Statistical sentiment analysis of Twitter as a review platform

Ankitkumar Jain
Masters of Computer Science
NC State University
Raleigh, North Carolina
aajain2@ncsu.edu

Chirag Jain
Masters of Computer Science
NC State University
Raleigh, North Carolina
csjain@ncsu.edu

Nirav Jain
Masters of Computer Science
NC State University
Raleigh, North Carolina
najain@ncsu.edu

Pratik Kumar Jain
Masters of Computer Science
NC State University
Raleigh, North Carolina
pjain22@ncsu.edu

Rishabh Jain
Masters of Computer Science
NC State University
Raleigh, North Carolina
rjain10@ncsu.edu

ABSTRACT

Reviews are perhaps the most important thing in deciding whether to visit a place or not and thus having a genuine review is essential. Generally, reviews available on websites and platform such as Yelp are more inclined to the positive aspects of a place. Moreover, the amount of usage of social media platforms has increased. Also, these platforms contain posts which express user's opinion in a more expressive way. Twitter is an important contributor, understanding the reliability of such tweets of being reviews is also important. To examine this, we have devised various approaches using natural language processing to analyze the tweets and also perform a statistical analysis of the metadata obtained from these tweets.

Keywords

Twitter, Sentiment Analysis, Reviews, Data Mining, Web Scraping, Tweet, Natural Language Processing, Abbreviation Expansion, indices, Yelp

1. INTRODUCTION

Reviews are the first thing explored by people before traveling to any place. Due to the digitalization, this can be collected from multiple sources. Some of these sources are Yelp, Zomato, and Google. A common observation found in these reviews is that they are mainly positive which can mislead the person and also create confusion as all the options are positively rated. Also, the number of reviews from these sources are less, which might not be desirable to obtain a better understanding of the place.

On the other hand, the use of social media has skyrocketed in the past few years. This can be proven by the figure (as shown in Figure 1) below which shows the growth of the number of users on various social media platforms from 2008 to 2017 in the US. [9]

Also, the posts on such social media platforms align with the user's opinion as they are more expressive on such platforms. Therefore, these posts have a potential to be used as a reliable source for the reviews. As a majority of such

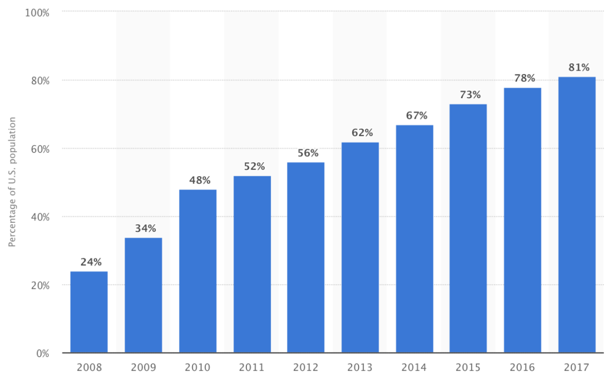


Figure 1: Percentage of U.S. population with a social media profile from 2008 to 2017

posts are contributed by Twitter, we have done a statistical sentiment analysis of tweets to verify whether tweets are a reliable source for reviews. Based on our study, we have found that indeed Twitter is a viable platform for review mining. We have used Natural Language Processing to do the same and obtained the polarity and subjectivity of such tweets. We have also created and defined three more indices associated with every tweet. They are review relevance index, support index and influential index.

2. CASE STUDY

In today's world, there are multiple social media platforms such as Facebook, Twitter, Instagram etc. Out of all these, on Twitter, the communication is more public so it is easier to connect to other people compared to other platforms. Also, Twitter, being focused on providing news about various events happening in the world using the hashtag feature, obtaining a tweet containing information about a place is easier.

Twitter is being used by almost all categories of users(as shown in Figure 2). [1] From the figure we can see, the distribution of Twitter usage among Age and Gender Demograph-

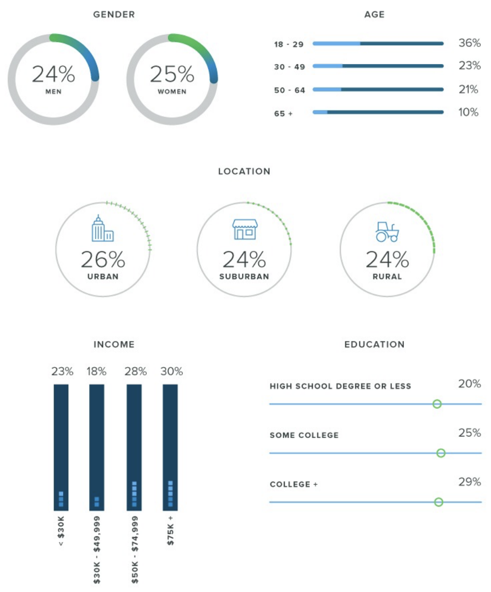


Figure 2: Twitter usage among key Demographics

ics, Location Demographics, Education Demographics and Income Demographics. This supports our choice of Twitter for our analysis as the tweets are contributed from almost all categories of users.

3. CURRENT SYSTEM

Performing sentiment analysis on the reviews from Yelp is not much useful, since the rating for a particular location is already available on Yelp. Therefore, the current system is mining reviews from social media websites like Twitter and Google places to perform sentiment analysis.

Current system is making use of JavaScript, HTML, CSS for front-end development and Python for the back-end development. The system has also included few libraries such as Textblob, python-twitter, python-google-places, spacy and standard libraries for python. Also, have used few APIs namely Google Places, Google Maps, Twitter API.

Some of the challenges in the current system are:

- **Vocabulary Correction** : Users do not use standard vocabulary therefore the system needs to find a way to translate user vocabulary into standard vocabulary.
- **Different languages** : Users use different languages therefore the system needs to understand the context in different languages.
- **API check** : There is a limited number of reviews that can be extracted from Twitter/Google places daily.
- **Unrelated tweets** : Some tweets mention the location of a place but are not reviews.
- **Different names for a location** : Users can refer to same location using different names or different location using same name.

4. INITIAL USER SURVEY

In this section, we described all the different questions we put into our survey, the reason why we asked those questions and what we concluded from the answers. We have used Google Forms to collect the data.

Question 1: Which source you generally use to obtain reviews before visiting a place?

We asked this question to divide the user group into categories based on the platforms they used to check the reviews before visiting any place. The users had an option to choose between a platform containing reviews or a social media platform.

