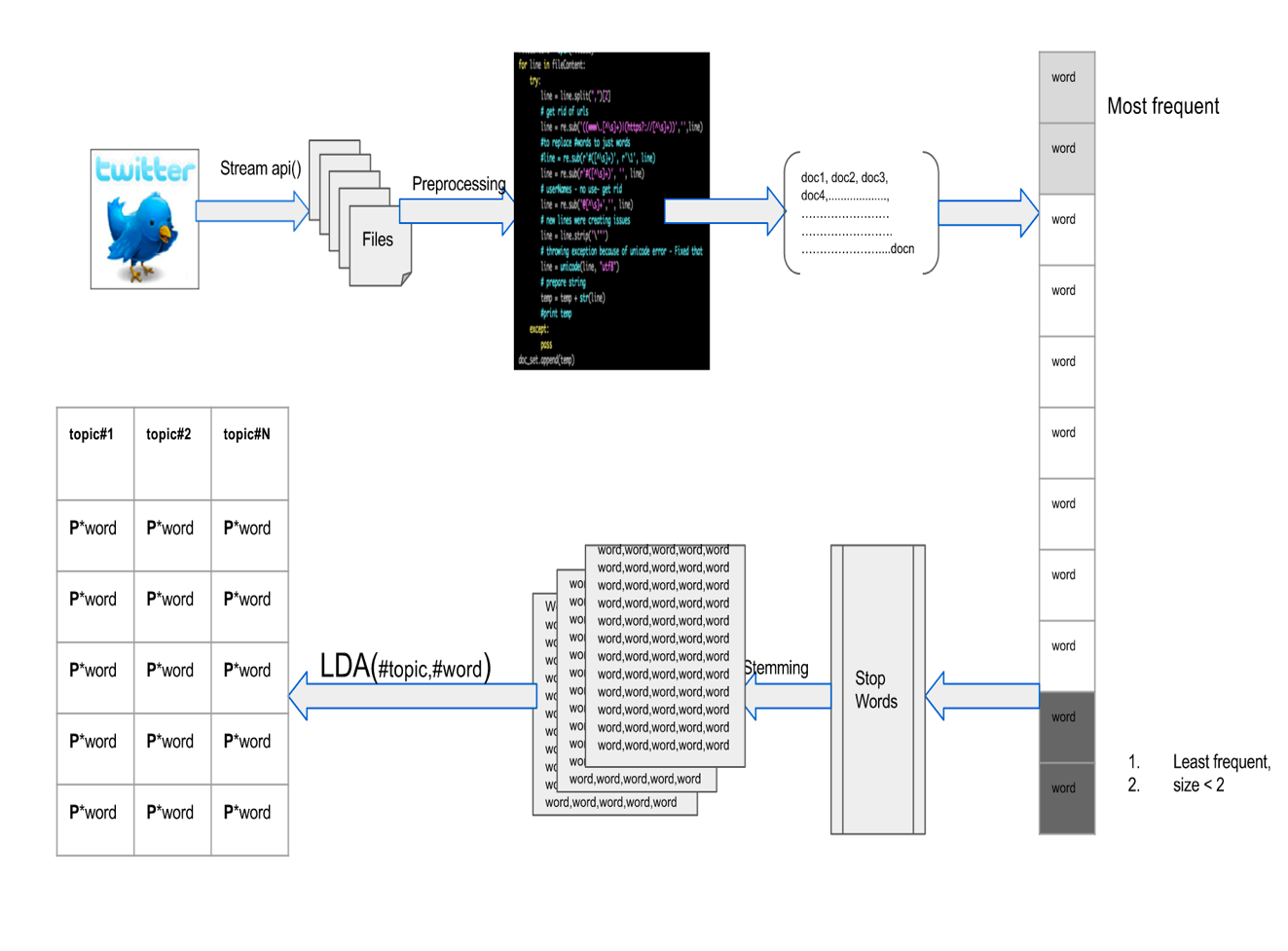


Figure Topic Modelling Pipeline Design

Steps involved in doing topic modelling over tweet corpus:

1. Data was downloaded as mentioned in section 3.1.
2. On downloaded data, preprocessing steps were applied as discussed in detail in section 3.2. The outcome of this result is to have a clean text corpus for each user, also knows as ‘document’ in this context.
3. Once we had document for each user, tokenize the document set to words.
4. Remove most frequent words as well as least frequent words. This is important to get a good LDA model as the model inherently works on bag of word [40] concept.
5. Remove words which had size < 2 (This step was experiment driven)
6. All stop words had to be removed. We used stopwords from Natural Language Toolkit’s (nltk) stopword's module.
7. We also added few words exclusively as they were meaningless words and must have been generated as a side product of some online rhetoric going on. Here is the list of words we extracted based on experiment output.



