TRUMP FACTS

Erik Lybecker, Love Marcus, Robin Maillot, Lucas Rodés-Guirao
KTH Royal University of Technology
Stockholm, Sweden
{ejhly,lovema,maillot,lucasrg}@kth.se

Simon Stenström
Findwise AB
Stockholm, Sweden
simon.stenstrom@findwise.com

ABSTRACT

Social Networks has redefined how companies, public institutions and personalities relate with the society and citizens. Some public figures avoid mainstream media in favor of using these new platforms for establishing a direct communications channel with the public. The U.S. president Donald J. Trump's use of twitter is a clear example of this new communication methodology. During the 2016 presidential election Trump's Twitter account gained notoriety and his impact in this social network increased. With more than 30 000 published tweets, this data can be used to extract relevant features in order to analyze behaviors and detect user patterns.

1 INTRODUCTION

Social Networks are today used as meeting points where users can exchange ideas, react to events, ask questions to customer services, follow the latest news related to celebrities, politicians or athletes etc. Yes, pictures of cats are also available. Twitter[1], Facebook[2], Instagram[3] and many other sites have become mainstream communication channels, which are present in our daily lives and relations with friends and relatives. As of March 31, 2017 there were 1.94 billion monthly active users of Facebook[4] and 313 million monthly active Twitter users[5]. Nowadays, these platforms serve as political, crisis and brand communication, since it is an effective and direct manner to convey information.

Hashtags are an important and elemental component of these sites. They are used as labels so that users can easily exchange comments about related topics. A pound sign '#' is used before the label for this purpose. Although the origin of the pound usage might be deeper, as highlighted by Bruns & Burgess (2011) and Chang (2010), it was Mr. Chris Messina who proposed this tagging system in Twitter in 2007

> How do you feel about using # (pound) for groups. As in #barcamp [msg]?
> @chrismessina, August 23, 2007 [6]

Motivated by the large amount of data that it is generated daily in Twitter[7], we make a deep dive into Donald J. Trump's Twitter account @realDonaldTrump analyzing his behavior in this social network. We provide visualizations with the hope that they can provide insights from the data. Fig. 1 shows a screen-shot from tweet containing the sentence *Make America great again*, which Donald has been using throughout his 2016 US presidential campaign.

To this end, we have used the popular Elastic software such as Elasticsearch, Gormley & Tong (2015), and Kibana, Elasticsearch (2016). The former is a distributed, RESTful search and analytics engine, which provides scalable search and has a near real-time search. The latter, provides visualization tools on top of the data indexed by ElasticSearch. The data retrieved from Donnald J. Trump's Twitter was first run through a series of filters and classifiers to retrieve and index only relevant information. These tools will be further explained in subsequent sections.

[1] http://twitter.com
[2] http://facebook.com
[3] http://twitter.com
[4] https://newsroom.fb.com/company-info/
[5] https://about.twitter.com/company
[6] https://twitter.com/chrismessina/status/223115412
[7] http://www.internetlivestats.com/twitter-statistics/

Figure 1: Sentence made famous by Donald Trump during 2016 US Presidential Elections

1.1 Contribution

In brief, our contributions are

- Clear and detailed explanation of our methodology, allowing for other researchers to replicate, and potentially improve, our work. We also provide the source code [8].
- Insights and interpretations about the data.
- Combination of State of the art algorithms to find correlations between relevant features of Tweets
- Design of a rule-based clustering algorithm for important words

1.2 Organization

Section 2 takes a brief look at work that has been previously done in Social Network data mining and analysis. Section 3 explains in more detail our approach and the software tools that we have used. Finally sections 4–5 concludes the report with the results obtained and a short discussion.

2 Related Work

The recent adoption of social network platforms has created a suitable environment for researchers, where they have access to large amounts of data to test the performance of algorithms extracting relevant information from users. Users post own opinions, reflections and thoughts about products, companies and organizations, which opens the opportunity to apply techniques such as Sentiment Analysis, to extract insights from the data. Kwak et al. (2010) were able to crawl the entire Twitter site and perform some exploratory data analysis on it. They ranked users by number of followers, by PageRank, Page et al. (1999), and by number of retweets. Back then, they say, there were 106 Million tweets in the whole Twitter domain... Today more than 500 million tweets are sent per day[9].