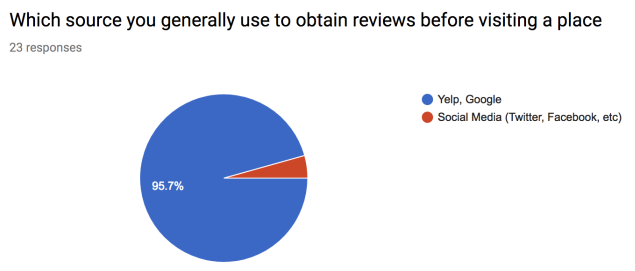


Figure 3: Responses for Question 1

Question 2: Do you write a tweet about a place while you are at that particular place?

We asked this question to see whether the users use Twitter more frequently while they are at that place or they tweet about it when they have time.

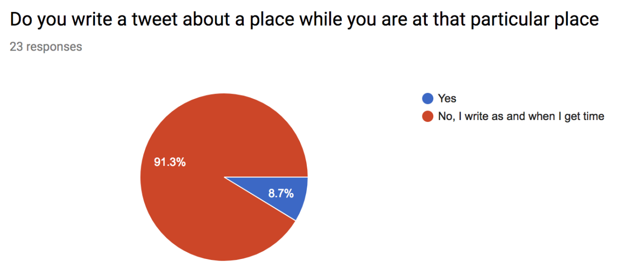
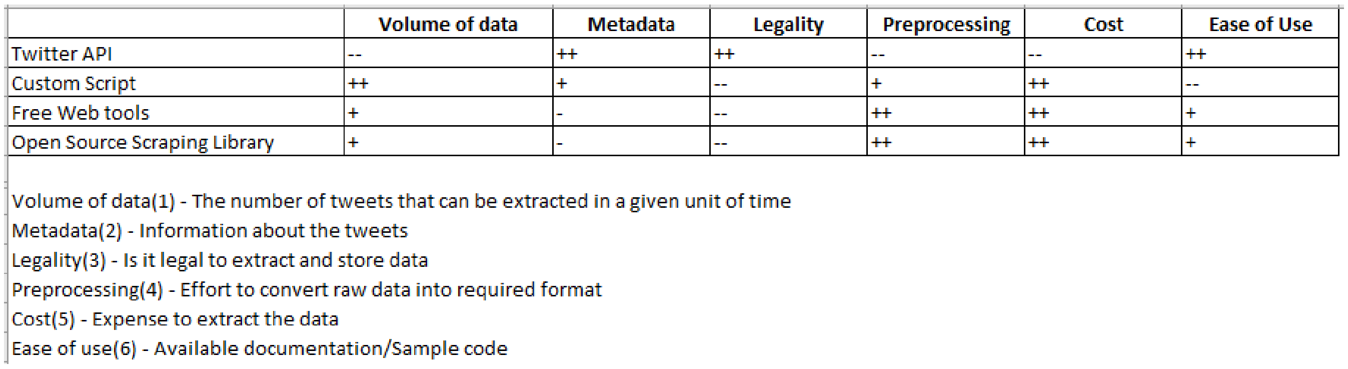


Figure 4: Responses for Question 2

Question 3: Would it matter if the review (tweet) about a location is from a celebrity?

We asked this question to obtain a user group who are influenced by tweets from a celebrity or a highly influential person.



	Volume of data	Metadata	Legality	Preprocessing	Cost	Ease of Use
Twitter API	--	++	++	--	--	++
Custom Script	++	+	--	+	++	--
Free Web tools	+	-	--	++	++	+
Open Source Scraping Library	+	-	--	++	++	+

Volume of data(1) - The number of tweets that can be extracted in a given unit of time
Metadata(2) - Information about the tweets
Legality(3) - Is it legal to extract and store data
Preprocessing(4) - Effort to convert raw data into required format
Cost(5) - Expense to extract the data
Ease of use(6) - Available documentation/Sample code

Figure 5: Assessment of various products/services

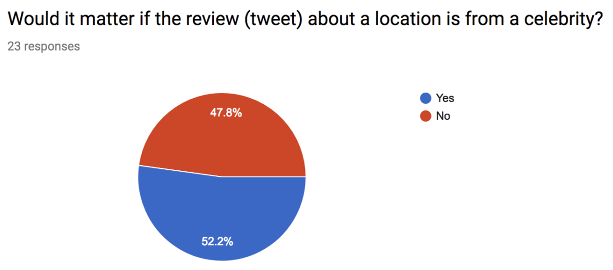


Figure 6: Responses for Question 3

4.1 Survey Conclusion

Based on the 23 responses we observed that the majority of people preferred review-based platform as compared to a social media platform. Also, the survey showed that a great number of users writes tweets when they have time and not when they are at that particular place. Along with that, the influence of a celebrity on the review of a place affects almost half of the users who took the survey.

5. IMPLEMENTATION

5.1 Technologies used

1. **Twitter API**: This was used to extract tweet text along with the additional metadata for each tweet.

2. **Python**: This was used to write the wrapper script around the Twitter API. Python was also used to store tweet related data into the database and also perform the calculation for each index.

3. **Cassandra**: Cassandra is a column-oriented NoSQL database that stores data of a particular column in physically contiguous locations. This was important for us because we are building several indices for performing analysis and for building 1 particular index we need data from just 1 or 2 columns from the entire database. Extracting data for just 1 or 2 columns from a huge database can be done efficiently using Cassandra.

4. **Spark**: This was used to get faster results for various analysis and visualizations as Spark can process the requests in parallel

5. **Tableau**[3]: After we created all the indices and stored in the database, Tableau helps us connect with Cassandra and view the data and perform different analytic tasks to identify the trends in indices. Tableau is an interactive data visualization tool that enables you to create interactive and apt visualizations in form of dashboards, worksheets to gain business insights for the better development of your company.

6. **DBeaver**[5]: This is a Tool that provides a user interface to run the Cassandra queries and see the formatted results.

5.2 Data Collection Techniques

This is a very crucial step of the proposed system. The volume of data or tweets collected will determine the accuracy of our results and the generality of the evaluation. Also, the amount of metadata collected in this step will help us define the proposed Indices [Section 5.5] better and with precise thresholds. The various available approaches for data collection are as follows:

1. Custom python script for web scraping

2. Twitter API and library

3. Open Source Scraping Libraries

4. Free Web-based tools

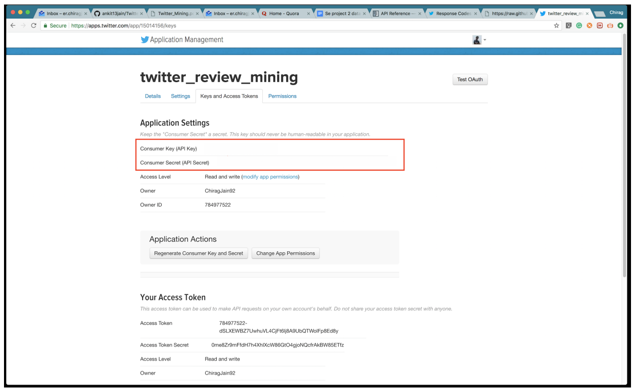
Each method has its own advantages and disadvantages as shown in Figure 5. Method 1 will give us full control over the volume and amount of information captured for each tweet. This will also enable us to extract more data with parallel processing. However, one of the major disadvantages of this approach is that the amount of metadata captured will be less compared to what the official Twitter API[15] provides.