Figure Stopwords used for filtering words

1. Next we perform stemming using PorterStemmer module of NLTK package. Stemming helps extensively in increasing the word quality. As it reduces the multiple variation of words to single one. This helps in keeping the words in dictionary to minimum. E.g. likes, like, liking get converted to same word like after applying stemming.
2. Once cleaning stage is completed, to generate an LDA model, we need to understand how frequently each term occurs within each document. Hence, we need to construct a **document –term matrix.**  We achieve that using package genism. The output of this dictionary creation stage would be a dictionary made of unique tokens. The Dictionary() function assigns a unique integer id to each unique token and in parallel get the wordcount as well.
3. Next the dictionary has to be converted in to bag-of-words. The result corpus, is a list of vectors which will be equal to the no of document (which is the user’s count). doc2bow() function does that.
4. LDA model takes this corpus as input and generates LDA model. It takes input
   1. num\_topics: based on this input LDA model generates topic
   2. id2word: needs the dictionary we created in step-9
   3. passes: this is an optional parameter. Typically, this value has to be chosen based on experiments. High value is good, but a very high value can make the algorithm go slow.

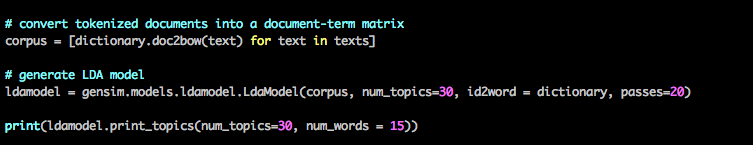


Figure Code Example of How LDA model is Computed

1. Output of above step is list of topics generated by model with probability value aligned with every word generated in each topic. Which is also the last step of Figure-21. We run this for multiple options of num\_topics. We experimented with 10, 20, 30, 40, 50 topics.
2. Once a model is generated, it can be saved for it to be used later with a simple command.

**ldamodel.save**('./data/lda\_twitter\_6310user\_30topics.model')

1. The result of above step 11 is (assuming num\_topic = 10, num\_word = 10)

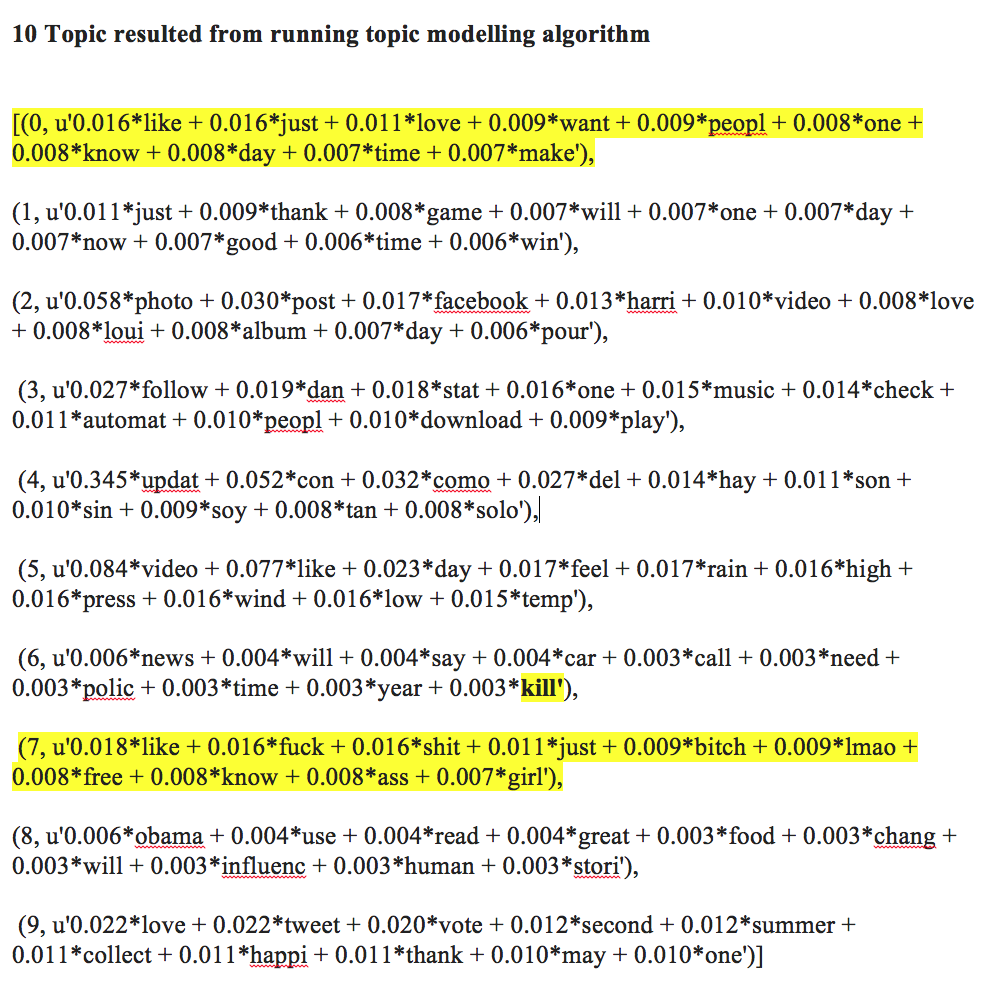


Figure 10 - Topics generated by LDA

1. Once the model generates topic and saved as per step 13, now we can compute the individual document feature vector on the basis of its correlation with topic. For that we compute bag of word for each document and pass it trained LDA model. The output of this step is as follows.

