Analysis of March 2016 Twitter Data

Anitha Subramania
Computer Science
Georgia State University,
Atlanta
asubramanian5@student.gsu.edu

Henry Woodyard
Economics
Georgia State University,
Atlanta
hwoodyard1@student.gsu.edu

Nikhil Gupta
Data Science
Georgia State University,
Atlanta
ngupta9@student.gsu.edu

Sonam Dawani
Data Science
Georgia State University,
Atlanta
sdawani1@student.gsu.edu

Susanth Dasari
Computer Science
Georgia State University,
Atlanta
sdasari3@student.gsu.edu

***Abstract*—In this paper, we perform five analyses on a sample of Twitter data from March 2016. We conduct a visual analysis, test the *six degrees of separation* hypothesis, implement a topic model, cluster users based on description, and classify misspelled words.**

Keywords— big data, Twitter, visualization, six degrees of separation, topic modeling, analysis, data mining, clustering

I. Introduction

Twitter is one of the most popular social media platforms in the world and is used around the globe. Such a vast base of users and activity by each user generates huge amounts of data which can yield valuable insights if the correct tools are applied. In this paper, we perform five analyses on a sample of Twitter Data from March 2016. First, we visualize the distributions of our variables, the evolution of Twitter activity over time, and the frequency with which hashtags were used. Next, we test a well-known hypothesis, *six degrees of separation*, which states that every individual is only six social linkages apart. Afterward, we attempt to model the topics of tweets using the Latent Dirichlet Allocation model. Then, we cluster users based on their self-reported description and perform analytics on the clusters. Finally, we examine the most misspelled words on Twitter.

II. Data Preparation

For this project, we use one month of Twitter data from March of 2016. The raw format of the data is a json file with each tweet being a single json object. The raw data is in a nested format natively, which poses some difficulties. With that being said, the data is rich in features, and includes, for instance, the text of the tweet itself, number of likes and favorites, time of posting, and user characteristics.

In addition to the nested data structure, the size of the data poses another problem – compressed, the raw data is 36.24gb. To deal with this issue, we use Microsoft Azure HDInsight, which allows creation of temporary Spark clusters. Distributed computing through Spark allows us to work with large and messy data relatively quickly. Furthermore, most analyses throughout the rest of the paper are performed using Spark with PySpark.

To preprocess the data, we flatten each nested json object and retrieve only the features that we may use. In addition, we filter to only English tweets. Ultimately, the final preprocessed data is still in json format rather than csv, as Spark can use json files without issue. The table below summarizes the preprocessing.[1]

	Raw	Processed
Number of Tweets	107,430,615	27,786,026
Number of Features	329	96
Languages	All	English
Size on Disk (gb)	36.24	6.68

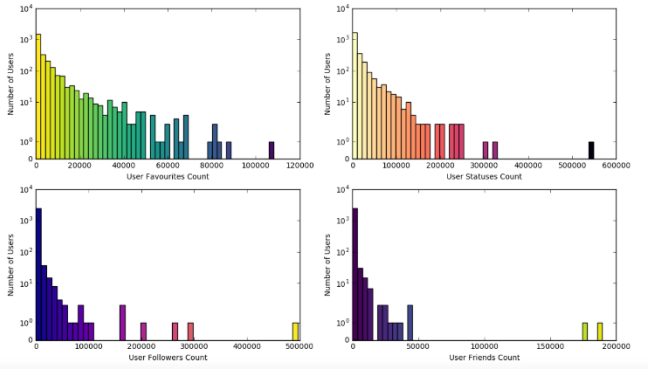
III. Visual Analysis of Tweets

Because we tend to be able to process visual information better than raw data, the first step in our analysis of Twitter data is to construct visual representations. By using visual elements like graphs, charts and images, we can see and understand the trends, patterns and outliers in data. In addition, this allows us to quickly understand complex patterns in our Twitter data and, more generally, in big data.

A. Distribution of features using histogram

There are several key features on Twitter that are used to measure tweet impressions. Such features are number of friends, number of followers, number of favorites, and the count of statuses by a user. We can see the distribution of these features using histograms.