Method 2, which is using the official Twitter API and libraries, provides a lot of metadata about a tweet[10]. More than the amount, the structure of data is well formatted eliminating the need for pre-processing. However, one of the major setbacks for this approach is the rate limitation[11] for 'Standard Subscription'.

Other methods mentioned in the list have all the disadvantages of Method 1 but with the incentive that they provide data scraping with minimum or no configuration overhead.

The amount of metadata and search query customization in these methods vary but are generally same.

We used a hybrid approach to incorporate the advantages of all the above methods into our tool. We developed a custom python script that gives us both the high volume of data that is similar to the web scrapper. Essentially, at the core of the program, we have used Twitter API. To tackle the problem of data volume we changed the token that we are using to call the API. To search at a rate greater than 18K tweets/15 mins the solution is to use Application only Auth instead of the Access Token Auth. Application only Auth has higher limits, precisely up to 450 requests/sec and again with a limitation of requesting maximum 100 tweets per request, this gives a rate of 45,000 tweets/15-min, which is 2.5 times more than the Access Token Limit as shown in Figure 7.[6]



https://apps.twitter.com/app/15014156/keys

twitter_review_mining

Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)

Consumer Secret (API Secret)

Access Level Read and write (modify app permissions)

Owner ChiragJain92

Owner ID 784977522

Application Actions

Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token 784977522-dSLXEWBZ7UwhuVL4CjFt6lj8A9UbQTWolFp8Ed8y

Access Token Secret 0me8Zr9mFfdH7h4XhlXcW86GtO4gjoNQcfrAkBW85ETfz

Access Level Read and write

Owner ChiragJain92

Figure 7: Application only Auth

To further optimize this, we are using the check-pointing mechanism in our search queries to the API. This is really helpful because we want the program to repeatedly fetch newer results since the last run. This checkpoint is the tweet_id for the latest tweet from the last run.

As the script is stateful/resumable, we extracted data from twitter till the API limit was not over or there we no new tweets to fetch for a particular business that given day. The next API call would then only return the new results, hence eliminating the need for handling duplicates at the API service level. This checkpoint is unique to each business thus providing the ability to extract high volume tweets for multiple locations with just one credential. Since the data is provided by the Twitter API returned, we have the entire metadata for each tweet.

5.3 Building Tweet Repository

As the free version of the Twitter API gives tweets of just last 7 days, we obtained the tweets of a few locations and food merchandise at regular intervals and built a database repository to perform analysis tasks. We realized that the different indices, that we would be building, would each perform analysis on just a small subset of columns respectively and we wanted the system to efficiently scale on a large dataset hence we decided to use a column family NoSQL database. In a columnar database, each field is stored by the column. This makes queries for analytics really fast. Example, in our case while finding if the tweets on twitter are highly polar, we were interested in only the polarity column and we can quickly obtain those for entire data as they are stored together rather than scanning each entire row for 1 field of interest.

The second advantage, which is also very important for large databases, is that column-based storage allows better compression since the data in one specific column is indeed homogeneous than across all the columns.

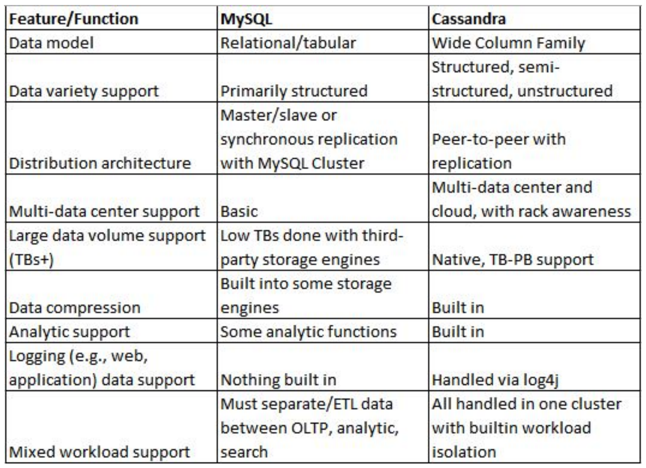
We have used Apache Cassandra to build the repository. Apache Cassandra is an open source NoSQL distributed database management system. Cassandra is designed to handle big data OLTP workloads across multiple data centers with no single point of failure, providing enterprises with continuous availability without compromising performance. We found Apache Cassandra is a standout among other NoSQL offerings for the following technical reasons:[4]

- **Massively scalable architecture** - Cassandra's masterless, peer-to-peer architecture overcomes the limitations of master-slave designs and allows for both high availability and massive scalability.
- **Transparent fault detection and recovery** - Cassandra clusters can grow into the hundreds or thousands of nodes. Because Cassandra was designed for commodity servers, machine failure is expected.
- **Flexible, dynamic schema data modeling** - Cassandra offers the organization of a traditional RDBMS table layout combined with the flexibility and power of no stringent structure requirements. This allows data to be dynamically stored as needed without performance penalty for changes that occur. In addition, Cassandra can store structured, semi-structured, and unstructured data.
- **Data compression** - Cassandra supplies built-in data compression, with up to an 80 percent reduction in raw data footprint. More importantly, Cassandra's compression results in no performance penalty, with some use cases showing actual read/write operations speeding up due to less physical I/O being required.
- **CQL (Cassandra Query Language)** - Cassandra provides a SQL-like language called CQL that mirrors SQL's DDL, DML, and SELECT syntax. CQL greatly decreases the learning curve for us who are coming from RDBMS world.

Figure 8 describes the difference between traditional RDBMS system MySQL and Cassandra.