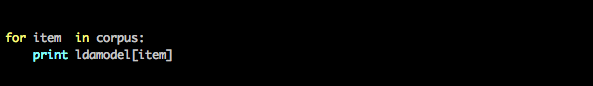


Figure Code Snippet to Print User's Individual Feature vector

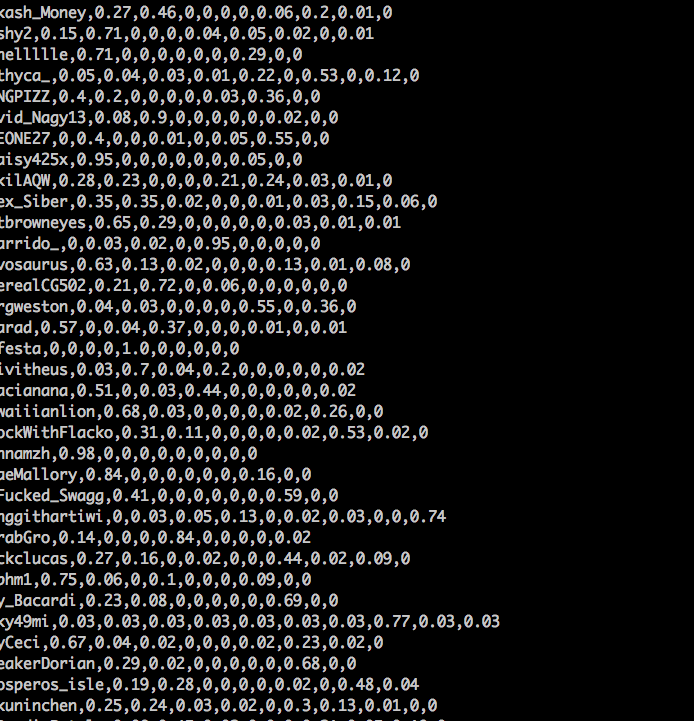


Figure User Feature Vector in Correlation with Topics

CHAPTER 4

**RESULTS AND EVALUATIONS**

**4.1 Data Distribution Sense**

To get a better sense of data, we plotted the feature distribution across data set.