Pang et al. (2008) analyzed different techniques for opinion-oriented information-seeking systems. In this regard, they also evaluated the privacy, manipulation and economic impact of the development of such methods. Furthermore, Yang et al. (2007) used Support Vector Machine (SVM) and Conditional Random Field (CRF) to classify the sentiment of web corpora using *emojis* contained in text, shedding light to web data sentiment analysis. Pak & Paroubek (2010) presented a method to classify web corpora and label them as positive, negative or objective without human intervention. To this end, they used statistical linguistic analysis to build a model from a given corpus and later evaluated its performance on unseen microblogging posts. Bruns & Stieglitz (2013) proposed metrics to study *hashtagged* Twitter conversations. They also pointed to open source tools which were meant to help other researchers in applying such metrics or further develop them.

Hao et al. (2011) presented an approach which included a visual interface, where the end user was able to observe the sentiment of large volumes of twitter comments to a given tweet. They developed a novel topic-based text stream analysis technique which was able to provide a Sentiment Geo map visualization. Baucom et al. (2013) explored how good Twitter was as a "world mirror". For this purpose, they analyzed the reactions of users in this platform after real-world events. In particular, they focused on NBA games, where supporter's reactions highly depends on their geo-location.

[8] http://github.com/lovemarcus/Trump-Facts
[9] http://www.internetlivestats.com/twitter-statistics/

During the US presidential election 2016, twitter played a crucial role, for the reasons that have been previously pointed out. Bovet et al. (2016) analyzed twitter user's opinion about the 2016 Presidential Candidates by looking at large amount of user interactions. Their result, show evidence of how Twitter can uncover real-world social trends.

Furthermore, MacDonald & Villamayor (2017) used several machine learning techniques (such as Naive-Bayes, SVM, J-48 Decision Trees) to build classifiers to discriminate `@realDonaldTrump` tweets from other user's tweets. Their results showed high accuracies. Ahmadian et al. (2017) took a detailed look at how Donald Trump conveyed information during the 2016 Republican Party presidential primaries. To this end, they analyzed speech transcriptions using Pennebaker's Linguistic Inquiry and Word Count (LIWC) and had speech recordings rated by trained raters. They concluded that communication-style was key in his success during the primaries.

3 METHOD

A typical approach is to use the defined Elastic Stack, which consists of using Logtash as the web data collection pipeline, ElasticSearch as the search and analytics engine to store the retrieved data and finally Kibana to visualize the results.

3.1 DATA COLLECTION

Because of the rate and date limitations that Twitter imposes, using an API is not practical. To bypass this, the tweets are retrieved from `@bpb27`'s GitHub repository [10], which is updated hourly with new tweets. Besides the inclusion of deleted tweets, the information retrieved is identical to the one available directly using an API. This section first describes what elements are parsed from each tweet, then how this raw information is used to obtain more useful features, such as sentiment analysis. The final features used are available in Table 1.

Table 1: Fields indexed for each tweet.

Field name	Description
`date`	Date tweet was posted
`text`	Tweet content as a string
`users_mentioned`	Twitter users mentioned
`hashtags_mentioned`	Hashtags mentioned in the tweet
`retweet_count`	Number of times the tweet was re-tweeted
`words`	Nouns and adjectives
`NER_PERSON`	People mentioned in the tweet
`NER_LOCATION`	Locations mentioned in the tweet
`NER_ORGANIZATION`	Companies mentioned in the tweet
`sentiment`	Attitude (Sentiment) of tweet
`geo_location`	Coordinates of locations mentioned in the tweet

3.1.1 PARSING

The information retrieved is given in a set of JSON files, each of them containing the tweets from a specific year (2009 - now). Since the raw data has over 60 fields per tweet, we loop over the data in order to extract only relevant information and index it to ElasticSearch with a unique id corresponding to the tweet, which is calculated at run time.

