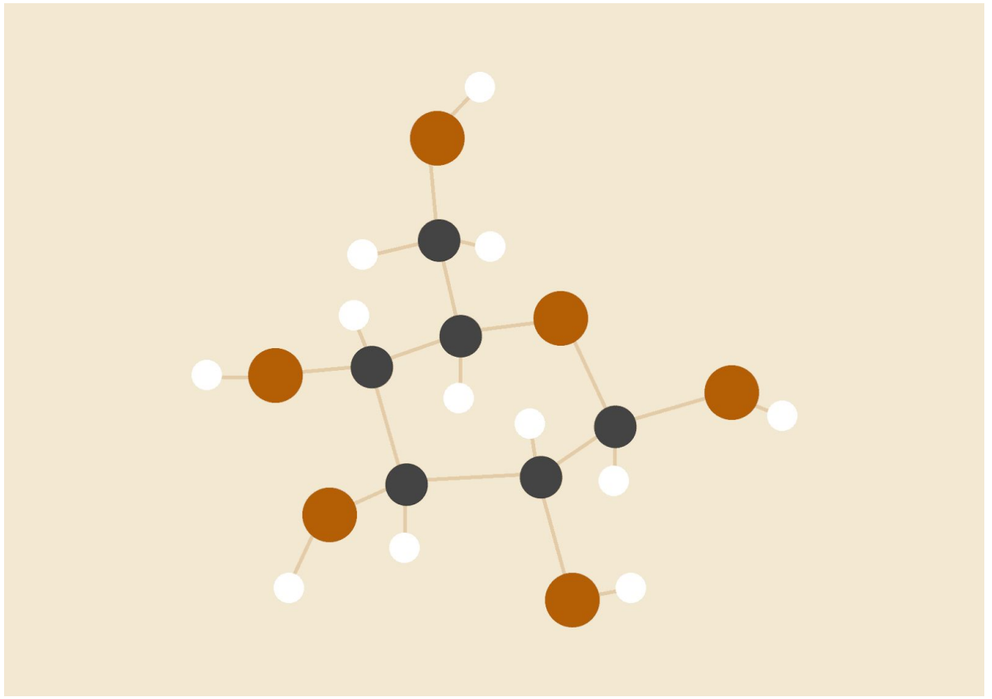
TORONTO TWITTER ANALYSIS

A summary of Toronto Twitter Data to determine if there is anything of value



Jennifer Johnson

27.04.2019
CSDA1050: Capstone

INTRODUCTION

This document summarizes the complete process and analysis of Toronto Twitter data. The purpose of this analysis was to initially find if there was a relationship between Twitter data in comparison to Housing and Rental prices in the area, if some neighbourhoods cost more had different twitter sentiment or activity. Unfortunately due to the limitations to the data on both sides there was no way to conclusively determine. So instead the analysis came to be what are people in Toronto talking about on Twitter and are there some commonalities. This was not a way to solve the world's problems with Twitter but to see if there was any useful data that could provide any insight into Toronto Life.

STREAMING TWITTER DATA

To start gathering the data for analysis, a program was created to collect streaming Twitter data. The tool of choice to use was Python and used the existing library Tweepy (https://www.tweepy.org/), an easy to use library to access the Twitter API. The program, twitter_stream_post.py, opened a Listener to start the streaming and applies a filter to capture Tweets within the defined location Polygon. This is the defined polygon area that captures Toronto and surrounding GTA, from Oakville to Whitby.

```
stream.filter(locations=[-79.639319,43.403221, -78.905820,43.855401])
```

Unfortunately, with this polygon it also captures some of New York due to the polygon area. Future parts of this analysis filter out the New York data.

HOW MUCH DATA COLLECTED

Data was collected from March 17, 2019 and April 10, 2019. While working out some bugs at the beginning and some technical difficulties there were some breaks in the data.

CLEANING AND PARSING THE DATA

After streaming the data, some cleaning, flattening, and parsing of the data was done, as only some of the data was used for analysis. The program, twitter_clean_data_v2.py, took all the files from streaming Twitter and splits the data into two files, Users and Tweets. These were split to perform separate analysis, but the focus of this analysis was on Tweets. The following cleaning was done on the data and lists the fields in the dataframe

USER DATAFRAME

FIELD NAME	TWEETS FILE MAPPING	CLEANSING DONE
id	tweet['user']['id']	
user_name	tweet['user']['name']	EncodeDecode to 'utf-8' and cleanEmoji to text
screen_name	tweet['user']['screen_name']	
location	tweet['user']['location']	EncodeDecode to 'utf-8'
description	tweet['user']['description']	EncodeDecode to 'utf-8' and cleanEmoji to text
protected	tweet['user']['protected']	

verified	tweet['user']['verified']	
followers_count	tweet['user']['followers_count']	
friends_count	tweet['user']['friends_count']	
listed_count	tweet['user']['listed_count']	
favourites_count	tweet['user']['favourites_count']	
statuses_count	tweet['user']['statuses_count']	
created_date	tweet['user']['created_at']	
geo_enabled	tweet['user']['geo_enabled']	
language	tweet['user']['lang']	
contributors_enabled	tweet['user']['contributors_enabled']	

TWEETS DATAFRAME

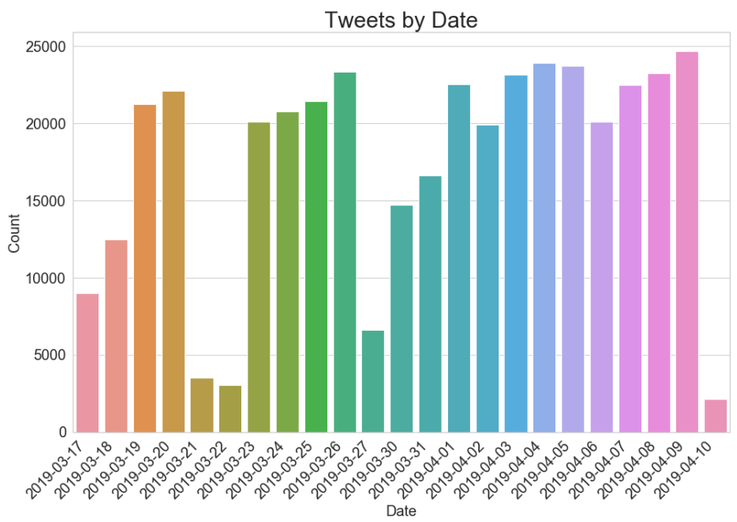
FIELD NAME	TWEETS FILE MAPPING	CLEANSING DONE
id	tweet[‘id’]	
tweet_date	tweet[‘timestamp_ms’]	
tweet_text	tweet[‘extended_tweet’][‘full_text’] tweet[‘text’]	Use extended_tweet text when populated as this will have complete tweet message, cleansing applied to apply unicode of ‘utf-8’ and convert emoji characters to text
latitude	tweet[‘coordinates’][‘coordinates’](0)	
longitude	tweet[‘coordinates’][‘coordinates’](1)	
source	tweet[‘source’]	Apply some regular expressions to remove special characters around the URL. Source represents the URL of the source tweets
hashtags	tweet['entities']['hashtags']	Apply function to create clean list

mentions	tweet['entities']['user_mentions']	Apply function to create clean list
reply_to_user	tweet['in_reply_to_screen_name']	
reply_to_status	tweet['in_reply_to_status_id']	
quote_status	tweet['is_quote_status']	
language	tweet['lang']	
place_type	tweet['place']['place_type']	
place_name	tweet['place']['full_name']	
place_country	tweet['place']['country']	
place_bbtype	tweet['place']['bounding_box']['type']	
place_bbcoordinates	tweet['place']['bounding_box']['coordinates']	
sensitive	tweet['possibly_sensitive']	
quoted_text	tweet['quoted_status']['text']	cleansing applied to apply unicode of 'utf-8'
quoted_id	tweet['quoted_status_id']	
user_id	tweet['user']['id']	
user_name	tweet['user']['screen_name']	
emojis	tweet['extended_tweet']['full_text'] tweet['text']	Using function, findEmoji, find all the emoji icons in the tweet text and put in a list for future analysis
sentiment	tweet['extended_tweet']['full_text'] tweet['text']	Using function, getSentiment, calculate the sentiment of the text

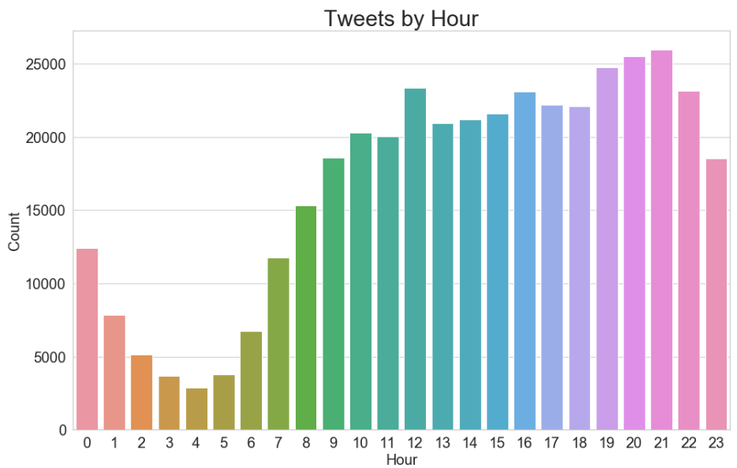
EXPLORATORY DATA ANALYSIS

This section explores the analysis of all the tweets. The analysis done initially, Tweet_Analysis_v1.ipynb, looked at all Twitter data streamed include tweets from areas of New York, further analysis v2 and v3 captured Toronto Only and Geotagged Toronto Only respectively.

All Polygon Tweets (400928, 24)

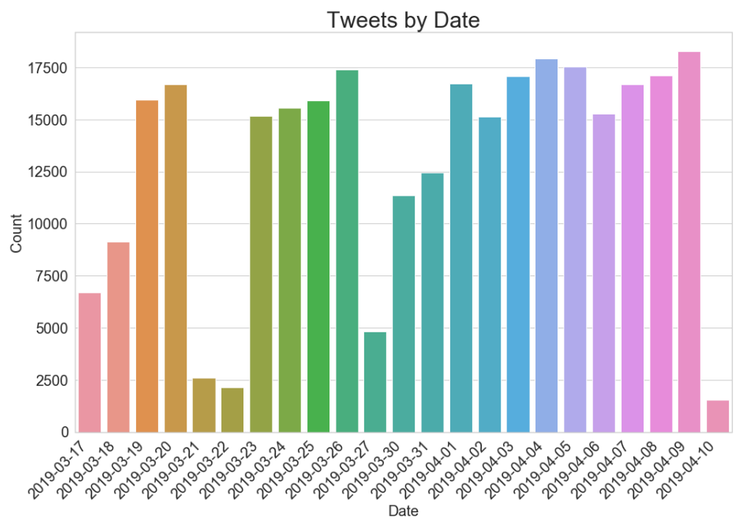


On complete days there is between 20,000 and 25,000 tweets a day within the polygon area. March 17, 18, 21, 22, 27, 30, 31, and April 10 all have partial data collected

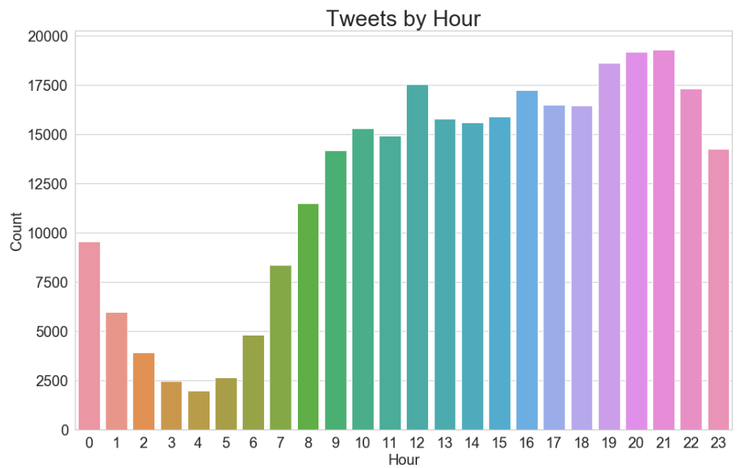


After applying timezone change on datetime of tweet, the tweets by Hour of Day show more accurately that tweets tends to reduce after 12AM, dropping to the lowest tweets at 4am, with a peak time of 7-9pm

Toronto (GTA) Only (299111, 24)

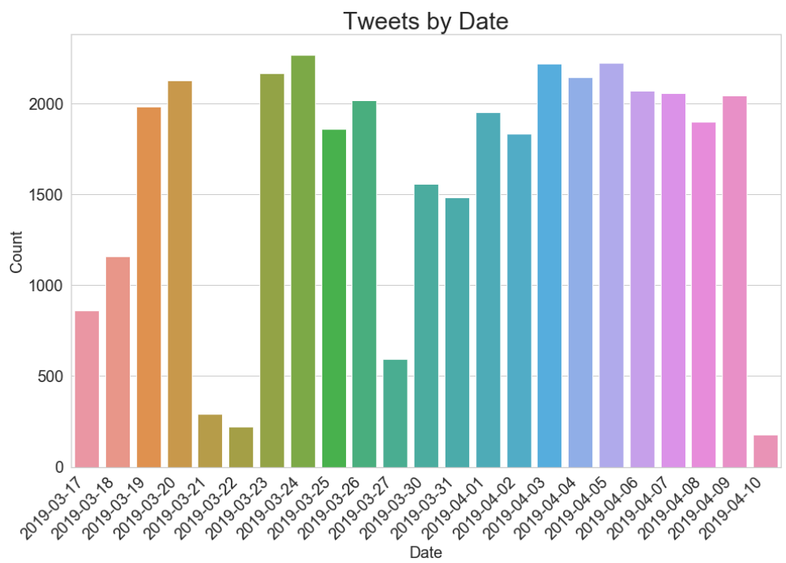


On complete days there are between 15,000 and 18,000 tweets per day showing that a quarter of the tweets streamed came from outside of Canada (Toronto)

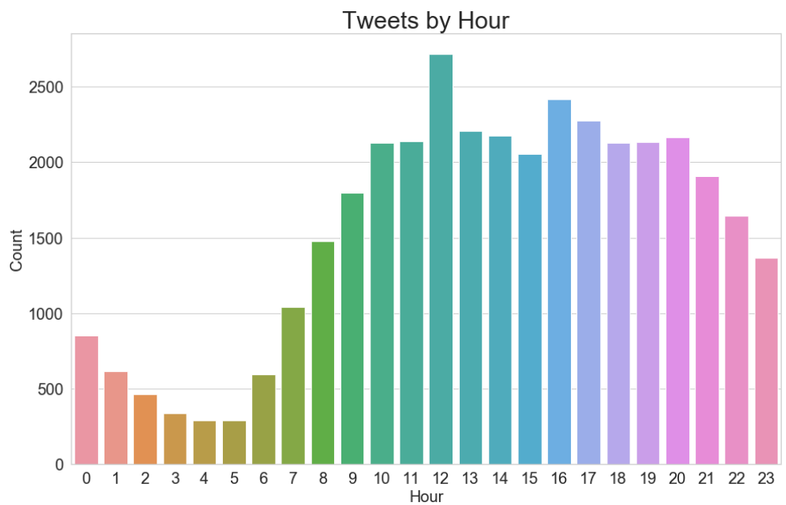


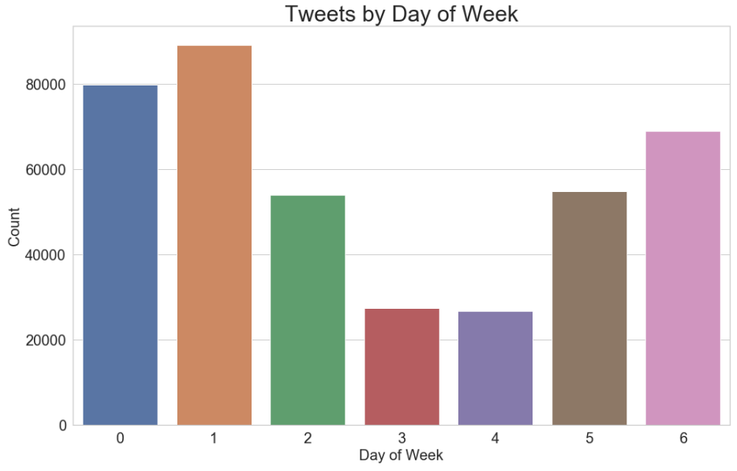
There is no real difference on the time of tweets for just Toronto

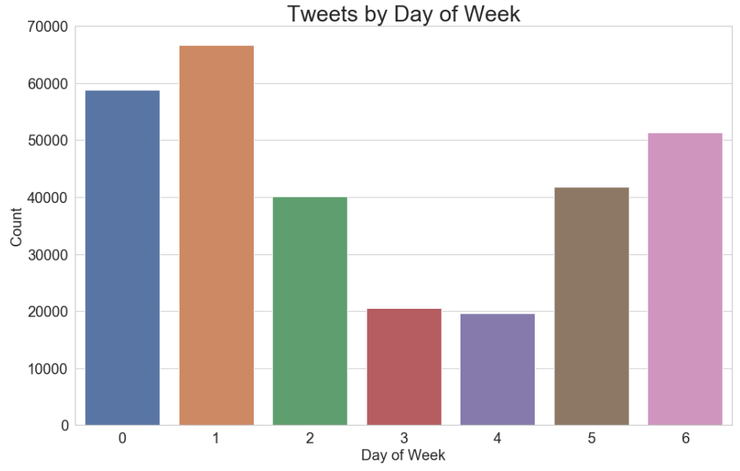
Geo Tagged Only (37220, 24)