The key attributes of the metadata of a tweet returned by the twitter search API that we are storing in the repository are as follows: [13] [14]

- tweet_id - ID of the tweet on twitter. (Primary key)
- tweet_text - the actual tweet
- search_query - The location or merchandise to which this tweet belongs
- favorite_count - Indicates how many times this Tweet has been liked by Twitter users.
- retweet_count - Number of times this Tweet has been retweeted



Feature/Function	MySQL	Cassandra
Data model	Relational/tabular	Wide Column Family
Data variety support	Primarily structured	Structured, semi-structured, unstructured
Distribution architecture	Master/slave or synchronous replication with MySQL Cluster	Peer-to-peer with replication
Multi-data center support	Basic	Multi-data center and cloud, with rack awareness
Large data volume support (TBs+)	Low TBs done with third-party storage engines	Native, TB-PB support
Data compression	Built into some storage engines	Built in
Analytic support	Some analytic functions	Built in
Logging (e.g., web, application) data support	Nothing built in	Handled via log4j
Mixed workload support	Must separate/ETL data between OLTP, analytic, search	All handled in one cluster with builtin workload isolation

Figure 8: Cassandra vs MySQL[4]

- lang - The language of the tweet text
- geo - The location from the tweet was created
- retweeted - Indicates whether this tweet has been Retweeted
- user_id - ID of the user who created the tweet
- user_followers_count - The number of followers this account currently has
- user_friends_count - The number of users this account is following (AKA their 'following')
- user_statuses_count - The number of Tweets (including retweets) issued by the user
- user_verified - When true, indicates that the user has a verified account
- user_favourites_count - The number of Tweets this user has liked in the account's lifetime.

Apart from the above fields we calculate and store the tweet subjectivity, polarity, review relevance index, support index and influence index which will be explained in the subsequent sections of this report. We also store the normalized value for every index in the database.

5.4 Building Bag of Words for Yelp Reviews

Bag of words is a model where every word in the text is represented along with the frequency of that word in the text. This simplified representation is used in natural language processing and information retrieval (IR).

We have used Yelp dataset for creating the bag of words model. Yelp is a platform which publishes crowd-sourced reviews about local businesses and a bag of word model based on reviews from such platforms will help in analyzing the tweets. The Yelp dataset consists of six files which has 5,200,000 reviews, 174,000 businesses, 200,000 pictures and 11 metropolitan areas [20]. The file that is important for the model is "review.json" which contains the review in string format in 'text' attribute.

The bag of words is generated using the Scikit-Learn library of python. From sklearn, we used the feature extraction module which consists of various different sub-modules such as text, image, etc. From these, text submodule consists of the CountVectorizer class which was used for creating the bag of words [22]. Before passing the text for creating the bag of words, we preprocessed the text of the reviews by tokenizing, stemming and removing stop-words. To improve the model, we removed the words that were not in the English dictionary. As the number of reviews was large, creating the model for all the reviews required a lot of processing power so we created a mechanism to store the model and reuse the same and trained it multiple times. To obtain words with high frequency, we normalized the count with the maximum and minimum frequency and considered words with normalized values greater than 0.1 as a threshold. The normalization formula used is *(x - min) / (max - min)*

5.5 Tweet Analysis Indices

Twitter corpora is a humongous database of tweets, however, every tweet is not a review. Therefore, it is important to filter tweets that are reviews and perform sentiment analysis only on those tweets. Also, Twitter consists of various types of tweets, such as General Tweets, Retweets, Quote Tweets, Promoted Tweets and so on [17]. It also consists of various types of users, such as general users, bots, and celebrities. The current implementation does not consider these factors while analyzing tweets for sentiments. However, we need to handle the bias of tweets introduced by these various factors instead of ignoring them, in order to generate correct sentiment about the reviews. Based on the meta-data of the tweets, we define the following indices to weigh a tweet for its review quality

5.5.1 Review Relevance Index

Review Relevance Index is a measure of how closely a tweet resembles a review. In order to identify whether a tweet can be considered a review or not, we first need to understand what is a review. There is no universal guideline for defining a "review", thus we will use machine learning for this task. Based on our initial survey, we identified that Yelp is a popular review platform. Therefore, we created a bag-of-words model on data extracted from Yelp with a normalized frequency score associated with each word, as explained in [Section 5.4]. Then, for every word in the tweet, we add this frequency score and later normalize the sum again to get a value between 0 and 1. Note that we are not using term frequency because Twitter is a micro-blogging website and therefore, checking the frequency of words in the tweet will produce poor results compared to term presence. Thus, this index provides us with a numeric value between 0 and 1. The higher the value, more closely does that tweet resemble a review. Note that normalization formula used here is standard normalization formula i.e. *(x - min) / (max - min)*. Figure 9 shows the visualization of normalized review relevance index, with the number of tweets for that value.

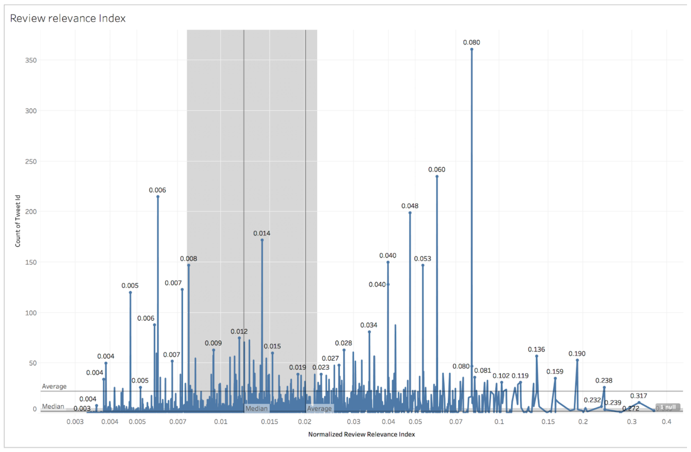


Figure 9: Tweet count vs Normalized Review Relevance Index for Chipotle

5.5.2 *Tweet Support Index*

Metadata about a particular tweet, such as the number of retweets, and the number of likes are important factors to consider while mining reviews from a tweet. If the number of likes of a negative tweet, such as "I hate the traffic near Empire State Building", is high then it means that many people feel this way about traffic around Empire State Building. Therefore, it is crucial to not disregard such information, but rather include it while analyzing a tweet. For this reason, we define a tweet support index associated with every tweet. It is a measure that tells us if people agree with the opinion present in the tweet or not. It is defined as follows: *support-index = (retweet-count * retweet-factor + fav-count) / past-tweet-count.* Each term in the formula can be referred to in [Section 5.3]

Once we calculate the support index for each tweet, we normalize them to get a value between 0 and 100 using formula: *100 * (x - min) / (max - min).* We multiply by 100 in order to preserve the precision up to 2 digits of the normalized values. Figure 10 shows the visualization of normalized support index, with the number of tweets for that value.