3.1.2 SENTIMENT ANALYSIS

Sentiment analysis is the act of trying to find the mood conveyed by a certain piece of text. One of the major difficulties is being able to predict the feeling the author is trying to depict in different types of texts (e.g. novels, microblogs or plays). Here, we only focus on tweets which composed of one or two sentences, which can convey one or more sentiments. On top of the usual vocabulary, *emojis* [11] and slang are often used in social networks, which can cause difficulties if not considered properly. Due to this, we choose to use VADER, a sentiment analyzer developed by Hutto

[10] https://github.com/bpb27/trump_tweet_data_archive
[11] http://unicode.org/emoji/charts/full-emoji-list.html

& Gilbert (2014) at GeorgiaTech. Not only is VADER available for free online with a practical Python API, but it was specifically designed for micro-blog use. VADER uses a lexicon of key features (words that convey sentiment) with associated polarities that is specifically tailored to micro-blog posts such as tweets. The lexicon was developed using several already existing sentiment word banks (Bradley & Lang (1999). Pennebaker et al. (2001)) as well as *emojis* and slang lists. VADER distinguishes positive, neutral and negative sentiments hence each word in the lexicon is given a value between -4 and 4 corresponding to its valence (4 corresponding to the most positive). These values are a combination of previous work and new human validation.

VADOR also leverages modifiers that can increase or decrease the valence of the words. These modifiers are combined with 5 context aware rules:

- Punctuation such as *!* can increase variance while keeping polarity.
- ALL-CAPS words have higher variance than non-capitalized words.
- Degree adverbs can increase or decrease variance of surrounding words.
- *but* represents a change in polarity in the sentence and the following sentiment is dominant.
- Negation of words is captured using tri-gram representation (90% accurate).

Using this combination of a $75,000$ word lexicon, modifiers and 5 rules, VADER yields better results than other available methods, including lexicon, valence and Machine Learning based techniques. Recent methods using SVMs give good results but are relatively slow compared to VADER, which enables real time indexing.

In this paper each tweet is considered as one piece of text and associated with one valence value from VADER. This approach is limited and tweets with two contradicting sentences end up with a neutral value. But because of the short nature of tweets (see Fig. 1) their sentiment can usually be represented by one value. Interesting statistics from a corpus of collected tweets include:

- Average sentiment of tweets including a word (as well as most positive and negative Tweet including that word).
- Average sentiment of Twitter account.
- Average sentiment of tweets posted at a certain time (morning, day, night etc).

The examples in Table 2 show how Tweets differ from normal text and justifies the use of a specifically tailored sentiment analyzer.

Table 2: Examples of VADER sentiment analysis on Trump's tweets

Score	**Tweet**
-0.934	What an awful piece on #realsports about @realDonaldTrump HBO should be ashamed. #trump2016" Very dishonest piece by racist Gum
-0.929	Spitzer failed as A.G., failed as Governor in disgrace, and was fired on all T.V. shows (boring and zero ratings), and he's at it again
0.972	From Donald Trump: Wishing everyone a wonderful holiday & a happy, healthy, prosperous New Year. Lets think like champions in 2010!
0.986	Good luck and best wishes to my dear friend, the wonderful and very talented Joan Rivers! Winner of Celebrity Apprentice, amazing woman.

3.1.3 Part-Of-Speech tagging

Part-Of-Speech (or POS) tagging is the act of assigning each word in a text into a grammatical category. Because the grammatical category depends on the sentence, POS tagging is context specific and categories have to be determined at run time. Taggers can be either rule based or statistical. The NLTK package for Python, Bird (2006), comes with text parsing and one can load a POS tagger to run on this pre-parsed sentence. Two of the most common taggers

available are Greedy Averaged Perceptron and Maximum Entropy classifier. Because of the time constraint, we did not have time to train any classifier specifically on Twitter data so the default classifier was used. We choose the Maximum entropy classifier[12] because it yielded good results, it was quick and it was simple to use without any additional training. Ratnaparkhi et al. (1996) show how a MxEnt classifier can yield good results for POS-tagging.