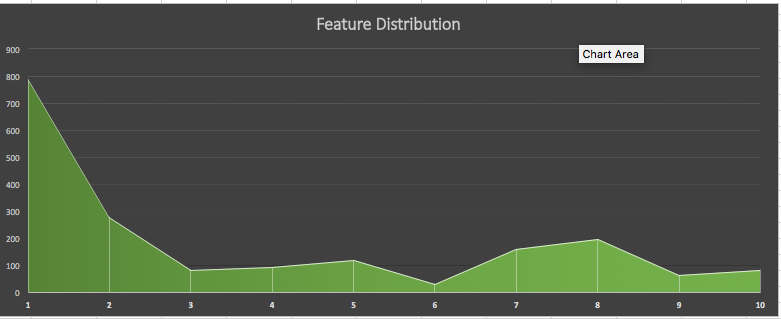


Figure Feature distribution – 10 topics

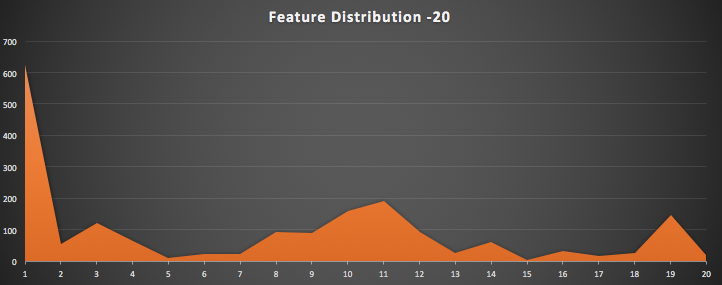


Figure Feature distribution – 20 topics

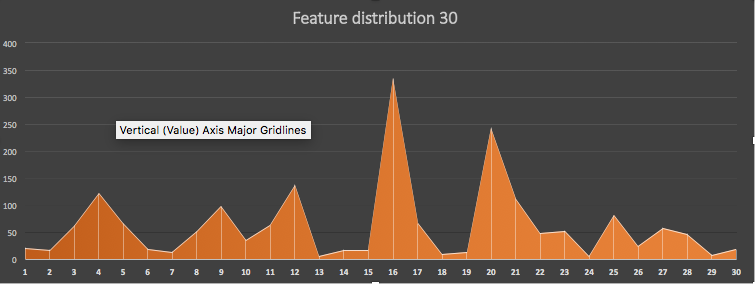


Figure Feature distribution – 30 topics

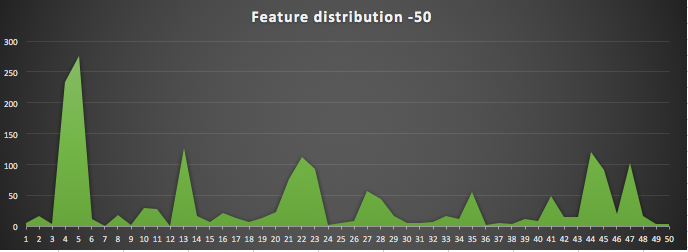


Figure Feature Distribution 50Topics

This study was done to get a sense of distribution of each topic.

**4.2 Feature Distribution and Assigning class**

As per Figure-26, we had user’s feature vector available. This could be also seen as user’s feature having 10 dimensional data where each dimension of feature represents one topic in the same order. Hence we plotted the feature vectors of top 20 negative and top 20 positive users to get a sense of feature vector distribution.

Figure Top Negative 20 Feature vector distribution

above graphs are very very interesting in any aspects and following observation can be made

1. Negative users’ tweets seem to be concentrated in two dimension, dimension 0 and dimension 7
2. However, positive sentiment feature vector spans across all dimensions and is not limited like Top Negative users.

Based on feature vector, we tried to give one of {Negative, NonNegative} class to each user. This decision of labeling a user was taken based on intuition of any user whose negative feature value in the user feature vector was higher than median, then that was marked as Negative.

This is how the distribution of each feature vector looked like.

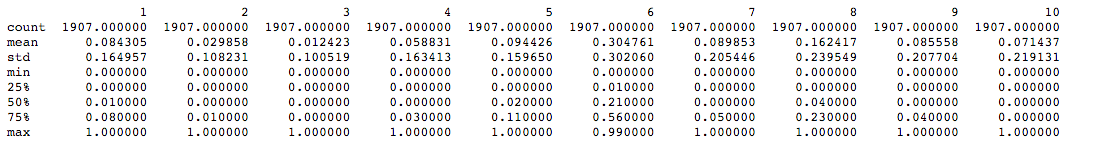
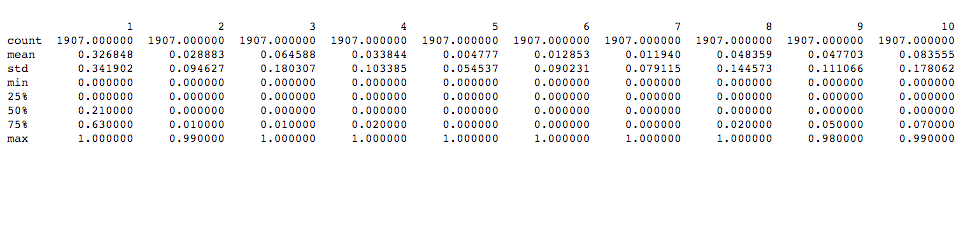


Figure Statistics related with 10 features



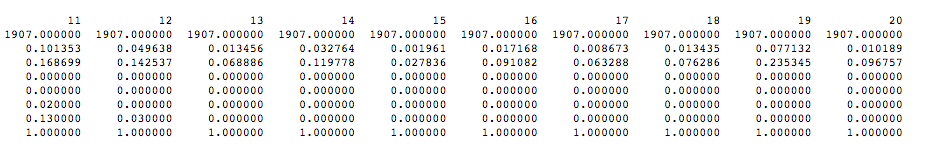


Figure Statistics related with 20 features

Similarly, we computed it for for 30, 40, and 50 features. To find the feature value which represents negative one, we simply look in to the topic distribution as shown in Figure-24 and as from topic it seems obvious that Topic 8 (7 if 0 included) is the negative one. Look at the distribution of feature 8 in Figure-32. Standard deviation is 0.23. Now look at the feature vector of each user in Figure-26. So a simple logic, which takes the feature-8’s value, compare it with the median value. If the the value is greater than the median we marked the user of class “N” (Negative) else “NN” (Non-negative).