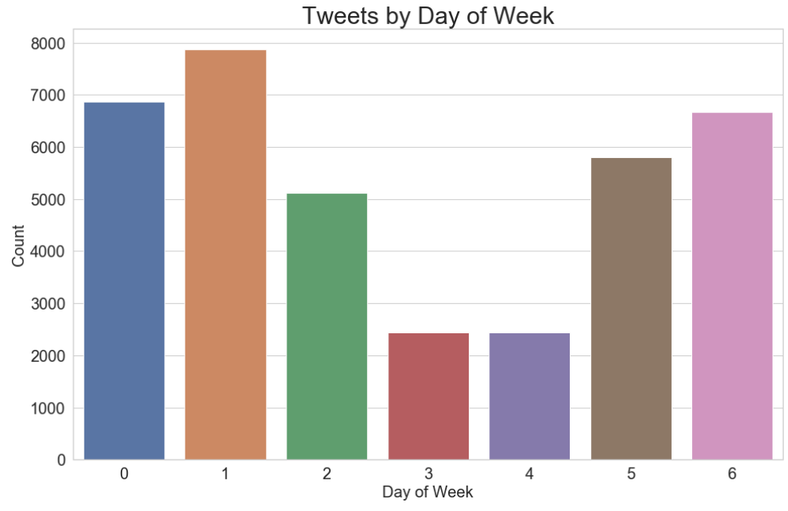


Summary of results

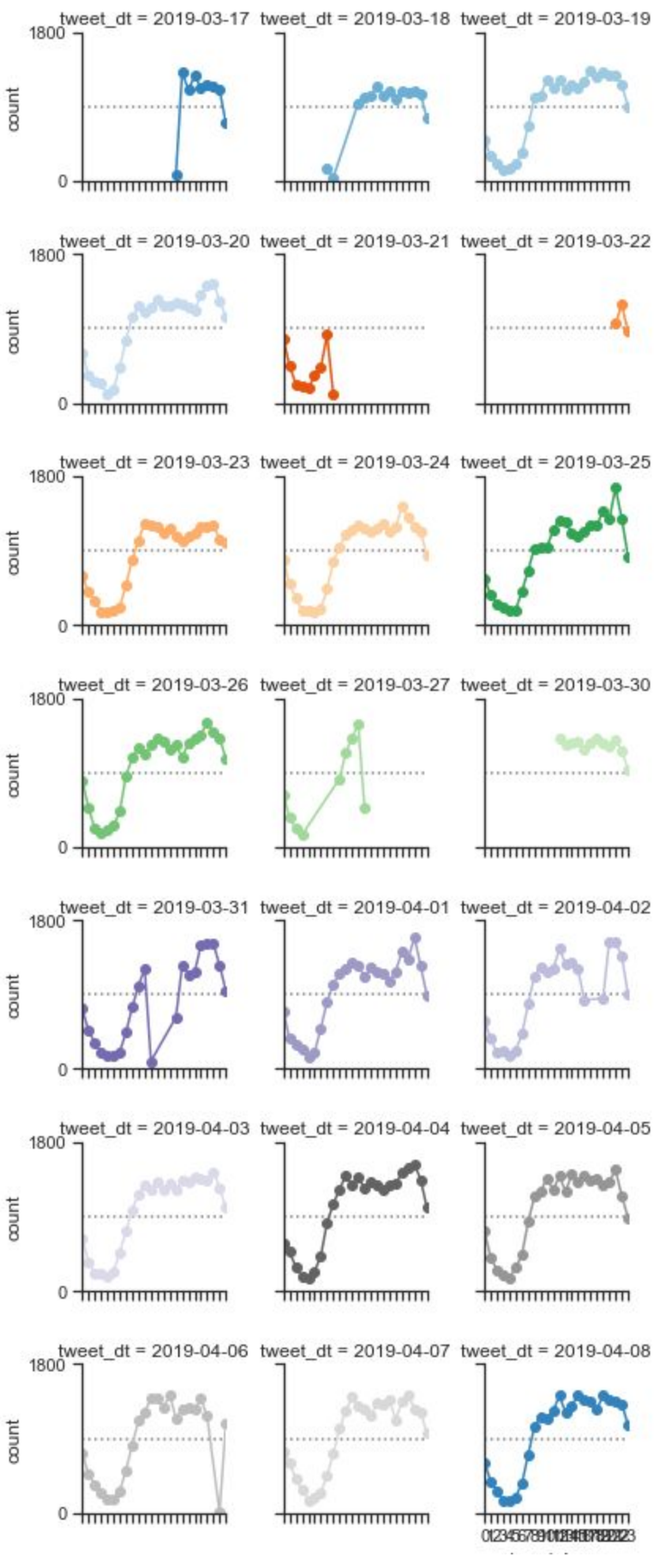


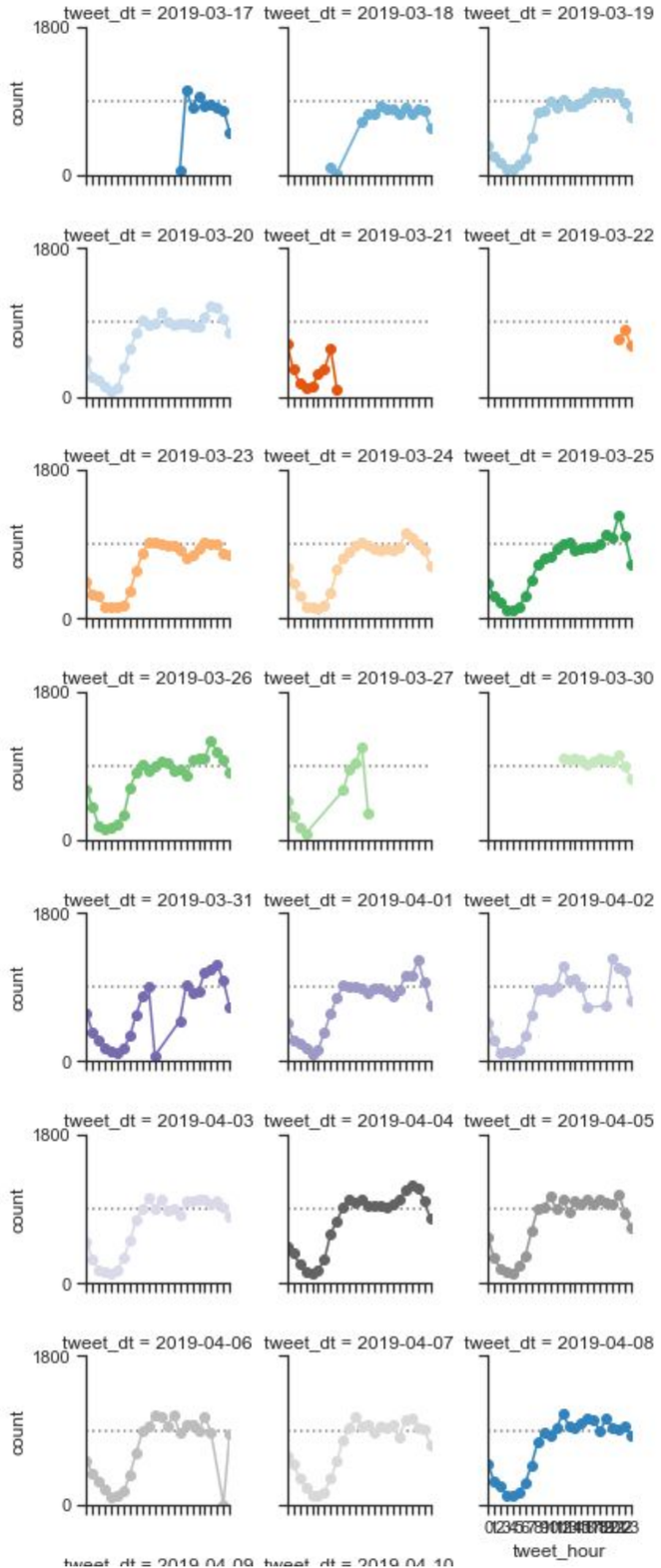


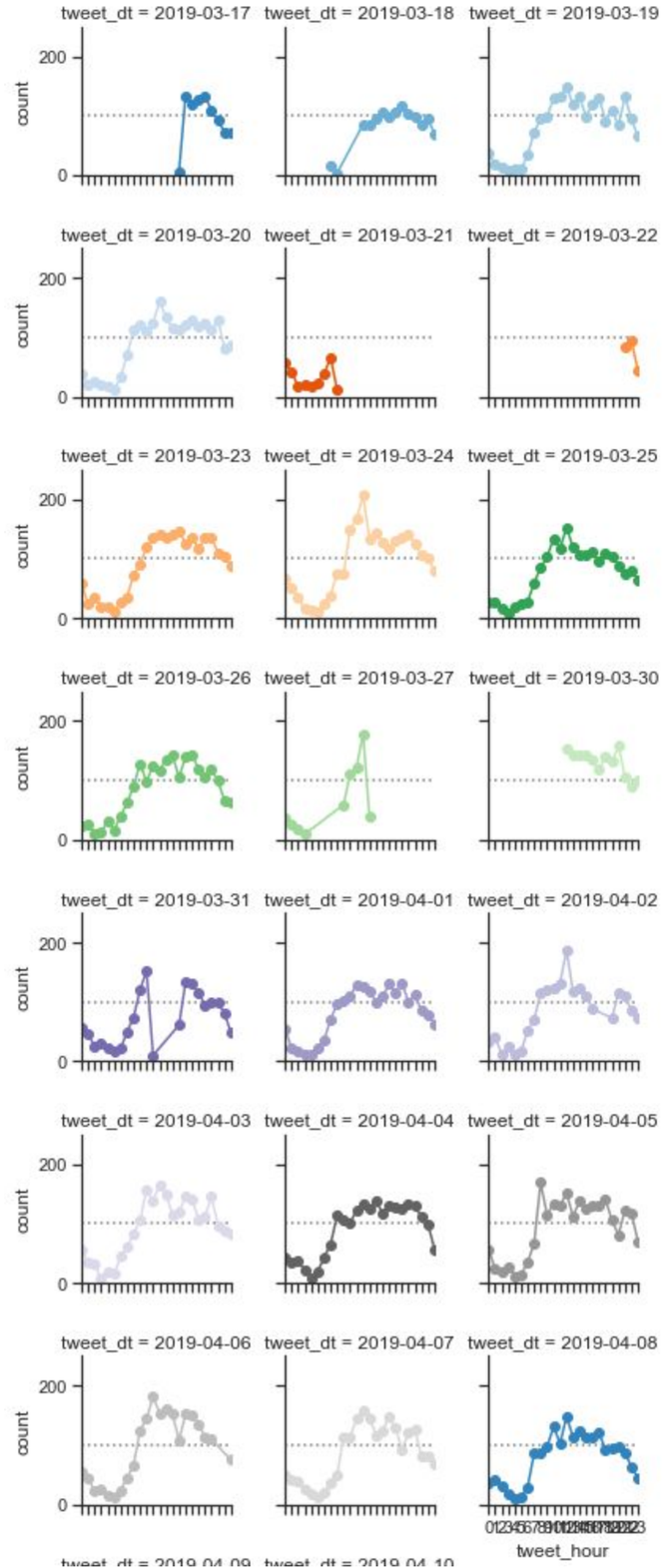




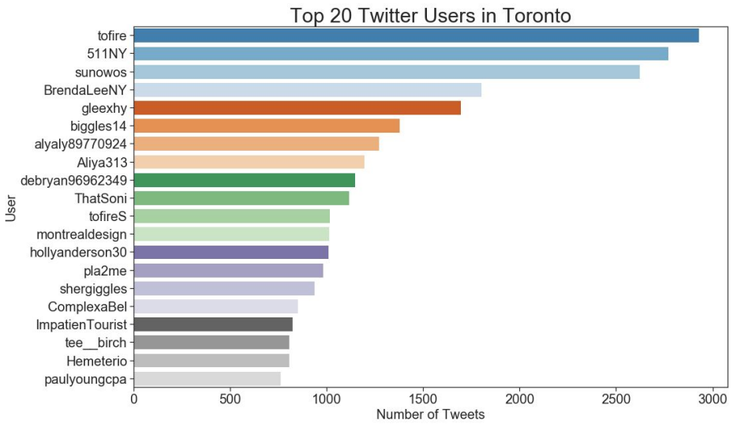
Tweets by Day of Week (Monday to Sunday) shows the lowest amount of tweets collected in the middle of the week. This is due to the breaks in the load of the data and on the days of class, the data was not being streamed and collected.


tweet_dt = 2019-03-17
tweet_dt = 2019-03-18
tweet_dt = 2019-03-19
tweet_dt = 2019-03-20
tweet_dt = 2019-03-21
tweet_dt = 2019-03-22
tweet_dt = 2019-03-23
tweet_dt = 2019-03-24
tweet_dt = 2019-03-25
tweet_dt = 2019-03-26
tweet_dt = 2019-03-27
tweet_dt = 2019-03-30
tweet_dt = 2019-03-31
tweet_dt = 2019-04-01
tweet_dt = 2019-04-02
tweet_dt = 2019-04-03
tweet_dt = 2019-04-04
tweet_dt = 2019-04-05
tweet_dt = 2019-04-06
tweet_dt = 2019-04-07
tweet_dt = 2019-04-08
count
1800
0
tweet_hour


tweet_dt = 2019-03-17
tweet_dt = 2019-03-18
tweet_dt = 2019-03-19
tweet_dt = 2019-03-20
tweet_dt = 2019-03-21
tweet_dt = 2019-03-22
tweet_dt = 2019-03-23
tweet_dt = 2019-03-24
tweet_dt = 2019-03-25
tweet_dt = 2019-03-26
tweet_dt = 2019-03-27
tweet_dt = 2019-03-30
tweet_dt = 2019-03-31
tweet_dt = 2019-04-01
tweet_dt = 2019-04-02
tweet_dt = 2019-04-03
tweet_dt = 2019-04-04
tweet_dt = 2019-04-05
tweet_dt = 2019-04-06
tweet_dt = 2019-04-07
tweet_dt = 2019-04-08
count
1800
0
tweet_hour

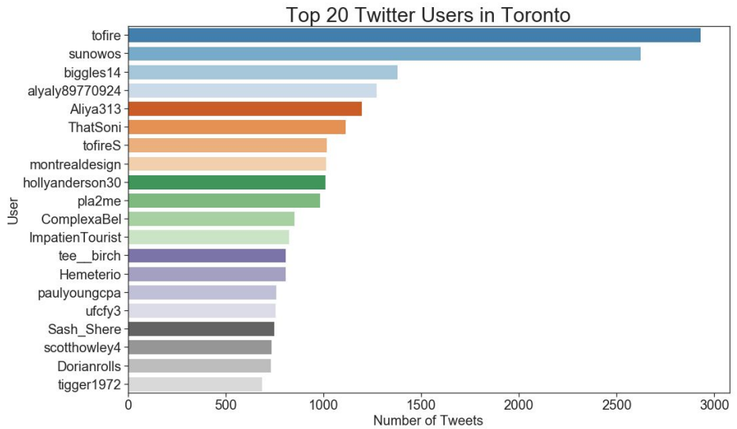

tweet_dt = 2019-03-17
tweet_dt = 2019-03-18
tweet_dt = 2019-03-19
tweet_dt = 2019-03-20
tweet_dt = 2019-03-21
tweet_dt = 2019-03-22
tweet_dt = 2019-03-23
tweet_dt = 2019-03-24
tweet_dt = 2019-03-25
tweet_dt = 2019-03-26
tweet_dt = 2019-03-27
tweet_dt = 2019-03-30
tweet_dt = 2019-03-31
tweet_dt = 2019-04-01
tweet_dt = 2019-04-02
tweet_dt = 2019-04-03
tweet_dt = 2019-04-04
tweet_dt = 2019-04-05
tweet_dt = 2019-04-06
tweet_dt = 2019-04-07
tweet_dt = 2019-04-08
count
200
0
tweet_hour

Tweets by date and Hour of date shows more clearly the breaks in the data when data was not collected

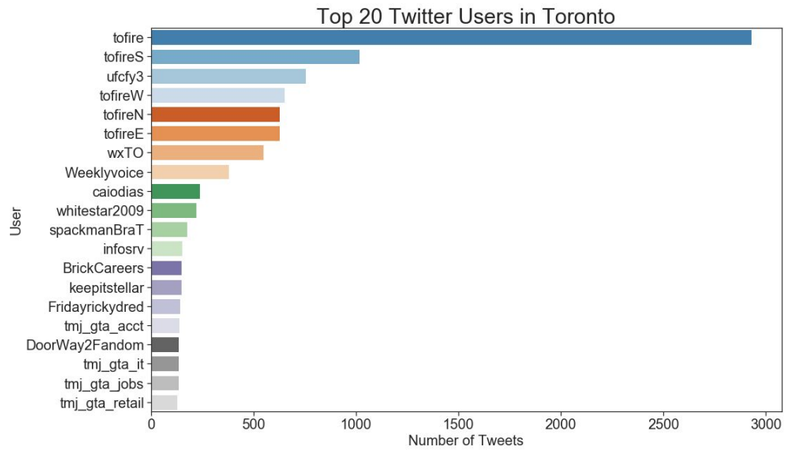


The top users shows Toronto Fire, New York 511. Emergency services seem to post the most

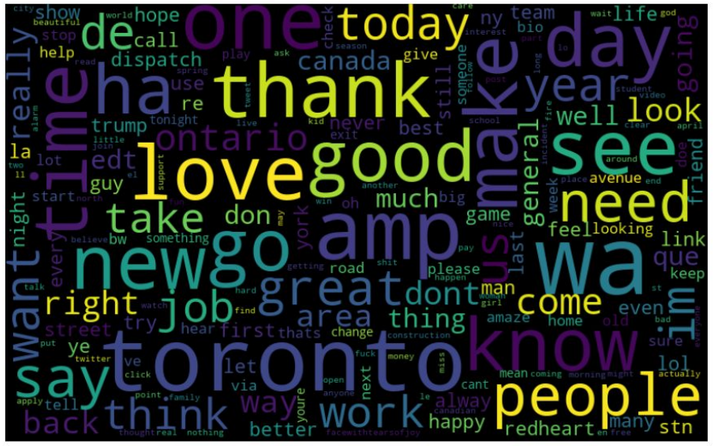
Toronto data shows around 900 tweets per hour were captured during the day 9am to 11pm.

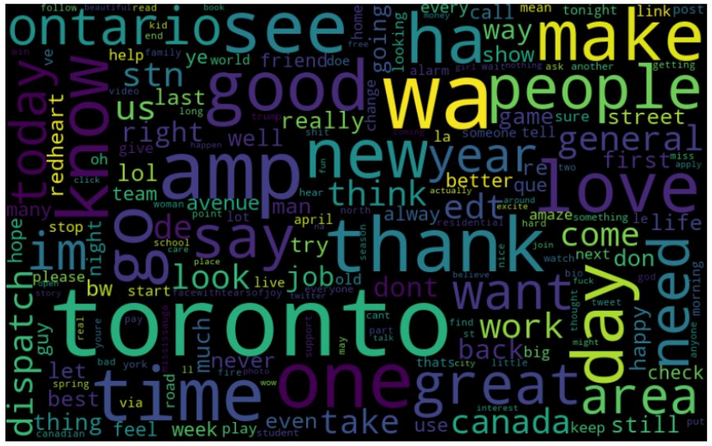


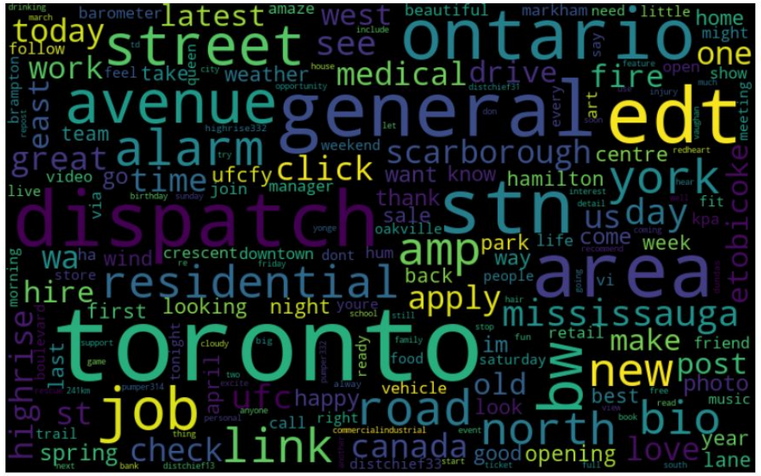
The top users removed 511NY, BrendaLeeNY, gleexhy, debryan96962349, shergiggles with filter; ufcfy3, Sash_Shere, scotthowley4, Dorianrolls, tigger1972 added.