A maximum entropy classifier is a machine learning technique that results in a log-linear classifier. The classifier does not require any information on the prior distribution. In the case of NLTK, training was done on ACE 2004[13] and the resulting statistics needed to build the maximum entropy classifier are provided by means of a pickle file which can be easily imported in Python. MaxEnt is feature based, including features such as the last three letters of the word, category of surrounding words etc.

Using this classifier, a list of key words were extracted from each Tweet. This included all the nouns and adjectives. With Kibana, word-clouds can be generated displaying the top key words associated with a specific search. The MaxEnt POS tagger is quick enough to be used directly at indexing time (typically a few milliseconds per tweet on a laptop) and efficiently filters out irrelevant words such as adverbs, pronouns and conjunctions.

On top of this, a rule based token clustering algorithm was run. The goal was to cluster tokens that represent the same word as one token. The algorithm was developed using human evaluation. The goal was to keep an accuracy close to 100% while increasing the recall when one token is searched. The algorithm is shown in Algorithm 1. The algorithm is naive because the dictionary of existing tokens is created dynamically at run time, which means that the order we receive the tokens in does matter. The list of the last letters to ignore was built empirically by checking which tokens were not accurate. The final ignore list used was "e,y,g". This method effectively reduced the number of different tokens from 48764 to 40096 (based on 30000 Tweets) while keeping the accuracy almost constant.

Input: $token, dict, ignore_{list}$
Output: $token_{new}$
Parameters: n: minimum length of token to filter
$token_{new} \leftarrow token$;
$end \leftarrow len(token) - 1$;
if $token \notin dict$ **then**
  if $token$ not $uppercase$ **then**
    $token \leftarrow lowercase(token)$
  end
  if $(len(token) > n) \wedge (token[end] \notin ignore_{list})$ **then**
    if $token[0 : end - 1] \notin dict$ **then**
      if $token[0 : end - 2] \notin dict$ **then**
        $d[token] \leftarrow$ *True*;
      else
        $token_{new} \leftarrow token[0 : end - 2]$;
      end
    else
      $token_{new} \leftarrow token[0 : end - 1]$;
    end
  else
    $d[token] \leftarrow$ *True*;
  end
else
  $d[token] \leftarrow$ *True*;
end
return $token_{new}$;

Algorithm 1: Rule-based naive Token Clustering Algorithm

3.1.4 Named-Entity Recognition

Named-entity recognition (NER) is a subfield of entity extraction within language processing, the general task of NER is to extract and label entities within a text. These labels are predefined through labeled training data, and can thus be

[12] the default tagger in NLTK is called `maxent_ne_chunker`
[13] https://catalog.ldc.upenn.edu/LDC2005T09

Figure 2: Result from using Stanford Named-Entity Recognizer

redefined as long as enough labeled data is available. There is not one single approach for Named-entity recognition, there are both linguistic grammar approaches as well as probabilistic machine learning methods. In this paper we will use a probabilistic machine learning method through a package that has been developed by The Stanford Natural Language Processing Group and Jenny Rose Finkel & Manning (2005). They are using Linear Chain Conditional Random Field Sequence Models (linear chain CRF) which is able to predict a sequence of labels, thus highly effective in language processing. Linear chain CRF were pioneered by John Lafferty & Pereira (2001) and it is their research that is the foundation of the Stanford model. The Standford Named-entity recognizer is trained using a mixture of the CoNLL[14], MUC-6[15], MUC-7[16] and ACE[17] data sets. The larger model is able to successfully identify and label the categories of: Location, Person, Organization, Money, Percent, Date, Time. In this paper we are using one of the smaller models that identifies locations, organizations and people. Below is an example result of NER on Donald J. Trumps tweet from 28 of April 2017.

