Análisis de sentimiento en eventos con contrincantes

Ashley J Naveso Cranford

Director: Rafael Caballero Roldán

Resumen

En este trabajo consideramos el problema de la caracterización de mensajes y usuarios en las redes sociales, en este caso el servicio de microblogging Twitter. Para ello hemos descargado 13,3 millones de tweets utlizando como contexto las elecciones generales estadounidenses que tuvieron lugar en el mes de noviembre de 2016. A partir de los tweets descargados construimos un conjunto coherente mediante un proceso de limpieza de datos. Una vez construido, analizamos el conjunto utilizando la técnica del análisis de sentimiento para asignar de forma automática una etiqueta descriptiva a cada tweet que indica si el autor del tweet apoya a alguno de los candidatos o, por lo contrario, se opone activamente a alguno de los candidatos.

Una vez compilados los datos obtenemos una serie de resultados, especialmente donde analizamos el comportamiento de los usuarios que apoyan a un candidato con respecto al oponente. Encontramos que los seguidores del candidato republicano Donald Trump fueron más activos en Twitter y más beligerantes contra la candidata Hillary Clinton que viceversa.

Finalmente, comparamos nuestro conjunto de datos con un estudio similar en Twitter.

Palabras clave: análisis de sentimiento, Naive Bayes, Twitter, elecciones, contrincantes, clasificadores.

Abstract

This paper considers the problem of categorizing text and users in the context of social networking, in this case the microblogging service Twitter. To to this we downloaded over 13 million tweets in the days leading up to the 2016 U.S presidential elections.

Using the tweets we downloaded we used a process known as data cleansing to build a coherent dataset. Once the dataset was built, we used sentiment analysis to automatically assign a label indicating whether the tweet's author supports one of the candidates, or on the contrary, actively opposes them.

Once every tweet was labeled we compiled a series of results, the most interesting of which being analyzing the behavior of users who support a candidate while opposing the other. We found that supporters of the republican candidate Donald Trump were more active on Twitter and more belligerent against the democratic candidate Hillary Clinton than vice versa.

Finally, we compare our dataset to the dataset of a similar study.

Keywords: sentiment analysis, sentiment classification, Naive Bayes classifiers, Twitter, elections, candidates.

Table of contents

1 Introducción 7

2 Introduction 9

3 Data Capture 11
- 3.1 Data Collection 11
- 3.2 Data Storage 12

4 Data cleaning 15
- 4.1 Remove data to obtain a coherent set 15
- 4.2 Adding information 16
- 4.3 Data quality dimensions 17
  - 4.3.1 Completeness 18
  - 4.3.2 Uniqueness 18
  - 4.3.3 Timeliness 18
  - 4.3.4 Validity 18
  - 4.3.5 Accuracy 19
  - 4.3.6 Consistency 19

5 Dataset description 21
- 5.1 Users 21
- 5.2 Tweets 22

6 Tweet classification 27
- 6.1 Precision and Recall 30

7 Classification results 33
- 7.1 Tweet results 33
- 7.2 User results 37

8 One million follower fallacy 41

9 Conclusions 43

10 Conclusiones 45

Chapter 1

Introducción

Las redes sociales están cambiando cómo se comparten noticias e información en la red. No solo en el campo del periodismo, sino muchos otros campos como por ejemplo la política. Además, plataformas como el servicio de microblogging Twitter hacen posible que personalidades políticas hagan llegar sus mensajes a los usuarios directamente, que a su vez pueden crear contenido, así fomentando debate en la red.

Nuestro trabajo se propone analizar este debate en el contexto de las elecciones generales estadounidenses de noviembre de 2016. Para ello descargamos durante la semana que tuvieron lugar las elecciones todos los tweets que mencionaron a alguno de los dos candidatos para la presidencia: Donald Trump (@realDonaldTrump) y Hillary Clinton (@HillaryClinton). Durante la semana de captura de datos descargamos alrededor de 13 millones de tweets.

Como parte novedosa en el campo nos proponemos caracterizar el tipo de usuarios que apoya a cada candidato. En particular, la novedad de este trabajo es que nos interesamos por el sentimiento que los usuarios muestran hacia el candidato opuesto, es decir, al que no apoyan.

Antes de poder utilizar estos tweets hicimos una limpieza de datos para obtener un conjunto coherente. A continuación comprobamos el grado de calidad del conjunto mediante un proceso de auditoría descrito en el capítulo 4.

El análisis de sentimiento es una técnica común que se ha utilizado en muchos trabajos para obtener y clasificar automáticamente el significado de textos. Esta técnica se ha utilizado en redes sociales[1], en el contexto comercial para analizar opiniones de usuarios, y, como en nuestro caso, para estudiar las opiniones de los usuarios de Twitter en elecciones políticas[2].

Este trabajo contempla el problema de clasificar automáticamente millones de tweets utilizando un clasificador probabilístico conocido como clasificador Bayesiano ingenuo. Estos clasificadores pueden emplearse para tomar decisiones binarias simples, como por ejemplo decirnos si un nombre es masculino o femenino, o si un texto habla de un producto de manera positiva o negativa.

Nuestro caso es más complicado ya que un tweet puede tener sentimiento negativo hacia Trump y positivo hacia Clinton o tener sentimiento neutral hacia ambos. Esto da lugar a muchas posibilidades y es un problema que afrontamos en el capítulo 6. Para minimizar el número de posibilidades de clasificación abordamos el problema de forma paralela. Es decir, clasificamos cada tweet una vez por candidato.

Una vez clasificados todos los tweets podemos proceder a analizarlos y extraer resultados aplicados a tweets y usuarios como veremos en el capítulo 7. Para aplicar los resultados a los usuarios exportamos la clasificación de tweets para clasificarlos por usuarios. Estos resultados se encuentran en la sección 7.2.

Una vez terminado el proceso encontramos unas diferencias significativas entre los seguidores de ambos candidatos y veremos que en la mayor parte de los casos, los números benefician al candidato Donald Trump.

Chapter 2

Introduction

Social networking is changing how news and information is shared online. Not only in journalism, but in fields like marketing and politics as well. Social networking also gives politicians a platform that allows them to directly interact with users, who at the same time create content, leading to online debate.

This paper analyzes this debate in the context of the November 2016 U.S elections. In the week leading to the elections we downloaded over 13 million tweets published by users who mentioned one of the two presidential candidates: Donald Trump (@realDonaldTrump) and Hillary Clinton (@HillaryClinton).

Previous works have studied Twitter but few have tried to characterize the type of users who support each candidate. The novelty of this paper is that we examine the sentiment that users show toward the candidate they do not support.

The tweets we used could not be used in their raw format and we used a technique known as data cleansing to build a coherent dataset. We followed this up by assessing the quality of the data, a process described in chapter 4.

Sentiment analysis is a method that has been frequently used in other studies to automatically obtain and classify the meaning of texts. This method has been used in social networking[1], and, like in our case, to study the opinions of Twitter users during political elections[2].

Once the coherent dataset was built we encountered the problem of classifying every tweet. This was done using a probabilistic classifier known as a Naive Bayes Classifier. These classifiers ca be used to make simple binary decisions. For example, we could use a classifier to determine whether a name is male or female, or if a product review is positive or negative.

Our case is more complex because a tweet can have negative sentiment toward Trump and positive sentiment toward Clinton, or a tweet could simply be stating a fact and not have sentiment at all. The number of classifying choices makes the decision harder to make and is more thoroughly discussed in chapter 6. We made the classifier's job easier by taking a parallel approach, meaning each tweet was classified one time for each candidate.

When every tweet in the dataset was classified we were able to survey them and make a list of results for both tweets that can be found in chapter 7. In section 7.2 these results are exported and applied to users.

Our results show significant differences between the supporters of each candidate and, in most cases, the candidate Donald Trump comes out ahead.

Chapter 3

Data Capture

The first section this chapter will discuss how we gathered our data by working with an API provided by Twitter. The second section will detail what data we chose to store, the format in which the data was stored, and the database program we chose to store it in.

3.1 Data Collection

Twitter provides developers with an Application Programming Interface (API)[1] to connect their applications to the Twitter Platform using OAuth[2] to provide authorized access.

OAuth is an open protocol that allows secure authorization in a simple and standard method from web, mobile and desktop applications. This allows a third party to obtain limited access to an HTTP service as well as to simply publish and interact with protected data.

To obtain access we registered our application with Twitter and obtained a set of tokens. Twitter offers three different types of API:

- REST APIs allows the user to read and write Twitter data: create a new tweet, conduct singular searches, read user profile and follower data, and more.
- Streaming APIs give low latency access to Twitter's real-time tweet data. Using these APIs it is possible to follow specific users and subjects using public. streams[3].
- The Ads API allows partners to integrate Twitter's advertising platform their own.