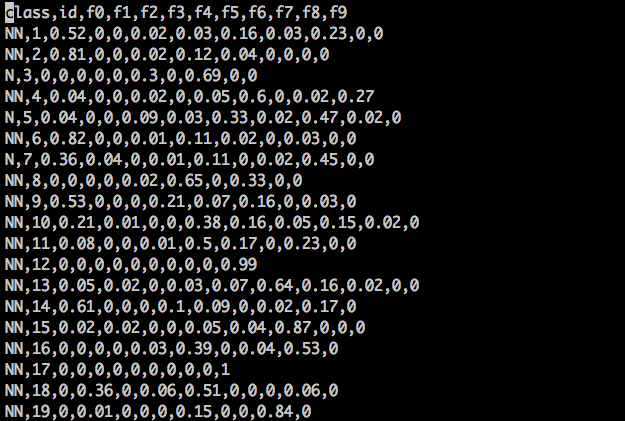
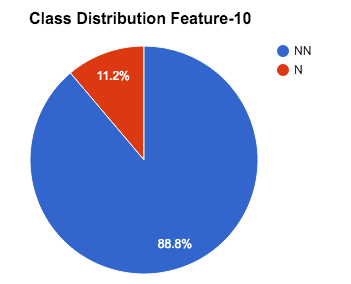
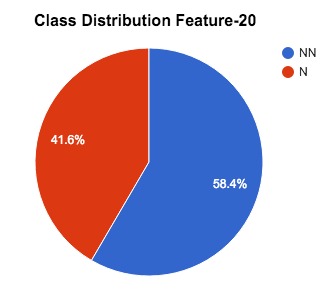
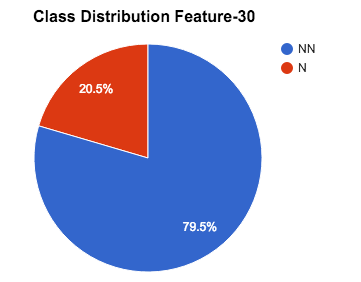
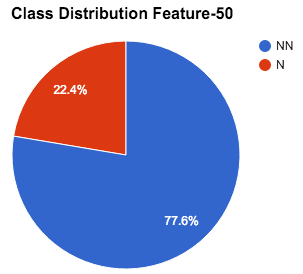


Figure Class included with feature vector

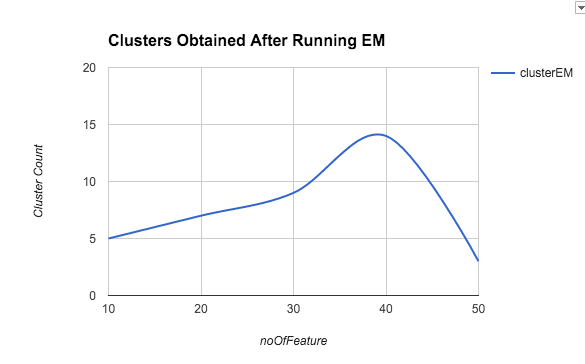
After doing this, to get a sense of class distribution we plotted different feature vectors.

Looking at the class distribution, we can say that this data distribution looks near real time, as data percent of people who use abusive language hovers from 10-20. Hence, next we wanted to see what kind of clusters exists, like we did it for traditional twitter data in section 3.3.8.

**4.3 Clustering**

We attempted clustering on data represented in Figure 26, to see how many clusters are present.



We studied clustering performed on 10 – topic to understand it in detail.

**4.3.1** **Clustering on 10 Dimensional Feature vector – Explained**

We ran the EM algorithm on Figure-34 data by excluding class feature. EM algorithm represented the entire data set in 5 clusters.

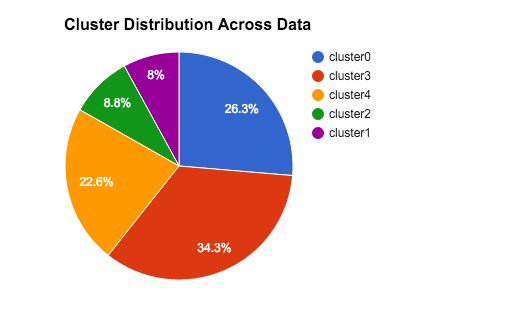


Figure Cluster distribution across data

Next we picked top 20 negative users who were sorted on their negative sentiment score.

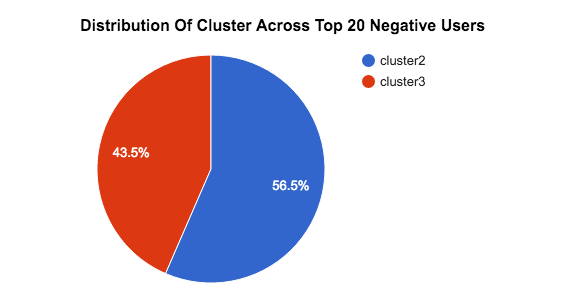


Figure Distribution of cluster among top 20 negative users

Its obvious from graph that the users in top most negative sentiment belonged to either cluster 3 or cluster 2, but we were also interested in understanding the login behind the clustering. Hence, we plotted the feature vector.