This example highlights interesting aspects of NER and how the three categories are defined. In this example Stanford's NER is able to identify North Korea and China as locations, however, "President" is not identified as a person. In this specific sentence president is being referred to as a person but for some reason the linear chain CRF fails to recognize it as such. As mentioned NER is a probabilistic and as we are using a light weight version we wont get perfect results.

3.1.5 Comparison between Stanford-NER and NLTK-NER

To validate the choice of the Stanford-NER it was compared to the default NER available in the NLTK toolbox. The results are shown in Table 3. The results are much better using the Stanford-NER but it is also slower. Because accuracy is more important we choose to keep Stanford-NER. An example of the words returned for a specific search is given in Fig. 3.

Figure 3: Word-clouds corresponding for the search: *election*. Both word clouds are similar but the Stanford (right) NER is more accurate than the NLTK one (left)

[14] http://www.conll.org/
[15] http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html
[16] http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_toc.html
[17] https://catalog.ldc.upenn.edu/LDC2006T06

Field name	Method	Accuracy	Top errors
Locations	Stanford	0.97	Trump, Atlantic, Miss USA
	NLTK	0.86 [18]	Best, Black, Dean
Organisations	Stanford	0.94	United, Aberdeen, Trump National Golf, Trump International Golf
	NLTK	0.49	BarackObama, MittRomney, MAKE
Persons	Stanford	0.99	Fox
	NLTK	0.77	Enjoy, Fox, Happy

Table 3: Table comparing the accuracy results of the default NLTK and Standford Named Entity Recognition classifiers. All results are obtained using the 100 most occurring terms.

3.1.6 Geo-Localization

In order to visualize locations that have been extracted for the Named-entity recognition we want to find their corresponding coordinates. This was done with the help of the python package geopy which accesses the Google API to request coordinates. When conducting this conversion between locations and coordinates we weren't able to convert all locations. This was partly due Trump misspelling locations or that the NER had extracted to specific locations e.g some of Trumps hotels.

3.2 Elasticsearch and Kibana

Elasticsearch, Gormley & Tong (2015), is a full-text search engine with an HTTP web interface and Json based data trasfer. There are official clients for a variety of different programming languages, among others python, and it is distributed open source under the Apache Licence. Kibana, Elasticsearch (2016), is an interactive graphical tool for generating viasualizations based on the information provided by the Elasticsearch search engine.

4 Results

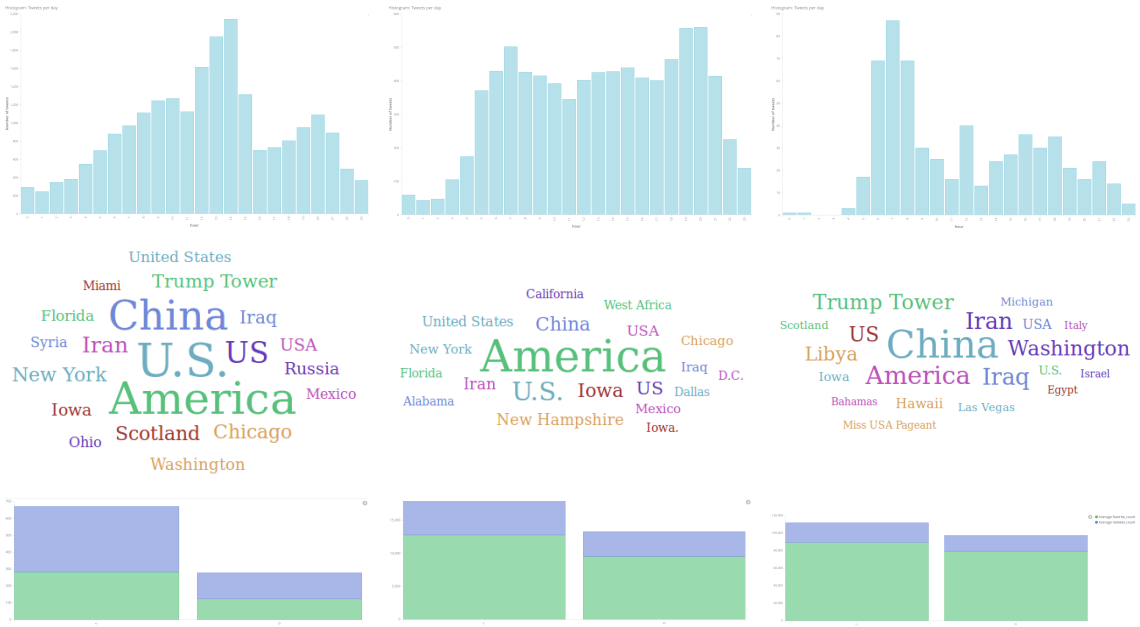
Donald Trump's tweets are revealing about his, and his supporters', reactions and mentality, especially when the results obtained by data analysis are combined with other related news stories and events. One of the recent turning points during Donald Trump's twitter career is his presidential election. The campaigning was formally launched on 16 June 2015 and ended with the election on 8 November 2016. The results presented here are just an example of what information can be gained from visualizing the data.

