Geolocating Twitter users using Twitter meta data

Oscar Rydh
Computer Science
Faculty of Engineering
Lund University
Lund, Sweden
Email: psy13ory@student.lu.se

Joel Klint
Computer Science
Faculty of Engineering
Lund University
Lund, Sweden
Email: dat13jkl@student.lu.se

***Abstract*—The purpose of the project is to geolocate tweets. To set a scope, we will focus on tracking Trump's reach on Twitter. We have applied a model where we, with a modular probabilistic approach, calculate the position of a Twitter user. We have managed to geolocate Twitter users with a accuracy of 63% based on data collectd over a seven day period**

I. Introduction

As the digital age emerges as the next industrial revolution we as humans have become more interconnected then ever before. One large phenomenon that has tagged along this new age is social media. Today large business such as Facebook and Twitter have millions of users and earn billions of dollars every year. Social media has gone from a hipster and niched area, to engaging large departments on marketing bureaus and being used as a major influence tool by political parties, governments and representational people. Therefore, it is of great interest to study each social media platforms impact on our world, and how it reacts.

This study is made as a part of a project course in computer science at the faculty of engineering, Lund University. The goal of the course is to allow students in computer science to test their skills as master students on unsolved problems provided by the department.

II. Background

To delve into the depths of all platforms is an immense task. One is bound to be lost in the data, drawing conclusions where none exist, if one even has the time to study the technical details of each area. Instead we have in this study focused on one platform, Twitter.

A. Twitter

Twitter is a platform designed to create interactions via small statuses called *tweets*. Each user has the ability to post tweets, which are up to 140 characters long. The content of a tweet is allowed to be almost anything at all, and mostly depends on the users interest. For example, a regular user is Swedens former foreign minister Carl Bildt. His tweets often contain political remarks, satiring or commenting the latest news in the area.

The reason you are even on twitter is to spread your tweets to your intended audience. There are multiple ways of achieving this. The first and foremost is by the use of followers. Each user has the ability to follow any other user. This results in the user who decides to follow subscribes to all tweets coming from the user he is following, effectively getting a notification each time something is written. This allows each influencer to build a follower base to which he can spread his message.

Alternatives to pure following is the ability to retweet someone else's tweet. This means that if a user likes what another has written, they can share it to their followers using the retweets. The retweets themselves can be designed in mutliple ways. They can be responses to what was in the original tweet, just pure passing along of the tweet, or sharing it with their own network. Using these functions a tweet can reach around the globe, spreading from user network to user network thereby impacting a multitude of people.

B. Geolocation

As each tweet has the ability to make a global impact, we took interest in locating users based on a tweet. If we were able to create some sort of system were one is able to see which users have interacted with a tweet, it would be possible to somewhat track the actual reach of an influencer. This could then be used to see which users who make an actual impact on the world, with the use of twitter.

Though designing such a system is not trivial. For example, twitter does not supply any exact location data. Instead it gives some meta data, as well as the data each user freely supplies. Therefore, some form of system is needed which uses the available data and generates a location.

C. Related Work

Trying to geolocate tweets is nothing new, and has basically been attempted since twitter became an international phenomenon. While reaserching related work there were basically two approaches which sparked our interest. The first was a study made at IBM were they trained classifiers in a neural network to predict user locations based on the twitter data.[2] We discussed using the same approach for this study, but

quickly concluded that it was out of scope for this course in the needed time consumption.

Instead we took inspiration from another study which was conducted by collaboration of two German universities.[1]. The Shchulz et all. tried to geolocate the tweets using an approach they called the *Multi-Indicator Approach*. The main idea was to identify multiple spatial indicators in the meta data and information around a tweet and its user, which then could be used for geolocation. As this approach intrigued us more, and felt more in line with the scope of the project we decided to take much inspiration from this paper.

III. Research Question

Most of the previous work has been related to geolocating tweets in general. One mayor benefit in doing this is that you get a broader dataset containing much information. However, we were interested in if it was possible to locate users in specific areas. Since we have a great enthusiasm in politics, we decided to limit ourselves to this area. The easiest way to do this, we believe, is to follow Donald Trump. This is a man who as used social media platforms to great effect in building opinion and influencing the United States, and the world. For example, he contributes his success in the American Election to twitter and social media itself. [3] And still, even after elected, he keeps creating headlines around the world with his controversial tweets.

Therefore, to limit the scope of the project, we decided on the following:

Is it possible to geolocate the users interacting with Donald Trump on twitter?

IV. Method

We decided to take an offline approach to the problem. This resulted in the outcome of three distinct phases.