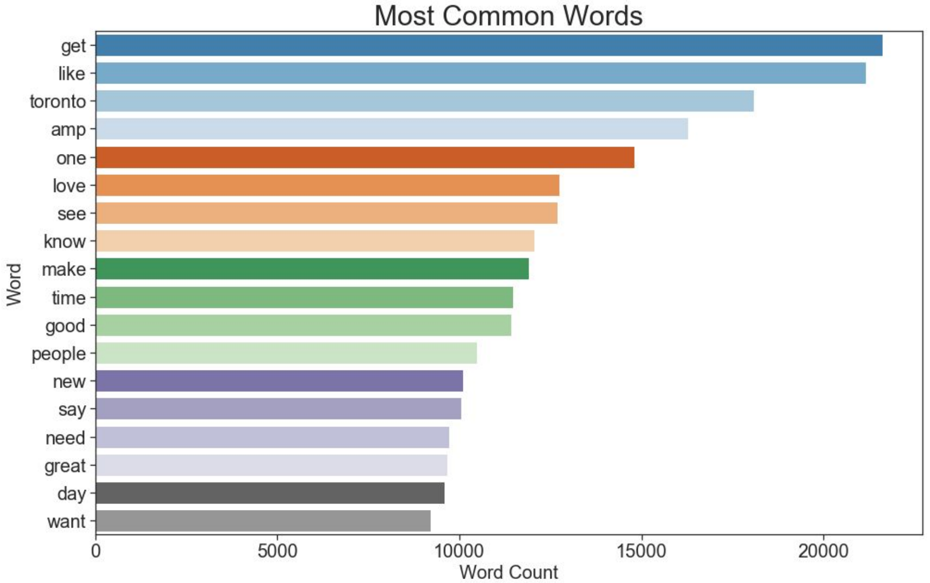


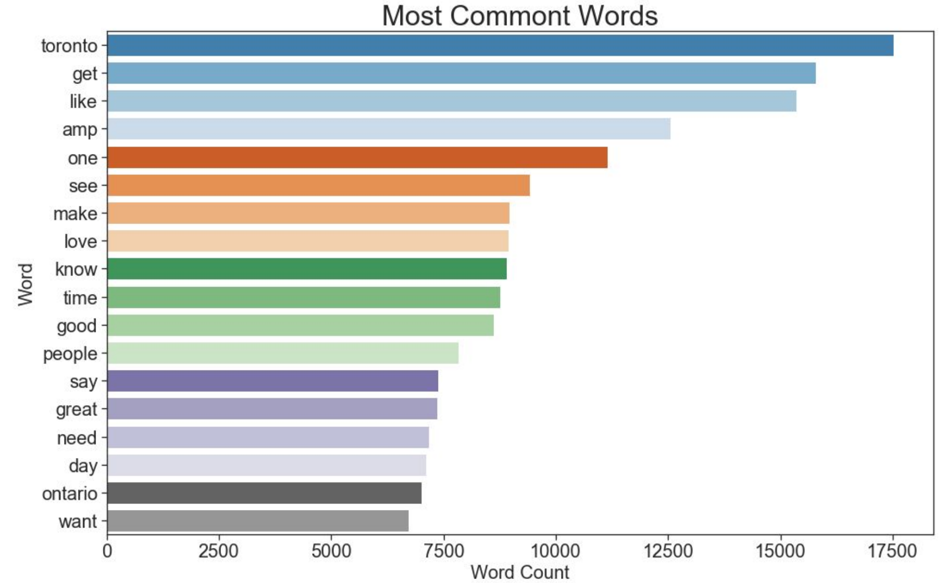
Word Cloud of Top words used in Tweets

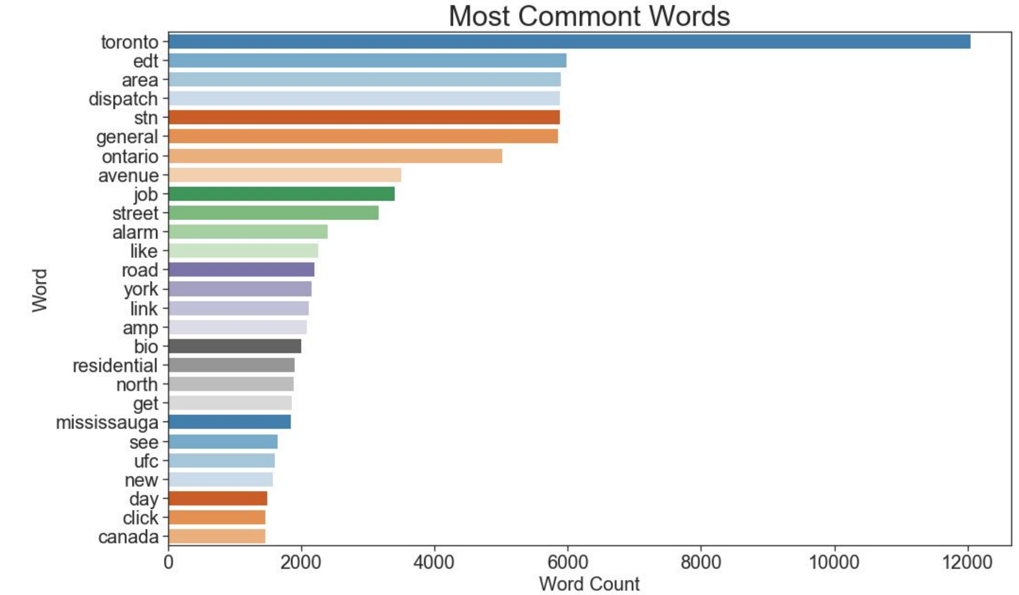




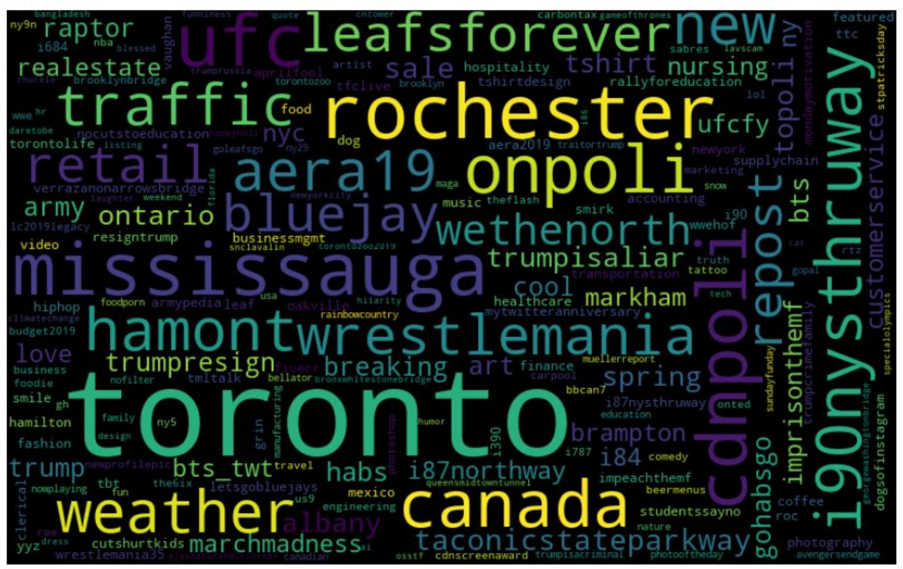


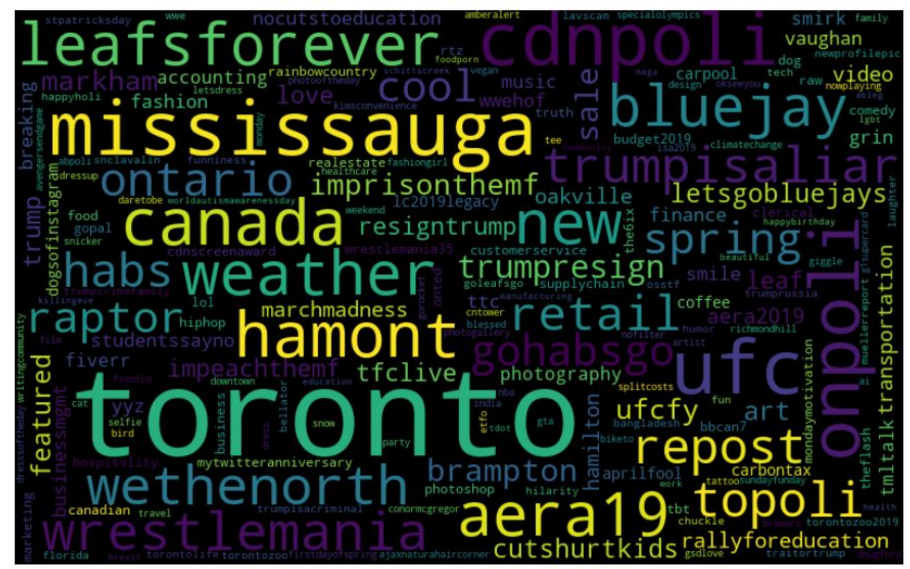


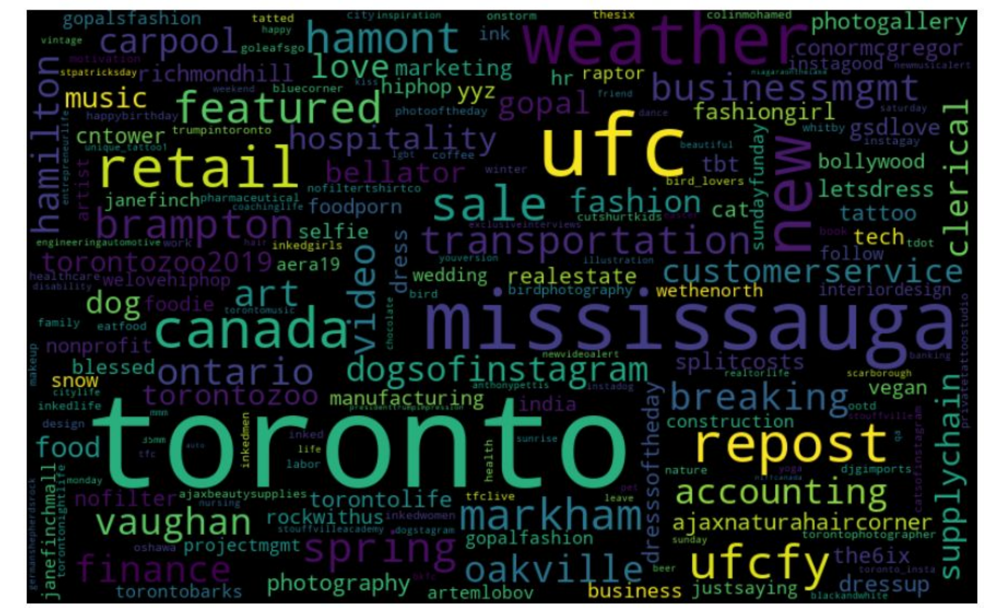


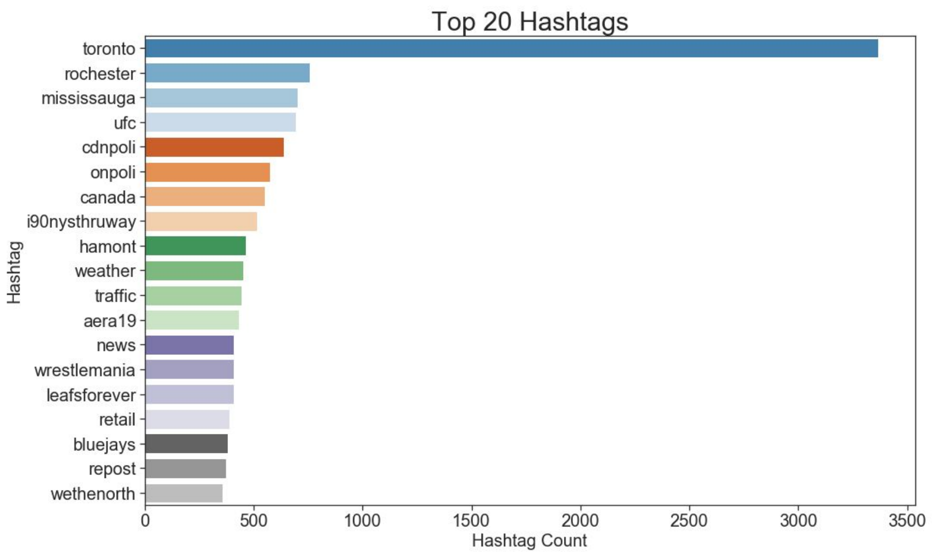


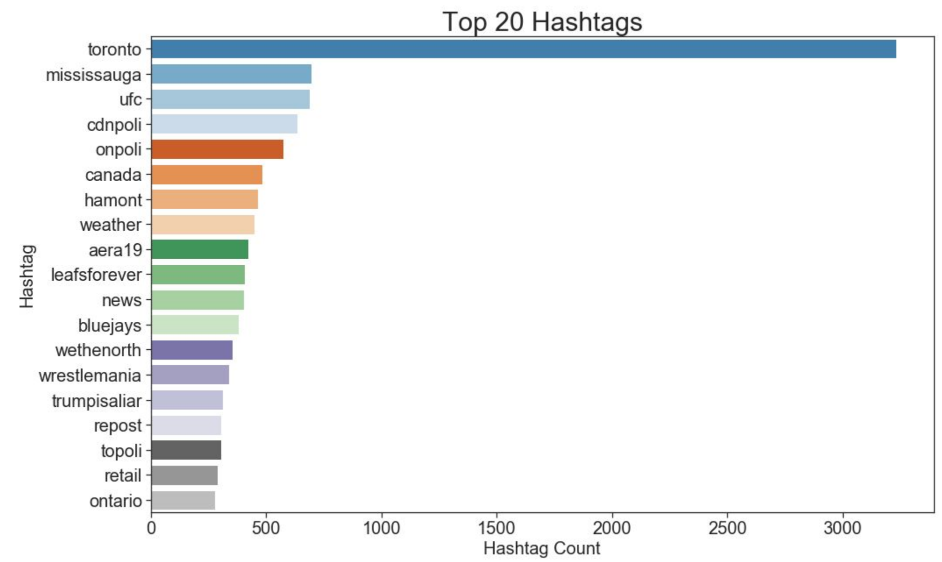
Word cloud of top Hashtags

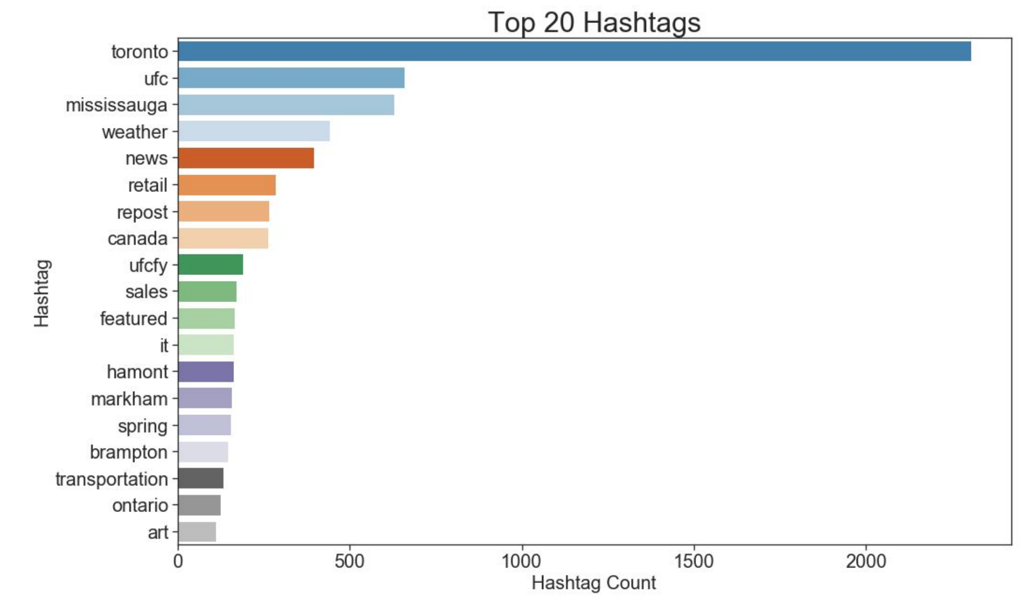


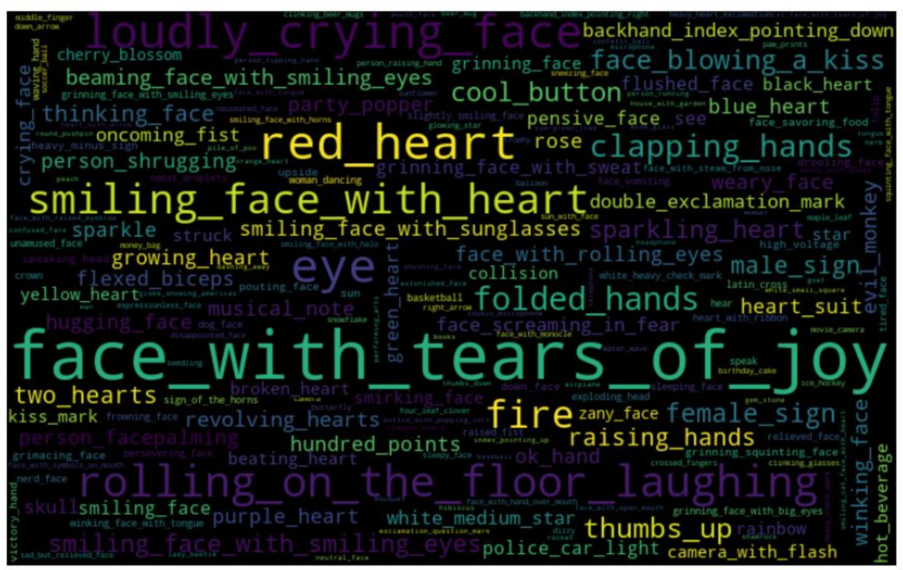


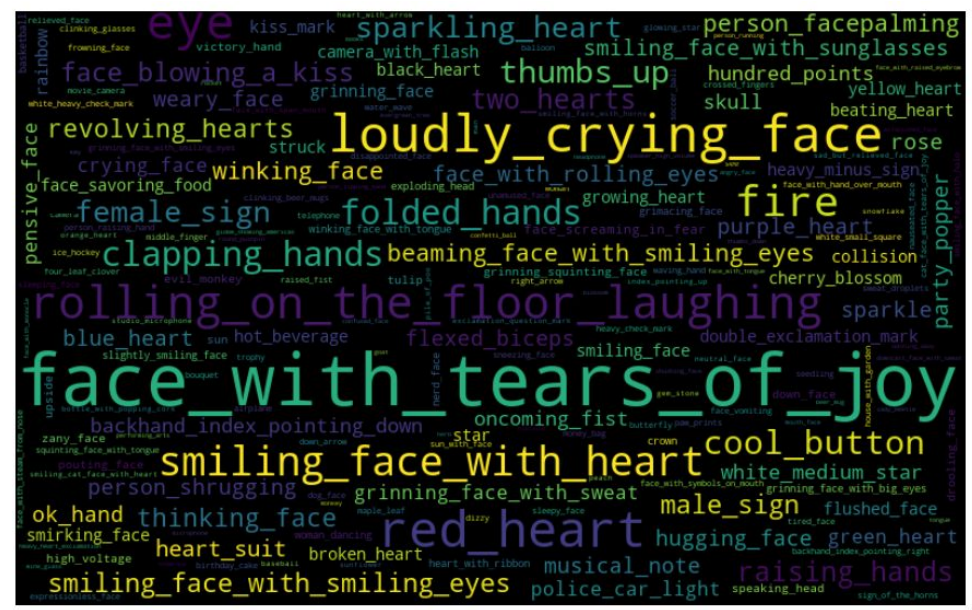


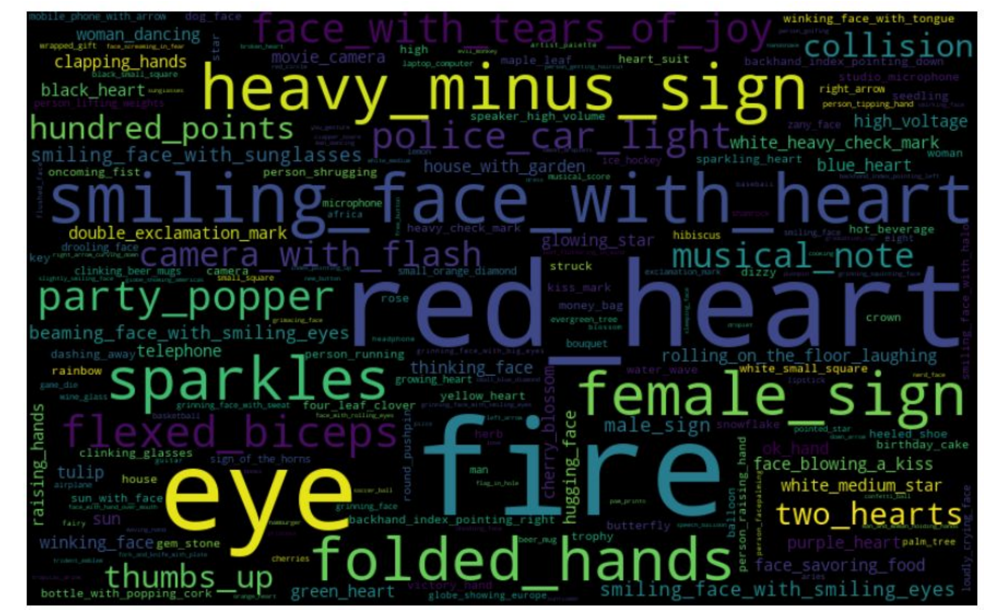

Top 20 Hashtags
toronto
rochester
mississauga
ufc
cdnpoli
onpoli
canada
i90nysthruway
hamont
weather
traffic
aera19
news
wrestlemania
leafsforever
retail
bluejays
repost
wethenorth
Hashtag
0
500
1000
1500
2000
2500
3000
3500
Hashtag Count