We chose to use the Streaming APIs because our intention is to process and store tweets as they are published. It's worth noting that Twitter's REST API

[1]https://dev.twitter.com/overview/api
[2]https://oauth.net/
[3]https://dev.twitter.com/streaming/public

is rate limited, that is to say, users are allocated a certain number of requests per 15 minute window. Conversely, the Streaming API doesn't have this limitation and we were able to capture data unhindered.

There are many different libraries that support the current Twitter API. Twitter itself has only built and maintains libraries for Java and Objective-C. However, a wide variety of libraries across different programming languages have been built for the Twitter Platform. We chose the Python programming language because of its ease of use, familiarity with the language and because the Twitter API returns JSON structures, which are very simple to use and build in Python and are convenient to store as documents using MongoDB[4].

In order to obtain the dataset, we chose the Python library Tweepy[5]. Tweepy is an open-sourced library that enables Python to communicate with Twitter and use its API to provide programmatic access to Twitter data. We used Tweepy's Streamlistener class to track tweets mentioning to two specific users: @HillaryClinton and @realDonaldTrump in a time frame spanning from 1 November 2016 at 11:46Z to 13 November 2016 at 8:45Z, thus procuring Twitter data immediately before and after the 2016 US presidential elections.

3.2 Data Storage

MongoDB is a free, open-sourced, document-oriented NoSQL database. NoSQL databases provide a mechanism for storage and retrieval of data which is modeled in means other than the tabular relations used in relational databases. Other examples of NoSQL databases are Cassandra, HBase, Voldemort, and Neo4j, among many others. We used MongoDB because it's flexible, lightweight, and prior work and experience with it.

When a new tweet containing @HillaryClinton or @realDonaldTrump is detected, the MyStreamListener class of the Tweepy library returns a JSON structure comprised of a series of fields known as a status. Each incoming message was read, processed and stored in a MongoDB collection. We built two collections using this method: one composed of tweet documents, and the other composed of user documents.

Tweet documents contain the following fields:

_i : unique identifier
text : the tweet's text
created_at : the UTC time in YYYY-MM-DD-T-HH-MM-SS format when the tweet was created
user : the user's attributes

[4]https://www.mongodb.com/
[5]http://www.tweepy.org/

_id : the user's unique identifier
verified : when true, indicates that the user has a verified account
screen : the screen name, handle, or alias that this user identifies themselves with

In traditional SQL databases the user's details would not be part of a tweet document and this information would be obtained using a join operation to combine both collections. However, these operations are very costly to do in NoSQL databases and it is more efficient to store the author's details along with the tweet.

User documents contain the following fields:

_id : unique identifier
lang : the BCP 47[6] code for the user's self-declared user interface language
verified : when true, indicates that the user has a verified account
screen_name : the screen name, handle, or alias that the user identifies themselves with
url : a URL provided by the user in association with their profile
created_at : the UTC time when user account was created on Twitter
time_zone : a string describing the time zone this user declares themselves within
followers : the number of followers this account currently has
location : the user-defined location for this account's profile
following : the number of users this account is following
favourites_count : the number of tweets this user has favorited in the account's lifetime
geo_enabled : when true, indicates that the user has enabled the possibility of geotagging their tweets

A summary of the data collected can be found in figure 3.1.

Collection	Tweets	Users
Num. of documents	13,358,219	1,511,054
Time frame	1 November 2016 - 13 November 2016	

Figure 3.1: Summary of the data collected

[6]https://tools.ietf.org/html/bcp47

Chapter 4

Data cleaning

english

We gathered over 13 million tweets but soon realized that not all of them were valid for the purpose of our study. To obtain a coherent set we had to examine and clean our data. After this was done we also added a few more fields to our tweet and user documents.

This task was accomplished by using a JavaScript script along with MongoDB commands. We chose to use JavaScript because the MongoDB shell supports this language and therefore it's simple to integrate with MongoDB shell commands.

Finally, the last section of this chapter will discuss and evaluate the quality of our dataset.

4.1 Remove data to obtain a coherent set

A retweet is a reposted or forwarded tweet by another user. To examine retweets the original tweet has to be in the dataset. Some of the retweeted tweets were posted before the data capture began and the majority of tweets in our dataset are RTs so we had to discard RTs whose original tweet we don't have.

Knowing whether a tweet is a retweet is trivial as every retweet contains the substring ' "RT @ '. If a tweet is determined to be a retweet, the tweet is marked as a retweet and it's discarded if its source tweet is not in the dataset. A total of 2.44 million tweets were discarded. In many cases we stored several instances of tweets because of autocopy. Repeated tweets amounted to 600.000 tweets in the dataset that were also removed. Tweets published by accounts that were suspended or no longer existed were also removed. The problem of duplicate users did not occur because once a user was identified, user documents were only inserted into the dataset if they weren't in the dataset. Once the cleanup was completed our tweet dataset is composed of 7.803.405 tweets.

A flowchart of the cleanup process can be found in figure 4.1

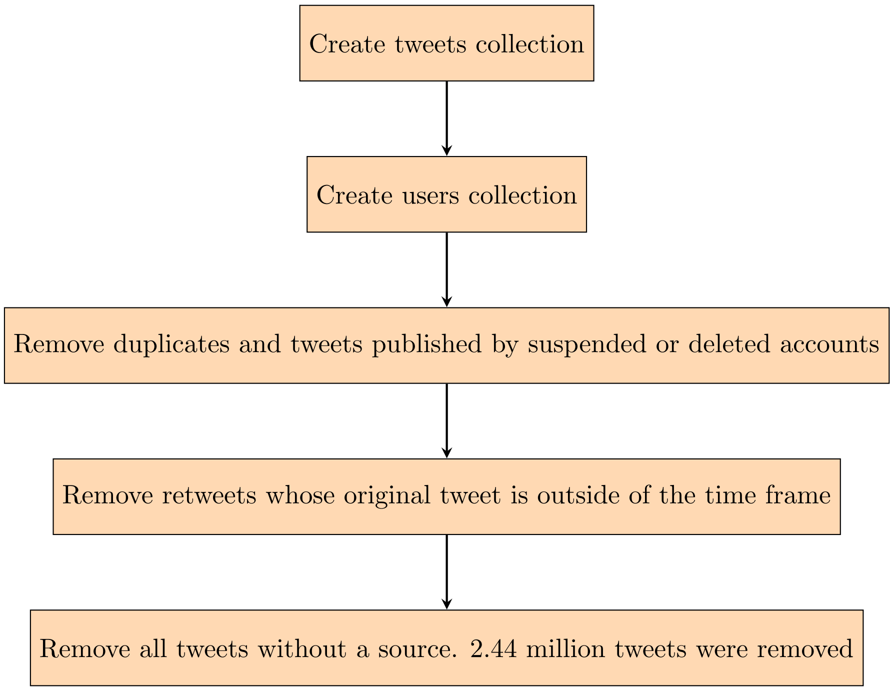

Create tweets collection
Create users collection
Remove duplicates and tweets published by suspended or deleted accounts
Remove retweets whose original tweet is outside of the time frame
Remove all tweets without a source. 2.44 million tweets were removed

Figure 4.1: Data removal cleanup

4.2 Adding information

To have a better understanding of the dataset we also added additional fields to both user and tweet documents. We added three fields related to retweets to every tweet document:

RT_screen for tweet documents. Indicates the original user's Twitter handle. Its value is nothing("") if the tweet was never retweeted
source the original tweet's ID. Its value is nothing("") if the tweet was never retweeted
RTin number of times the tweet has been retweeted. 0 if the tweet was never retweeted

We also compiled a summary for each user:

tweets : the user's Twitter activity report

- **all** : user's tweet count at the time of their first tweet in the time frame
- **total** : user's tweet count in the time frame. total = RT + original
- **RT** : number of retweets in the time frame
- **original** : number of original tweets in the time frame

Finally, fields related to sentiment analysis were added during the classification process. These fields will be discussed in chapter 4: Tweet classification.

A detailed illustration of the steps taken to add information to the dataset can be found in figure 4.2.