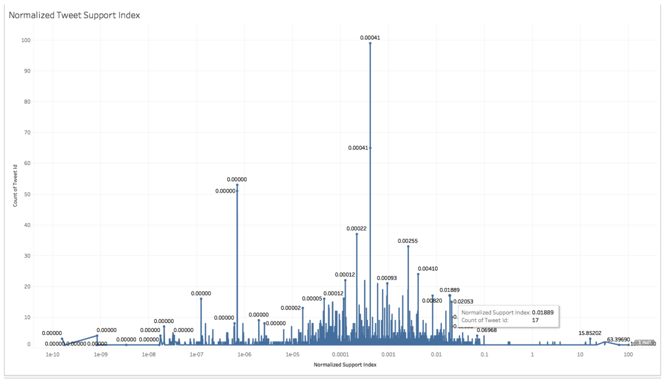


Figure 10: Tweet count vs Normalized Support Index for Chipotle

5.5.3 *Influential Index*

Influencers on Twitter are the people who inspire people in their industry. A good review of such an influencer will introduce influential bias in tweets of their followers. Consider the following tweet by Elon Musk: "Dinner in an old Belgian ironmongery. Best menu art ever." [16] Since Elon Musk loves the menu of this restaurant, chances are that Elon Musk's followers would also love this menu. It is important to factor in this information while analyzing the reviews of this particular restaurant because influencers may be bought for marketing purposes and thus their tweet might not reveal the real underlying sentiment. For this purpose, we will associate an influential index with every tweet. A straightforward approach to identify an influencer is to simply consider the number of followers of the user who is tweeting. We define a tweet support index associated with every tweet, as follows: *influential-index = 100 * (numberOfFollowers - minInfluence)/(maxInfluence - minInfluence).* A large number of followers will lead to a high influential index. Also, we have normalized the values of the influential index between 0 to 100 because the values are quite small. This is due to the fact that celebrities usually have a very large number of followers(in millions) while most of the common folks don't. Figure 11 shows the visualization of influential index, with the number of tweets for that value.

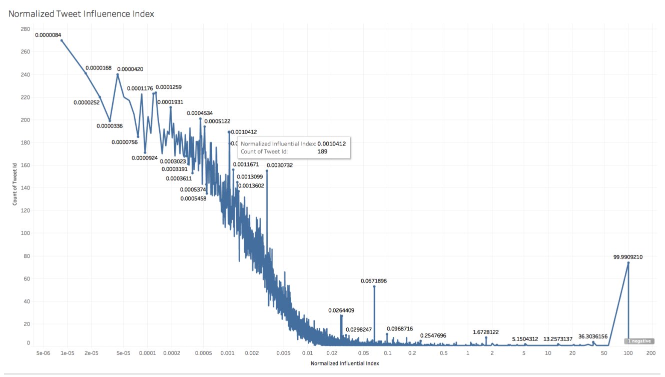
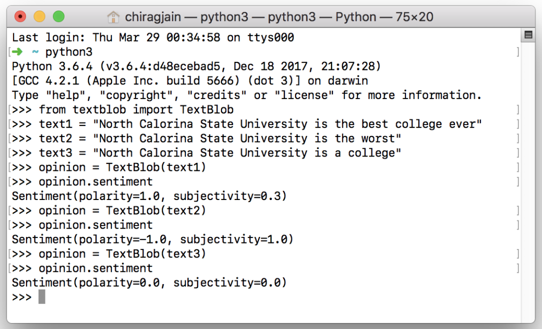


Figure 11: Tweet count vs Normalized Influential Index for Chipotle

5.6 Sentiment Analysis

Sentiment analysis or emotion AI refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to the voice of the customer materials such as reviews and survey responses, online and social media, and health-care materials for applications that range from marketing to customer service to clinical medicine [8].

For our project, we plan to perform sentiment analysis on the Tweet's text. The process provides us information about the subjectivity and polarity of the text. This is done using the existing python library TextBlob [12]. Once we have the value indicating the subjectivity of a particular tweet, we can determine whether the tweet is a general factual statement or a polar opinion or comment. The polarity value will give a sense of negative or the positive tone of the tweet. Figure 12 shows the example of possible types of opinion which can be tweets and their corresponding sentiment's polarity and subjectivity. The value of subjectivity is within the range 0 to 1. 0 means that there is no subjectivity in the opinion presented and it is purely fact-based whereas 1 indicates the contrary. Similarly, Polarity lies within the range -1 to 1. -1 indicates that the opinion presented in the tweet has negative emotion whereas 1 indicates that the opinion is positive.



```
Last login: Thu Mar 29 00:34:58 on ttys000
➜  ~ python3
Python 3.6.4 (v3.6.4:d48ecebad5, Dec 18 2017, 21:07:28)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from textblob import TextBlob
>>> text1 = "North Calorina State University is the best college ever"
>>> text2 = "North Calorina State University is the worst"
>>> text3 = "North Calorina State University is a college"
>>> opinion = TextBlob(text1)
>>> opinion.sentiment
Sentiment(polarity=1.0, subjectivity=0.3)
>>> opinion = TextBlob(text2)
>>> opinion.sentiment
Sentiment(polarity=-1.0, subjectivity=1.0)
>>> opinion = TextBlob(text3)
>>> opinion.sentiment
Sentiment(polarity=0.0, subjectivity=0.0)
>>>
```

Figure 12: Sentiment Analysis on various opinions

We choose the default implementation as understanding the linguistic semantics of tweets is difficult given the large vocabulary and use of acronyms. Moreover, the accuracy of the default sentiment analyzer from the TextBlob library was under the acceptable limits.

Contrary to the notion that the tweets are generally expressing extreme emotions, for a majority of the tweets that we extracted for 'Chipotle', the polarity was neutral. This is both good and bad for review mining because if the tweets contain only high positive and high negative polarity the contrasting opinion which compares both the good and bad things about the business is lost. Also, if there isn't enough polarity, it becomes difficult to assess whether the review is presenting the good or bad aspects of the business. Figure 13 depicts the above.

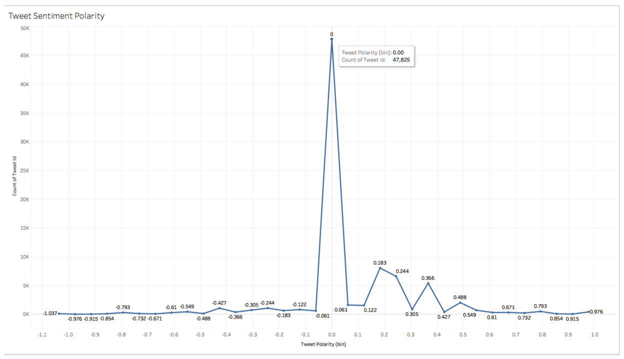


Figure 13: Tweet count vs Polarity for Chipotle

Similarly, for subjectivity, the results indicated that a large number of tweets were not subjective and were instead fact based. However, there is a uniform distribution of tweets across the subjectivity values indicating the various types of reviews that can be extracted from tweets. A tweet with low subjectivity would be very formal and descriptive of the services and features of a business that can be quantified e.g., A tweet about the menu and prices. On the other hand, highly subjective tweets would point out a personal experience that is unique yet informative. Figure 14 depicts the above.