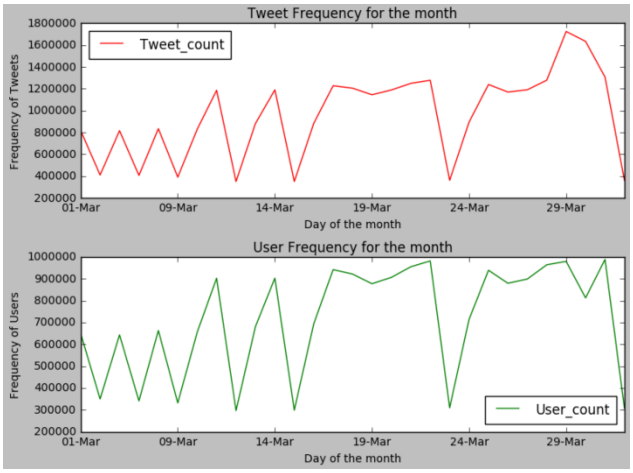
[1] Please note that each analysis also does further cleaning.

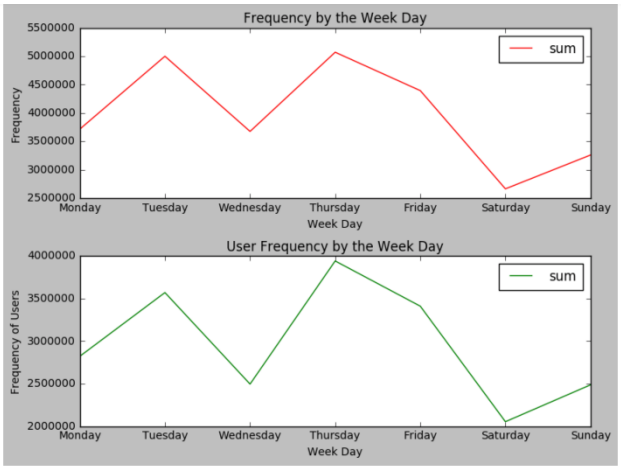


Note: from the above chart we can easily see that approximately 2000 Twitter users have around 2000 favorites count, 10000 statuses count, 10000 followers count and 3500 friends count. The outliers in the histogram represent a very small subset of popular Twitter users.

B. *Tweets and user frequency using time series*

To understand how the number of tweets and users tweeting change over time, we can use simple time series graphs.





From the above graphs, we can see that March 29th had the highest number of tweets in the month (around 1800000 tweets that day). Similarly, March 22nd and March 29th had the most users those days (around 1000000 users).

One might assume that there will be more tweets and users on weekends compared to weekdays. However, according to our data from March of 2016, it is evident that Tuesdays and Thursdays are the most popular days to use Twitter.

C. *Word Cloud Hashtag*

Word clouds – also known as text clouds or tag clouds – are generally used for visually presenting text data. This type of visualization is popular for text analysis.

The first step in constructing a word cloud is to extract the hashtags used in every tweet using regex and store it in a text file. Using this text file as input, the word cloud generator creates a word cloud with the hashtags that were most used. This functionality was achieved by using generate from frequencies method of word cloud package.

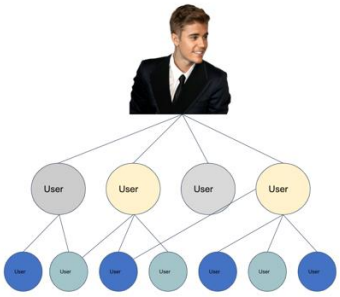
This word cloud depicts the most frequently used hashtags in March 2016 Twitter data. Larger text represents more popular hashtags. From this we can see that #iHeartAwards and #BestFanArmy are the most popular Twitter hashtags in March 2016.

IV. Six Degrees of Separation

The idea of *six degrees of separation* is that everyone in this world is only six social connections away from each other. Historically, this theory has been tested more than once in different scenarios with some tests even being successful. Recently the most popular environment to test this theory has been social media.

D. *Network Analysis*

Twitter is one of the biggest social media platforms with over 330 million real-time users as of 2019. In such a platform it is evident that there will be more than one relationship of interest between users. The users and the relationships between them make a network. By harnessing this network data properly, we can test the six degrees of separation phenomenon. Graph theory provides us with the tools to analyze these relationships efficiently and effectively.



For this analysis, we are only considering the relationship of retweet between the users.

E. Retweets

A retweet is a feature on Twitter through which, when one user posts a tweet to his network, another user can share the tweet to his own network by giving the original user credit.

This feature can be accessed in two ways. First, "quoting" is when the user shares the tweet after adding some of his own words to the original, by still giving credit to the user who originally posted it. Alternatively, "retweeting" is when the user shares the tweet without making any modifications by giving credit to the user who originally posted it.