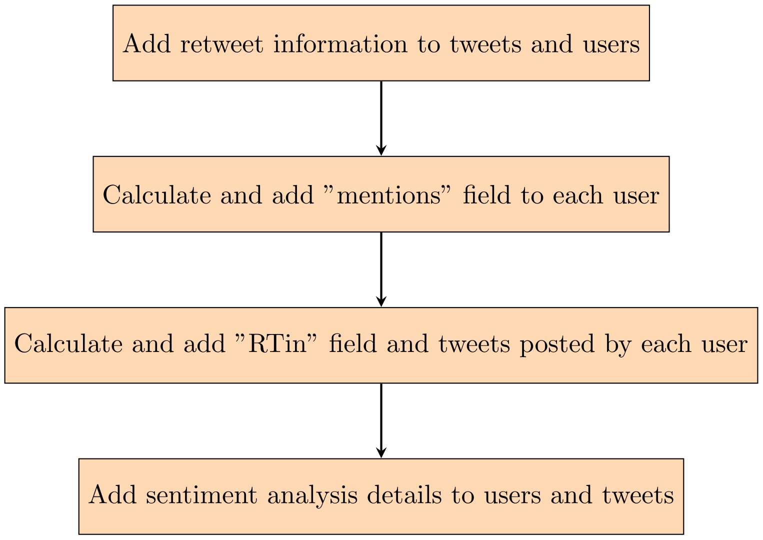

Add retweet information to tweets and users
Calculate and add "mentions" field to each user
Calculate and add "RTin" field and tweets posted by each user
Add sentiment analysis details to users and tweets

Figure 4.2: Data addition flowchart

4.3 Data quality dimensions

Once we cleaned the dataset and added new fields to the documents, we examined the dataset to measure the quality of the data. To do this, we based our analysis on a paper titled *The Six Primary Dimensions for Data Quality Assessment*[3]. Another interesting book on data quality is *Data Quality Assessment*[4]. This paper outlines six key 'dimensions' recommended to be used when assessing or describing data quality. In this context, the authors take the term *data quality dimension* to mean: some thing that can either be measured or assessed in order to understand the quality of the data. The six dimensions are:

- Completeness
- Uniqueness
- Timeliness
- Validity
- Accuracy
- Consistency

4.3.1 Completeness

Completeness is defined as *The proportion of stored data against the potential of "100% complete"*. In our case, this would be calculated by dividing the total number of tweets and users in our database by 100% of the tweets published in the time frame and the active users.

During the data capture phase our collection program experienced several short pauses during which we weren't able to collect data. Because our program was not running, we are unable to know how many tweets were missed, and thus unable to accurately measure completeness. However, we estimate that the data we missed is a small percentage of our dataset, and that the missed data is not critical.

A way we can calculate the completeness of our dataset is by comparing the initial data to the final usable dataset.

$$Completeness = \frac{7.803.405}{13.358.219} \times 100 = 58.4\%$$

4.3.2 Uniqueness

Uniqueness is defined as *No thing will be recorded more than once based upon how that thing is identified.* One way of calculating the degree of uniqueness would be to divide the number of unique documents by the number of total documents. Our case is special because we store original tweets and retweets. If we were to consider retweets exact copies of tweets, the uniqueness of the dataset would be of 71.1%.

However, in our case we consider each retweet to be unique because we don't only store the tweet's text, but data about the users who are forwarding the tweets. With that in mind, we can say that our dataset's uniqueness is of 100%.

4.3.3 Timeliness

Timeliness is defined as *The degree to which data represent reality from the required point in time.* This can be understood as the time that has elapsed between data collection and data storage. Our data collection program builds and stores the MongoDB documents as soon as they are detected, so there is a high degree of timeliness.

4.3.4 Validity

Validity is defined as *Data are valid if it conforms to the syntax (format, type, range) of its definition.* In our case validy means that our data conforms to a series of rules. One of the most important aspects of our dataset are that dates and times are stored in the same timezone. Because of this, all of our data is stored using Zulu time and date.

4.3.5 Accuracy

Accuracy is defined as *The degree to which data correctly describes the "real world" object or event being described.* We can apply this measure to our dataset by asking one question: how closely do our tweet documents resemble the tweets themselves in the "real world"?. We're able to store only what the Twitter API provides us. In some cases for long tweets, Twitter didn't provide the tweet's full text. Instead, it's cut off and replaced by an ellipsis (...). We believe the rest of our data to be accurate.

4.3.6 Consistency

Consistency is defined as *The absence of difference, when comparing two or more representations of a thing against a definition.* The unit of measure is percentage and it can be obtained by comparing the number of inaccurate records against the total number of records. An interesting characteristic about this dimension is that consistency can be achieved without validity or accuracy. If we apply this measure to our dataset we can say that it's consistent because of the way the database was built. If a user's document specifies they have published ten tweets, ten distinct tweets will appear in the tweet collection by that user. This makes our dataset 100% consistent.

With these data quality dimensions in mind, we feel confident of the quality of our dataset.

Chapter 5

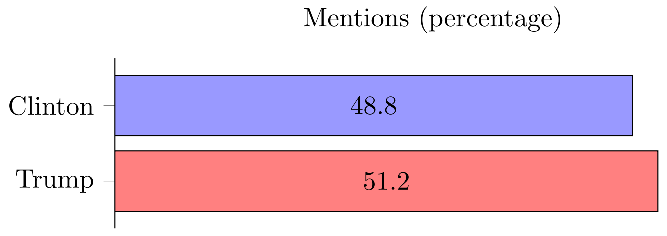
Dataset description

To have a better understanding of the data obtained we queried the dataset and compiled a number of statistics that will be illustrated in this chapter. There will be a subsection for the two aforementioned entities: users and tweets.

5.1 Users

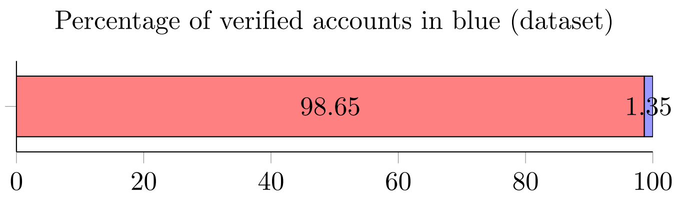
To compare the users tweeting each candidate we separated users by grouping them together depending on which candidate they mentioned: @HillaryClinton or @realDonaldTrump.

The obvious first step is to see what percentage of users mentioned each candidate in their tweets. Both candidates were mentioned equally in our dataset; no discernible difference was observed.



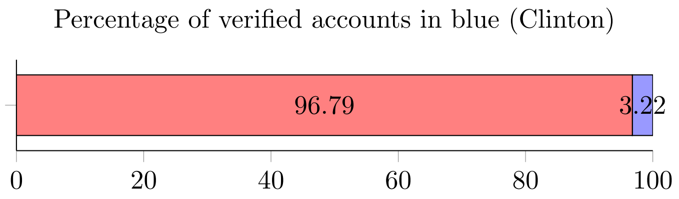
Verified accounts are those that Twitter considers to be of public interest. The owners of these accounts are usually news networks, celebrities, performers, users specializing in key interest areas, and others. Twitter uses a blue verified badge to let others know that an account of public interest is authentic.

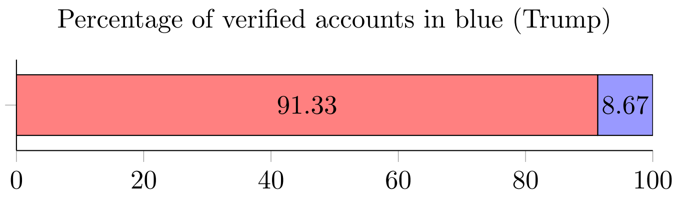
It is not easy to obtain a verified badge; the vast majority (98.65%) of users in our dataset are unverified users. Accounts must meet a variety of requirements for them to be verified, the most important being that all published tweets by the account must have an open privacy setting, meaning tweets by the account are able to be seen by any Twitter user. Additionally, the owner of the account must fill out a form and submit it to Twitter, who assesses each request individually and approves or denies it. If the request is denied, users must wait 30 days before submitting another.



Since verified accounts are deemed to be of public interest, they attract more users and therefore have a far larger average amount of followers than regular accounts do. The average number of followers for verified accounts is 173054 whereas unverified accounts have an average of 1387 followers.

If we perform the same calculation on each candidate's set we saw a considerable difference:

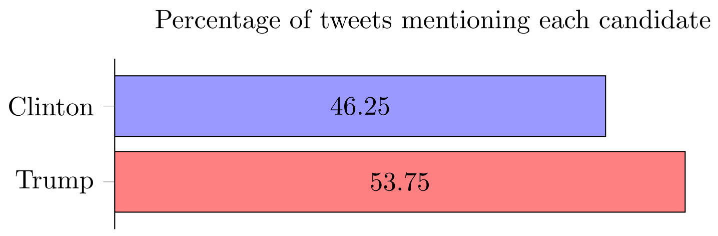




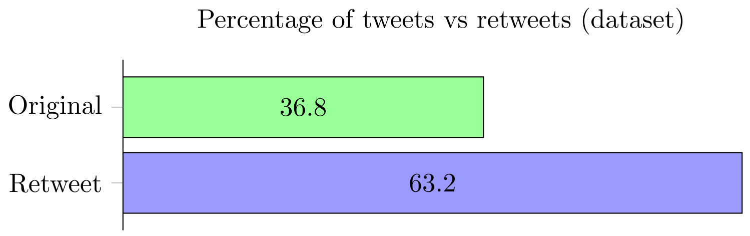
This difference is important because, as we've seen, verified users are more influential than standard users and in this case it could be said that tweets mentioning Trump had more exposure than tweets mentioning Clinton.