1) Collect data
2) Geo locate data
3) Visualize result

A. Phase 1: Collect data

Twitter has three API:s that are of relevance for this project.

1) REST API
2) Streaming API
3) Firehose API

The REST API works with historic data but is heavily restricted on the amount of data that can be fetched. That leads us to Firehose as it is unlimited in the allowed amount of data fetching, but it comes with a massive price tag. This narrows our choice down to the streaming API.

With the streaming API, we construct a filter of data we are interested in and establish a constant connection which allows Twitter to push data to us, as it happens in real time. We designed a PostgreSQL database to hold this data. After this preparation phase, we started the collection of data and let the process go on for a week. We experiences some connection issues at unfortunate times which resulted in hiccups in the continuity of our data. Se figure 1 for a report of the times when we collected data.

Status	Start	End
Included	2017-03-27 17:34:00	2017-03-31 02:49:00
Included	2017-03-31 20:00:00	2017-04-01 00:21:00
Included	2017-04-01 01:09:00	2017-04-01 07:46:00
Included	2017-04-01 09:32:00	2017-04-02 01:27:00
Included	2017-04-02 17:33:00	2017-04-03 08:51:00

Fig. 1. The times that we captured data from twitter, on a minute accuracy level

The following is a report of what data we received from Twitter on tweet level.

- Text
- User
- Place
- Whether is is a retweet/response
- Language

The following is a report of what data we received from Twitter on user level.

- Screen name
- Location
- Description
- Followers/Following count
- Timezone
- Language
- Timezone
- URL

All in all, this resulted in 1144233 tweets, from 331887 distinct users. 36 of these tweets were from Donald Trump.

B. Phase 2: Geo locate data

Now we were to figure out how to translate all this data to a location on earth. We read papers form earlier studies, looking for inspiration. We liked Schulz [1]idea as we wanted a modular solution. We decided to create our own, simplified version of that model given the limited time of the course. The model works like this. We create a matrix representing the world. We then build a couple of functions that takes interesting data values and translates them to bounding boxes that represents possible locations. By stacking these layers on each other, we create altitude curves, that represents the probability of the user being at a certain location. We then assume that the user is on the tallest point.

There is a trade off regarding the size of the matrix that we experimented with. The larger the matrix, the higher precision is allowed, but the longer the time of calculation. We started out with a $360*180$ matrix, which gives a pretty low accuracy. We moved further on to a $3600*1800$ matrix but this increased the calculation time by a large factor. Since we worked on a country level (state for US), we decided on settling for the $360 * 180$ matrix since its accuracy was enough for us. Now we will talk a bit about each layer we developed, and how we constructed them.

1) Language: Twitter returns language definitions in various formats. Character length varied between 2 and 5, so we started by clipping the length at 2. This means e.g. "en-AU" and "en-GB" will be translated to "en". The reason for this was that we was not sure how much we could trust the language information. Geonames[5] has a file *countryInfo.txt*, which provides you with languages that a country speaks, given a language. We want it the other way around, so we started by creating a dictionary that reversed the order of information. With this dictionary we could quickly find what countries spoke a language. After this, we used a python package [4] that returns interesting information of countries. We used this to get bounding boxes for all relevant countries, which is what defined this layer.

2) User Timezone: Twitter returns information regarding the timezone of the user. How they receive this data is still unclear, but we have two theories.

- Automatically received based on the position of the user
- Manually decided by user in its profile

This information is a string with various formats.

- *Continent/City*
- *Country/City*
- *City*
- *Special cases (Such as Eastern Time(US&Canada) and UTC*

There is a well established database of time zones and their distribution in the world called TZDB[6]. It is created by IANA(Internet Assigned Numbers Authority) and intended primarily for use in computer programs. We used the file *zone.tab* to find what countries use a given time zone. After that, we used the bounding box python package[4] to find the bounding boxes. That defined this layer.

There were some twitter time zones that was not defined in this file. We listed these time zones sorted by frequency in our database. We then created custom bounding boxes for all time zones with a frequency larger of 140 or greater. The rest is left undefined, and no bounding boxes is returned.

3) User Location: Twitter returns information regarding the location of the user. This information is in the form of a string and Twitter gathers it by allowing their users to set this manually in their profile. It can contain any string such as "The moon", "California, US" or "x+a0d99/". To translate this information to a position is a challenge, and a big part of the challenge is to determine which strings are true positions and which are gibberish. We used Geonames[5] for this as well. Geonames provides an HTTP API for it's data. We used the endpoint $http://api.geonames.org/searchJSON?q=<INSERT-QUERY-HERE>$. This returns the most relevant entry, from which we chose to save the geoname_id. This is a unique id that every entry in the geoname database has. We then downloaded the entire database *allCountries.zip*. This way we have all information from the entire database locally. Each user received a point this way, and not a bounding box. We decided to pad this point in order to make it an area. We experimented with different values and decided on padding the point with two degrees in each direction. That defined this layer.