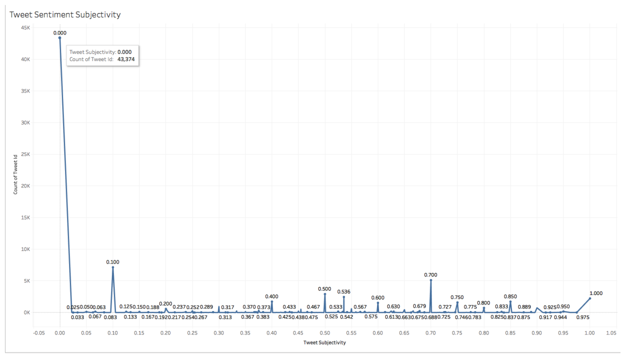


Figure 14: Tweet count vs Subjectivity for Chipotle

6. IMPROVEMENTS OVER THE PREVIOUS SYSTEM

6.1 Optimized implementation around Twitter API

Our implementation of the process to extract and store tweets for a particular business was better in terms of volume and avoiding redundant API calls. This resulted in a better utilization of the API limits. Details about this are mentioned in the section 5.2.

6.2 Leveraging Tweet Metadata to create indices

The previous implementation relied only on the text of the tweet. It performed sentiment analysis on the tweet and tried to gauge the polarity of the review. Use of single tweet feature was not the best idea as the tweet text is not always English like. It has a lot of acronyms, grammatical error, linguistic intricacies like shorthand vocabulary etc. We used the metadata along with the tweet text to find useful reviews for a particular business.

Since the metadata is very rich, a lot of statistical analysis can be done to derive quantitative measures to classify or rate each tweet as a review or not with respect to other tweets. To implement this we formulated indices which use the tweet metadata, the details about their individual implementation is mentioned in [Section 5.5].

6.3 Focused only on Twitter

We concentrated our development to develop a process to mine reviews only from Twitter. The previous implementation did not present the best reviews in form of tweets for multiple reasons mentioned above. Thus, requiring an alternate platform/service that can reinforce the reviews, they used Google Places along with Twitter. However, focusing only on Twitter helped us explore methods to overcome the specific problems that the previous implementation had with Twitter. It also helped us leverage twitter metadata, as explained in [Section 6.2]

6.4 Visual analysis

We made heavy usage of Tableau[3] to understand how the metadata can be used to extract meaningful reviews from tweets. It also helped us distribution of the values of the normalized indices.

7. EVALUATION

7.1 Visualization and Statistical Analysis

After calculating values of indices for each tweet, we used Tableau to visualize every index and analyzed them to find threshold values for each index where the tweets were more review like. We used Tableau by connecting it to the Cassandra database using Spark to get the data and did analysis on every index.

7.2 Indices

After calculating and visualizing all our indices, we performed manual experiments to evaluate each of them. We did this, by dividing each index into 5 equally spaced buckets based on its value. Then, we checked random tweets retrieved for every bucket and assigned a score to it. Based on the average score, we determined good buckets that we then identify as sweet spots. Thus, these sweet spots are defined for each index. They can be configured as threshold values in order to filter out tweets. Some more observations we had while evaluating our indices are as follows:

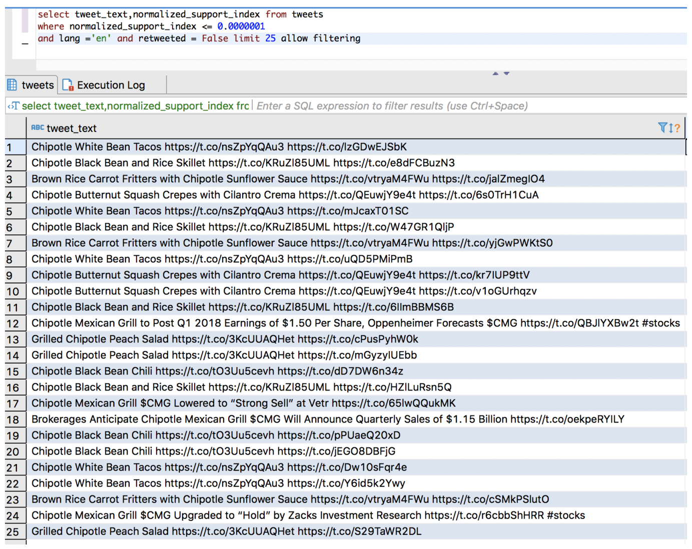
- After analyzing around 83000 tweets for Chipotle, we found that most of the tweets are less polar. Tweets with high polarity seem to provide more specific information about products and services. To get more review-like tweets, the polarity of the tweets required is high, thus a large number of tweets need to be mined for the same.

- Tweets with highest values of normalized influential index seems to be promotional tweets by official Twitter channels.

- Tweets with lower values of normalized support index are tweets posted by food critics, review platforms or businesses. They tweet about products and a corresponding link for exhaustive review as shown in Figure 15.

7.3 User Evaluation

Once we identified the tweets that were review-like and that were not review-like using different indices, we sent a small sample of those tweets for 'Chipotle' to the users to validate our results without mentioning it to the users that the tweet is review like or not to get an unbiased opinion from the users. We just asked them to rate a tweet based on whether it provides good info about the products and services of Chipotle. And we found that the tweets that were review like as per our system were rated high by the users and vice-versa. We sent a total of 20 tweets to the user for evaluation. Some of the responses that we received are in the figures 16-19

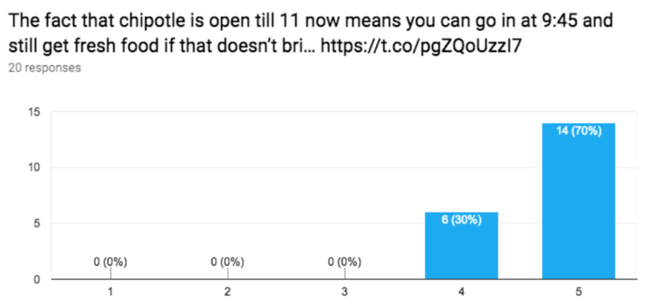


```
select tweet_text,normalized_support_index from tweets
where normalized_support_index <= 0.0000001
and lang ='en' and retweeted = False limit 25 allow filtering
```

tweets
Execution Log
select tweet_text,normalized_support_index frc
Enter a SQL expression to filter results (use Ctrl+Space)

