ADA project - Mining democracy

Ahmad Ajalloeian, Gergely Odor, Ramtin Yazdanian
EPFL, Switzerland

I. Introduction

Twitter is a tremendously popular social network (also notorious if you count Donald Trump's account), based on 140 (recently changed to 280) character pieces of text called ”tweets”. Twitter has not only been used in a more personal capacity, but also in the direction of political agendas and activism.

The Swiss tweets dataset offers many intriguing questions, especially given the fact that it has geolocation tags. First of all, the locations allow us to understand how Swiss tweeters are distributed throughout the country. Secondly, given Switzerland's relatively rare status as a country with multiple official languages, tweets by Swiss users offer a unique opportunity to gain insight into the relationship between the languages, contents, and locations of tweets. In addition, we are interested in gaining insight into Swiss political activity on Twitter and its distribution across languages and locations. This project will aim to find answers for - or at least gain insights into - the questions listed below.

A. Research questions

1) Does the density of the location of tweets correspond to population densities? Or are the tweets significantly more concentrated in cities?
2) Can we reconstruct the Röstigraben only based on the language of tweets?
3) Are there any spikes in activity and do they correspond to particular events (e.g. referendums)
4) How involved are swiss people in politics on twitter? Which areas are most involved?

B. Literature review

Related to the last question we did a short literature review. We did not find anything on the political activity of the population, however we found articles on the twitter activity of Swiss politicians. It has been reported that in Switzerland, politicians were quite late to adopt twitter; by the end of 2012 only 30% of Swiss politicians had a twitter account [**?**]. However, this figure has seamed to changed since 2012 since according to a recent news article “For Swiss politicians Twitter profiles are no longer a novelty” [**?**].

In the final days before the deadline, we found a preprint that had a large overlap with our fourth research question [**?**]. The only difference between this work and ours is that we worked with a downsampled local dataset, whereas the authors there used the Twitter API to specifically download the political tweets. Also large part of the search for political tweets in [**?**] were done manually.

II. Methods

The code for the project is available at [**?**].

A. Acquiring the datasets

We primarily use with the swiss tweets dataset provided by the ADA team (originally by Swisscom). The tweets were queried using a bounding box around Switzerland, which means that not all tweets are actually swiss. We will explain how we filter the ones outside of Switzerland in section II-B2.

We also used several supporting datasets, which were not straightforward to locate and acquire. The geojson file for the boundary of swiss municipalities was downloaded from [**?**]. The population data was acquired from [**?**] and the population densities were downloaded from [**?**]. For detecting political activity we used the data of [**?**] available at [**?**].

B. Preprocessing data

1) Reading the data: Parsing the provided dataset was in itself a demanding challenge, so we decided to dedicate a subsection to it. The data was provided in the tab separated values format; columns were separated by \t and rows by \n. However, many of the tweets contained tabs and newlines, these were denoted by \\t and \\n. To make it more complicated the last column (userLocation) sometimes ended with a large number of backslashes which resulted in further ambiguity. Since even after resolving all these exceptions and a few others, we still got rows with varying length, we decided to follow a different approach.

The first three columns of the dataset (id, userid, createdAt (date)) were all not nullable and very robust in terms of syntax. We decided to match a regular expression on

```
\d{4,}\t\d{4,}\t\d{4}\-\d{2}\-\d{2}\s\d{2}:\d{2}:\d{2}
```

which has worked with 100% accuracy. This allowed us to find the beginning of each row, however, separating the cells was still difficult. We decided to split on \t, and merge together the cells [3:-16] as the text cell, since we expected that the last 16 and the first 3 columns had no \t, only possibly 4^{th} (text) column. It turned out that we were wrong, the last (20^{th}) column (userLocation) also had \t, but these exceptions only involved less than 20000 datapoints (out of $\sim$ 20 million), so we neglected these. We will see later that this is a tiny fraction compared to the fraction of datapoints we had to exclude due to other issues with the dataset.

2) Processing geolocation: Out of the $\sim$ 20 million tweets, about $\sim$ 18 million were geolocated. Only about 10 million these geolocations were in Switzerland (the rest came from neighbouring countries in the bounding box). Since our goal was to analyse only swiss tweets, the rest of the datapoints (about 50% of the whole) had to be dropped.

For our later analyses, we had to detect the municipality each tweet came from. We made use of the geojson map which contained the boundaries of each municipality as a polygon, and we used the *shapely* python library to detect which polygon each geolocation is in [**?**]. Since the 10 million tweets came from only less than

15000 unique locations (probably since the geolocations were not the exact locations of the users, rather maybe the location of the transmissing tower each cellphone was connected to), we could compute the municipality information in less than 30 mins.

3) Language detection: Language detection was a crucial step especially for research question 2, but also for detecting political content. For this task, we used the *langdetect* python library, a port of Google's language detection code based on a naive Bayesian filter [**?**]. This is the only task for which we had to use the hadoop cluster. Our simple spark code took 3 hours on ?? nodes.