Fig. 4 displays data about Trump's Twitter activity relative to the election in a 3x3 grid format, where each column represents a different period of time and each row illustrates extracted information using different formats.

An interesting change can be seen in the histograms, how the tweeting has been concentrated to the morning hours after the election. Before he presented his candidacy, his activity was concentrated during the afternoon while his morning activity was very poor. This is probably due to Trump trying to get coverage before the election and then reverting to work tasks that do not allow a constant tweeting. The lack of tweets during the small hours is most likely a consequence of him traveling less abroad after becoming the president. We also note his almost-uniform activity during the elections, probably meant to make himself visible to as many people as possible.

The third row in Fig. 4 also shows an interesting development in connection to the presidential election. Before Trump declared his candidacy the relation between the appreciation of his more positive and more negative tweets was more skewed towards the negative tweets than after his candidacy had been declared. The scales are, however, radically different on the three graphs as it is apparent from Fig. 5, displaying the average number of shares and re-tweets over time. There are many possible reasons for this, for example that he changed focus from mostly focusing on himself and his products to more politicized questions, another is that he gained supporters that wanted to aid his campaign by re-tweeting and favoring his message.

Furthermore, from the word-clouds in Fig. 4, we observe that the geographical focus of his tweets also changed over time, he seems to be more focused on the U.S. during the election process than after becoming president. This is probably because his focus during the election was mostly about making *America Great Again*, but his role as president is an international one. Interesting to note is that Mexico does not show up in the word cloud after the election, probably because Trump has had no success in building the wall. The overall global spread of Trump's tweets can be seen in Fig. 6, from which it is apparent that Trump is tweeting about global affairs, but mainly about the U.S.



(a) Before election (b) During election (c) After election

Figure 4: The first period of time comprises from 2009 until the official start of the campaigning, the second and third periods are during the election process and the period after the election, respectively. First row: Histogram depicting the hourly Trump's tweets distribution. Second-row: Word-clouds displaying the most frequently mentioned locations in his tweets using NER. Third row: Bar graphs describing the relative distribution of re-tweets and favorites depending on the text sentiment (blue: average number of re-tweets, green: average number of favorites; left: low sentiment score, right: high sentiment score).

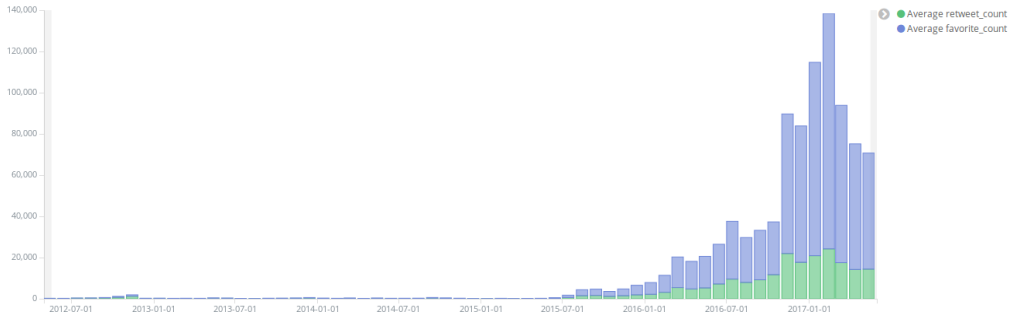


Figure 5: Monthly average of re-tweets and favorites per month. Blue is favorites and green re-tweets.