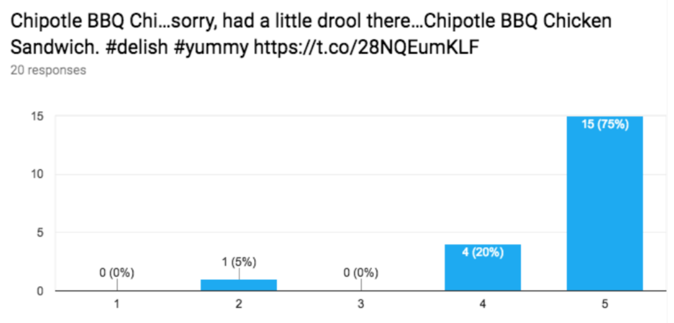
	tweet_text
1	Chipotle White Bean Tacos https://t.co/nsZpYqQAu3 https://t.co/lzGDwEJSbK
2	Chipotle Black Bean and Rice Skillet https://t.co/KRuZl85UML https://t.co/e8dFCBuzN3
3	Brown Rice Carrot Fritters with Chipotle Sunflower Sauce https://t.co/vtryaM4FWu https://t.co/jalZmeglO4
4	Chipotle Butternut Squash Crepes with Cilantro Crema https://t.co/QEuwjY9e4t https://t.co/6s0TrH1CuA
5	Chipotle White Bean Tacos https://t.co/nsZpYqQAu3 https://t.co/mJcaxT01SC
6	Chipotle Black Bean and Rice Skillet https://t.co/KRuZl85UML https://t.co/W47GR1QljP
7	Brown Rice Carrot Fritters with Chipotle Sunflower Sauce https://t.co/vtryaM4FWu https://t.co/yjGwPWKtS0
8	Chipotle White Bean Tacos https://t.co/nsZpYqQAu3 https://t.co/uQD5PMiPmB
9	Chipotle Butternut Squash Crepes with Cilantro Crema https://t.co/QEuwjY9e4t https://t.co/kr7IUP9ttV
10	Chipotle Butternut Squash Crepes with Cilantro Crema https://t.co/QEuwjY9e4t https://t.co/v1oGUrhqzv
11	Chipotle Black Bean and Rice Skillet https://t.co/KRuZl85UML https://t.co/6llmBBMS6B
12	Chipotle Mexican Grill to Post Q1 2018 Earnings of $1.50 Per Share, Oppenheimer Forecasts $CMG https://t.co/QBJlYXBw2t #stocks
13	Grilled Chipotle Peach Salad https://t.co/3KcUUAQHet https://t.co/cPusPyhW0k
14	Grilled Chipotle Peach Salad https://t.co/3KcUUAQHet https://t.co/mGyzylUEbb
15	Chipotle Black Bean Chili https://t.co/tO3Uu5cevh https://t.co/dD7DW6n34z
16	Chipotle Black Bean and Rice Skillet https://t.co/KRuZl85UML https://t.co/HZlLuRsn5Q
17	Chipotle Mexican Grill $CMG Lowered to "Strong Sell" at Vetr https://t.co/65lwQQukMK
18	Brokerages Anticipate Chipotle Mexican Grill $CMG Will Announce Quarterly Sales of $1.15 Billion https://t.co/oekpeRYlLY
19	Chipotle Black Bean Chili https://t.co/tO3Uu5cevh https://t.co/pPUaeQ20xD
20	Chipotle Black Bean Chili https://t.co/tO3Uu5cevh https://t.co/jEGO8DBFjG
21	Chipotle White Bean Tacos https://t.co/nsZpYqQAu3 https://t.co/Dw10sFqr4e
22	Chipotle White Bean Tacos https://t.co/nsZpYqQAu3 https://t.co/Y6id5k2Ywy
23	Brown Rice Carrot Fritters with Chipotle Sunflower Sauce https://t.co/vtryaM4FWu https://t.co/cSMkPSlutO
24	Chipotle Mexican Grill $CMG Upgraded to "Hold" by Zacks Investment Research https://t.co/r6cbbShHRR #stocks
25	Grilled Chipotle Peach Salad https://t.co/3KcUUAQHet https://t.co/S29TaWR2DL

Figure 15: Result of Low Support Index


The fact that chipotle is open till 11 now means you can go in at 9:45 and still get fresh food if that doesn't bri… https://t.co/pgZQoUzzI7

20 responses

Figure 16: User Evaluation


Chipotle BBQ Chi…sorry, had a little drool there…Chipotle BBQ Chicken Sandwich. #delish #yummy https://t.co/28NQEumKLF

20 responses

Figure 17: User Evaluation

8. FUTURE SCOPE

8.1 Normalization with respect to time

Currently, as per [Section 5.5] the indices values are built by normalizing them with respect to the followers count, favourites count, re-tweets count, etc but all these counts are taken from the day the tweet was tweeted this can lead to biased result for the indices because the tweets can be

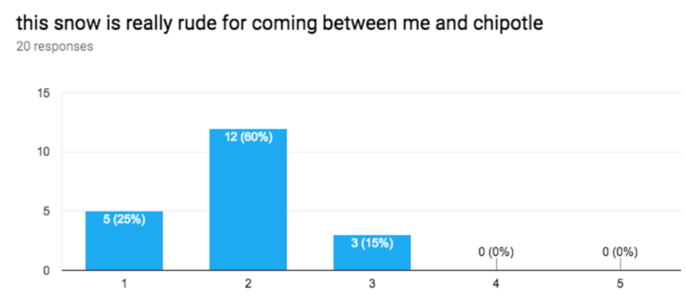


Figure 18: User Evaluation

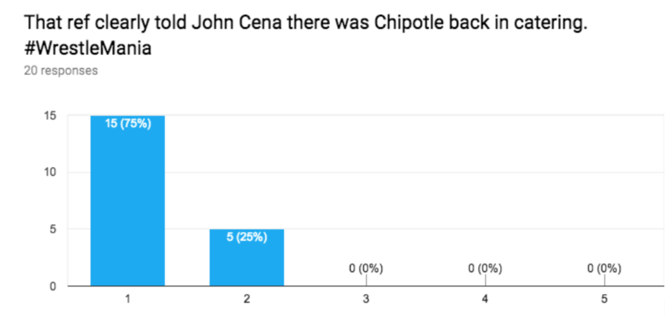


Figure 19: User Evaluation

from the Day 1 whereas the other tweets can be from Day 7, so ensure proper results we can normalize the results by the number of days before the current day when the tweets were tweeted by the user. This will ensure that all indices values will be from tweets which have a similar time period for the users to read.

8.2 Composite Index

As of now, the indices produces results individually and classify the tweet as review-like or not. But to improve the efficiency of the system to classify them, we can generate the results by combining these indices and create one new composite index which will be a single formula where the trends of all the other indices will be taken into consideration. Such index will have the sweet spots of all the indices used. This can lead to better results and better classification.