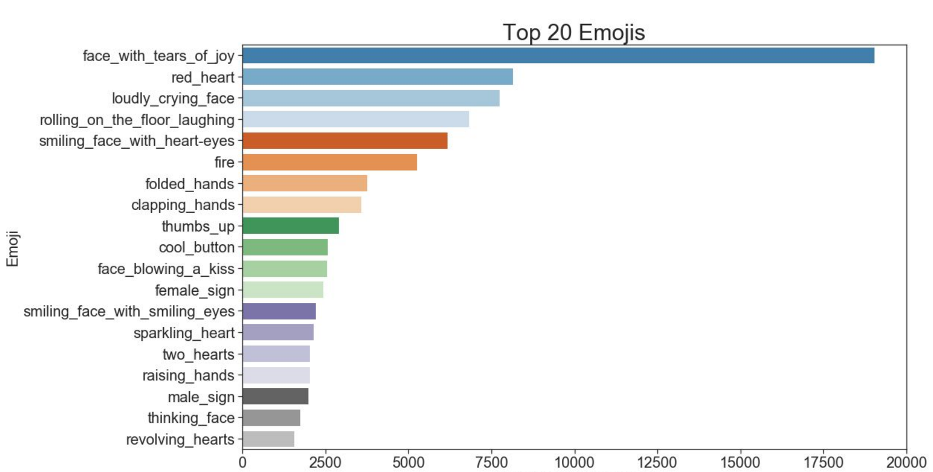

Top 20 Hashtags
toronto
mississauga
ufc
cdnpoli
onpoli
canada
hamont
weather
aera19
leafsforever
news
bluejays
wethenorth
wrestlemania
trumpisaliar
repost
topoli
retail
ontario
Hashtag
0
500
1000
1500
2000
2500
3000
Hashtag Count

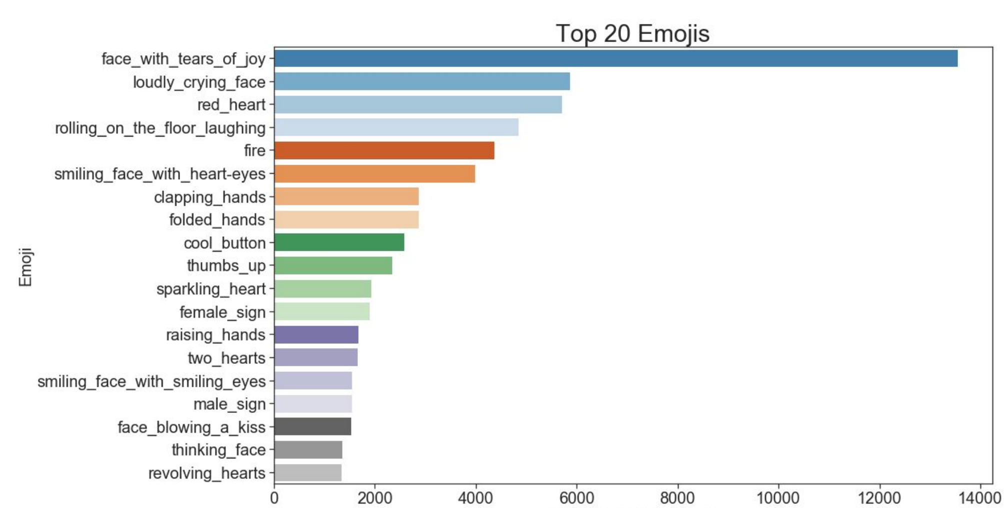

Top 20 Hashtags
toronto
ufc
mississauga
weather
news
retail
repost
canada
ufcfy
sales
featured
it
hamont
markham
spring
brampton
transportation
ontario
art
Hashtag
0
500
1000
1500
2000
Hashtag Count

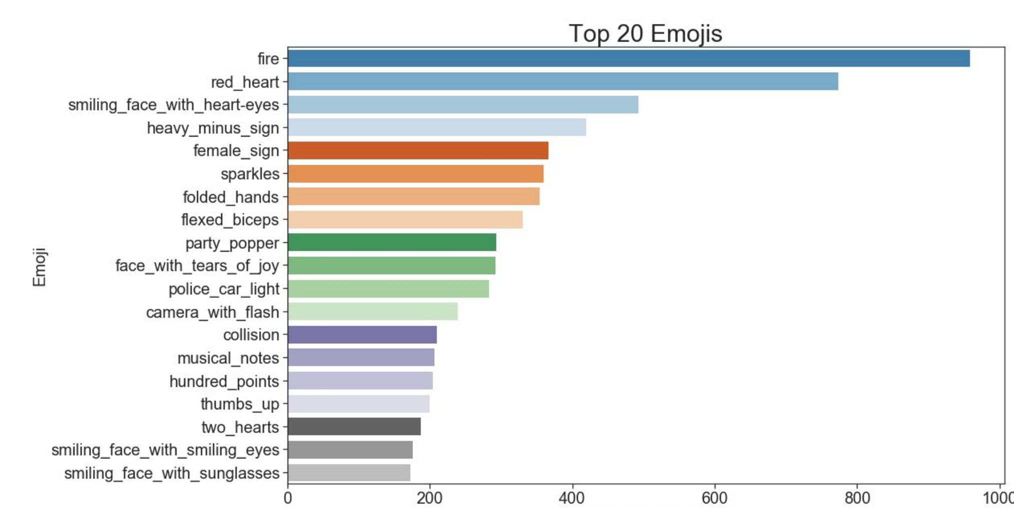

loudly_crying_face
face_with_tears_of_joy
red_heart
smiling_face_with_heart
rolling_on_the_floor_laughing
clapping_hands
folded_hands
fire
eye

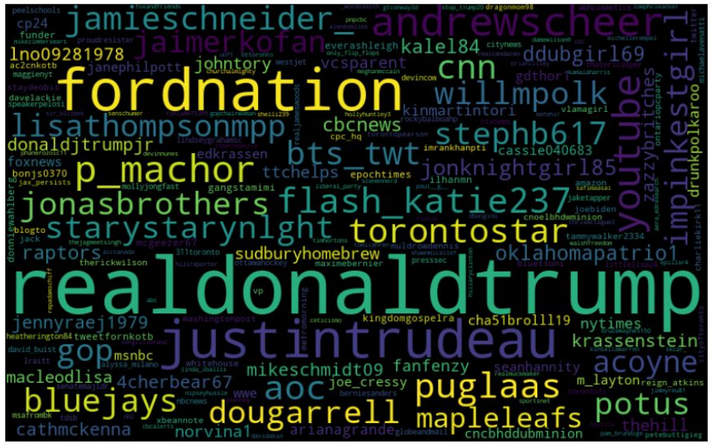

face_with_tears_of_joy
loudly_crying_face
rolling_on_the_floor_laughing
smiling_face_with_heart
red_heart
folded_hands
clapping_hands
fire

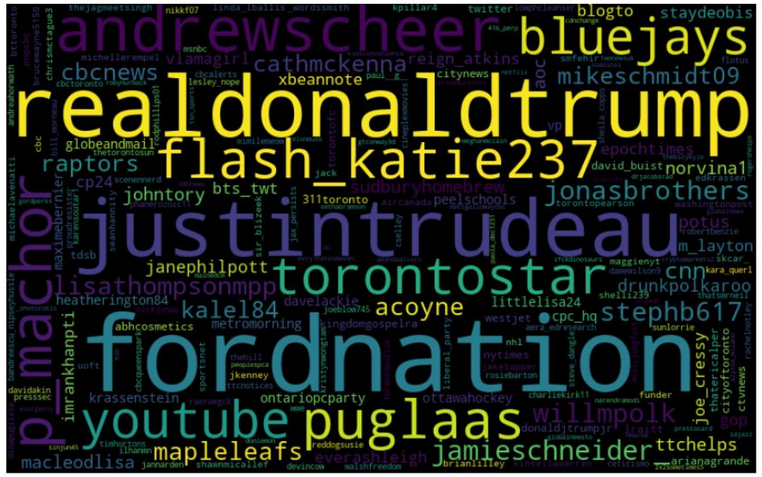

red_heart
smiling_face_with_heart
heavy_minus_sign
fire
eye
folded_hands
sparkles
female_sign
flexed_biceps
party_popper

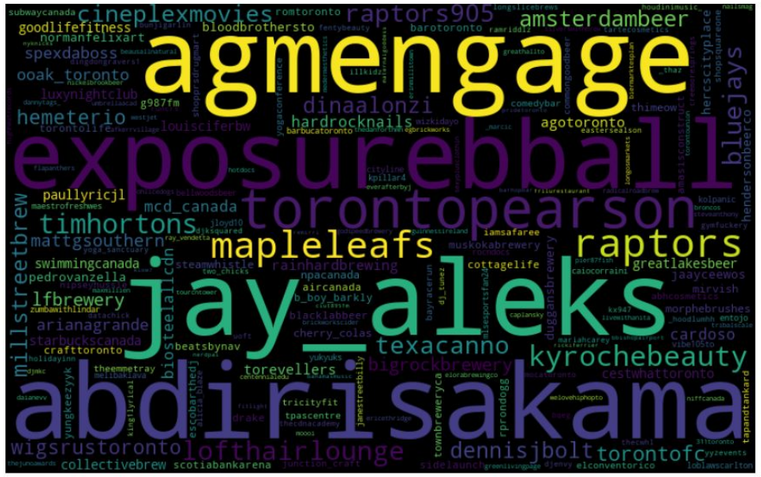

Top 20 Emojis
face_with_tears_of_joy
red_heart
loudly_crying_face
rolling_on_the_floor_laughing
smiling_face_with_heart-eyes
fire
folded_hands
clapping_hands
thumbs_up
cool_button
face_blowing_a_kiss
female_sign
smiling_face_with_smiling_eyes
sparkling_heart
two_hearts
raising_hands
male_sign
thinking_face
revolving_hearts
Emoji
0
2500
5000
7500
10000
12500
15000
17500
20000
Number of Times Used

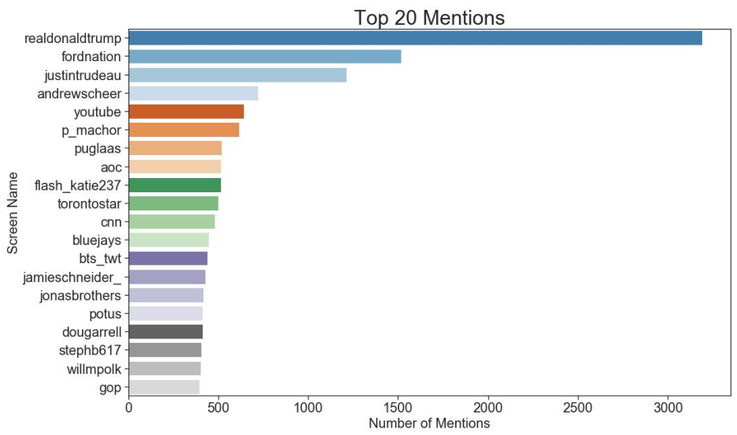

Top 20 Emojis
face_with_tears_of_joy
loudly_crying_face
red_heart
rolling_on_the_floor_laughing
fire
smiling_face_with_heart-eyes
clapping_hands
folded_hands
cool_button
thumbs_up
sparkling_heart
female_sign
raising_hands
two_hearts
smiling_face_with_smiling_eyes
male_sign
face_blowing_a_kiss
thinking_face
revolving_hearts
Emoji
0
2000
4000
6000
8000
10000
12000
14000
Number of Times Used

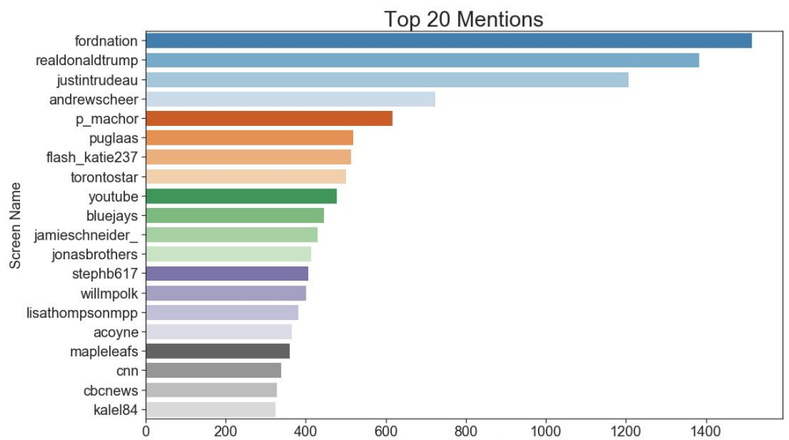

Top 20 Emojis
fire
red_heart
smiling_face_with_heart-eyes
heavy_minus_sign
female_sign
sparkles
folded_hands
flexed_biceps
party_popper
face_with_tears_of_joy
police_car_light
camera_with_flash
collision
musical_notes
hundred_points
thumbs_up
two_hearts
smiling_face_with_smiling_eyes
smiling_face_with_sunglasses
Emoji
0
200
400
600
800
1000
Number of Times Used

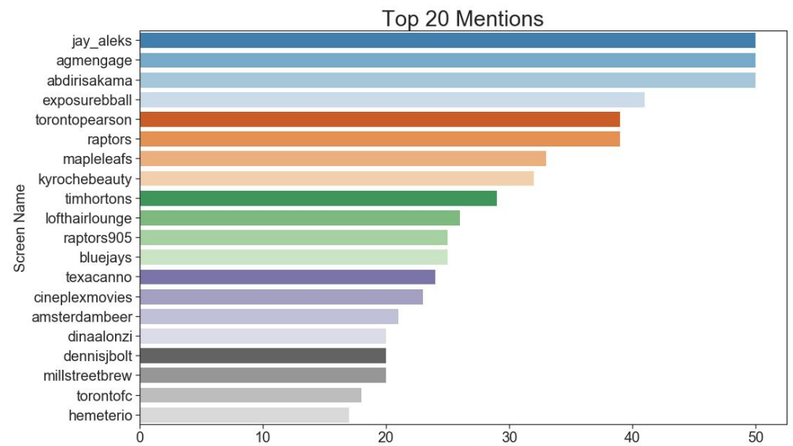










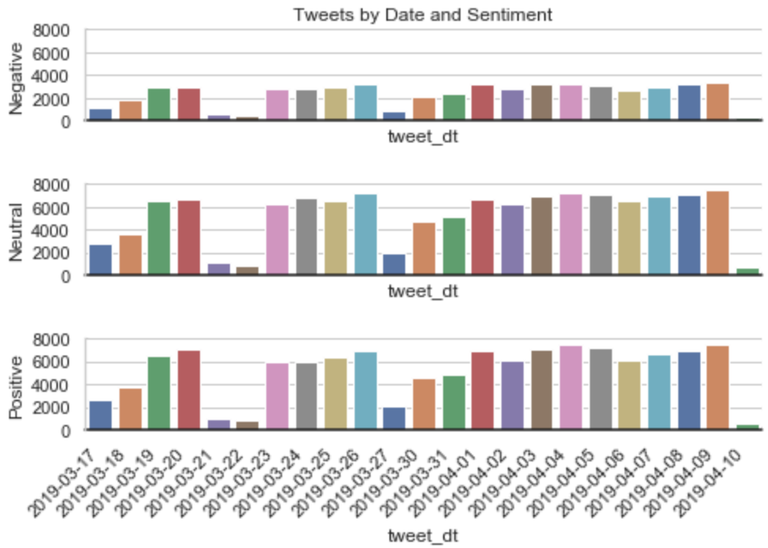


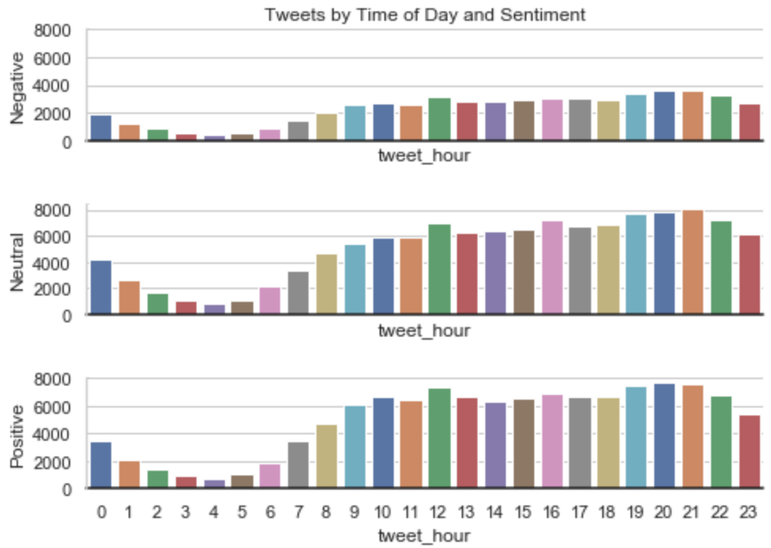
SENTIMENT ANALYSIS

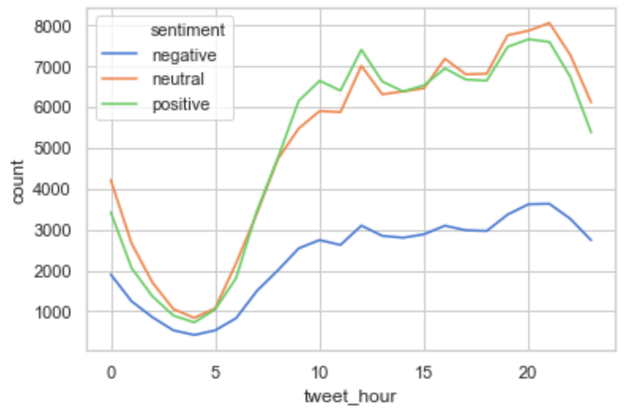
Sentiment was calculated during the parsing phase, after the data was streamed and before any analysis was done. Sentiment was calculated using `SentimentIntensityAnalyzer` from the `vaderSentiment.vaderSentiment` python library. See the following article for more information about this library and its use with social media, https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f.

After doing all the analysis and multiple ways of doing data cleansing in the analysis, it is possible that the sentiment could have been different if calculated later.

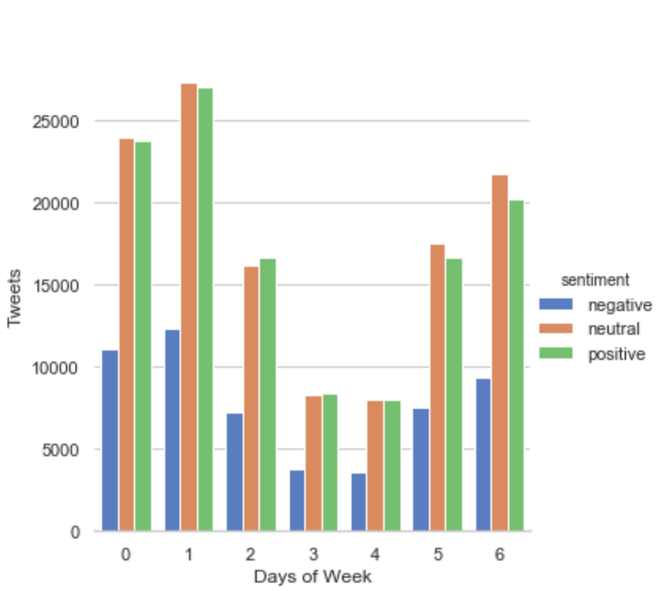
Analysis 1

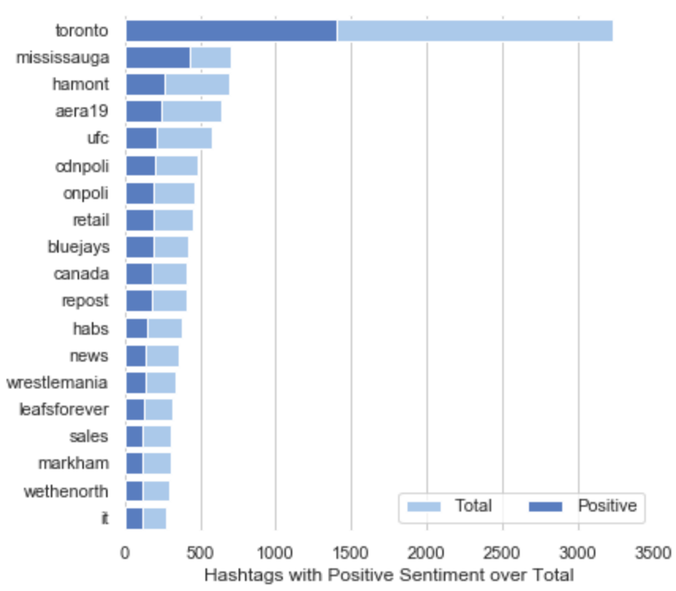


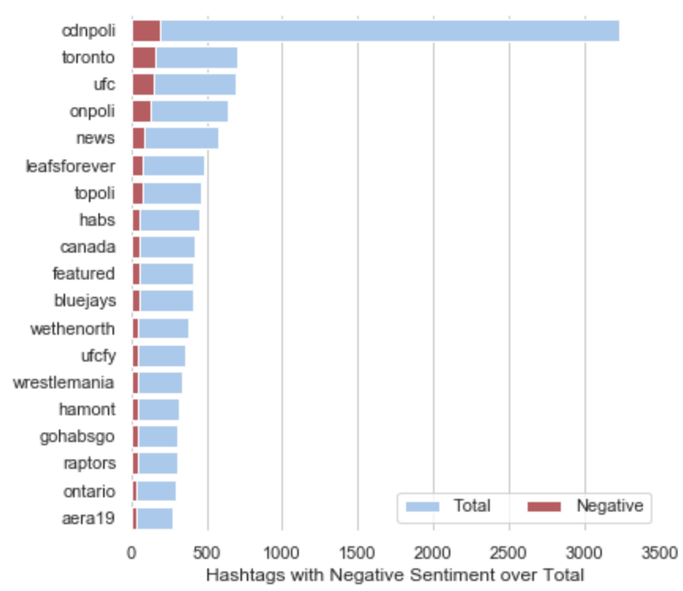




The above graphs were created to try and see if sentiment changed on certain dates and times. Are there more negative or positive tweets? There are more positive tweets. Are there different patterns in time of day of date, no difference in pattern between the different sentiments.