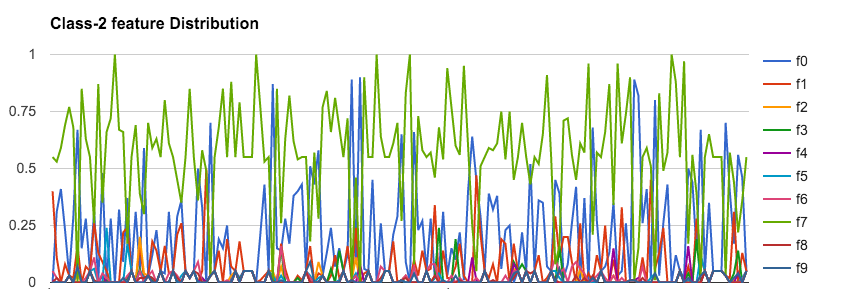


Figure Feature Distribution of users classified with class-2

The graph was suggesting clearly the dominance of f7 (feature-7) as a reason for classifying these users in class-2 category.

We also plotted users who were classified with class-2 with the labels (N/NN) we had created.

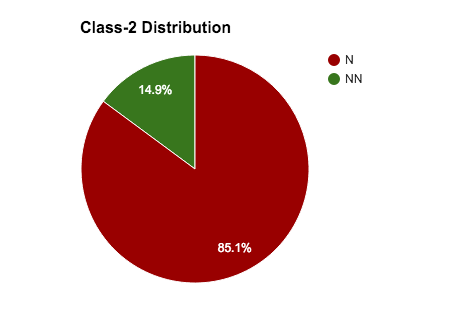


Figure Class -2's distribution on labels N/NN

Which was suggesting very strongly that most of the negative users were marked as class-2. And because the top 20 negative users were marked either class-2 or class-3 we did the similar analysis of class-3

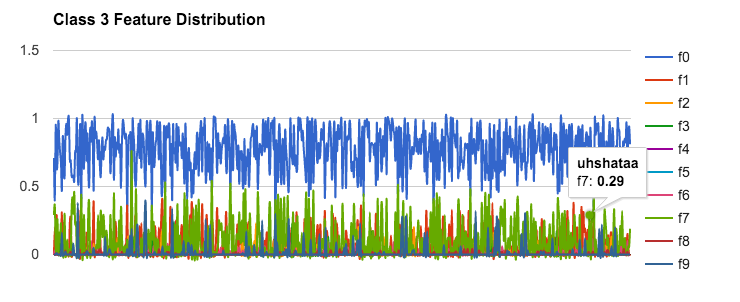


Figure 36Feature Distribution of users classified with class-3

Which was suggesting that while these users had presence of feature 7, feature-0 was dominant.

Hence, next we wanted to know, what percentage of total negative users, these class-2 and class-3 hold.

Total N in entire set: 213

Total N with combined classification of class-2 or class-3: 204

i.e. 95.7 % of users with N tag were placed in these class with a logic that

* if user had a dominant negative feature: class 3
* if the user had negative feature but not dominant: clsss-2

We did a quick analysis on 30 dimensional feature vector to cross verify if clear trend seen in 10 dimensional data was not merely coincidence and on 50 dimensional, because we wanted to confirm 2 things.

1. On what basis EM came up with just 3 clusters.
2. How well the trend seen with 10 and later 30 holds when the dimensionality of feature vector goes as high as 50

**4.3.2** **Clustering on 30 Dimensional Feature vector – Explained**

We first had the 30 topics generated from the same word corpus. Later we again picked the same top 20 with most negative sentiment users and plotted their feature vector.This is how the feature vector distribution looked like.

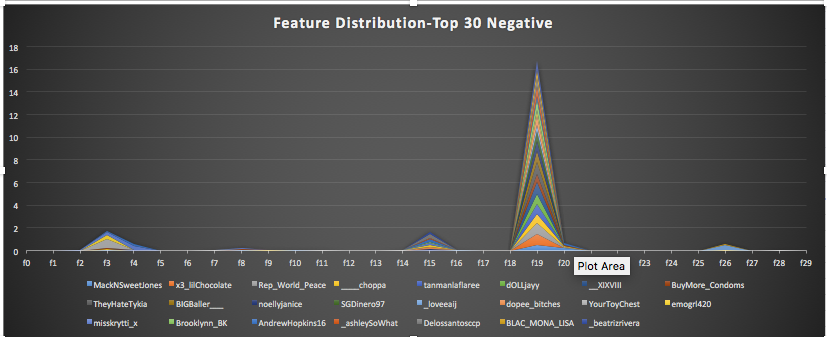


Figure Top 20 Negative Feature vector distribution

From feature distribution its evident that, for top 20 most negative sentiment users have a strong f19, and f3 and f15 of similar strength. Here is what those features are.

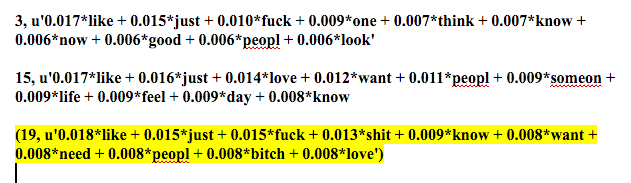


Figure Topics which were showing dominance for most negative users

This is good, as this is inline with what the analysis suggested for 10 dimensional user feature vector. To cross check we also plotted the top 20 user’s feature distribution, and this is how that looked like.