To add some insights, Fig. 7 displays a bar graph of the sentiments over time, estimated using VADER toolkit. The combined height of the bars show the total relative number of tweets and the colors the relative part within a a specific sentiment range. Interestingly, Trump most intensive tweeting periods were not during the election period and the total number of tweets has been steadily decreasing since the end of 2015.

Finally, Fig. 8 displays the relative frequency and sentiment of tweets including the terms *Clinton*, *Obama*, *fake news* and *thanks*. As can be expected the tweets containing the word *thanks* are generally more positive than the other terms that are either his political opponents, or as in the case of *fake news*, something inherently bad. The usage of the term *Clinton* does as expected increase over the election period, but surprisingly the relative sentiment is more positive towards the end of 2016. Another notable relation is that Trump stops using the word *thanks*, not only in absolute numbers, but also relative to his total tweeting.

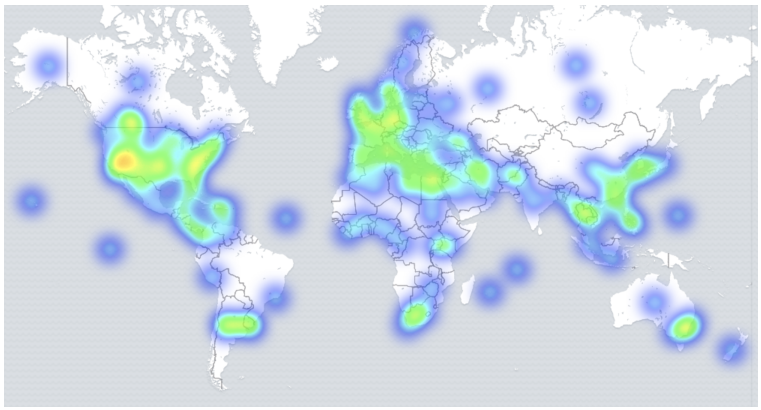


Figure 6: Geographic visualization of locations extracted from his last 5 years' tweets using Stanford NER.

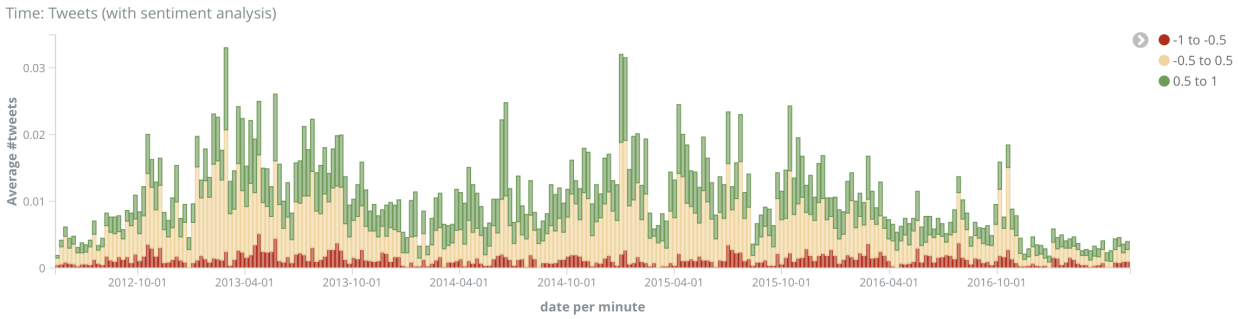


Figure 7: Sentiment of Donald Trump's tweets using VADER toolkit.

Figure 8: Sentiments of a few terms over time. UL: "Clinton", UR: "Obama", BL: "Fake news", BR: "Thanks".