Unfortunately, by manual inspection we saw that the accuracy of *langdetect* was far from perfect (even though the website claimed 99% accuracy). One of our concerns was that possibly the swiss german dialect could cause problems, so we downloaded a swiss german corpus and tested *langdetect* on the sentences of the text. On the collection of swiss german blogposts we saw an accuracy above 82%, whereas on the "Blick am Abig" newspaper from Zürich our accuracy was above 95% (the majority of tweets were classified as simply german in both cases). After this confirmation, we concluded that the observed errors are probably due to the fact that tweets are short and don't follow the rules of the languages very closely.

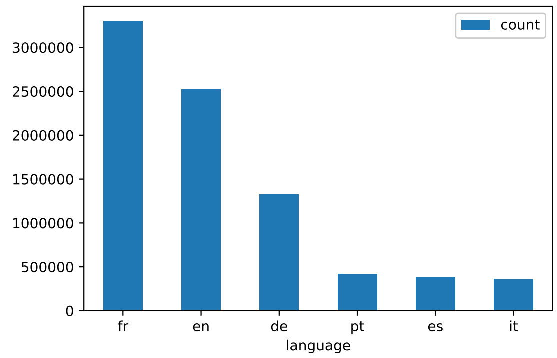

Figure 1. The number of tweets in the top5 languages in the dataset

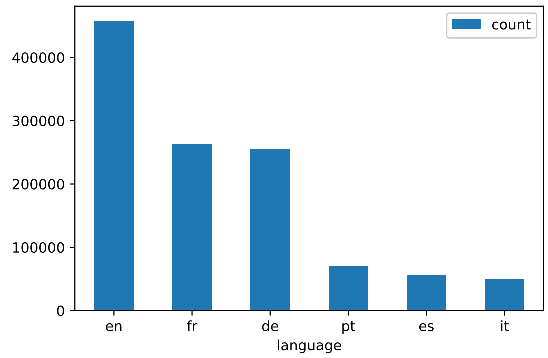

Figure 2. The number of tweets in the top5 languages only for 2016

C. Detecting political content

Our aim is to collect a set of political tweets, which then can be used to detect trends through time. Given the lack of labels, our approach to collecting these tweets will have to be unsupervised. We have two options at our disposal:

1) Collect a list of known political users (i.e. politician accounts), use their tweets as the political tweet set.
2) Using a set of known political words (e.g. ideologies, parties), collect all tweets that contain these words as the political tweet set.

The first option has the downside that politicians may also have non-political tweets, gathering the set can be time-consuming, and also that this will not capture political tweets made by the people. The second option has the downside that gathering a large list of words may be very time consuming, and that the words need to be high precision, because we do not want to capture too much noise by having words that give us many political tweets (i.e. high recall), but also many non-political ones. Therefore, to gather a sufficiently large set of tweets with high precision, we propose the following scheme:

1) Create an initial set, consisting of a few distinctly political words and party names (paying extra attention to other abbreviations that could match these party names, such as SVP - which is both a party name in Switzerland, and the shortened form of "s'il vous plait" in French).
2) Find tweets that contain words in the political word set and call it the political tweet set.
3) Compare the probability of words in the political tweet set with their probabilities in the entire tweet set using pointwise KL-divergence, which is calculated as $KL(w|d_1, d_2) = P(w|d_1)log(\frac{P(w|d_1)}{P(w|d_2)})$, where d_1 and d_2 are the two language models (the unigram model for the political tweet set and the unigram model for the whole dataset, respectively) and w is the word in question. Those words that attain a higher score are words that have a much higher frequency in the political set than in the entire set, and are also sufficiently frequent (so we don't capture words that occur only once across the whole dataset).
4) Choose the top-scoring words, add them to the political word set.
5) Repeat 2-4 as long as desired (e.g. as long as clearly low-precision words do not start appearing in the set).

Given the fact that we also have a set of politician accounts available to us [CITATION TO THE POLITICIAN ACCOUNT SET], we augment our initial set by first retrieving all "politician tweets", ranking their words (using the KL-divergence method described above) and taking some of those top-ranking words to our initial set.
Some details of our method are as follows:

- We do the bootstrapping separately for French and German, because we do not want to find French words in German tweets and vice versa. Therefore, we have one French word set and one German word set.
- Before doing the bootstrapping, we remove stopwords and punctuation.
- In order to increase precision, among the top-scoring words we only consider the hashtags and disregard the rest (both in

the politician tweets, and in the actual bootstrapping steps). Our ultimate aim is to see whether after collecting this political tweet set, significant political events (such as elections or referendums) create spikes in the plot of these tweets against time and whether by looking at those spikes, we can detect those event. This will establish our method as a framework for detecting significant political events based on spikes in political tweet activity. This will be further discussed in the Results section.