5.2 Tweets

After cleaning up the data we had a sizable dataset of tweets that we divided into two groups in the same manner as we divided the users: using mentions. These groups are not disjoint; a tweet that mentions both candidates will be part of both groups.

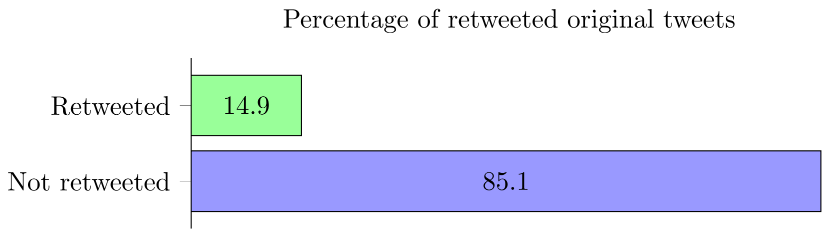


An important aspect about our dataset of tweets is how many of the tweets are actually original and how many tweets are simply retweets, or forwarded tweets.



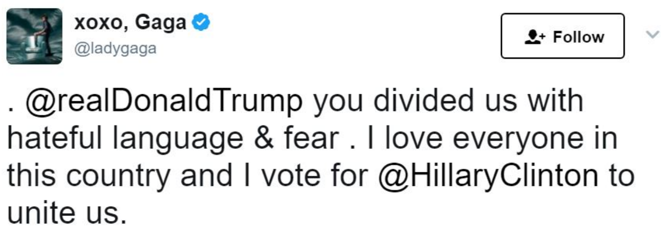
We found that during the time we built the dataset, users interacted with each other by forwarding others' tweets, rather than posting their own original tweets. We also performed the same calculation on each candidate's tweet dataset and no significant difference was observed. Indeed, the results are almost identical to the dataset of all tweets.

We can divide the original tweets in the dataset in two categories: original tweets with retweets and original tweets that weren't retweeted.



Of our dataset of several million tweets, only 36.8% of tweets are original tweets. Only 14.9% of those original tweets were retweeted. This means that 63.2% of the tweets in the dataset are actually retweets of a small 5.49% of the entire dataset. This leads us to believe that a vast majority of tweets fall on deaf ears.

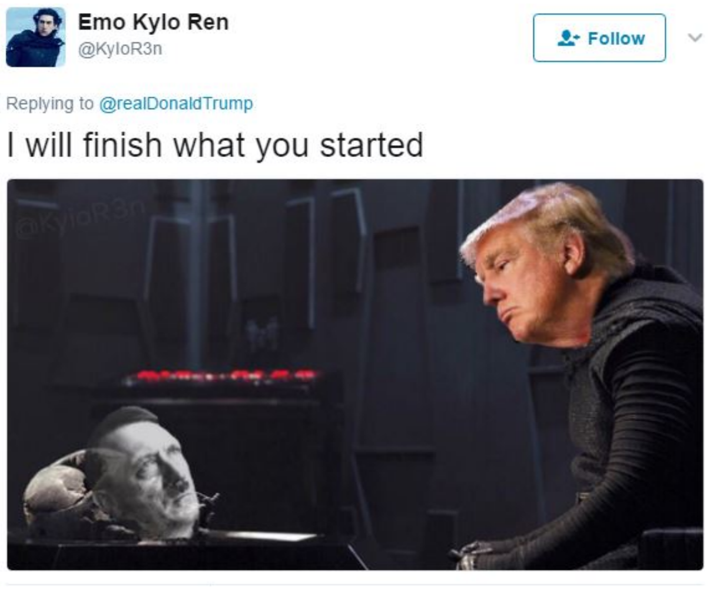
The most retweeted tweet mentions both candidates, so it tops the list of the most retweeted tweet of both candidate's dataset. The author of this tweet is @ladygaga, the second most followed account on Twitter and the tweet was forwarded 17993 times.



xoxo, Gaga
@ladygaga

. @realDonaldTrump you divided us with hateful language & fear . I love everyone in this country and I vote for @HillaryClinton to unite us.

The most retweeted tweet by an unverified author is part of @realDonaldTrump's dataset, and was retweeted 17143 times:

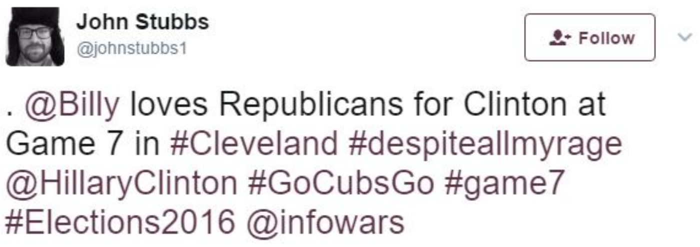


Emo Kylo Ren
@KyloR3n

Replying to @realDonaldTrump

I will finish what you started

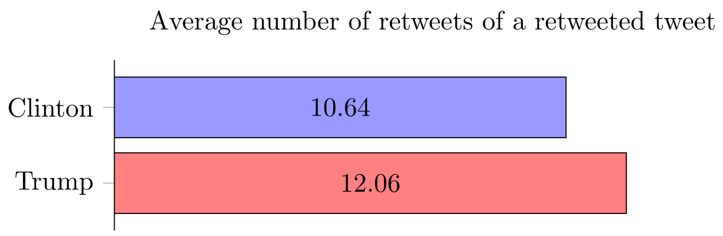
Conversely, the most retweeted tweet of @HillaryClinton's dataset by an unverified author was retweeted 9546 times:



John Stubbs
@johnstubbs1

. @Billy loves Republicans for Clinton at Game 7 in #Cleveland #despiteallmyrage @HillaryClinton #GoCubsGo #game7 #Elections2016 @infowars

The average number of retweets of a tweet with at least one retweet is 11.5 retweets. If we compare the candidate's dataset we can see that retweeted tweets mentioning @realDonaldTrump are forwarded more times than those mentioning @HillaryClinton.



With the data at hand we were able to plot the frequency of tweets by time and date shown in figure 3.1. We can clearly see the frequency of activity rise during the day and fall at night, as well as a considerable surge in Twitter activity after election day (November 8) once the results came in.

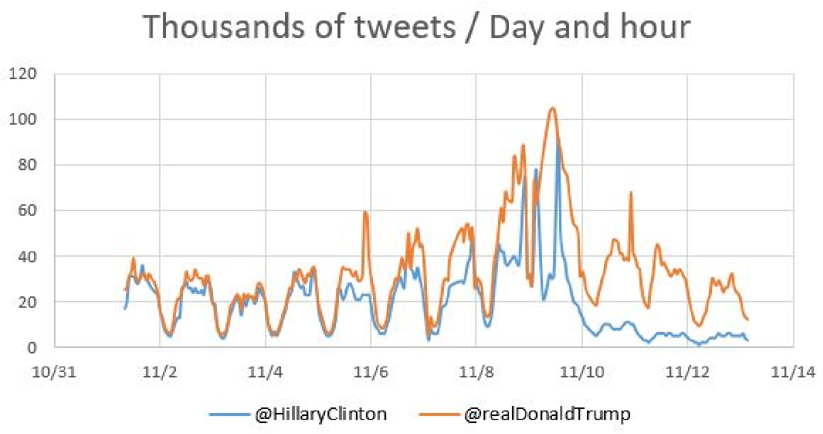


Figure 5.1: Thousands of tweets per day and hour

Chapter 6

Tweet classification

To identify and categorize opinions (sentiment) we created a set of labels to assign to each tweet. The tweet's sentiment isn't always clear; it can be ambiguous, dependent upon the context, the tweet may be sarcastic or ironic, the tweet is simply the statement of a fact and has no sentiment at all, the tweet could not be understood, or the tweet may satisfy two labels at the same time. For example, a tweet's sentiment can be favorable toward one candidate and negative toward the other.

Each tweet was assigned up to two of five labels:

zero: tweet isn't biased in favor of or against either candidate
TPos: sentiment is favorable toward Trump
TNeg: sentiment is negative toward Trump
HPos: sentiment is favorable toward Clinton
HNeg: sentiment is negative toward Clinton

Manually classifying nearly 8 million tweets is not practical so the dataset classification was conducted automatically using a machine learning technique known as supervised classification. For the purpose of this study, classification is the task of choosing the correct label for each tweet. A classifier is supervised if it has been trained using a set of manually labeled tweets and their basic features. Tweets written in a language other than English were labeled "zero". To classify our dataset we used a Naive Bayes classifier.
Naive Bayes classifiers are based on Bayes' theorem. In statistics, Bayes' theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event. Bayes' theorem is stated with the following equation:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

where A and B are events and P(B) $\neq$ 0.