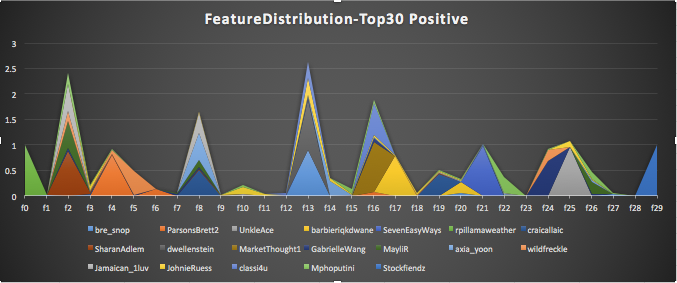


Figure Top 20 Positive Feature vector distribution

Which is also inline with our early observation that, while negative user’s tweets are really focused, positive ones like to talk about almost everything. Next we wanted to see how the over all cluster distribution looked like.

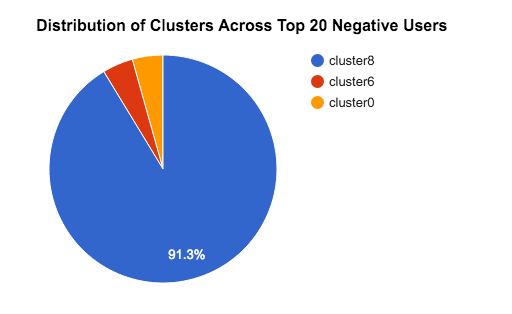
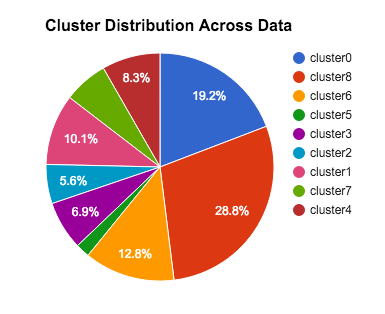


Figure Cluster distribution across data and Top 20 Negative users

Figure-43 suggested that negative users were clusters in group cluster-8. Hence, we also wanted to have another look at the total negative user distribution across clusters.

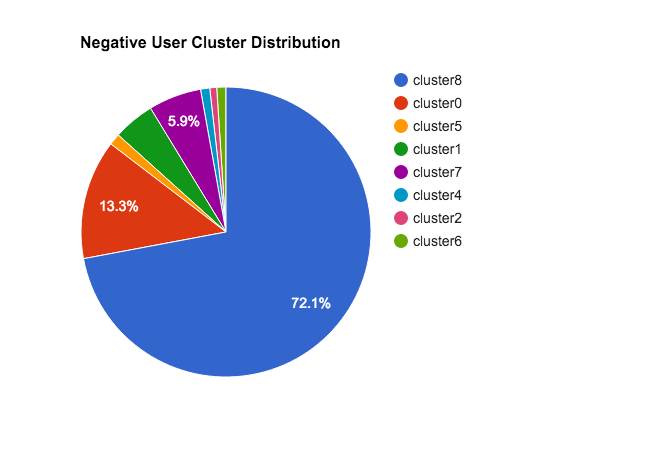


Figure Negative user cluster distribution

Above infographics suggested that, 85 % of the total negative users were concentrated in two the clusters {cluster 8, cluster0}. Which is always an indication of algorithm being able to group similar users in some space. Further analysis on cluster-8, gave a reasoning on clustering.

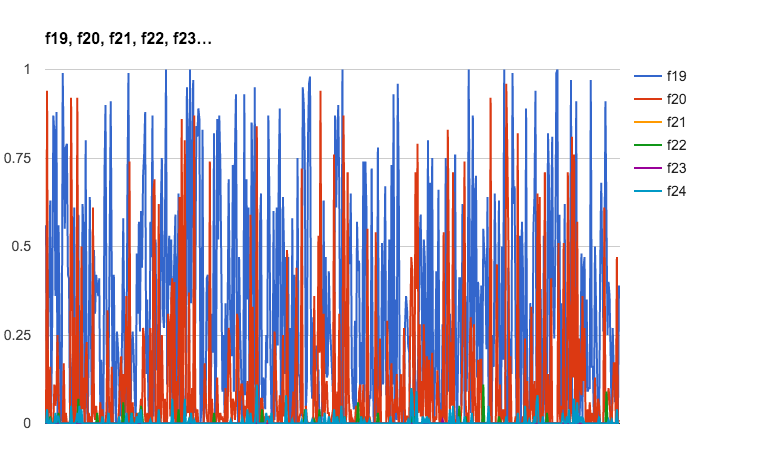


Figure Cluser-8 was designed based on f19 being the dominant feature

Upon experimenting further with feature vector sized 20, 40, and 50, we realized that the trends were just getting repeated. Hence we choose not to include that as a part of this document. Next challenge was to view these datasets in form of cluster.