F. Graph Mining with Apache Spark

Given the large number of users and the relationships between them, sketching the network as a graph gives us the advantage of utilizing graph algorithms such as Page ranking or shortest path which perform better than the algorithms for linear data structures.

Because Apache Spark doesn't have a native implementation to work with graph data, we use GraphFrames. The GraphFrames API provides us with the set of tools that we require to analyze data in a graph structure with methods for additional operations. The advantage pf GraphFrames is that it works with the data frames of the PySpark-SQL instance. This makes it accessible to use, as it is easy to switch between data available to us in storage and in the GraphFrames graph structure.

G. Preprocessing Twitter Data

1) Extracting user information

The data we have from Twitter for March 2016 has the user data embedded into the tweet, with each tweet being a different row of data. So if the same user has posted a tweet twice, once at the beginning of the month and once at the end of the month, the embedded user information inside the tweet might have conflicting information, since some of the values like follower count can change over the course of the time.

If a tweet is a retweeted/quoted tweet, the same row contains more than one user's information. The user who posted the tweet originally and the user who shared the tweet. To get the right information, we extract the user information from the sections of root tweet, retweeted_status and quoted_status and carry forward the date the tweet is posted. By sorting on the date, we set each user's information to the latest information available in our data for that user.

2) Extracting user relations

The relationship in this case is which user retweeted which user's tweet. The tweet itself is not of concern for this analysis, but the users associated with it are.

3) Finding the optimal user

We select the top ten most popular users based on number of followers. Then, for these top 10 users, we calculate the *degree centrality*. Degree centrality is the number of connections a node has in a network. We consider the user with the highest degree centrality as the origin of our test. Henceforth he will be referred to as the *optimal user*.

H. Testing 6 degrees of separation

After selecting the optimal user, we find the users who are directly connected to him in the first iteration. This new list of users becomes the first-degree connections to our optimal user. From the next iteration, we combine the optimal user and the list of his first-degree connections and find all the users who are connected to them. The user list from this iteration becomes the second-degree of connections to the optimal user.

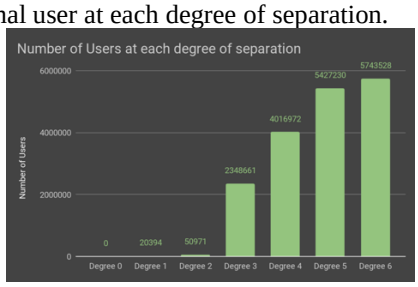
The users from every subsequent iteration become the subsequent degree of connections to the optimal user. After six iterations, if the resultant list of users comprises all of the users in the network, we can say that the six degrees of separation hypothesis holds true for the Twitter network.

For this purpose, we chose SQL joins instead of the native pattern-based filtering offered by GraphFrames called *motifs*. Motifs work by using a cross product of the users and the relations, while by using SQL left outer joins we can achieve the same results. By doing this, we increase the performance and decrease the memory usage of the test significantly.

I. Results

Once the SQL joins are performed 6 times, the final list of users is all the users that are connected to our optimal user within six degrees of social connections. The total number of users within the retweet network of our data is 6,871,346.

The figure below depicts how many users are connected to the optimal user at each degree of separation.



We can see that between degree 2 and degree 3 there is a huge jump in the number of users. Finally, at degree 6, the total number of users at 6 degrees of separation from the optimal user is 5,743,528. Therefore, the number of users 6 degrees away from the optimal user is 83% of the total users in our network**.**

J. Conclusion

Since we are only able to comprise 83% of the total users, there are still 17% of users who are not connected within 6 degrees of connections. Given that the March data for Twitter is only a sample stream of the entire data, there

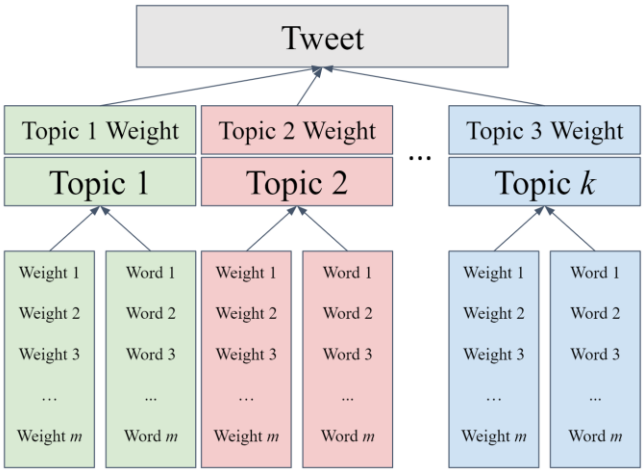
might be a few crucial connections missing. Because of this, we cannot assume that the six degrees of separation hypothesis is false. Instead, we can only accept that it doesn't hold true in our case.