In the context of a Naïve Bayes classifier the formula's elements are used as follows:

$$P(label|features) = \frac{P(features|label)*P(label)}{P(features)}$$

- P(label) is the probability of the label appearing
- P(features) is the probability of the features appearing
- P(features|label) is the probability of the label appearing given a feature set

Features are the most important or defining parts of the tweet that make a contribution towards deciding which label should be assigned to it. Tweet features are in the form of n-grams, which are a contiguous sequence of n words. Our classifier uses unigrams, which are standalone elements and also bigrams, which are pairs of consecutive words.

Naive Bayes classifies documents to their right category or class. Tweets are represented by a features vector $d = (d_1, \ldots, d_n)$ and C is a class assigned to d, where

$$P(d|C_k) = \prod_{i=1}^{n} P(d_i|C_k)$$

Bayes classifiers use a normalizing factor behind the scenes, usually referred to as λ,which is defined as a normalizing factor and it's a small number that's used to nullify probabilities that are zero. The value we used was $\lambda = 0.0002$. This normalizing factor is used when the probability of a class being assigned to a feature is zero.

Naive Bayes starts by calculating the prior probability of each label based on how frequently it appears in the training set. Supervised Naive Bayes classifiers go through two stages: training and prediction. During training, the machine learning algorithm is fed the tweet's features as well as the correct label to create a classifier model. Once the model is created, the classifier can be fed the tweet's features. Now the classifier can generate the predicted label.

An appealing characteristic of Naive Bayes classifiers is that they return the probability of each label. This helps us compare results and discard them if the accuracy is below a certain probability threshold, which in our case is 0.999. If the classifier ascertains that a tweet has sentiment with a probability lower than the threshold, the result is rejected. This results in a large number of rejected results, which in more specific terms means that we sacrificed recall in favor of precision, measures that will be discussed in the next section.

An illustration from the *Natural Language Processing with Python* book[5] provides a good illustration of the process followed by Naive Bayes classifiers and can be found in figure 6.1.

To train the model we manually labeled a training set of 2000 tweets, the results of which can be found in figure 6.2.

The program in charge of classifying the tweets was implemented using the Java programming language. This program uses the Naive Bayes multiclass

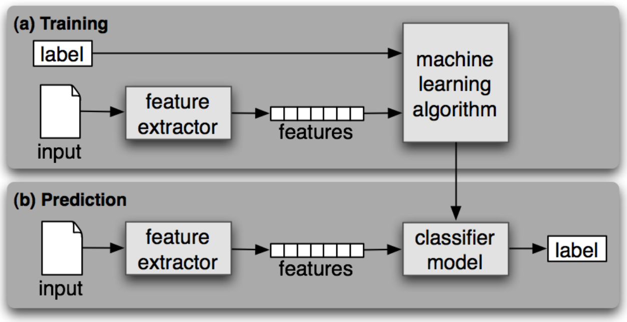


Figure 6.1: Naive Bayes classifier diagram

Label	Num. of tweets
zero	573
TPos	521
TNeg	292
HPos	215
HNeg	595

Figure 6.2: Manual classification of tweets

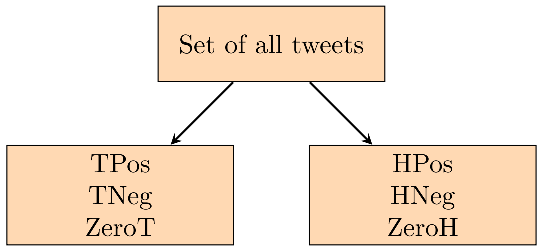
classification algorithm that can be found in the MLlib library. MLlib is Apache Spark's scalable machine learning library.[1]

Our initial classifiers didn't yield very good results. We thought that maybe a training set with only 2000 tweets wasn't enough so we manually labeled an additional 400 but didn't detect a significant accuracy increase.

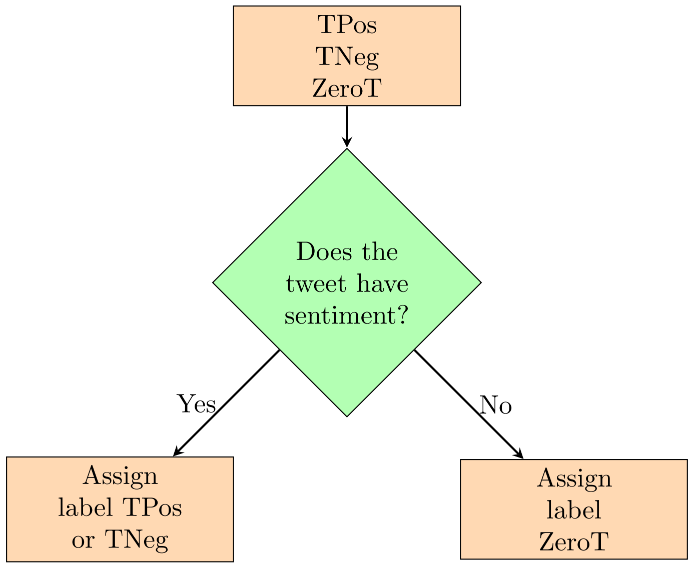
Many Naive Bayes classifiers can correctly predict the label approximately 80% of the time. However, they typically only have to guess correctly between two labels. For example: good or bad, happy or sad, male or female. On the other hand, our classifier must choose one of five labels.

Because we had five labels to choose from, we had to take a parallel approach in our classifier. To help it decide between labels we had it choose between positive, negative, or zero. With a starting set of entries of the form "Tweet + Tag", we created two disjoint sets: one set composed of positive for Trump, negative for Trump, and tweets with zero sentiment toward Trump. These sets are disjoint. A tweet could be classified as positive toward Trump and neutral toward Clinton, but this option is not considered in Naive Bayes classifiers. Naive Bayes classifiers consider each label to be independent from the rest. The same is applied to a second set of Clinton tweets:

[1]https://spark.apache.org/docs/latest/mllib-naive-bayes.html



This way we were able to run the Bayes classifier on each set using binary decisions, which made the prediction much more accurate than it would have been to run the classifier on the original set with five labels. The following is a graphical representation of the classification process for the Trump set:



To test the classifier's accuracy, the classifier takes 80% of the test set and tries to guess the correct labels for the other 20% of tweets. After each test the classifier creates an error report in which we can see what tweets were guessed incorrectly. In some cases, the classifier was correct in its prediction and the manually labeled tweet was incorrect.

Once the results on the training set were satisfactory, the classifier was run on a test set of 400 tweets. This is done to avoid overfitting, which occurs when a classifier has been extensively trained to correctly guess the labels for the training set but its prediction is poor for test sets.

6.1 Precision and Recall

Once each tweet was labeled by the classifier we had to verify the correctness of the results. To do this, we used performance measures such as precision, recall, and accuracy.

Precision is a measure we can use to check how exact the classifier's results are. A different way of defining precision could be to say that low precision deals with the idea of false positives, that is, tweets that were assigned a certain label incorrectly. If, for example, we take the set of tweets labeled TNeg (Negative sentiment toward Trump) we can use a simple calculation to obtain the precision:

$$Precision = \frac{Number\ of\ tweets\ correctly\ labeled\ TNeg}{Number\ of\ total\ tweets\ labeled\ TNeg}$$

The results of applying this measure to every possible label for the candidate Donald Trump can be found in figure 6.3.

Label	Precision
TPos	88%
ZeroT	90%
TNeg	82%

Figure 6.3: Precision results for the republican candidate Donald Trump

While the results of the same measure applied to the candidate Hillary Clinton can be found in figure 6.4.

Label	Precision
HPos	99%
ZeroH	82%
HNeg	88%

Figure 6.4: Precision results for the democratic candidate Hillary Clinton

Another measure we can use to measure the quality of our results is recall. Recall allows us to measure the proportion of correctly labeled tweets. If the recall value is low, this indicates that the classifier has produced a large number of false negatives, that is, tweets that were incorrectly labeled. We used the following formula to calculate the classifier's recall using the label TNeg:

$$Recall = \frac{Number\ of\ tweets\ correctly\ labeled\ TNeg}{Number\ of\ TNeg\ tweets\ in\ the\ dataset}$$

The results of applying the recall measure to every possible label for the candidate Donald Trump can be found in figure 6.5.

The results of applying the recall measure to every possible label for the candidate Hillary Clinton can be found in figure 6.6.

The overall accuracy results for both candidates are very high and satisfactory and can be found in figure 6.7.