4) Extra tweets: For users which we had gathered less than three tweets, we fetched more information via Twitter REST api. We fetched up to the 20 latests tweets for the user, and created bounding boxes based on the information from those tweets. We use the tweet language and timezone. This layer were to generate a fixed height of its bounding box. To achieve this, every tweet generated a bounding box, weighted depending on the total amount of tweets. Lets take an example to explain it in better detail. Lets say one user has 30 tweets. All of these has undefined time zones. 20 are in language English and 10 are in language Spanish. 2/3 of the total height will be in bounding boxes where English is spoken, and 1/3 of the total weight will be where Spanish is spoken. We translated the data values to bounding boxes with the same method as for user data described above.

5) Translate probability to a geographic point: We use a rather naive method for extracting a point out of the probability. All our bounding boxes are squares, which resulted in the top areas being squares as well, since we only used three layers. This is how we find the middle point. We find all coordinates with max value in the probability matrix. If there are more than one, we take the position which is in the middle.

$$midlat = minlat + \frac{maxlat - minlat}{2}$$

$$midlong = minlong + \frac{maxlong - minlong}{2}$$

This does not always result in the correct point though. If there are two separate polygons which are both in the max value, the wrong result will be returned. We added a safety check, that tests whether our calculated point has the max value of the matrix. If it does not, we save that to the database. Further than that, the case in left unhandled.

C. Phase 3: Visualize result

For the last phase, we plotted all calculated points on a map. Google Maps provides an API [7] which allows anyone, to plot points on a map free of charge. It also contains support for marker clusters, which means that one zoom out of the map, close markers will be clustered and visualized as a number, representing the amount of clustered markers in the area.

We constructed a website which presented aggregated results such as amount ratio between retweets, mentions and responses as well two maps. One map which displays all results instantly, and another map which displays them synchronous as they were created. We created this to give a sense of how the tweets were spread over the world.

V. Result

To calculate the accuracy of our model, we took a random sample and verified the positions. Data Undefined means that the there was not sufficient data to get a position at all.

Name	Amount	Ratio
Clean retweets	386 564	0.338
Commented retweets	209 341	0.183
Reply tweets	523 420	0.457
Mention tweets	21 287	0.019
Undefined	3 585	0.003
Trumps tweets	36	3.146×10^{-5}
Total Tweets	1 144 233	1

Fig. 2. Aggregated data from our database

Status	Amount
Correct	253
Incorrect	86
Data Undefined	62
Total	401

Fig. 3. A random sample describing the accuracy of our model

Based on this, we have a 63% $\frac{Correct}{All}$ chance of correctly finding the position of a random Twitter user. We also have a 75% $\frac{Correct}{All-Undefined}$ chance of correctly finding the position of a random Twitter user, given that a location can be found at all.

For a the visualization see figure 4.

VI. Discussion

The goal of this study was to geolocate users which interact with Donald Trump. This was done using the meta data provided by twitter stream api which was preprocessed and run through multiple services and programs. Though as in any projects it was not without some road bumps and hiccups.

A. Donald Trump as an influencer

Before we analyze the accuracy and efficiency of the geolocation techniques used in this study, it is quite interesting to take a step back and just look at the data gathered from twitter revolving the latest edition of the American president. Firstly, as seen in the results we gathered 1.1 million tweets from almost 330 thousand unique users revolving Donald Trump under just one week. Of these Trump was able to author 36 tweets which is about 5 each day.

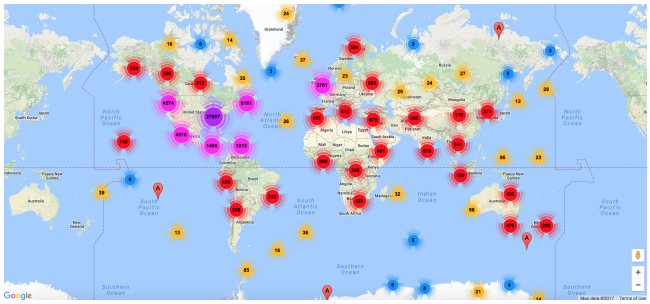

Fig. 4. The visualization result

.