V. Topic Modeling

Given the volume of tweets at hand, we want to have some measurement of the type of content of the tweets, and thus we seek to implement topic modeling. This will give us insight about the topics discussed in our tweets and could allow for future expansions which are outside the scope of this current project.

When deciding on the method to be used for topic modeling, the first - naive - approach considered was to base the topic on the hashtags. Using this approach, we could have used a supervised classification model to predict whether a new tweet belongs to a given topic or hashtag. However, this approach would have been, once again, naive. Instead of constraining our model in such a way, we opted instead to use an unsupervised method for topic modeling.

The algorithm used for topic modeling in this project is Latent Dirichlet Allocation (LDA). This model assumes that, given a body of *n* tweets, there are *k* underlying topics. Further, it models the topic of a given tweet as a weighted combination of the k underlying topics. The clear advantage of this unsupervised approach - compared to the naive approach mentioned previously - is the fact that it allows a tweet to belong to multiple subtopics. Each topic is modeled as a weighted list of words. For clarity, the figure below gives a graphical depiction of the LDA framework.



In preparing to implement an LDA model, we first filter to only original tweets by users, removing all retweets. This reduces the data from approximately 27 million tweets to nearly 15 million. Next, stopwords and short words are removed from each tweet, and then a term frequency - inverse document frequency (TF-IDF) is calculated for each tweet. In short, this quantifies the text data in each tweet in such a way that we have a metric of how frequently words are used without overweighting ubiquitous words.

Finally, we fit the LDA model on our prepared tweets using the LDA routine from PySpark's machine learning library, specifying the number of underlying topics as one hundred. The table below gives an example of five of the topics generated, showing the three most prominent words per topic. In addition, the next table after shows two tweets and the three words from that tweet's most prominent topic.

Topic Number	Word 1	Word 2	Word 3
1	visit	calls	talks
2	trump	without	donald
3	latest	sweet	marketing
4	photo	facebook	album
5	national	taken	political

Topic	Most Prominent Topic Word 1	Word 2	Word 3
"Come check out our full taster section for tank users & drippers alike!"	Eating	Hands	Article
"@willlutz @MiskovichJ I'm having Mitt write a letter to Congress now. Action must be taken!!"	Chance	Release	Community

As mentioned previously, the model predicts the topic of a tweet as a weighted combination of all 100 topics. Using Latent Dirichlet Allocation to model the topic of tweets showed us some of the difficulties inherent in both topic modeling and unsupervised learning. For instance, while the output generated by an unsupervised model is likely more robust than that of a supervised model, it is less comprehensible. Depending on the context in which topic modeling must be performed - for example, in business - it could be advantageous to also implement a supervised approach which is more easily understood and interpreted.

VI. User Clustering

We can't imagine any application without users - they are the driving force behind any consumer application. Because users are one of the important dimensions for Twitter as well, we analyze what kind of users are present in the data by clustering similar users together. We measure similarity based on the user's description, also known as a

user's bio. Further, we evaluate which kind of cluster is most followed.

A. Data selection

The first step in this analysis is to select relevant features. From the common preprocessed file[2], we filter the features to only those with user characteristics: user's ID, screen name, description, follower count, language, creation date, and whether the user has set a profile photo.

B. Preprocessing

As we are clustering users based on their description which is free text, it is obvious that it would have many elements which we won't require and have to remove. Hence, we remove stop words, mentions, hashtags, non-Latin characters, and words with less than 3 characters.

C. Data filtering

Furthermore, we filter to only users who have a user description, profile photo, and language as English.

As our data consists of tweets, every instance contains the tweet and its details, including the user information. We are extracting the user information from the tweet's details. In the case that a user has multiple tweets, we ultimately record a user's information multiple times. To prevent redundancy, we extract only the latest details for each user.