Label	Recall
TPos	85.68%
ZeroT	89.81%
TNeg	86.9%

Figure 6.5: Recall results for the republican candidate Donald Trump

Label	Recall
HPos	91%
ZeroH	97%
HNeg	58%

Figure 6.6: Recall results for the democratic candidate Hillary Clinton

Candidate	Overall accuracy
Donald Trump (R)	87.8%
Hillary Clinton (D)	86.4%

Figure 6.7: Overall accuracy results for both candidates

Chapter 7

Classification results

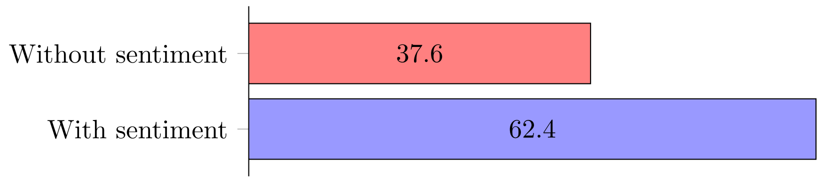
The classifier we described in the previous chapter assigned one or more labels to each tweet depending on its sentiment. Once the classifier was finished labeling tweets, we had a set of 7,8 million labeled tweets and 1,5 million users to analyze.

7.1 Tweet results

After the classifier was finished, three new fields were added to every tweet in the dataset:

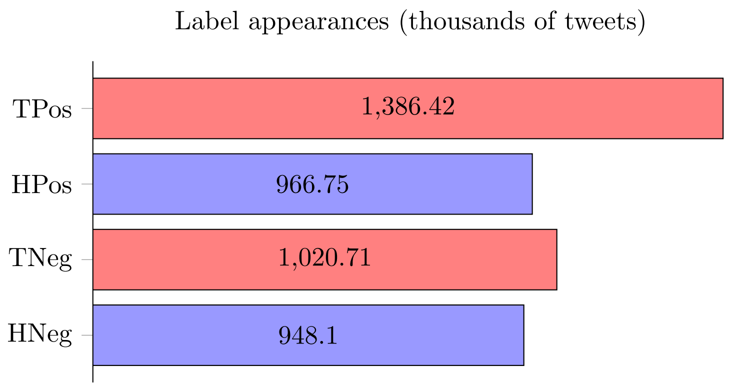
opinion : if the tweet has sentiment it's equal to 1, otherwise it's 0
hlabel : if the sentiment is positive toward Clinton the field takes a value of 1. If it's negative it will be -1, and if it has no sentiment toward Clinton it will be 0
tlabel : if the sentiment is positive toward Trump the field takes a value of 1. If it's negative it will be -1, and if it has no sentiment toward Clinton it will be 0

The first step the classifier took, as mentioned in chapter 4, was to decide if a tweet has sentiment or not. This results in two disjoint sets: one set of tweets with ascertainable sentiment and a second set of tweets without sentiment.



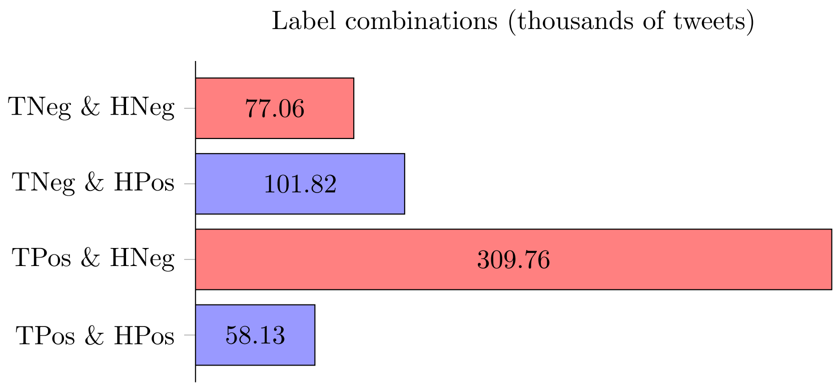
The classifier provided us with a set of 4,86 million tweets with sentiment, 62.4% of the original set. This is the set that we will consider in this chapter. The set of tweets without sentiment will not be discussed any further.

The following graph illustrates the number of tweets for each individual label, that is to say, only one label and no other, in thousands of tweets.



As we can see, there are clearly more tweets with positive sentiment toward Trump than Clinton, and the proportion of tweets with positive sentiment vs negative sentiment is larger for Trump than for Clinton. Indeed, Clinton's ratio is almost 1:1, which means that for every tweet with positive sentiment, there is another with negative sentiment.

Now we can take a look at the tweets that were assigned more than one label. Two combinations will not appear because they're complementary: TPos and TNeg, HPos and HNeg.

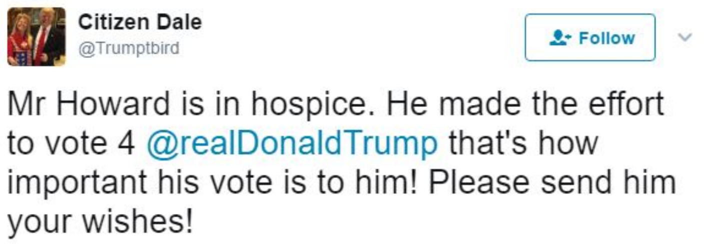


In this case one of the combinations immediately captures our attention. The number of tweets with a positive sentiment toward Trump and negative sentiment toward Clinton more than triples the second most observed combination of labels. We interpret this to mean that Trump followers, those who publish tweets with positive sentiment toward Trump, are far more likely to use Twitter to speak ill of Clinton than Clinton supporters are to speak ill of Trump.

Perhaps unsurprisingly, the least observed combination of labels was the combination of positive sentiment toward both candidates.

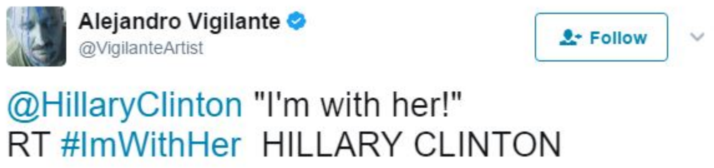
We queried our dataset to find the most representative tweet for each label, which we determined to be the one with the most retweets and a correctly guessed sentiment by the classifier. We will begin by finding the most representative tweets that were only assigned one label. The tweets were taken directly from Twitter in order to better illustrate them.
We'll also show how many times the tweet was retweeted at the time we finished capturing our data. If we were to search for the tweet today, the number of retweets would differ because the tweet continued to be retweeted after we finished the process of capturing data.

The most representative tweet for the label TPos was posted by the user @Trumptbird and it has 7705 retweets:


Citizen Dale
@Trumptbird

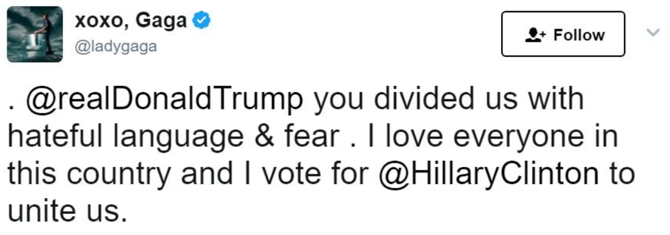
Mr Howard is in hospice. He made the effort to vote 4 @realDonaldTrump that's how important his vote is to him! Please send him your wishes!

The most representative tweet for the label HPos was posted by the user @VigilanteArtist and it has 16485 retweets:


Alejandro Vigilante
@VigilanteArtist

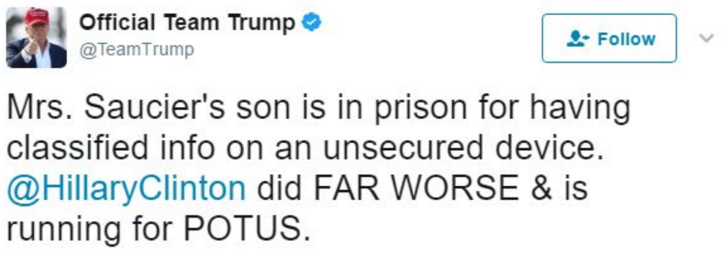
@HillaryClinton "I'm with her!"
RT #ImWithHer HILLARY CLINTON

The most representative tweet for the label TNeg was posted by the user @ladygaga and it has 17993 retweets. This is the same tweet we saw in the second section of the third chapter:


xoxo, Gaga
@ladygaga

. @realDonaldTrump you divided us with hateful language & fear . I love everyone in this country and I vote for @HillaryClinton to unite us.

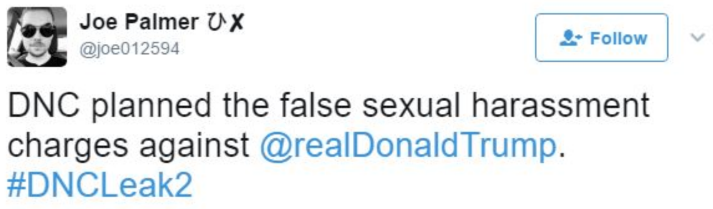
The most representative tweet for the label HNeg was posted by the user @TeamTrump and it has 11462 retweets:


Official Team Trump
@TeamTrump

Mrs. Saucier's son is in prison for having classified info on an unsecured device. @HillaryClinton did FAR WORSE & is running for POTUS.