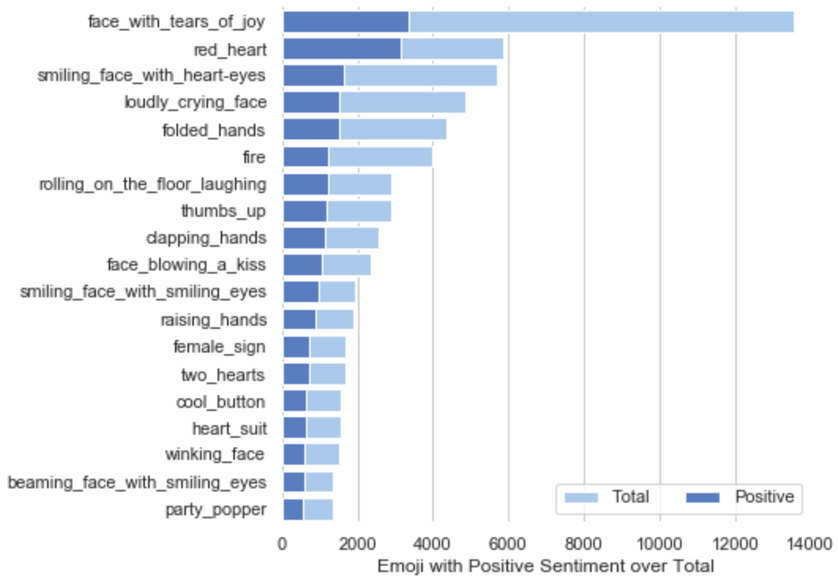


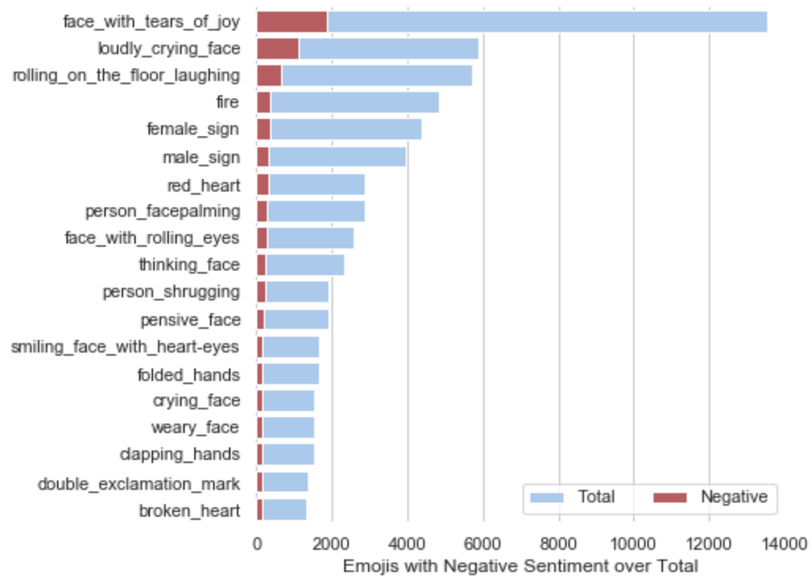




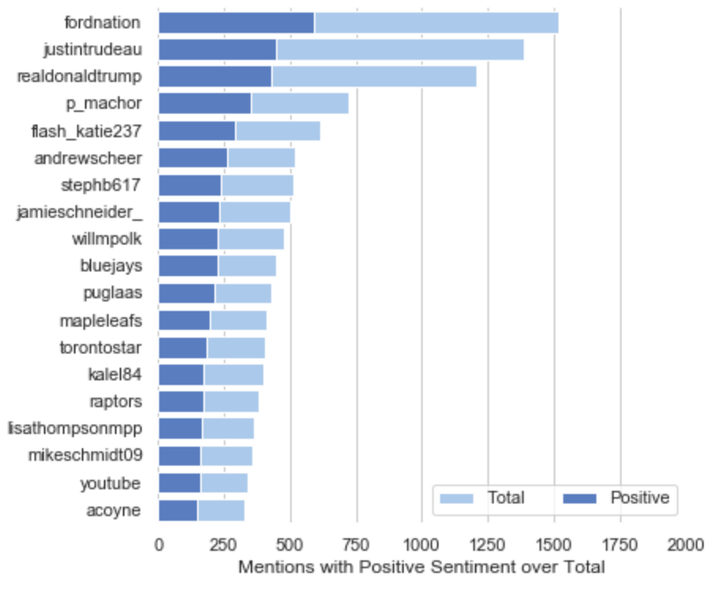
The above graphs shows tweets by day of week, to make this graph better it should be average tweets per day to accommodate the lower days. The second and third graphs shown above compare the positive and

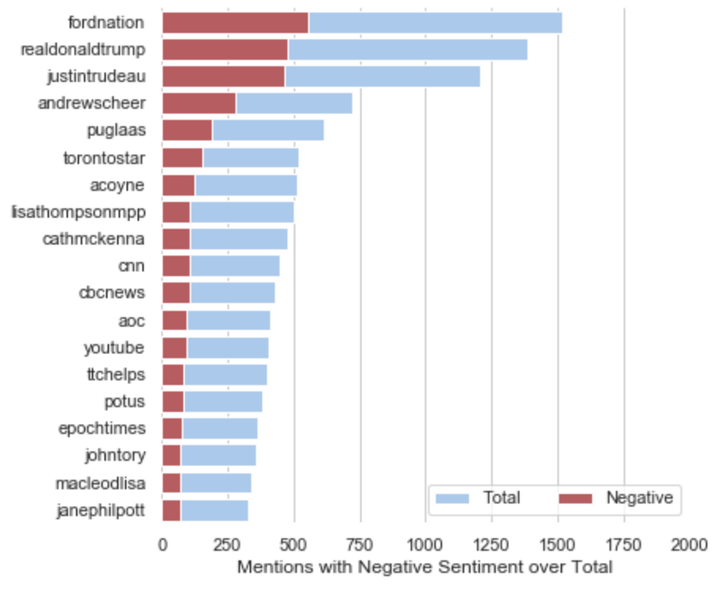
negative hashtags to the total number the hashtags is used. Negative sentiment is found more in referencing political hashtags, such as “cdnpoli”, “onpoli”, and “topoli” all appearing in the top 10. There is also a high frequency of sports related hashtags seen in both the positive and negative sentiment, but it appears more positive references appear for sports hashtags than negative, such as “ufc”, “bluejays”, “habs”, “wrestlemania”, “leafsforever”, and “wethnorth” in positive, with the addition of “gohabsgo” and “raptors” appearing in the negative sentiment.





The above graphs shows the comparison of positive and negative emojis to the total number the emoji appears in tweets. “Face_with_tears_of_joy” is the top used emoji for both positive and negative sentiment.



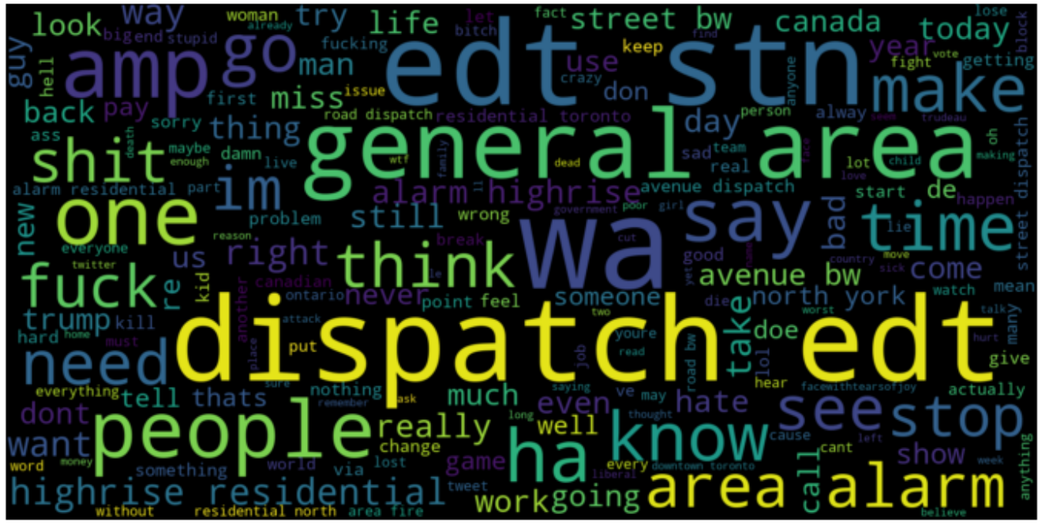


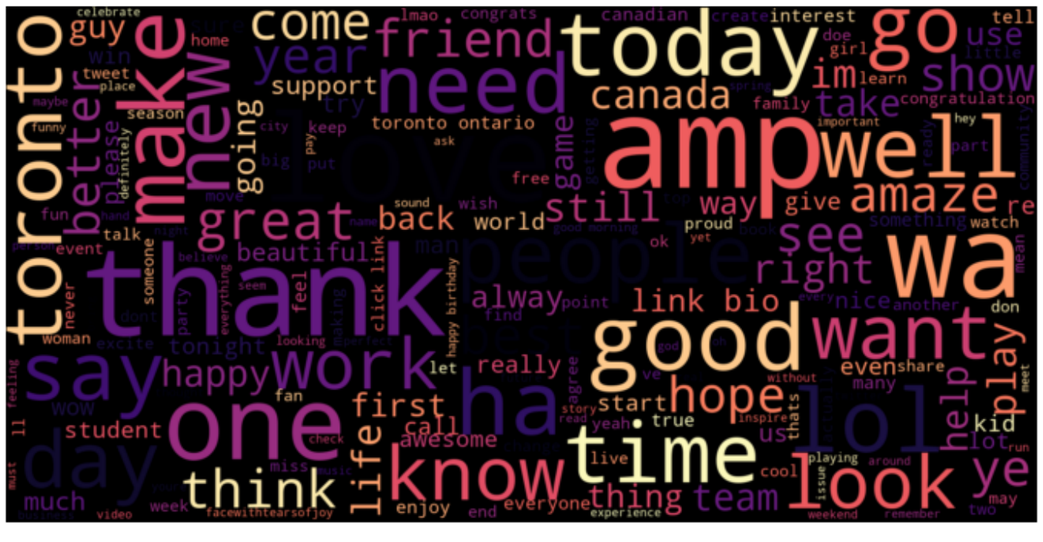
The above graph shows the comparison of positive and negative mentions to the total number of mentions appearing in tweets. It was surprising to see that there was a pretty even distribution of politicians mentions

both positive and negative, with the top 3 mentions on both side being “fordnation”, “justintrudeau”, and “realdonaldtrump” with a little higher number appearing in the negative side but the number appearing very close, around the 500 mark for all.

Analysis 1

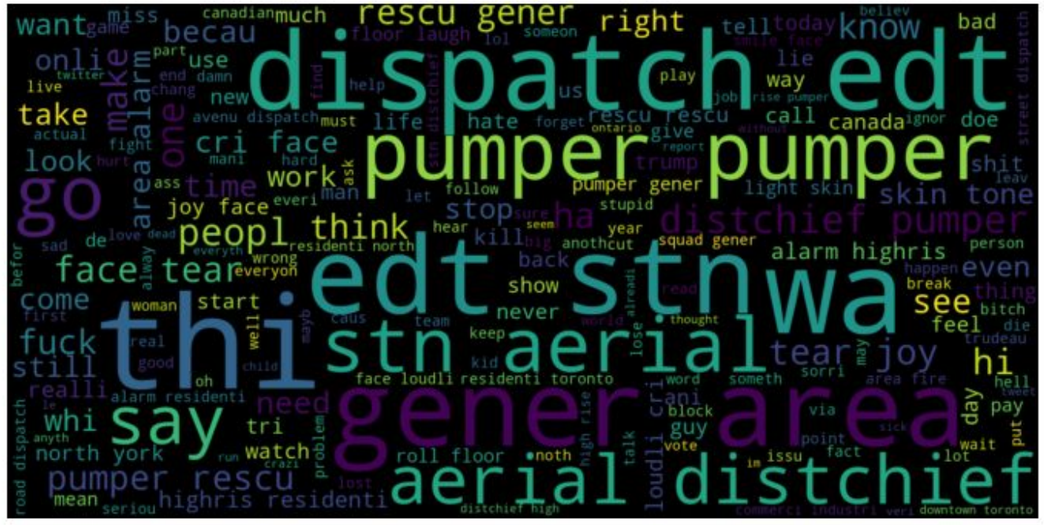
The difference between analysis 1 and analysis 2 is using different cleaning code for the text.

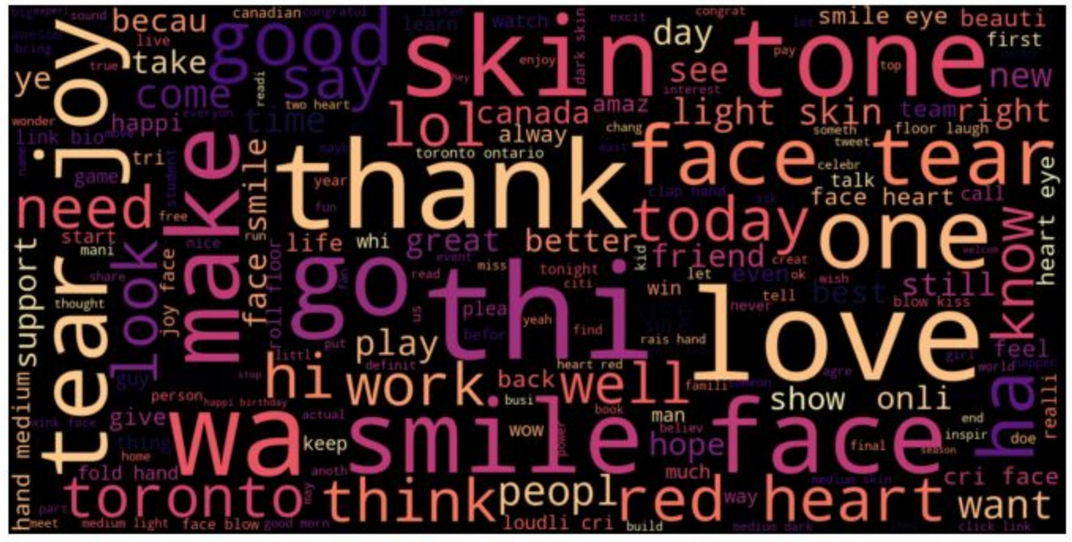




Analysis 2

Analysis 2 has more cleaning applied which provides better insight, removing numbers, punctuation, applying lemmatization and stemming