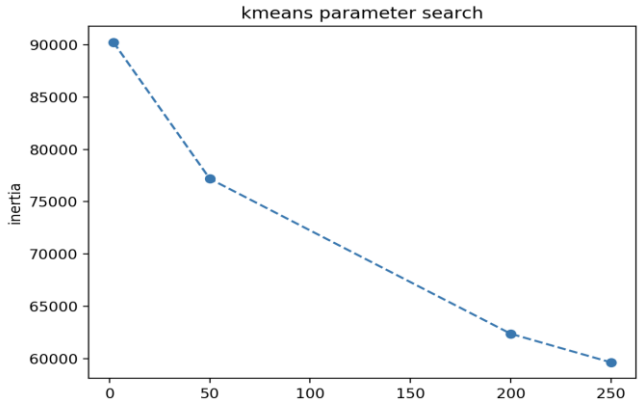
Moving forward, we cluster the top 100,000 users based on follower count.

D. Vectorization

As we are clustering based on user's description which is text data, we need to perform vectorization. Here we are performing term frequency – inverse document frequency (TF-IDF) vectorization. As explained earlier, a TF-IDF vectorizer creates a vector corresponding to each document (each user description in our case). Each vector consists of weight of the words in dataset. As discussed previously, using TF-IDF ensures that common words don't receive disproportionate weights.

E. Clustering

For clustering, we use a simple but effective unsupervised clustering algorithm, k-means.[3] Based on the inertia graph below, we decide to use 50 clusters.



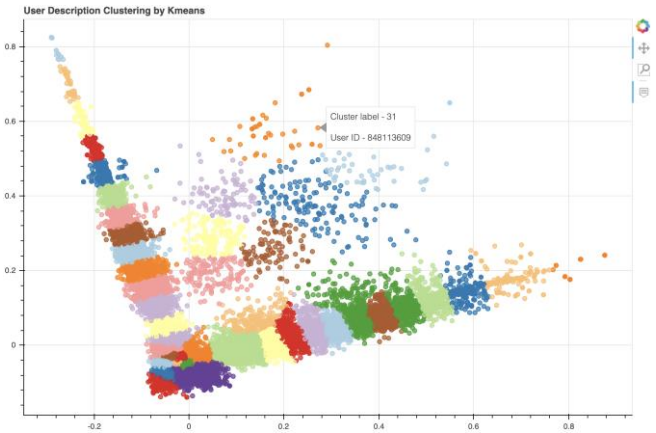
To check our results, we look at a few frequent words of some clusters.

```
Most frequent words for few clusters:

Cluster 11 -> music producer com new news entertainment lover
Cluster 12 -> hit music station follow com new show
Cluster 13 -> love follow people live things always much
Cluster 14 -> god love family first life people jesus
Cluster 15 -> artist recording com producer songwriter music booking
Cluster 16 -> contact com bookings booking business info inquiries
```

F. Visualization

Because we cannot visualize high-dimensional data, we reduce the number of dimensions to two using Principal Component Analysis (PCA). The following result is plotted using BokehJS.



G. Follower count analysis based on cluster

After creating clusters based on user descriptions, we further analyze clusters on the basis of average follower count. Because we have data on the number of followers per user, we calculated the average follower count for each cluster. The table below depicts the average follower count for the top five clusters.

CLUSTERS WITH HIGHEST AVERAGE FOLLOWER COUNT:

	cluster_label	users_count	Average_follower_count
44	44	596	293334
5	5	1748	262653
24	24	2391	217320
31	31	2193	216107
39	39	3315	202328

[2]I.e., the file which is saved in our storage in .json.bz2 format

[3]The clustering method used partly follows a routine from the following article:

https://twitterdev.github.io/do_more_with_twitter_data/clustering-users.html#clustering-users

Further, in order to check what words were most frequently used in each of the most followed clusters, we use the 'top_words_in_cluster' method. The results are depicted below.

```
TOP WORDS IN THE CLUSTERS ABOVE:

Cluster 44 -> actress model singer host com producer instagram song
writer writer lover wife bookings film author mother
Cluster 5 -> now available new album itunes single com get download
booking link free book music order
Cluster 24 -> account official twitter follow news university welco
me team football club world updates service tweets city
Cluster 31 -> twitter official feed home welcome follow tweets news
channel team university new updates club site
Cluster 39 -> news breaking source updates sports weather follow in
formation entertainment analysis local reviews team events tips
```