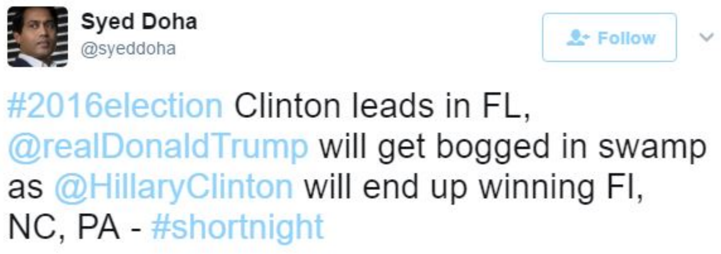
Now we can query the set to find the most representative tweets that the classifier assigned more than one label to.

In the case of the combination of the labels TNeg and HNeg the most representative tweet was posted by the user @joe012594 and was retweeted 8882 times:


Joe Palmer ひX
@joe012594

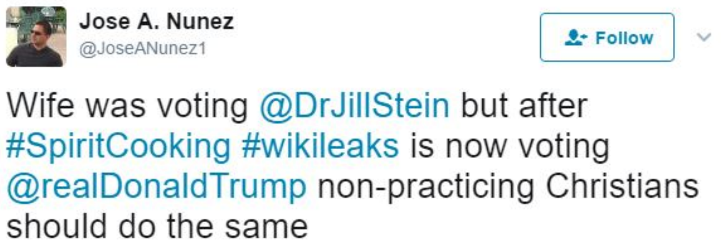
DNC planned the false sexual harassment charges against @realDonaldTrump. #DNCLeak2

In the case of the combination of the labels TNeg and HPos the most representative tweet was posted by the user @syeddoha and was retweeted 2845 times:


Syed Doha
@syeddoha

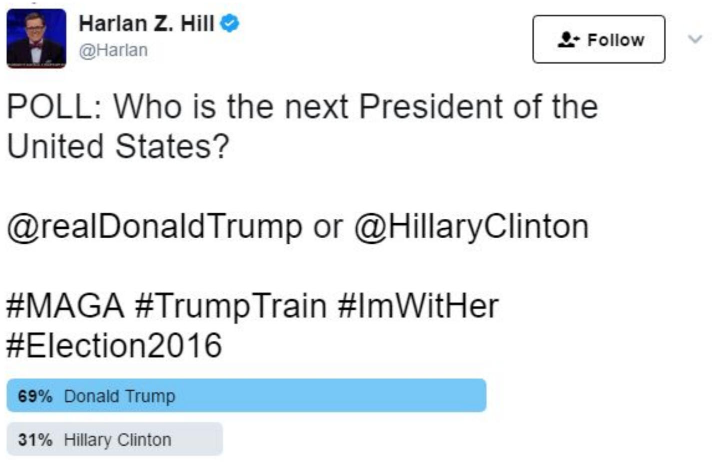
#2016election Clinton leads in FL, @realDonaldTrump will get bogged in swamp as @HillaryClinton will end up winning Fl, NC, PA - #shortnight

In the case of the combination of the labels TPos and HNeg, the most repeated combination of labels, the most representative tweet was posted by the user @JoseANunez1 and was retweeted 4555 times:


Jose A. Nunez
@JoseANunez1

Wife was voting @DrJillStein but after #SpiritCooking #wikileaks is now voting @realDonaldTrump non-practicing Christians should do the same

Finally, in the case of the combination of the labels TPos and HPos, the least frequent combination of labels, the most representative tweet was posted by the user @Harlan and was retweeted 3230 times:


Harlan Z. Hill
@Harlan

POLL: Who is the next President of the United States?

@realDonaldTrump or @HillaryClinton

#MAGA #TrumpTrain #ImWithHer #Election2016

69% Donald Trump
31% Hillary Clinton

7.2 User results

Once every tweet was labeled, a sentiment summary of the every user's published tweets was added to the user documents:

h0 : number of tweets published during the time frame with neutral sentiment toward Clinton

hp : number of tweets published during the time frame with positive sentiment toward Clinton

hn : number of tweets published during the time frame with negative sentiment toward Clinton

t0 : number of tweets published during the time frame with neutral sentiment toward Trump

tp : number of tweets published during the time frame with positive sentiment toward Trump
tn : number of tweets published during the time frame with neutral sentiment toward Trump
ph0 : h0/tweets.total
php : hp/tweets.total
phn : hn/tweets.total
pt0 : t0/tweets.total
ptp : tp/tweets.total
ptn : tn/tweets.total

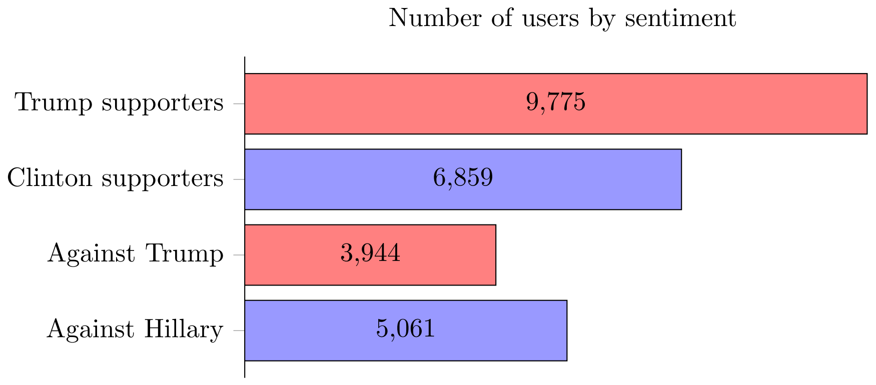
Where the following is true:
t0+tp+tn = tweets.total = tweets.RT + tweets.original
h0+hp+hn = tweets.total = tweets.RT + tweets.original
Using these fields we used a formula using a 20% margin for error to ascertain whether a user supports a candidate. If we were to find Trump supporters we would use this formula to find users that make it true:

ptp > (pt0 + ptn + 0.2) & tweets.total > 5

Conversely we can use the same formula to find users who actively oppose Trump:

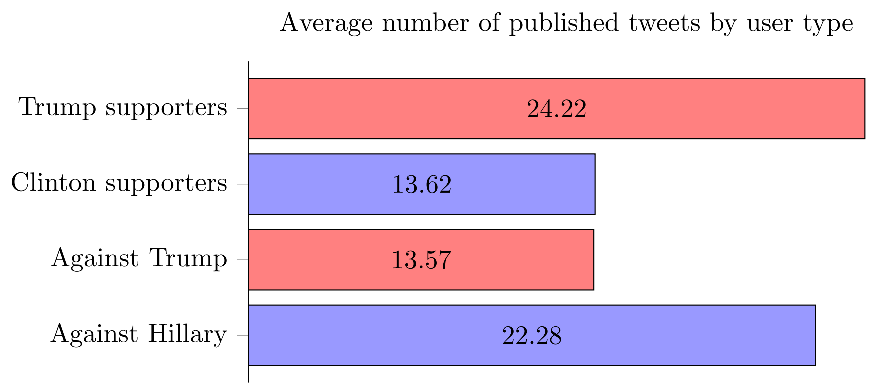
ptn > (pt0 + ptp + 0.2) & tweets.total > 5

The total amount of supporters is proportionally low to the total number of users. However, taking into account the margin of error of the classifier and the strictness of our measure we can be reasonably sure that these users have been classified correctly.
Using these measures we were able to build four subsets of users:

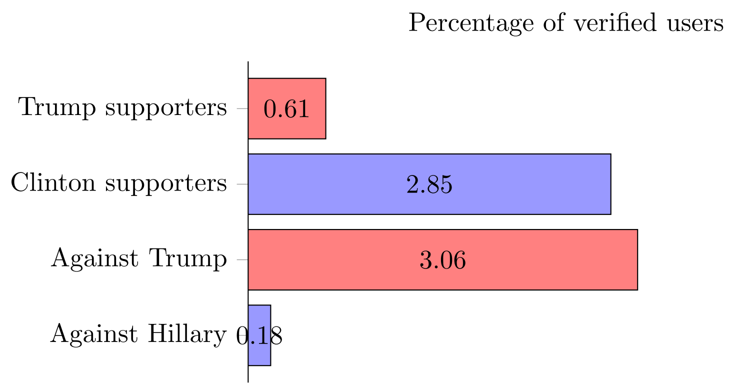


The results seem to confirm the data observed in the previous section: there are more Trump supporters in our dataset and more users are publishing tweets with negative sentiment toward Clinton than toward Trump.