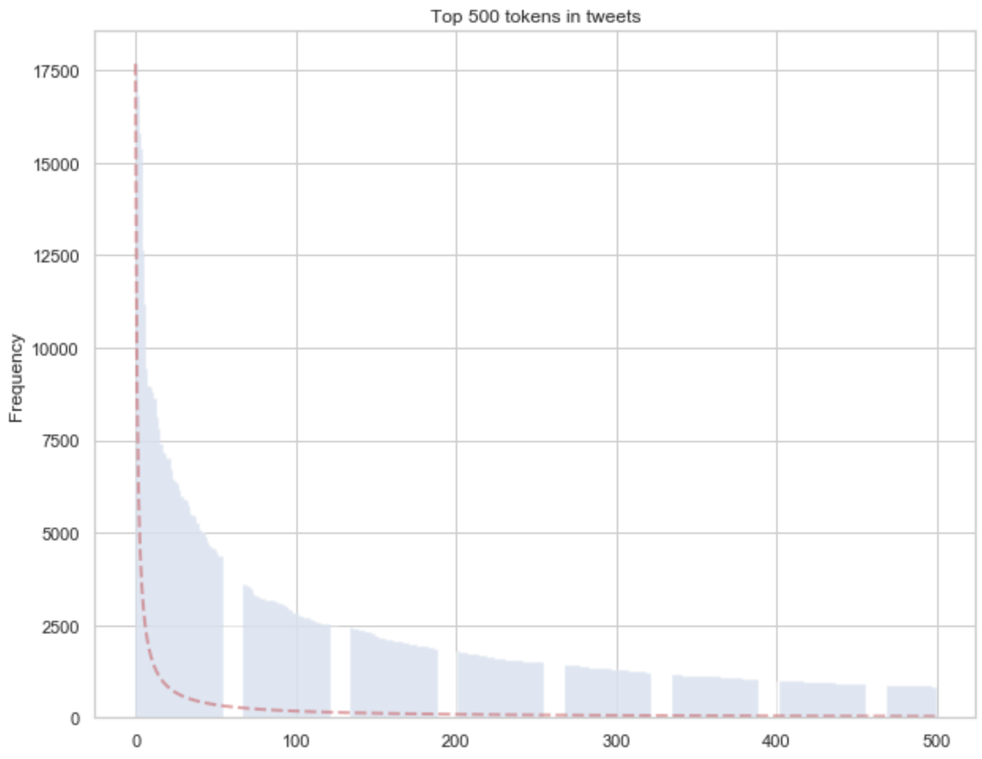


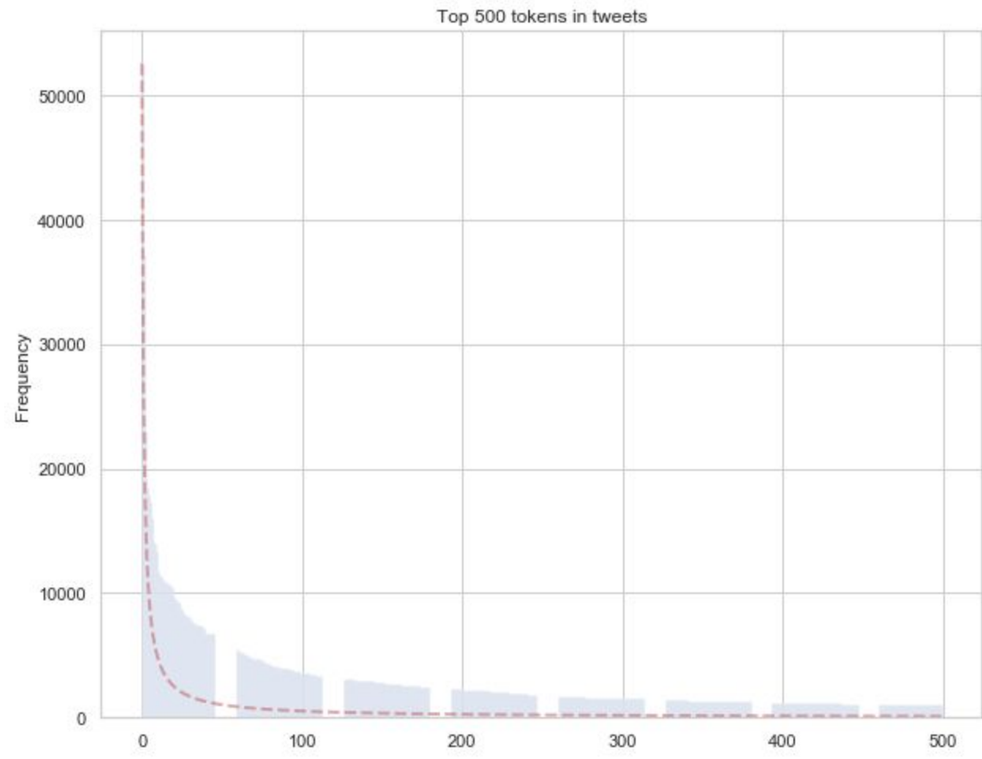
Zipf's Law

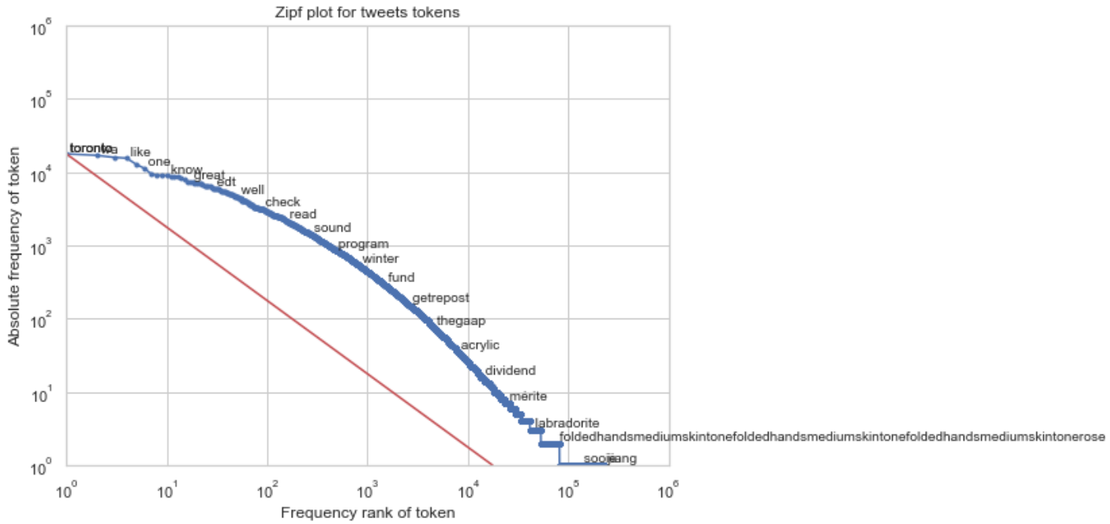
Zipf's Law is first presented by French stenographer Jean-Baptiste Estoup and later named after the American linguist George Kingsley Zipf. Zipf's Law states that a small number of words are used all the time, while the vast majority are used very rarely. There is nothing surprising about this, we know that we use some of the words very frequently, such as "the", "of", etc, and we rarely use the words like "aardvark" (aardvark is an animal species native to Africa). However, what's interesting is that "given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table. Thus the most frequent word will occur approximately twice as often as the second most frequent word, three times as often as the third most frequent word, etc."

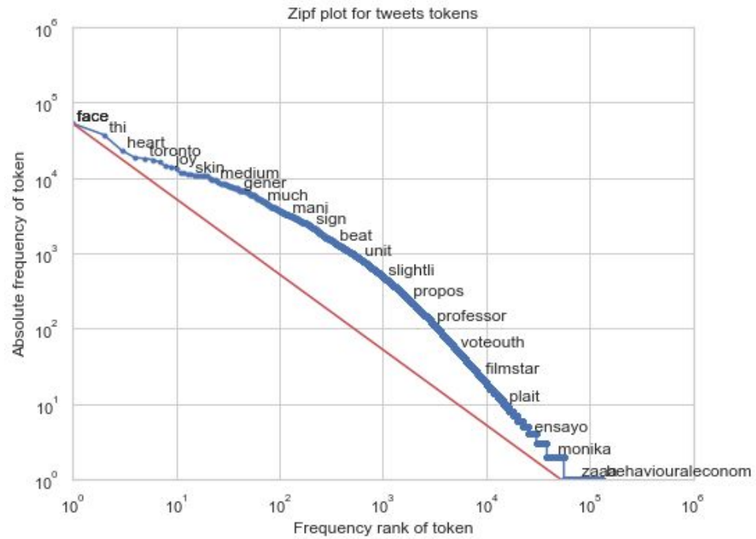
In other words, the rth most frequent word has a frequency f(r) that scales according to $${f(r)} \propto \frac{1}{r^\alpha}$$ for $$\alpha \approx {1}$$

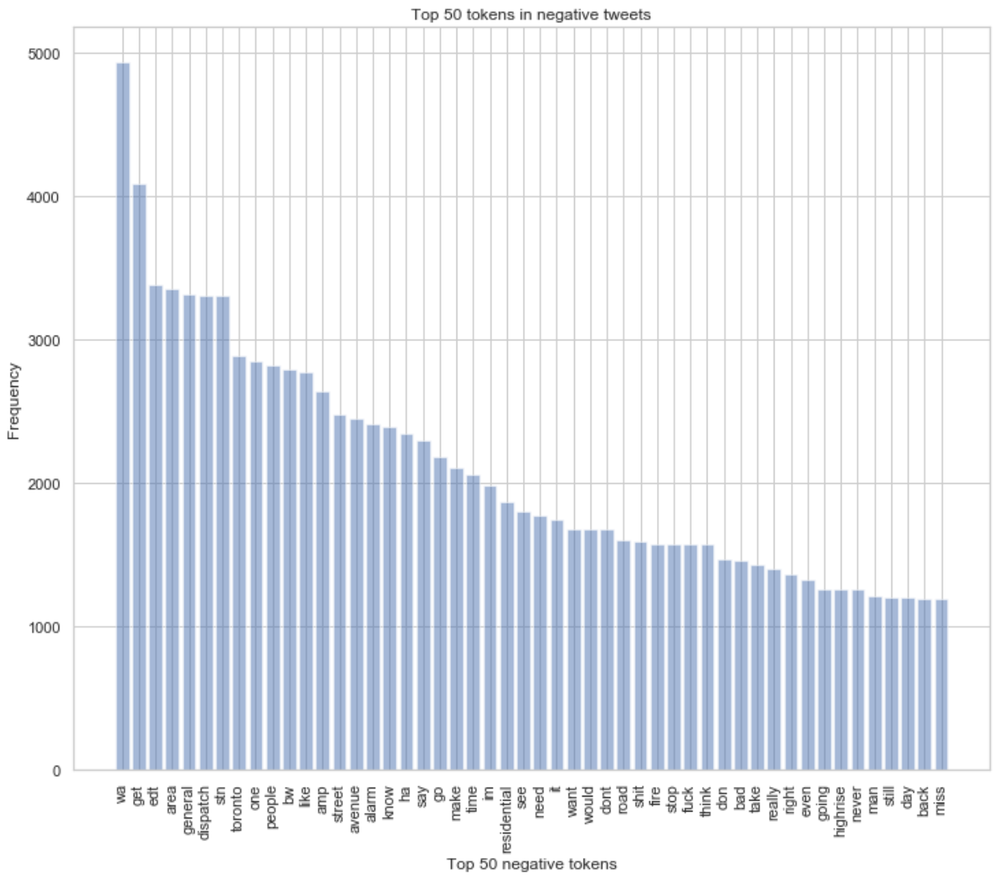
Let's see how the tweet tokens and their frequencies look like on a plot.

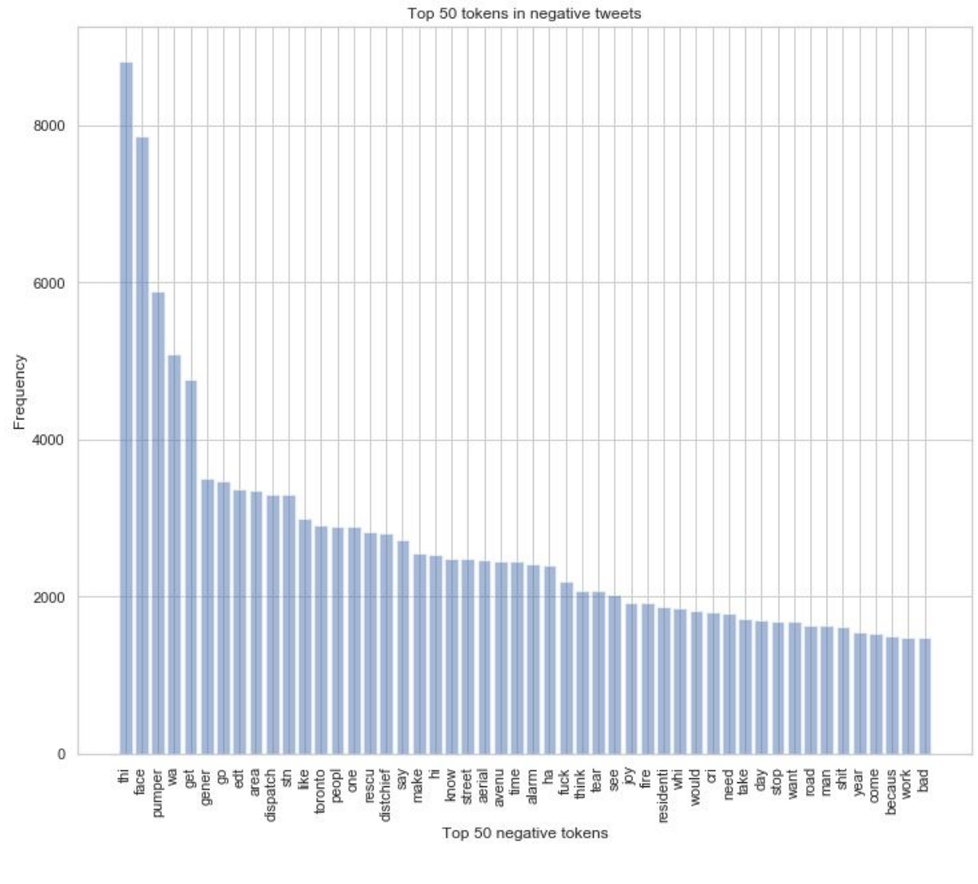


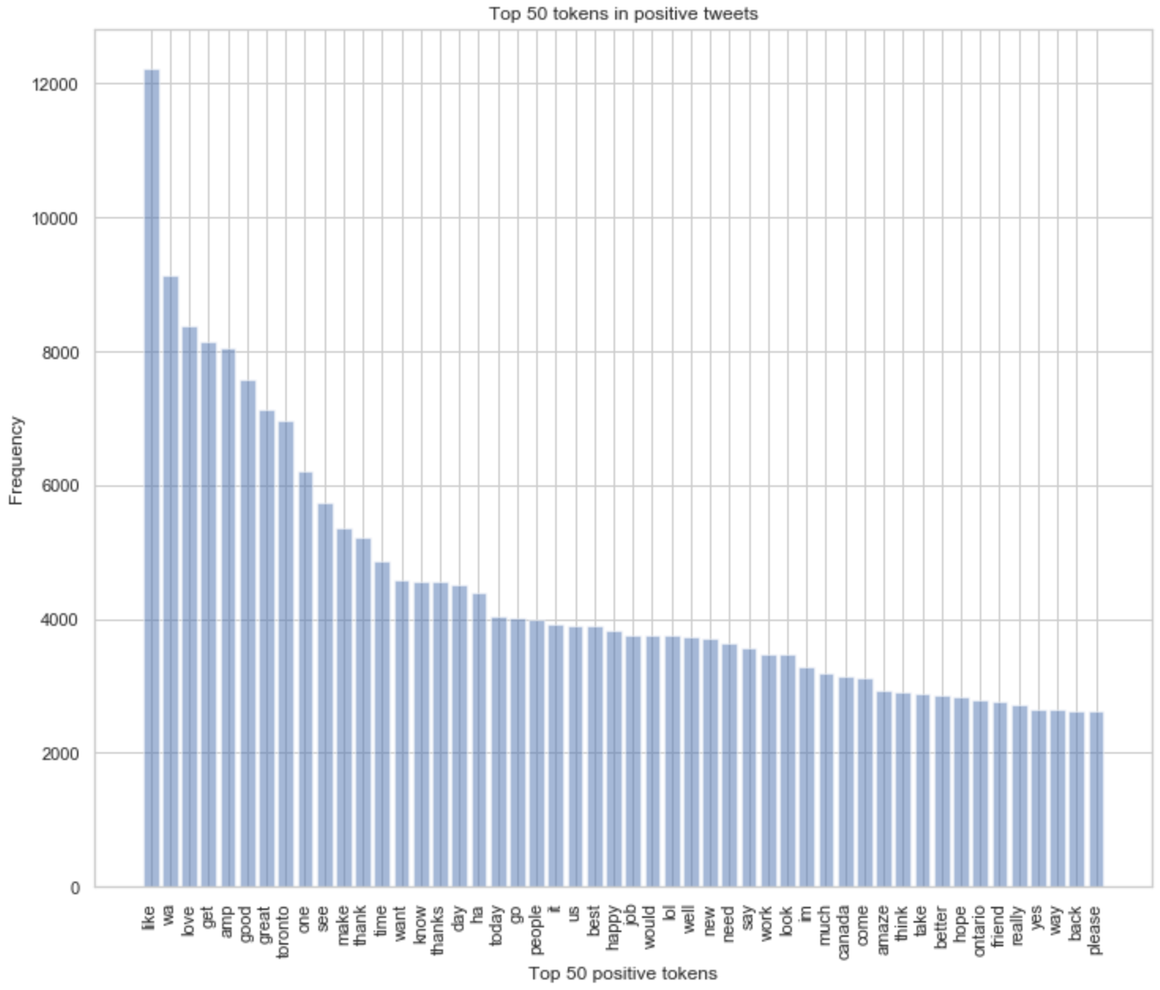


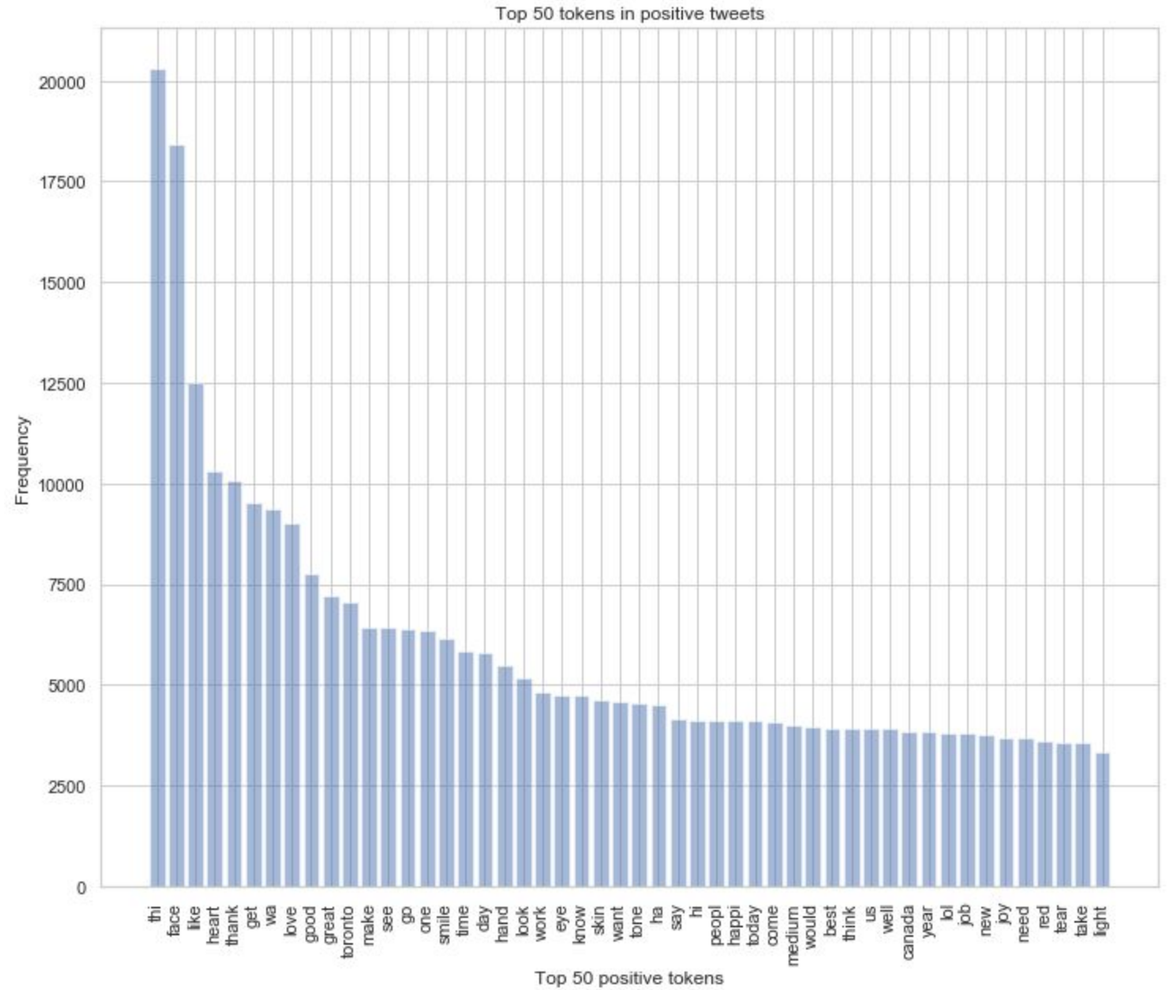

Zipf plot for tweets tokens
Absolute frequency of token
Frequency rank of token
toronto
like
one
know
great
edt
well
check
read
sound
program
winter
fund
getrepost
thegaap
acrylic
dividend
merite
labradorite
foldedhandsmediumskintonefoldedhandsmediumskintonefoldedhandsmediumskintonerose
socialang

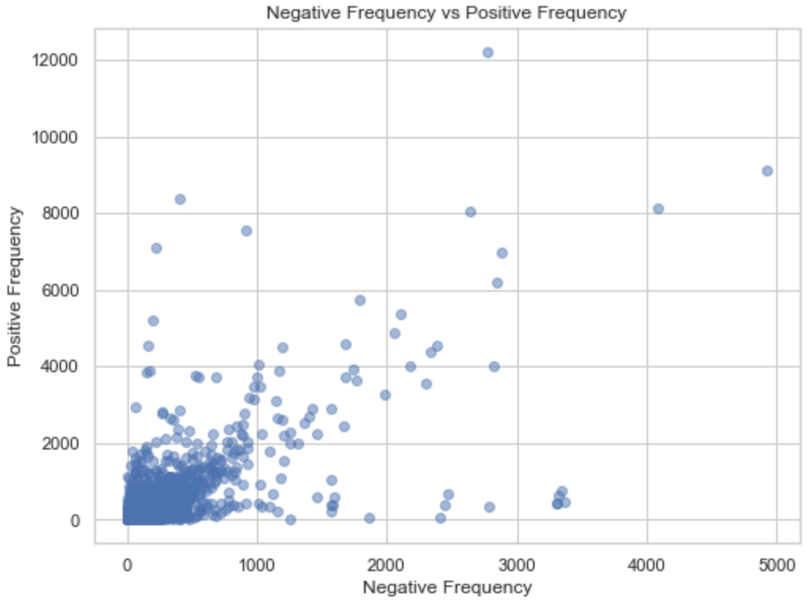

Zipf plot for tweets tokens
Absolute frequency of token
Frequency rank of token
face
thi
heart
toronto
joy
skin
medium
gener
much
mani
sign
beat
unit
slightli
propos
professor
voteouth
filmstar
plait
ensayo
monika
zaab
behaviouraleconom