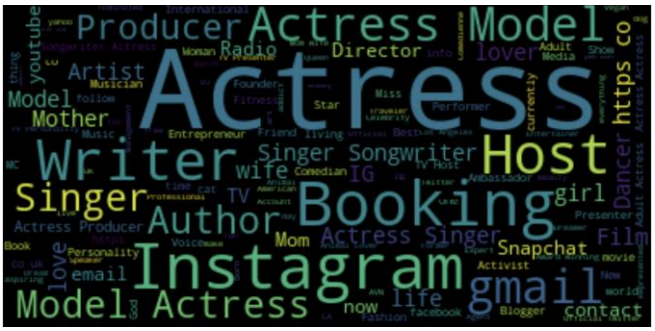
As we saw previously, the most followed cluster was Cluster 44. For this cluster we were interested to see the users' descriptions, which was the basis of clustering. A sample of the user descriptions for Cluster 44 are shown below.

```
SOME OF THE USER DESCRIPTIONS OF THE TOP MOST CLUSTER:

User's Descriptions for Cluster 44
```

	user_screen_name	user_description_org
2	ladygaga	jazz art pop punk actress fashion magazine edi...
76	annecurtissmith	Anne Curtis. Fil/Aussie. Actress/Host & Pursue...
89	EvaLongoria	Actress, Producer, Director, Activist, Philant...
192	juliaperrez	☆ Actress ☆ Singer ☆ Entertainer ☆ FDJ ☆ ...
211	JordinSparks	Singer/Songwriter/Actress/Host/NFL Fanatic. #R...
...	...	...
99414	Myslady	Actress, Forensic Psych, Writer, expert body l...
99438	VeronicaLavery	European ▪Model☆Actress▪Film Student★ instagra...
99828	GigiCappetta	@YEAAWARDS 2016 Best Supporting Actress Nom 20...
99865	caitlinsvoice	actress / voice actress, french bulldog mommy ...
99941	AnnaDeavereS	Actress,Playwright

Finally, to summarize the result, we create a word cloud of the words used in the user descriptions for the most followed cluster.



VII. Misspelled Words Analysis

Next, we want to map English words to their incorrectly spelled variants in the Twitter data. We want to see how people misspell words and how many variations they generally use to misspell a word. Our primary concern is the amount of variation in misspellings, not the frequency of them.

A. Data Cleaning[4]

Because of the size of the data, we utilize Spark data frames for this stage. In order to perform an analysis on the words, we discard everything else in a tweet – for instance, we remove mentions, hash tag, emoticons and special characters. We also remove stop words using a dedicated library. Finally, we group unique words (misspelled or correctly spelled) and their counts.

B. Data Processing

One of the most challenging parts of this analysis is to determine which word is correctly spelled. For example, "NYC" is not a word found in a dictionary, but it still cannot be considered a misspelling. Also, parts of speech needed attention as well. We utilize an industrial Natural Language Processing tool called SpaCy. For this project, we use the 'medium' size English language model developed by SpaCy. If a word cannot be represented as a vector in SpaCy, we consider it a misspelling. Finally, we discard all correctly spelled words. For this analysis, we consider only words which are between 3 and 15 letters in length. In total, 1,223,899 words are sent to SpaCy. [5]

```
tokens = nlp("GSU Whatsapp Shakira Wazzup")
for token in tokens:
    print(token.has_vector)
```

```
True
True
True
True
```

C. Correction

In order to correct a misspelled word, we use the 'autocorrect' package for Python. It corrects a misspelled word by comparing various similarity measures. We pass all the misspelled words to this tool and generate their correct spellings. In total, we send 1,031,894 words to autocorrect.

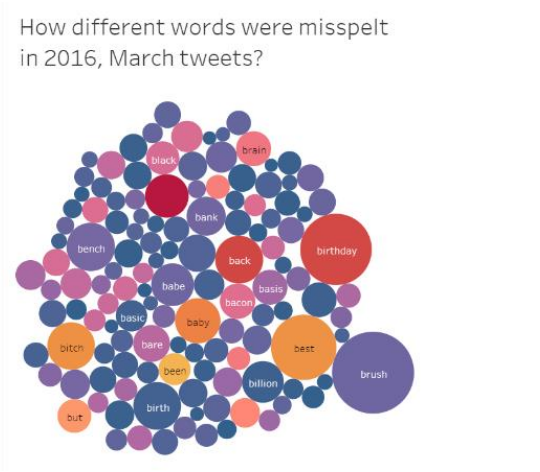
D. Mapping and Visualization

Having every misspelled word and its correct spelling, we group the data frame by incorrect word and map it with all of its misspelled variations and their respective counts in the Twitter data. We export this data to a .csv file to create a Tableau dashboard. This dashboard is a packed bubble chart which, upon hovering over a bubble, shows the variations of a word. This is depicted below.[6]

[4] Note: We already pre-processed the whole Twitter March 2016 data using Spark to include the column i.e. 'Tweet' for this analysis.