Using these four new subsets we can take a look at their characteristics and compare the types of user:



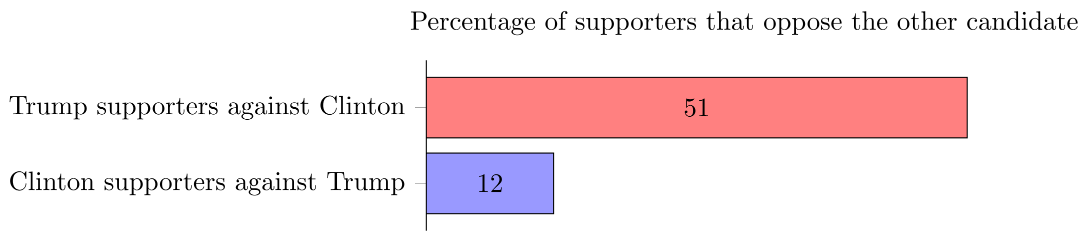
Again, this measurement seems to be in line with what we've observed so far: Trump has more supporters than opponents in our dataset, and the opposite is true for Clinton. Furthermore, Trump supporters publish considerably more tweets than Clinton supporters.



An interesting phenomenon takes place if we query the dataset to obtain the percentage of verified users for the different types of user. While verified users are a small fraction of the dataset we can see that the results differ from the ”pro Trump” trend: Clinton supporters and Trump opponents show a bigger proportion of verified users than Trump supporters and Clinton opponents.

A possible explanation for this is that Trump supporters are more active on Twitter and have more presence than Clinton supporters, while Clinton supporters are not as vocal but their tweets have more public interest than tweets published by Trump supporters.

Now that we have identified Trump and Clinton supporters we can query the dataset to find users who support a candidate and actively oppose the other. We consider a supporter to oppose the other candidate if at least 20% of their tweets have negative sentiment toward the other candidate.



The results are clear and they confirm what we suspected: Trump supporters are far more likely to publish negative tweets about Clinton than Clinton supporters are to publish negative tweets about Trump.

Chapter 8

One million follower fallacy

During the course of our research we encountered a term coined *million follower fallacy*[6] by Avnit (Avnit 2009) who claims that having a large amount of followers does not always equate to being influential in the Twitter world. We based our analysis on the paper[7] titled *Measuring User Influence in Twitter: The Million Follower Fallacy* that discusses and expands upon the original Avni paper.

First, we must define what having influence on Twitter means. The Cambridge Dictionary defines influence as "the power to have an effect on people or things, or someone or something having such power". Since there is no concrete way of measuring influence on Twitter, we follow Cha et al's example and underline three activities that we will use to measure influence:

- Followers: the number of followers the user has directly indicates the size of their audience.
- Mentions: the number of times the user has been mentioned indicates the user's means to interact with other Twitter users
- RTin: the number of times the user's original tweets have been shared indicates the user's ability to generate content

In order to investigate how the three measures correlate, we compared the relative influence of the top 2033 users based on number of followers.

The following table represents a graphical example of the dataset's top 10 users based on number of followers:

Screen name	Followers	Mentions	RTin
katyperry	93816740	5169	2661
ladygaga	64298611	5812	22304
nytimes	31153306	15916	479
MileyCyrus	30974762	1219	16050
CNN	29185114	67413	2567
BBCBreaking	26634986	1874	7825
ConanOBrien	22068465	115	76
BBCWorld	16378247	2434	950
Reuters	15211054	2336	403
RyanSeacrest	15054013	91	89

The follower, mention, and RTin data for the top 2033 users based on number of followers was stored in the CSV ("Comma Separated Values") file format and the correlations were calculated using R, which is a free software environment for statistical computing and graphics.[1]. R is very simple to use and the correlations were calculated by using a single command: cor(data).

The correlations are as following:

	Followers	Mentions	RTin
Followers	1	0.1953174	0.1110613
Mentions	0.1953174	1	0.7657633
RTin	0.1110613	0.7657633	1

We can see that there is no correlation across the pairs (follower influence, mentions influence) and (follower influence, RTin influence). However, we observe a high correlation (0.76) between mentions influence and RTin influence.

Indeed, only three users have a presence in all (followers, RTin, and mentions) top-100 lists: BernieSanders (politician), ladygaga (performer), and FoxNews (television news channel).

We conclude that the users with most followers are generally not the most influential when it comes to spreading their messages and engaging others in conversation.

[1]https://www.r-project.org/

Chapter 9

Conclusions

During the course of our work we showed that millions of posts of raw data in social networks during a major election can be analyzed using machine learning techniques that offer great potential.

While sentiment analysis is a valuable tool and many authors have compared techniques[8], steps can still be taken to improve it and in order to obtain more relevant results it's important to improve the accuracy of sentiment analysis classifiers.

We found that Twitter can be used to find defining characteristics shared by users who support a certain candidate. We have shown that the republican candidate Donald Trump had more supporters on Twitter than the democratic candidate Hillary Clinton, that Donald Trump had more active and vocal supporters during the days leading up to the election and that Trump supporters were more likely to publish negative tweets toward Hillary Clinton than vice versa. In most of the data considered Donald Trump has more favorable data than Hillary Clinton. However, as shown in section 7.2, a higher percentage of Hillary Clinton's supporters are verified Twitter users than Donald Trump's supporters. Most verified users are news organizations, celebrities, performers and other prominent figures. A possible explanation for this data could be that Hillary Clinton had more acceptance in the mainstream media than Donald Trump.

Further work needs to be done to establish the reason for these results: did Hillary Clinton's campaign fail? Or, on the other hand, did Donald Trump simply have more supporters online? A first step toward finding the reasons behind these results could be tracing the origin of the most common tweets and determine whether tweets are anonymous and spontaneous or whether the candidate's campaign or affiliated media are behind them.

Chapter 10

Conclusiones

Durante nuestro trabajo hemos visto que se pueden analizar millones de publicaciones de redes sociales utilizando técnicas de aprendizaje automático que ofrecen muchas posibilidades.

El análisis de sentimiento es una herramienta valiosa y muchos autores han comparado técnicas, pero se debe mejorar el porcentaje de acierto para obtener resultados relevantes.

Hemos comprobado que Twitter puede ser utilizado para caracterizar a usuarios de Twitter que apoyan a determinados candidatos políticos. Hemos visto que el candidato republicano Donald Trump tuvo más seguidores en Twitter durante la campaña electoral que la candidata democrática Hillary Clinton, que los seguidores de Donald Trump fueron más activos en la semana de las elecciones y que los seguidores de Donald Trump tenían más probabilidades de publicar mensajes en contra de Hillary Clinton que viceversa. En la mayor parte de los casos los resultados son más favorables para Donald Trump pero hemos visto en la sección 7.2 que hay una excepción: Hillary Clinton tiene mayor número de usuarios verificados entre sus seguidores que Donald Trump. Los usuarios verificados son frecuentemente organizaciones mediáticas, celebridades y otras personas conocidas. Una posible interpretación de estos resultados es que mientras que Donald Trump tuvo más seguidores, los seguidores de Hillary Clinton fueron más mediáticos.

Para dar con la causa de estos resultados se debe seguir investigando: ¿falló la campaña de Hillary Clinton? ¿simplemente Donald Trump tuvo más seguidores en las redes sociales? Un primer paso para encontrar la causa de estos resultados sería encontrar el origen de los tweets más comunes y determinar si fueron publicados de manera anónima y espontánea o si, por lo contrario, fueron gestionados por las campañas de los candidatos o por medios afiliados a ellos.

Bibliography

[1] Daniel M. Romero, Wojciech Galuba, Sitaram Asur, and Bernardo A. Huberman. Influence and passivity in social media. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 113–114, New York, NY, USA, 2011. ACM.

[2] Deb Roy Sophie Chou. Nasty, brutish, and short: What makes election news popular on twitter? 2017.

[3] DAMA UK Working Group. The six primary dimensions for data quality assessment. October 2013.

[4] Leo L. Pipino, Yang W. Lee, and Richard Y. Wang. Data quality assessment. *Commun. ACM*, 45(4):211–218, April 2002.

[5] Ewan Klein Steven Bird and Edward Loper. *Natural Language Processing with Python.* O'Really Media, Reading, Massachusetts, 2009.

[6] A Avnit. The million followers fallacy. 2009.

[7] Fabrício Benevenuto Krishna P. Gummadi Meeyoung Cha, Hamed Haddadi. Measuring user influence in twitter: The million follower fallacy.

[8] Hoda Korashy Walaa Medhat, Ahmed Hassan. Sentiment analysis algorithms and applications: A survey. 2013.