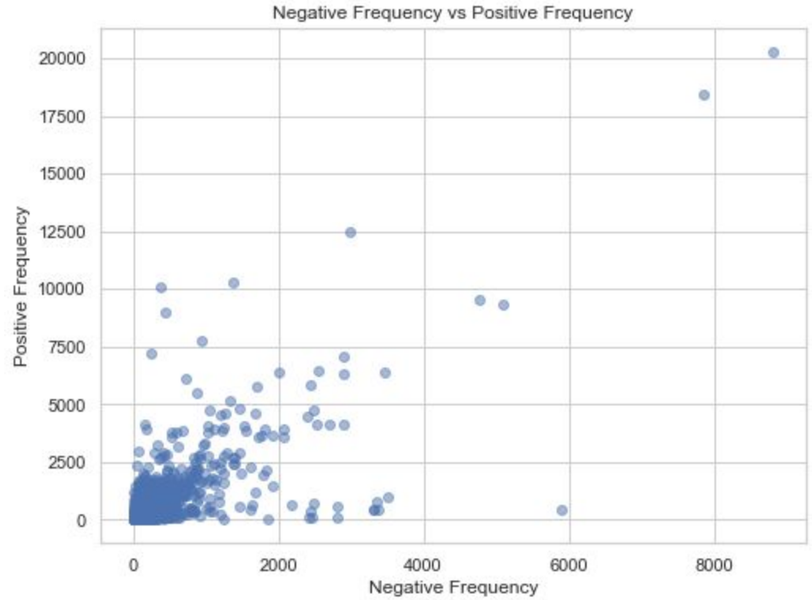

Top 50 tokens in negative tweets
Frequency
Top 50 negative tokens
wa
get
edt
area
general
dispatch
stn
toronto
one
people
bw
like
amp
street
avenue
alarm
know
ha
say
go
make
time
im
residential
see
need
it
want
would
dont
road
shit
fire
stop
fuck
think
don
bad
take
really
right
even
going
highrise
never
man
still
day
back
miss

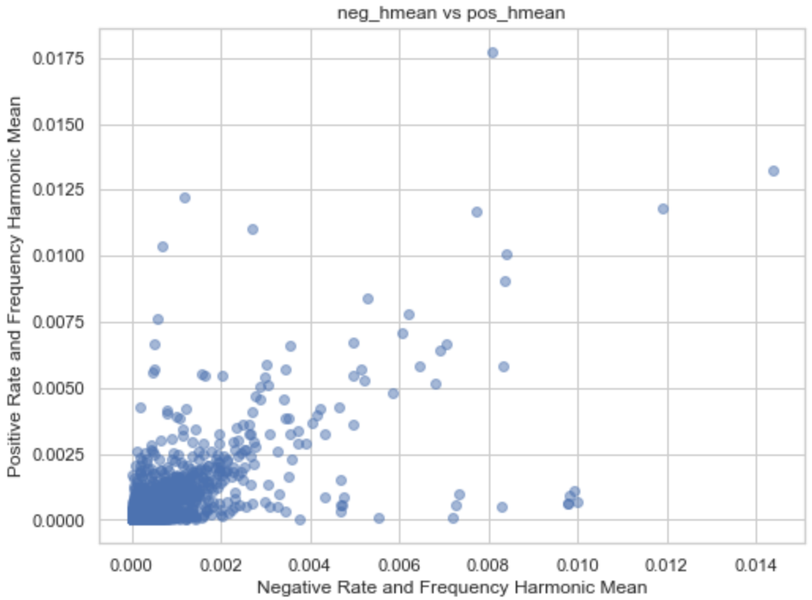

Top 50 tokens in negative tweets
Frequency
Top 50 negative tokens
thi
face
pumper
wa
get
gener
go
edt
area
dispatch
stn
like
toronto
peopl
one
rescu
distchief
say
make
hi
know
street
aerial
avenu
time
alarm
ha
fuck
think
tear
see
joy
fire
residenti
whi
would
cri
need
take
day
stop
want
road
man
shit
year
come
becaus
work
bad

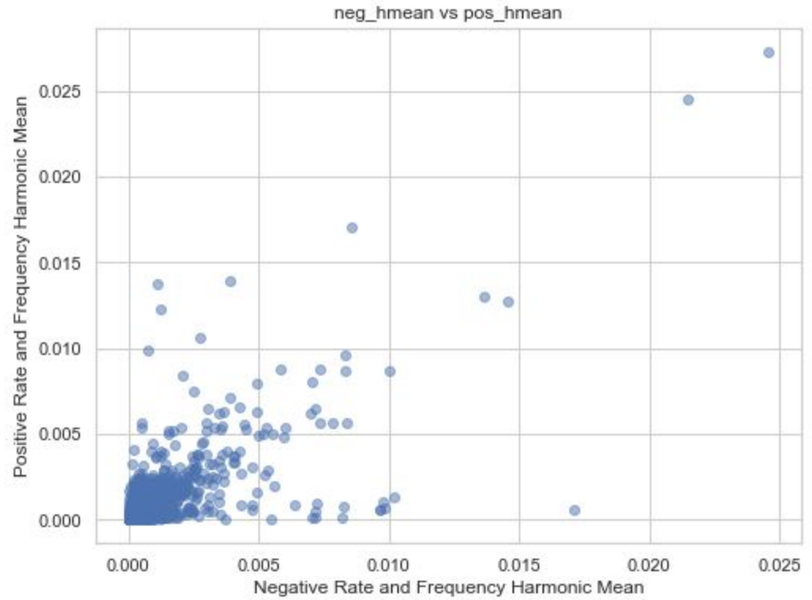

Top 50 tokens in positive tweets
Frequency
12000
10000
8000
6000
4000
2000
0
like wa love get amp good great toronto one see make thank time want know thanks day ha today go people it us best happy job would lol well new need say work look im much canada come amaze think take better hope ontario friend really yes way back please
Top 50 positive tokens

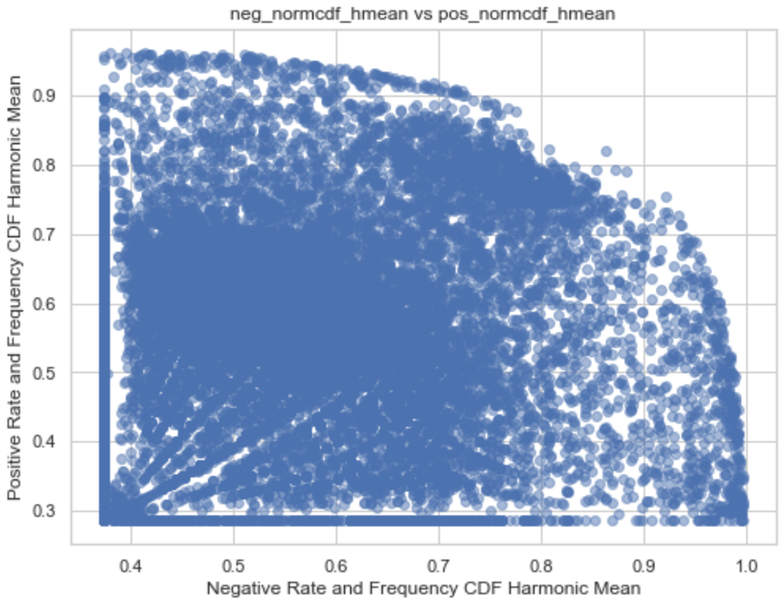

Top 50 tokens in positive tweets
Frequency
20000
17500
15000
12500
10000
7500
5000
2500
0
thi face like heart thank get wa love good great toronto make see go one smile time day hand look work eye know skin want tone ha say hi peopl happi today come medium would best think us well canada year lol job new joy need red tear take light
Top 50 positive tokens


Negative Frequency vs Positive Frequency
Positive Frequency
Negative Frequency
0
2000
4000
6000
8000
10000
12000
0
1000
2000
3000
4000
5000

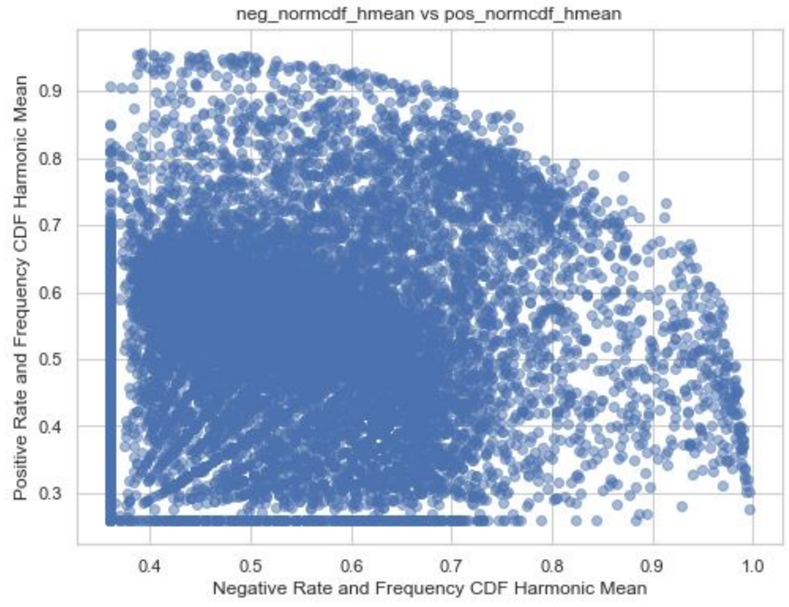

Negative Frequency vs Positive Frequency
Positive Frequency
Negative Frequency
0
2500
5000
7500
10000
12500
15000
17500
20000
0
2000
4000
6000
8000


neg_hmean vs pos_hmean
Positive Rate and Frequency Harmonic Mean
Negative Rate and Frequency Harmonic Mean
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
0.000
0.002
0.004
0.006
0.008
0.010
0.012
0.014

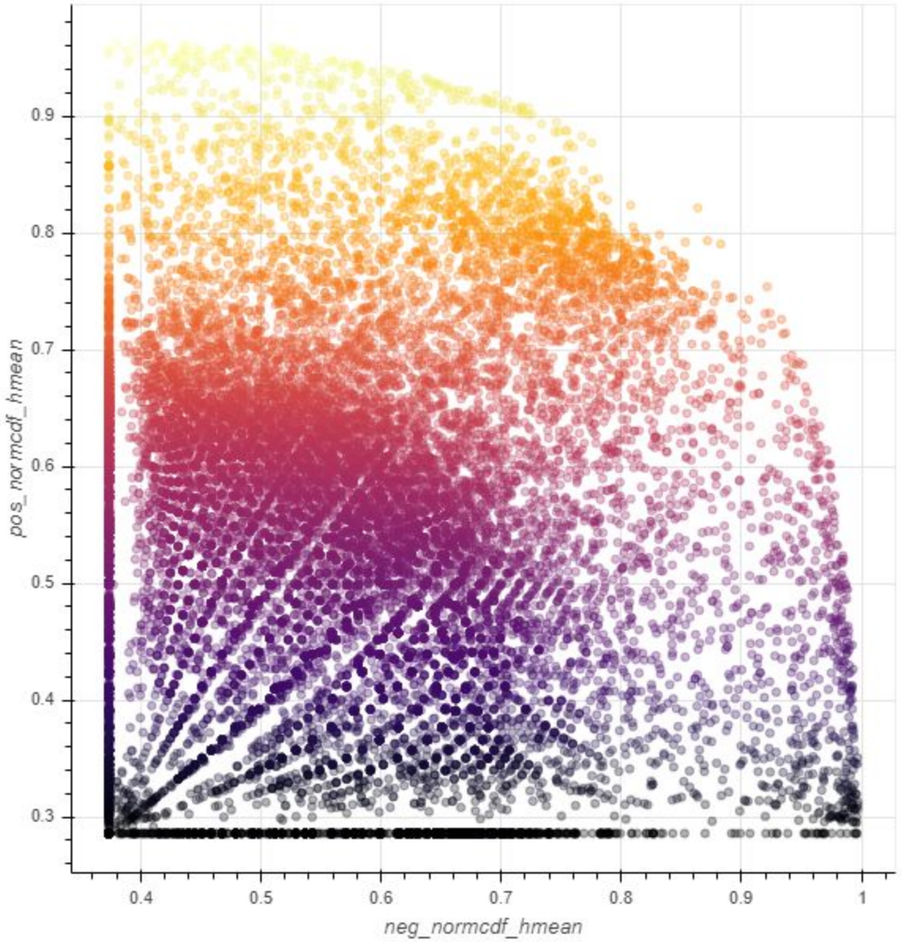

neg_hmean vs pos_hmean
Positive Rate and Frequency Harmonic Mean
Negative Rate and Frequency Harmonic Mean
0.000
0.005
0.010
0.015
0.020
0.025
0.000
0.005
0.010
0.015
0.020
0.025



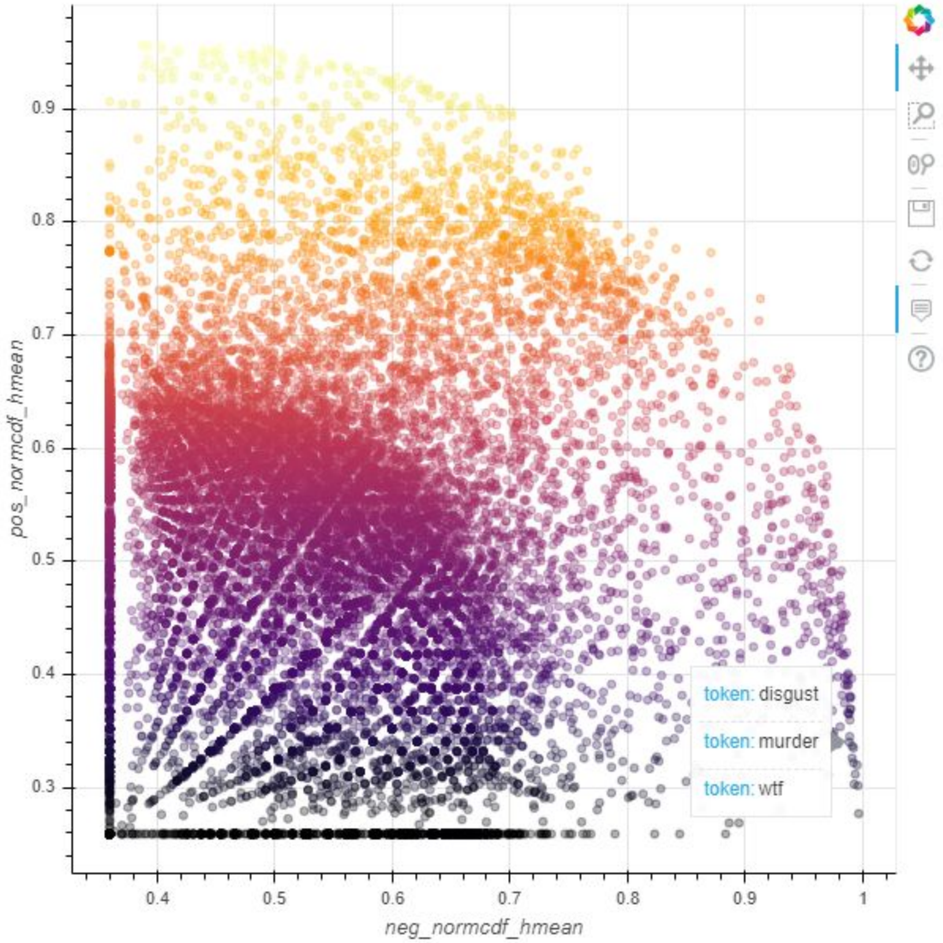
With text data, showing every token as just a dot is lacking important information on which token each data point represents. With 10,000 points, it is difficult to annotate all of the points on the plot.



With cleaner data there appears to be less dots present



Bokeh is an interactive visualisation library for Python, which creates graphics in style of D3.js. Bokeh can output the result in HTML format or also within the Jupyter Notebook. And above is the plot created with Bokeh. https://bokeh.pydata.org/en/latest/



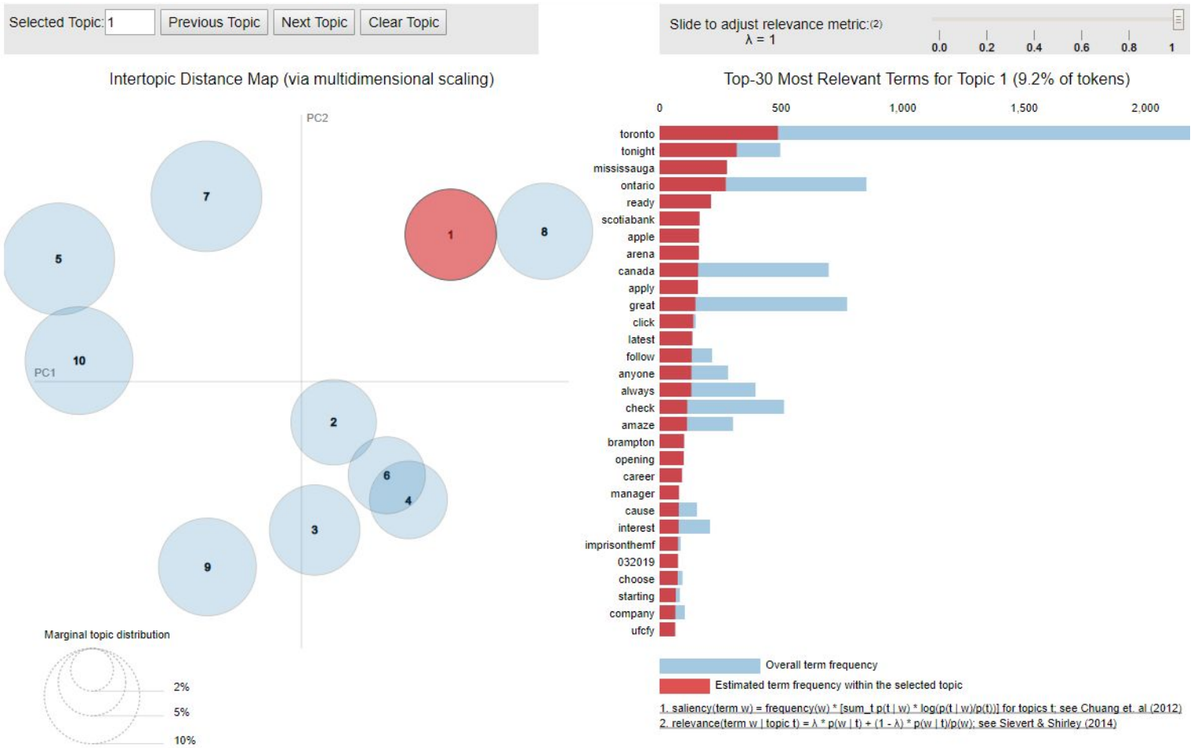
When hovering over the dots in the Bokeh graph, it will list the words associated. Words that appear on the lower right represent a higher negative sentiment based on the (neg_normcdf_hmean) than the words that appear in the top right, for higher positive sentiment based on the (pos_normcdf_hmean). As shown in the image the words “disgust”, “murder”, and “wtf” are shown as far lower right for the negative scale.