[5] Note: We use Swifter, a Pandas parallelization tool, to speed up the processing.

[6] The Tableau dashboard can be found here: https://shorturl.at/hEGPY



Corrected: birthday
Word: ['birtthday', 'birthdag', 'birthhdaay', 'berdthday', 'brihday', 'birthfdfay', 'behrthday', 'borthday', 'birthsary', 'birzhday', 'birrhday', 'birthafy', 'birthdsy', 'bitrthday', 'birthsday', 'birthday', 'birthhhda 'burthdae', 'birthdau', 'burthdah', 'birthdab', 'bhirtday', 'birthbae', 'birtjday', 'birthdaaay', 'birftday', 'brthday', 'birthdaayy', 'birthkay', 'birthdy', 'bithhday', 'birthdaya', 'birthdayyy', 'birhthday', 'bortday 'brothday', 'birtyhday', 'birthsat', 'bkirthday', 'birhday', 'birthdas', 'birhtfay', 'berthdey', 'birthfay', 'borthaday', 'birthdail', 'bearthday', 'berthday', 'birthder', 'bitrhday', 'bathday', 'borthdsy', 'birthhdayy', 'birbday', 'birthdayy', 'brithady', 'birthdya', 'birthdil', 'biethday', 'birthdqy', 'biirthdayy', 'burthday', 'birthdfay', 'birthdah', 'btrtoday', 'birthday', 'birthhay', 'biryhday', 'birffday', 'birtdaay', 'barthday', 'birrthdaay', 'birthdayu', 'birtgday', 'birthda', 'bierthday', 'birthhtday', 'bjrthday', 'bihtda 'birtbday', 'burthdayy', 'bilirthday', 'brirthday', 'borthda', 'birthhday', 'birthdae', 'birthray', 'byrthday 'birthtae', 'bitchday', 'biiirthday', 'birthdai', 'birthay', 'bithiday', 'burthdey', 'birtyday', 'brthdsy', 'birhhday', 'birthdey', 'burithday', 'birthsay', 'birthaay', 'birtfay', 'birthayy', 'bitthday', 'birthady', 'birsthday', 'birthsIday', 'bernthday', 'bvbirthday', 'birtrhday', 'barithday', 'birtdhayy', 'baerthday', 'buthday', 'birthdei', 'birthdayf', 'burtday', 'birthyday', 'birthgay', 'birthdeh', 'birsday', 'birthaday', 'birthdaymy', 'birthddaay', 'birthiday', 'birtdhay', 'birthmas', 'birghday', 'birthdee', 'britthday', 'birthyay', 'birthdat', 'birrrthday', 'birthdaay', 'brithay', 'birthdday', 'birthfday', 'birzday', 'bbirthday', 'bairthday', 'birtsday', 'birthdahy', 'birthdhay', 'birthbay', 'birthdyy']
Error Count: 1,389
Variations: 146

E. Results

Given the March 2016 Twitter data, we find that ‘beautiful’ is the word which is misspelled in the most ways, with 167 variants. Furthermore, ‘birthday’ was misspelled in 146 different ways. These misspellings are comprised of words which are popular on social media platforms like ‘birthdayyyy’ and ‘birthyay’. From a sample data of words which starts with letter ‘b’, the words with most variations are: beautiful, back, birthday and baby with 167, 147, 146 and 117 variations respectively.

VIII. Conclusion

In this paper, we performed five analyses on a month of Twitter data from March of 2016. We began by visualizing relevant features of the data. Then, we moved on to testing the *six degrees of separation hypothesis*, where we found that 83% of users are six degrees apart. Next, we modeled tweet topics using a Latent Dirichlet Allocation model. Afterward, we clustered users based on their description, and found that the most popular cluster is composed of actresses, models, and the like. Finally, we examined the most misspelled words on Twitter, and found that the word ‘beautiful’ is misspelled in the most ways. By undertaking this project, we gained experience in using the tools and techniques required for handling big data. In the future, we will be better equipped to tackle problems that conventional methods of data analysis cannot handle.