**4.4** **Cluster Visualization using t-SNE approach.**

t- Distributed Stochastic Neighbor Embedding (t-SNE) is a technique designed for dimensionality reduction, that is particularly well suited for the visualization of high dimensional dataset. This was perfect for us. Here is the visualization we got when we plotted them in unsupervised (by ignoring class N/NN) and later by including class definition.

|  |  |
| --- | --- |
| Figure tSNE viz 10 Dimensional data-Unsupervised | Figure tSNE viz 10 Dimensional data - Supervised |

|  |  |  |
| --- | --- | --- |
| Figure : 20 Dimensional feature Vector | Figure : 20 Dimensional feature vector | |
| Figure : 30 Dimensional feature vector | Figure : 30 Dimensional feature vector | |
|  | |  |

REFERENCES

1. <http://www.webmd.com/mental-health/>
2. Sayali Kale UGA Graduate Student Computer Science.
3. ‘Using Twitter to Track the Flu (“Dredze et al 2014”)

# Analysis of Twitter posts could provide fresh insight into mental illness trends (“Dredze et al 2015”)

1. Discovering Multi-Scale Co-occurrence Patterns of Asthma and Influenza with the Oak Ridge Bio-surveillance Toolkit." *(“Ramanathan et al* 2015”)
2. ‘SemEval-2015 Sentiment analysis in twitter (“*Rosentha et al 2015”* )
3. “Retweets but not just Retweets - Quantifying and predicting influence on twitter (Rosemann et al., 2012).

# Detecting suicidality on Twitter (“O'Dea et al 2015”)

1. Mining Twitter Data to Improve Detection of Schizophrenia (“McManus et al-2015”)
2. Measuring Post Traumatic Stress Disorder in Twitter (“Coppersmith et al - 2014”)
3. <http://spark.apache.org/>
4. <https://radimrehurek.com/gensim/>
5. <https://about.twitter.com/company>
6. <http://hadoop.apache.org/>
7. Predicting Postpartum Changes in emotion and behavior via Social Media (“Choudhury et al-2013”)
8. Apoorv Agarwal, Fadi Biadsy, and Kathleen Mckeown. 2009. Contextual phrase-level polarity analysis using lexical affect scoring and syntactic n-grams. Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), pages 24–32, March.
9. Luciano Barbosa and Junlan Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pages 36–44.
10. Adam Bermingham and Alan Smeaton. 2010. Classifying sentiment in microblogs: is brevity an advantage is brevity an advantage? ACM, pages 1833–1836.
11. C. Fellbaum. 1998. Wordnet, an electronic lexical database. MIT Press.
12. Michael Gamon. 2004. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. Proceedings of the 20th intern
13. Munmun De Choudhury, Scott Counts, and Eric Horvitz. 2013a. Major life changes and behavioral markers in social media: Case of childbirth. In Proceedings of the ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW).
14. Munmun De Choudhury, Scott Counts, and Eric Horvitz. 2013b. Predicting postpartum changes in emotion and behavior via social media. In Proceedings of the ACM Annual Conference on Human Factors in Computing Systems (CHI), pages 3267–3276. ACM.
15. Munmun De Choudhury, Scott Counts, and Eric Horvitz. 2013c. Social media as a measurement tool of depression in populations. In Proceedings of the Annual ACM Web Science Conference.
16. Munmun De Choudhury, Michael Gamon, Scott Counts, and Eric Horvitz. 2013d. Predicting depression via social media. In Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM).
17. Munmun De Choudhury, Andres Monroy-Hernandez, and Gloria Mark. 2014. ” narco” emotions: Affect and desensitization in social media during the mexican drug war. Munmun De Choudhury. 2013. Role of social media in tackling challenges in mental health. In Proceedings of the 2nd International Workshop on SociallyAware Multimedia, pages 49–52.
18. TextBlob:<http://textblob.readthedocs.org/en/dev/quickstart.html#sentiment-analysis>
19. <http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010>
20. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. The Journal of Machine Learning Research, 3:993–1022, 2003
21. Liangjie Hong and Brian D. Davison. Empirical study of topic modelling in twitter
22. K-Mean: <https://en.wikipedia.org/wiki/K-means_clustering>
23. <https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm>
24. <http://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster>
25. <https://en.wikipedia.org/wiki/T-distributed_stochastic_neighbor_embedding>
26. <https://www.digitalocean.com/>
27. <https://aws.amazon.com/ec2/>
28. <http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html>
29. <https://gephi.org/>
30. <https://en.wikipedia.org/wiki/Topic_model>
31. <https://en.wikipedia.org/wiki/Bag-of-words_model>