5 Discussion

It is apparent that much information can be gained from collecting and analyzing data. In this project we have focused on information in connection to the election process, but any event can be analyzed. Or just looking at trends in the data. Examples of other queries that can be interesting to visualize are how Trump's sentiment differs depending on location, if his general position on international conflicts can be evaluated based on the sentiment of his tweets.

REFERENCES

Ahmadian, Sara, Azarshahi, Sara, and Paulhus, Delroy L. Explaining donald trump via communication style: Grandiosity, informality, and dynamism. *Personality and Individual Differences*, 107:49–53, 2017.

Baucom, Eric, Sanjari, Azade, Liu, Xiaozhong, and Chen, Miao. Mirroring the real world in social media: twitter, geolocation, and sentiment analysis. In *Proceedings of the 2013 international workshop on Mining unstructured big data using natural language processing*, pp. 61–68. ACM, 2013.

Bird, Steven. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pp. 69–72. Association for Computational Linguistics, 2006.

Bovet, Alexandre, Morone, Flaviano, and Makse, Hernán A. Predicting election trends with twitter: Hillary clinton versus donald trump. *arXiv preprint arXiv:1610.01587*, 2016.

Bradley, Margaret M and Lang, Peter J. Affective norms for english words (anew): Instruction manual and affective ratings. Technical report, Citeseer, 1999.

Bruns, Axel and Burgess, Jean E. The use of twitter hashtags in the formation of ad hoc publics. In *Proceedings of the 6th European Consortium for Political Research (ECPR) General Conference 2011*, 2011.

Bruns, Axel and Stieglitz, Stefan. Towards more systematic twitter analysis: metrics for tweeting activities. *International Journal of Social Research Methodology*, 16(2):91–108, 2013.

Chang, Hsia-Ching. A new perspective on twitter hashtag use: Diffusion of innovation theory. *Proceedings of the American Society for Information Science and Technology*, 47(1):1–4, 2010.

Elasticsearch, BV. Kibana, explore & visualize your data, 2016.

Gormley, Clainton and Tong, Zachary. *Elasticsearch: The Definitive Guide*. " O'Reilly Media, Inc.", 2015.

Hao, Ming, Rohrdantz, Christian, Janetzko, Halldór, Dayal, Umeshwar, Keim, Daniel A, Haug, Lars-Erik, and Hsu, Mei-Chun. Visual sentiment analysis on twitter data streams. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pp. 277–278. IEEE, 2011.

Hutto, Clayton J and Gilbert, Eric. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth international AAAI conference on weblogs and social media*, 2014.

Jenny Rose Finkel, Trond Grenager and Manning, Christopher. Incorporating non-local information into information extraction systems by gibbs sampling. 2005.

John Lafferty, Andrew McCallum and Pereira, Fernando C.N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

Kwak, Haewoon, Lee, Changhyun, Park, Hosung, and Moon, Sue. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pp. 591–600. ACM, 2010.

MacDonald, Kent and Villamayor, Julius. Trumptweets: Using machine learning algorithms to classify tweets by donald trump. 2017.

Page, Lawrence, Brin, Sergey, Motwani, Rajeev, and Winograd, Terry. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

Pak, Alexander and Paroubek, Patrick. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, 2010.

Pang, Bo, Lee, Lillian, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.

Pennebaker, James W, Francis, Martha E, and Booth, Roger J. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001, 2001.

Ratnaparkhi, Adwait et al. A maximum entropy model for part-of-speech tagging. In *Proceedings of the conference on empirical methods in natural language processing*, volume 1, pp. 133–142. Philadelphia, USA, 1996.

@realDonaldTrump. Donnald j. trump official twitter account. https://twitter.com/realdonaldtrump. [Online; accessed 19-May-2017].

Yang, Changhua, Lin, Kevin Hsin-Yih, and Chen, Hsin-Hsi. Emotion classification using web blog corpora. In *Web Intelligence, IEEE/WIC/ACM International Conference on*, pp. 275–278. IEEE, 2007.