TOPIC MODELING ANALYSIS

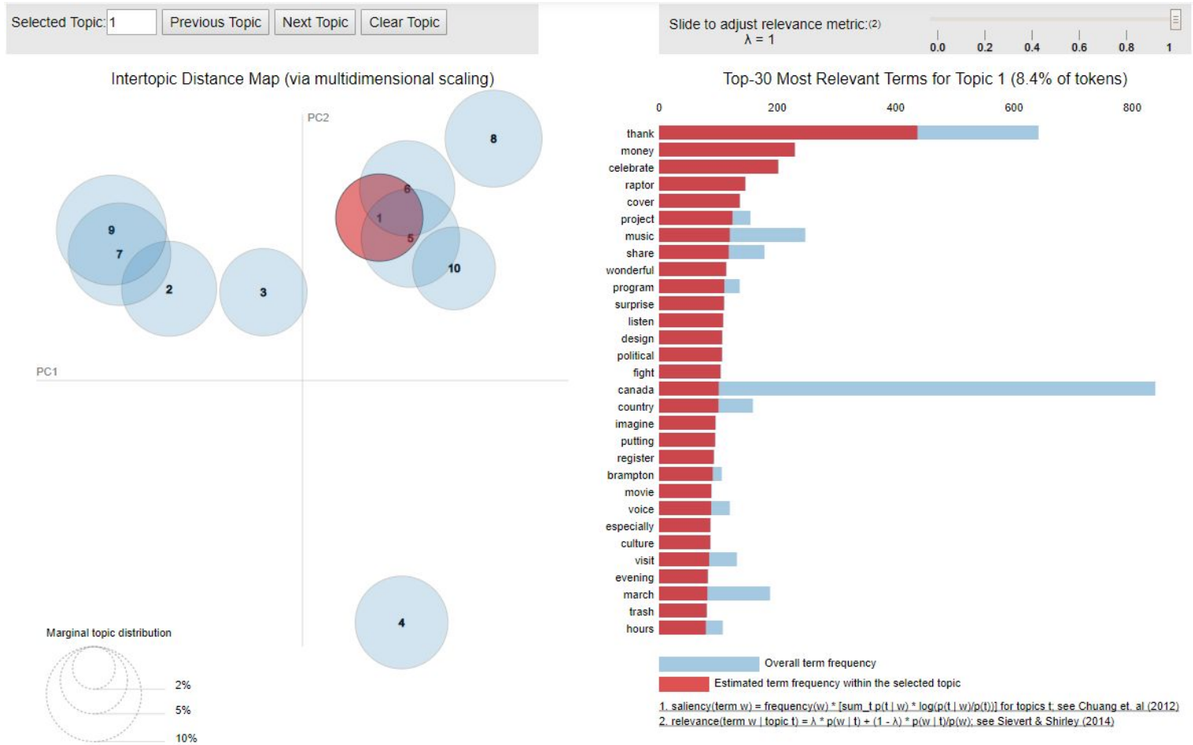
Using Latent Dirichlet Allocation (LDA): a widely used topic modelling technique and apply LDA to convert set of tweets to a set of topics.

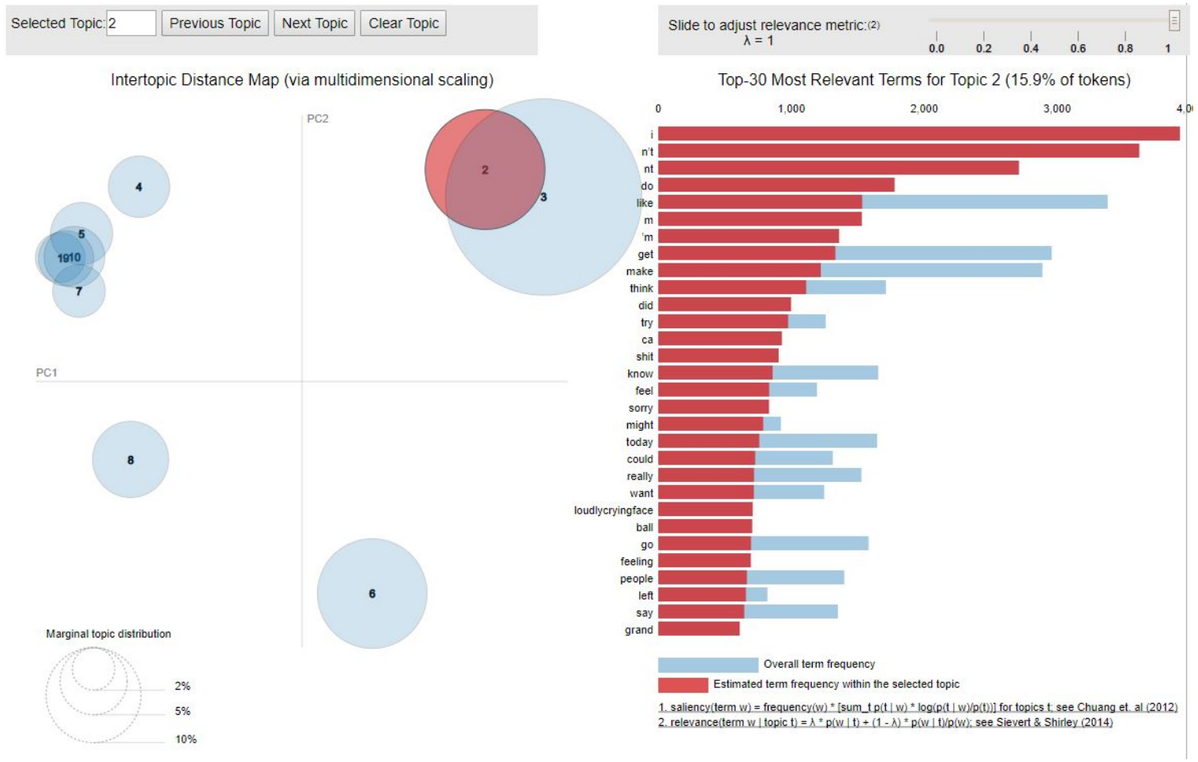
Topic Analysis 1 uses the initial data cleansing code to try and find topics in the data. The issues in v1 became more apparent when looking at the larger date ranges when there was a lot of punctuation and numbers showing up in the topic analysis. Version 2 was created to see if applying additional cleansing to the text to remove punctuation and numbers would improve the LDA to find words that make more relevant topics. After the additional cleansing, v2 did come up with more relevant topics that made more sense.

Topic Analysis 1



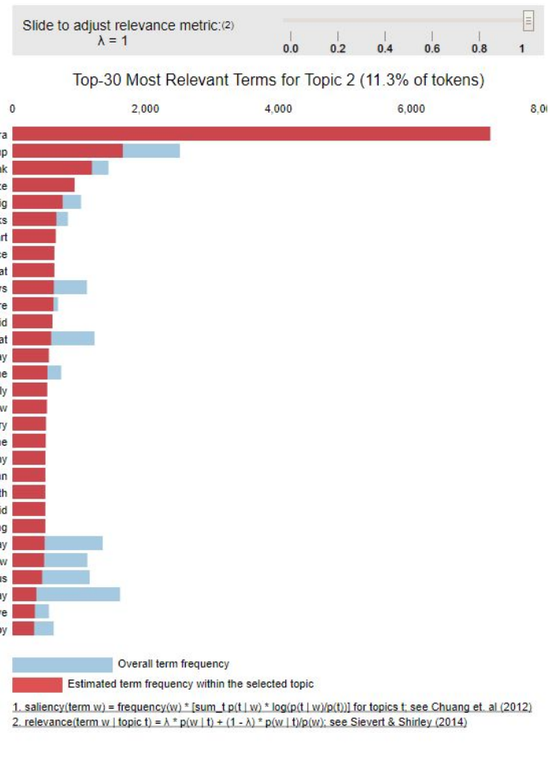
Topic Analysis 2



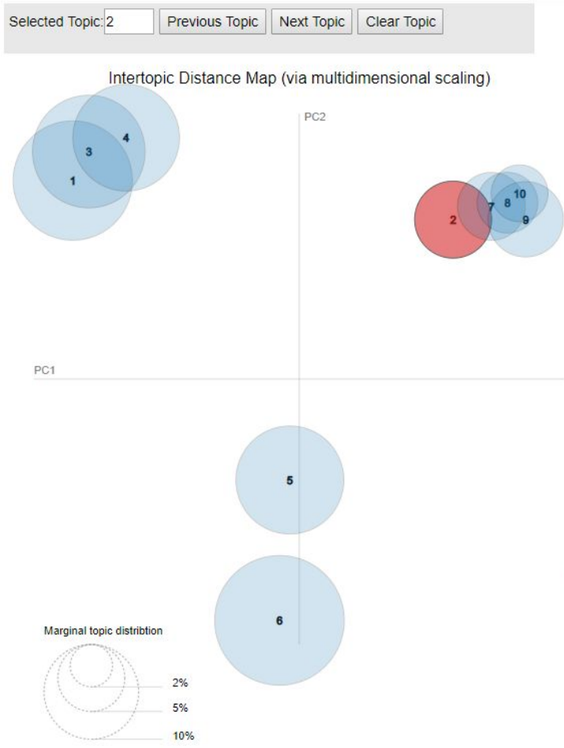

Selected Topic: 2
Previous Topic
Next Topic
Clear Topic
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
Marginal topic distribution
2%
5%
10%

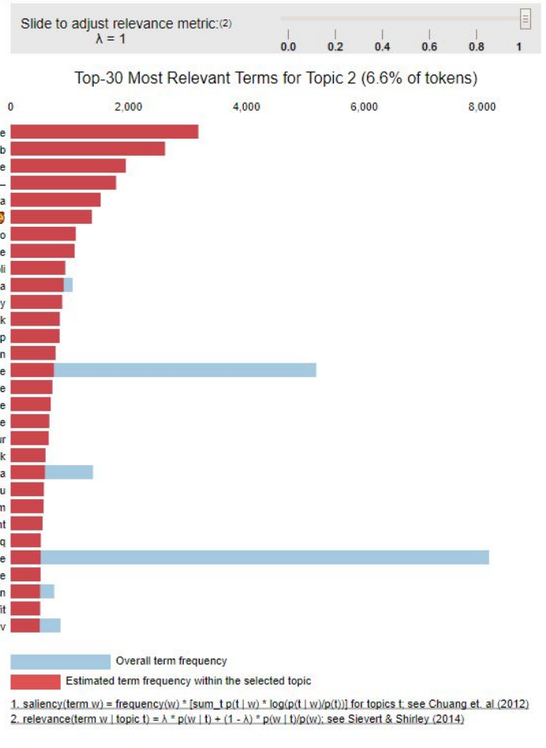
Slide to adjust relevance metric:(2)
λ = 1
Top-30 Most Relevant Terms for Topic 2 (15.9% of tokens)
i
n't
nt
do
like
m
'm
get
make
think
did
try
ca
shit
know
feel
sorry
might
today
could
really
want
loudlycryingface
ball
go
feeling
people
left
say
grand
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

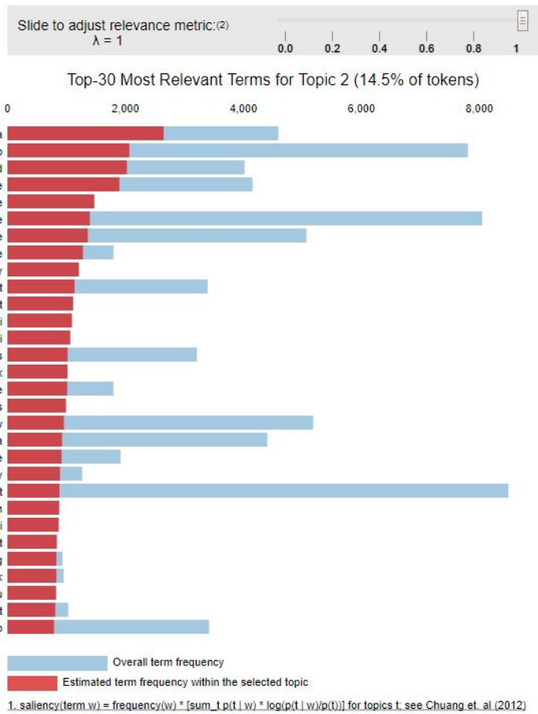

Selected Topic: 2
Previous Topic
Next Topic
Clear Topic
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
Marginal topic distribution
2%
5%
10%


Slide to adjust relevance metric:(2)
λ = 1
Top-30 Most Relevant Terms for Topic 2 (11.3% of tokens)
camera
amp
thank
amaze
big
thanks
art
piece
eat
always
tire
stupid
great
purpleday
become
fully
screw
contemporary
shotoniphone
travelphotography
cayman
photobooth
photogrid
photoshooting
day
new
us
today
approve
happy
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

000







Period 3: April 1-10, 2019

GEO ANALYSIS

Downloaded neighborhood data for Toronto from Open Data Toronto, the shape file and all associated file were found and could be used to do the analysis. With the help of some books and support from some people from the team was able to get the maps displayed straight. Due to the low volume of twitter data in Toronto with geo coordinates available, only xx tweets had geo-tagging. This limits the analysis available to be able to get any data that could be aligned with the other teams work on Toronto data.

Summary of the Visualizations below:

The first grid shows the volume of tweets by the AREA_NAME, this is the neighbourhood in Toronto, which shows that the highest number of tweets come from Flemingdon Park (44) with 6,316 tweets for the period.

The first map, shown in green beside the grid, gives a visual view of the Tweets by neighbourhood, and shows where on a map Flemingdon Park (shown in dark grey).

The second map, shown with the blue dot, gives a visual of all the points on the map where tweets came from. It can show you in all neighborhoods if there were tweets or not. It does not depict volume (number) of tweets in the geo area.

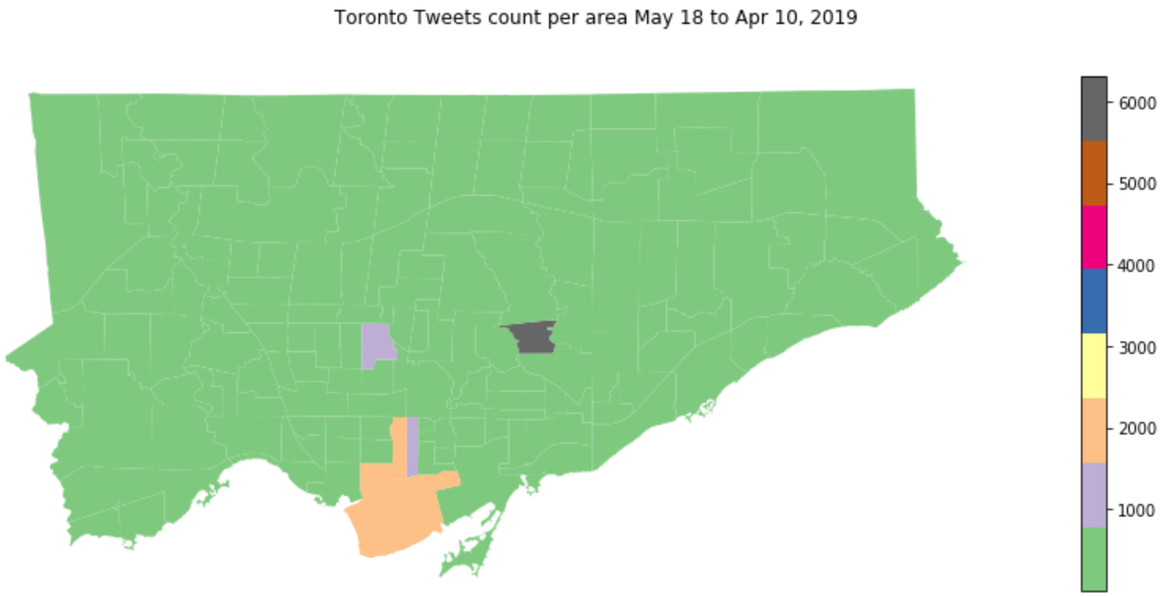
The third map, shown with the hexbins in orange, shows areas on the map where there is more than 5 tweets. This shows that some of the areas where in Map 2 there were dots present, there is no hexbin available. And it can be seen as one red dot on the map with high volume of tweets.

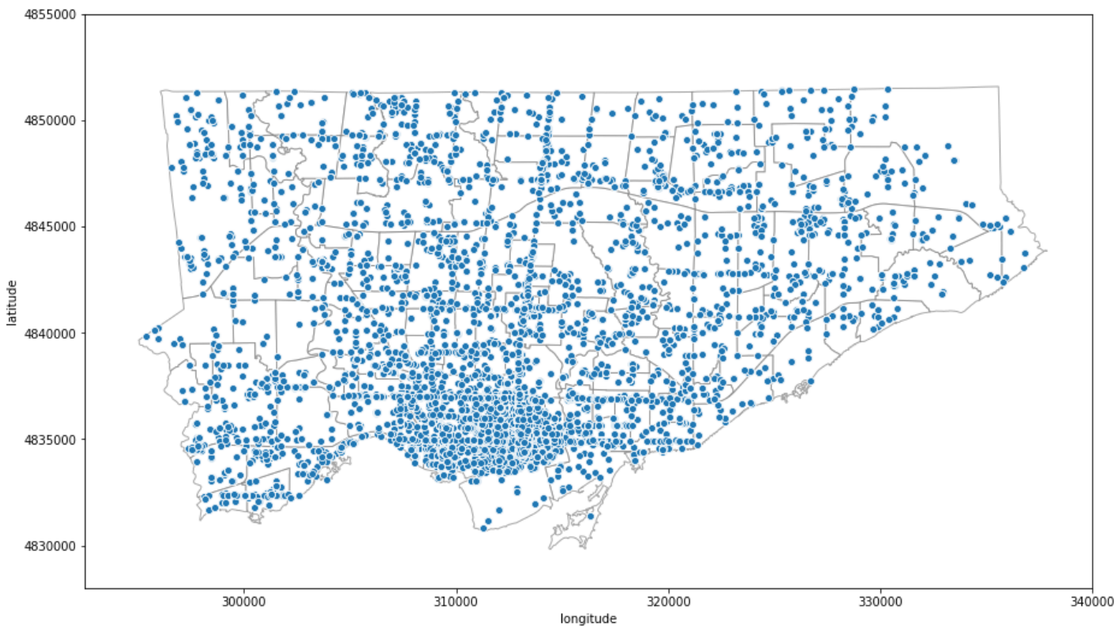
The forth map, is the final map and a zoomed in hexbin map, shows the same map as the third map but zoomed into the neighbourhood with the darkest red colour. This area when doing some investigation represents

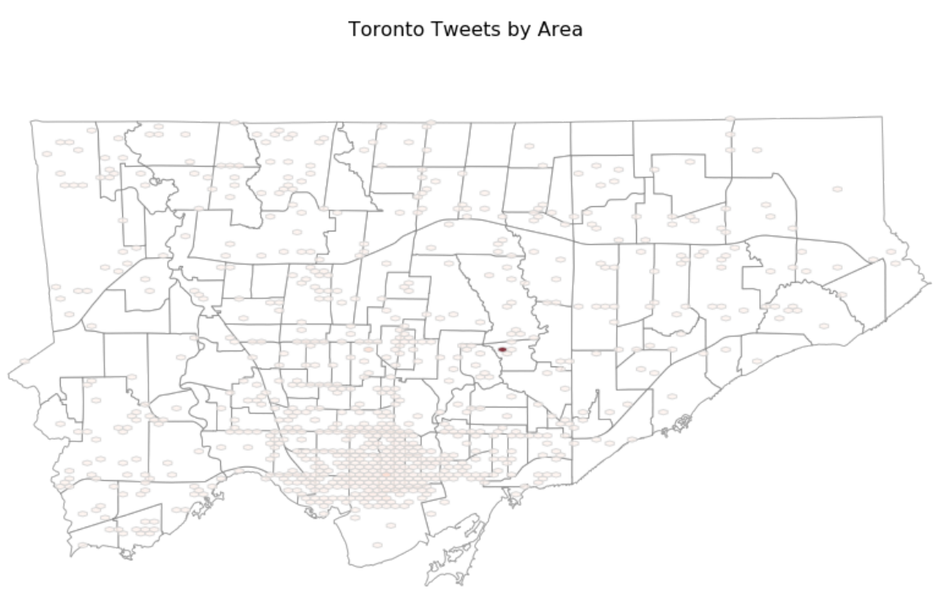
the user “tofire”, Toronto Fire who does the most tweets in Toronto.

Maps

AREA_NAME	
Flemingdon Park (44)	6316
Waterfront Communities-The Island (77)	2358
Bay Street Corridor (76)	2120
Church-Yonge Corridor (75)	870
Forest Hill South (101)	849
Annex (95)	608
Kensington-Chinatown (78)	584
Moss Park (73)	418
Niagara (82)	414
South Riverdale (70)	412
Islington-City Centre West (14)	355
University (79)	332
South Parkdale (85)	300
West Humber-Clairville (1)	287
York University Heights (27)	286
Trinity-Bellwoods (81)	280
Dovercourt-Wallace Emerson-Junction (93)	255
Banbury-Don Mills (42)	246
Woburn (137)	243
Mimico (includes Humber Bay Shores) (17)	214







OTHER REFERENCES AND CREDITS

1. Mike Parravani - CSDA1050-CAP, geo/maps of Toronto
2. Topic Modeling Analysis: https://towardsdatascience.com/topic-modeling-in-pythoon-with-nltk-and-gensim-4ef03213cd21
3. Sentiment Analysis:
    a. https://github.com/tthustla/twitter_sentiment_analysis_part2/blob/master/Capstone_part3-Copy1.ipynb
    b. https://github.com/tthustla/twitter_sentiment_analysis_part3/blob/master/Capstone_part3-Copy2.ipynb