A first analysis one can do is comparing the amount of reached users with the total number of monthly active users on twitter, which as of Q1 17 is equal to 318 million. [8] When looking at the percentage reached of all the active users, it is astonishingly low: 0.00001%. But this is quite a naive approach on looking on reach, since each user has a follower base. Every time a user interacts with trump, it will spread to all of its followers. When looking at the average amount of followers for the users in our database we get a count of approx 1200 users. Multiplying this with the total amount of users we get a reach of almost 400 million users, which is much more than the active user base. This if course an overblown number, since each users follower is not unique to that user. Still, lets assume that each user have a unique follower count of just 5%, resulting in 60 users. This would result in a reach to about 20 million users. As a comparison, it is estimated as of 2016 that there are 40 million weekly active users in the US. [9] [10]

One can also look into each tweet Trump makes and check its reach with the same approach. Since each user only can make one retweet, mention, etc for a tweet we can take the average amount of interactions during a week and multiply it with our estimated amount of unique followers. When looking through Donald Trumps twitter between May 20 - May 27 2017 we have an average of about 28 thousand interactions. Lets say that the users interacting have the same amount of followers as the ones in our data base. This would give Trump an average reach of about 1.7 million users for each tweet. Interesting is that Trump has a follower base of about 31 million users. So that leaves a question for future work, who are all of Trumps followers and how come not more react to Trumps tweets?

B. Geolocating a tweet using twitter meta data

As seen by our results we have been able to geolocate tweets with an accuracy of 65%. This was done via taking a random sample set of 400 geolocated values and comparing the estimated position with the meta data, were user location took precedence over the other. If the point was estimated to the same country (state in the US case) it was marked as correctly estimated. If the user location was undefined or some ambiguous string it was marked as undefined. The accuracy was then calculated as the number of correct positions divided by all estimations.

As noted, some data contains user locations which are undefined places in the world. Examples of such locations are: ”Turn on Post Notifications!”, ”Nicki Minaj Follows” and ”Fantasy Land”. If these samples are removed from the calculations we increase the accuracy rate to about 75%. This essentially means given valid data, we will predict the country or state a user is in 75% of the time.

1) Why not more layers?: The first question one might ask is why did we no include more layers in the algorithm to increase its precision. There are multiple reasons for this. Schulz et all describes the tweet text as the most significant layer [1], so how come we have excluded this layer? The reason for this is due to the area on which we implemented

the algorithm and the political nature of Donald Trump. In the previous work tweets were analyzed on a global scale, not limiting themselves to any area. This meant that tweet text might include statuses such as ”Having a lovely day by the Eiffel Tower” or ”The Niagara Falls are truly majestic”. In these tweets it is possible to draw conclusions on were a user might be, since the text contains key places around the world with well placed coordinates.

If we instead look at some random tweets gathered of users interacting with Donald Trump we have examples such as: ”We’re going to make America greater than ever before #believe #MAGA #mondaymotivation” and ”We need clean power plants!!! Stop destroying our planet.”. From these kind of tweets it is impossible to isolate and gather any form of localization indicators. Instead most tweets are directed at Trump either praising or condemning his tweets and actions. From our point of view this is kind of unfortunate since one of the most influential components of the algorithm could not be used. One could instead argue that political tweets are not a useful source for localization data.

Another reason to that no more layers were built were that we could not extract any useful localization data from that provided by twitter. Looking back to the list of the meta data labels the ones worth consideration are the ”description” tag and ”location”. The reason for not using the description tag was that, just as for the tweet text, we saw no useful indications when looking through them. The reason for not using the ”location” tag is more interesting. In some sense, using this field would solve all our problems, removing the need for any algorithm. Unfortunately the case is that twitter basically never gives this field, making it impossible for us to use.

2) Precision, picking data points and visualization: When implementing the algorithm we limited the precision of our localization to degrees without any decimals. The main reason for this was due to computational complexity and our limited time. This has some consequences. The most intuitive is that the localization is limited to a set amount of points on the map, essentially making it a discrete 180X360 grid. This means that our estimated locations are limited to one of the 64800 available points on the map. This is quite unfortunate, since many location fields are quite detailed all the way down to city level. We are confident that given more power, and some implementation optimization we could improve this algorithm such that a users city could be located. Instead, now we need to settle on only looking at countries and US states which still gives quite a nice picture of our data.

Still, another consequence of the use of imprecise coordinates is that locations close to country boarders, or very small countries, are hard to use. Lets take a country like Switzerland for example. Pinpointing this country with our low precision is quite hard, since the coordinate must be located the closest to one of the available points in that country. Another example is the city London. Here there are no available points right on the city, making it look like thousands of users are on the cities country side.