8.3 Custom Sentiment Analysis

Alternatively, instead of using the standard sentiment analysis module provided by TextBlob we can also train our own learning model to provide better control over the accuracy of the sentiment measure. This is because the TextBlob sentiment analyzer is trained on a dataset containing movie reviews. These are similar to reviews for a particular location but have a subtle difference in the range and variety of words that are used. A custom sentiment analyzer can be built and trained using the python NLTK library[7]. There is a great guide[21] about implementing the same.

8.4 Abbreviation and Acronym Expansion

Abbreviation and Acronym Expansion is the process of determining the correct expansion of an abbreviation / acronym in a given context.

One approach is to use a general library that maps abbreviations and acronyms to their expanded full-forms. However, the challenge with using a general library is acronyms are almost always domain dependent. That is why it is not a good idea to have a "general" library. NLP, for example, could mean 'natural language processing' or 'neuro-linguistic programming', depending on the domain. Another approach is if we have a collection of textual documents where both the acronym and its expansion occur you can apply an algorithm to extract (acronym, extension) pairs. That will not be possible as the data in context is *tweets*

An approach that we will explore is a method for predicting the most appropriate expansion of acronyms using Wikipedia.[19] It is based on tf-idf word frequencies. Given a document containing an unknown acronym, the algorithm makes its prediction in two general phases. First, training phase where Wikipedia documents are scraped and acronyms and their expansion is discovered. Second, tweets are fed into the system containing unknown abbreviation. The input is converted to a tf-idf frequency vector. For each possible expansion (class), it computes the dot product of the class's learned coefficient vector and the tweet's frequency vector (and add the class's bias term). This results in a set of scalar values, one for each possible class. The class with the maximum value is predicted as the expansion of the current abbreviation/ acronym.[2]

8.5 Vocabulary Correction

Vocabulary Correction is the process of rectifying words like "abt" that stand for about and "ordr" which stands for "order". These words might occur in tweets because of the informal nature of the medium, character limit or mere typing error.

We would use n-gram model to find the most probable word and then use substitution. Example, we substitute the word "ordr" by the word "order".[18] An n-gram model is a type of probabilistic language model for predicting the next item in such a sequence. A probabilistic model of word sequences could suggest the desired substitution in tweets.

8.6 Using Machine Learning

The evaluation of the indices and finding the sweet spot is done by human intelligence. Currently, the sweet spots are found by a human by manually reading the text of the tweets and rating the tweet as per the knowledge of the person. Instead of using human intelligence, we can build a machine learning model which will classify the text as being a review or not.

For doing this, the model will have to be trained on the bag of words and context of the text. The model will be trained to understand the text, context of the text and whether the text has words that are present in the bag of words. Using this, the model can learn to classify the tweet and generate sweet spots for all the indices.

9. CONCLUSION

Contrary to the initial impression from the previous work, we strongly believe that Twitter is a viable platform to mine reviews for businesses in an automated fashion with certain configurations. We defined, calculated and evaluated various indices for the same purpose. Based on our study, we conclude that every index, when considered in isolation, has the potential to extract tweets which are like reviews. We figured out the "sweet spot" or the configurations as we may call it for each index and were able to mine reviews. However, these maybe different for different businesses. Also, more research is required to find a correlation between the indices.

10. ACKNOWLEDGEMENTS

Our special thanks to our Prof. Timothy Menzies and Teaching Assistant, Amritanshu Agrawal for helping and guiding us throughout the project development.

11. CHITS

HVW JPX TCE OZK ILN OVI KQE NMN RSS MBH RRU QMX LLA DCU QAU IFF NOB TBP APY NDY

12. REFERENCES

[1] Social Media Demographic . `https://sproutsocial.com/insights/new-social-media-demographics/#twitter`, 2017. [Online; accessed 28-March-2018].

[2] Acronym Expansion Disambiguation. `https://pdfs.semanticscholar.org/52d5/6c8de14b3d6640d048eb0fbd06ef95bd945d.pdf`, 2018. [Online; accessed 26-March-2018].

[3] Business Intelligence and Analytics. `https://www.tableau.com/`, 2018. [Online; accessed 26-April-2018].

[4] Cassandra | Datastax. `https://www.datastax.com/wp-content/uploads/2012/08/WP-DataStax-MySQLtoCassandra.pdf`, 2018. [Online; accessed 27-April-2018].

[5] DBeaver | Free Universal SQL Client. `https://dbeaver.jkiss.org/`, 2018. [Online; accessed 26-April-2018].

[6] How to use Twitter's Search REST API most effectively. `https://www.karambelkar.info/2015/01/how-to-use-twitters-search-rest-api-most-effectively./`, 2018. [Online; accessed 26-April-2018].

[7] Natural Language Toolkit. `https://www.nltk.org/`, 2018. [Online; accessed 25-March-2018].

[8] Sentiment Analysis. `https://en.wikipedia.org/wiki/Sentiment_analysis`, 2018. [Online; accessed 25-March-2018].

[9] Social Media Platform User's Increase Graph . `https://www.statista.com/statistics/273476/percentage-of-us-population-with-a-social-network-profile`, 2018. [Online; accessed 28-March-2018].

[10] Standard search API. `https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets.html#example-response`, 2018. [Online; accessed 25-March-2018].

[11] Standard search API | Resource Information. `https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets.html#resource-information`, 2018. [Online; accessed 25-March-2018].

[12] TextBlob: Simplified Text Processing. `http://textblob.readthedocs.io/en/dev/`, 2018. [Online; accessed 25-March-2018].

[13] Tweet Object. `https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object`, 2018. [Online; accessed 27-April-2018].

[14] Tweet User Object. `https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/user-object`, 2018. [Online; accessed 27-April-2018].

[15] Twitter Developer Platform. `https://developer.twitter.com/`, 2018. [Online; accessed 25-March-2018].

[16] Twitter Status | Elon Musk. `https://twitter.com/elonmusk/status/647209943515332608?lang=en`, 2018. [Online; accessed 26-March-2018].

[17] Types of Tweets. `https://help.twitter.com/en/using-twitter/types-of-tweets`, 2018. [Online; accessed 26-March-2018].

[18] Vocabulary Correction. `https://lagunita.stanford.edu/c4x/Engineering/CS-224N/asset/slp4.pdf`, 2018. [Online; accessed 26-March-2018].

[19] Wikipedia Abbreviations Expansion. `https://en.wikipedia.org/wiki/Wikipedia:Abbreviation_expansion`, 2018. [Online; accessed 26-March-2018].

[20] Yelp Dataset. `https://www.yelp.com/dataset`, 2018. [Online; accessed 27-March-2018].

[21] L. Luce. Twitter sentiment analysis using Python and NLTK. `http://www.laurentluce.com/posts/twitter-sentiment-analysis-using-python-and-nltk/`, 2012. [Online; accessed 25-March-2018].

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.