Finally, there are some points on the map that contain many more users then others. These are often either in the middle of countries or on random locations throughout the world. There are two reasons for this happening. The first case is a result on the way we are picking the final location to display in our visualization. Often when the algorithm has completed we get a large set of max points creating a bounding box for the most likely location. Since we do not want to print whole bounding boxes on the map we somehow need to pick a single point from all the available. The way we decided to do this was simply to pick the middle one in the middle. So when any user for example specifies its location to ”USA” it will plot it in the middle of the United States. As this is done for all user with the same locations, many users will be gathered in one point.

The reason for the random locations are partly the same as as above, though it is not the full story. The random locations are often a result of undefined data getting a hit in the geonames api. For example any sentence begging with ”In” will be set to India. When this then is combined with the other layers, locations which appear quite random are decided. To solve this issue someone needs to find a way to classify real locations, which we did not do in this project.

3) Using user data and its consequences: Throughout this project we have only worked with user defined meta data. This also has some mayor consequences, since we never get any official and perfect data from twitter itself. Instead we base all our geolocations on something a user sometime somewhere has written. Therefore, if a user says he is in Antarctica, we will place him in Antarctica. This results in us placing many users in places which are highly unlikely, such as North Korea, he Poles and the middle of Sahara. Though, if you do not pay twitter to gain access, we have not found any way around this problem.

VII. Future Work

There are many paths which we needed to leave unexplored due to the limited computing power and time for this project. One interesting new approach we believe to somehow incorporate machine learning and neural networks to decide what the values should be for each of the layers. The way we did it in this project was more or less based on reasoning and heuristics.

Another mayor project would be to implement some form of reasoning for a location and if it is reasonable. By doing this you would somehow be able to weight different layers on it’s likelihood. For example, if a user location is ”Antarctica” while its time zone is ”GMT” one would assume it’s more likely that this person is somewhere in the UK, since they have more users and such. How one could do this generally we do not know, and is probably quite a hard task.

It would also be interesting to see this approach added on completely different problems. One example we could think of was trying to find a persons true political position, by for example creating layers which indicated that a person was more likely to vote for specific types of politics. The problem here is to identify a political position in a tweet text

or description and encoding it in some form of layer. Finally one might combine these layers to get a probability on which party a person was most likely to vote

Finally, one could try to combine the twitter data with more data from other social networks a user is using, like Face Book or Instagram. By somehow identifying the users on all the platforms, one could combine the data to increase its precision and drawing conclusions on multiple areas.

VIII. Conclusion

In this project we have applied parts of the model described by Schulz et all [1] on a more limited subject, namely politics. We have also visualized the results with the help of Google maps. Finally, we have analyzed the accuracy of the model, and discussed its limitations when applied to such a narrow concept as following a single politician. We have produced a result which efficiently and quite accurately identifies a users location based on the meta data provided by Twitter. There are still some limitations which needs to be addressed, mainly the dependency and the need on trusting user specified data. Still, when looking at the visualization it is an clear that Donald Trump influences and impacts Twitter users all around the globe. As this is one of the first studies which tries to geolocate tweets based on politics we look forward to see other researches testing similar approach and creating more precise solutions.

Acknowledgments

We would like to thank our project supervisor Anamaria Dutceac Segesten at Lund University, for all the insight, interesting discussions, and of course for coming up with the idea of geolocating tweets.

We would also like to thank the authors of the paper *A Multi-Indicator Approach for Geolocalization of Tweets*[1] for their method.

References

[1] Axel Schulz, Aristotelis Hadjakos, Heiko Paulheim, Johannes Nachtwey, and Max Mhlhuser, *A Multi-Indicator Approach for Geolocalization of Tweets*, Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media

[2] Jalal Mahmud, Jeffrey Nichols and Clemens Drews, *Home Location Identification of Twitter Users*

[3] Lesley Stahl, *President-elect Trump speaks to a divided country* http://www.cbsnews.com/news/60-minutes-donald-trump-family-melania-ivanka-lesley-stahl

[4] Python package: country-bounding-boxes 0.2.3 https://pypi.python.org/pypi/country-bounding-boxes/0.2.3

[5] Geonames: Database of geographic data in the world http://download.geonames.org/export/dump/readme.txt

[6] TZDB: Databsae of time zones and their distribution in the world. https://www.iana.org/time-zones

[7] Google Maps: API for plotting geographic points and clustering them. https://developers.google.com/maps/documentation/javascript/marker-clustering

[8] Statistics on monthly active Twitter users https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

[9] Statistics on the frequency of active users in the United State https://www.statista.com/statistics/234245/twitter-usage-frequency-in-the-united-states/

[10] Statistics on the estimated amount of American Twitter users https://www.statista.com/statistics/232818/active-us-twitter-user-growth/