III. Results

A. Population density and tweeting location

The distribution of tweets over the municipalities is visualized on Figure 3. On Figure 4 we show a heatmap of twitter activity for only one month (over the whole dataset this was too computation intensive). From these maps show that indeed highly populated areas have higher twitter activity. However, our question was more subtle, we were interested if the

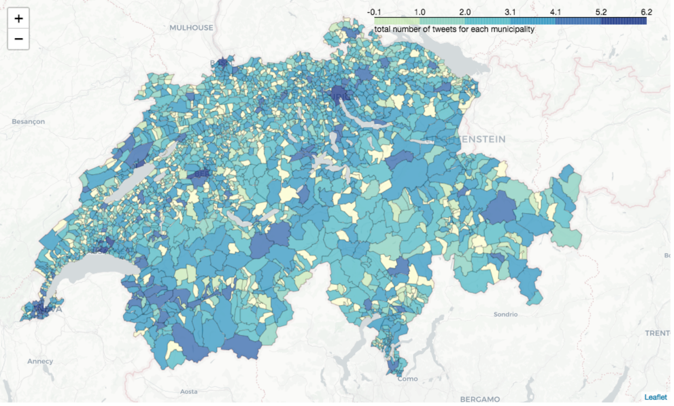

Figure 3. Colors on the map show the logarithm (base 10) of the number of tweets in the whole dataset coming from each municipality

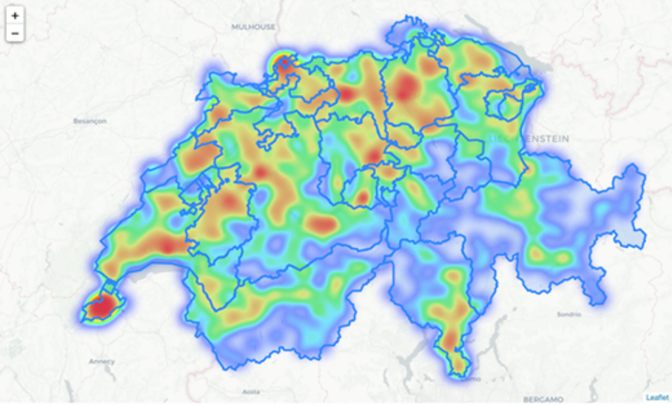

Figure 4. The heatmap of the number of tweets in September 2016.

B. Reconstructing the Röstigraben

Figure 5

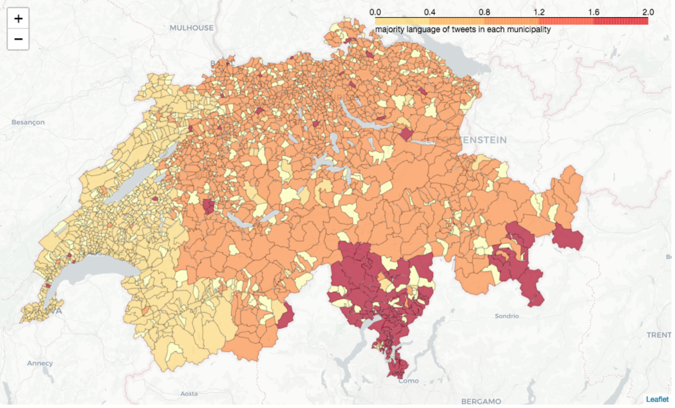

Figure 5. The Röstigraben reconstructed from the language of the tweets. The color encoding is yellow: French; orange: German, red: Italian

C. Political activity

We have tested four sets of political tweets:

1) Tweets collected using only the above initial sets, without any bootstrapping.
2) Tweets collected using bootstrapping using only the above initial sets, without making use of the political user data. The bootstrapping is 3 iterations, with 2 top hashtags that are not present in the existing set added at every iteration.
3) Tweets collected using the above initial sets and the best words extracted (using KL-div) from the political user tweets. We pick the top 15 French hashtags and the top 10 German hashtags.
4) Tweets collected with bootstrapping, using the union of the above sets with words extracted from political user tweets as the initial word set. We perform 3 iterations, picking 2 top hashtags not in the existing word set each time.

In the following figures, the y axis is the ratio of political tweets to all tweets, and the x axis is month and year. Therefore, the plots show the fraction of political tweets in each month. Division by total number of tweets in each month was necessary to account for the imbalances of the dataset.

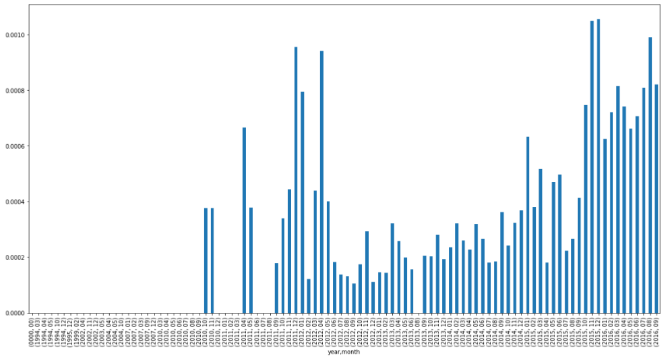

Figure 6. Fraction of political tweets per month in case 1: Only hand-made initial set, no bootstrapping

In cases 3 and 4 there are clear spikes around the two federal elections in 2011 and 2015, and also a significant spike at February

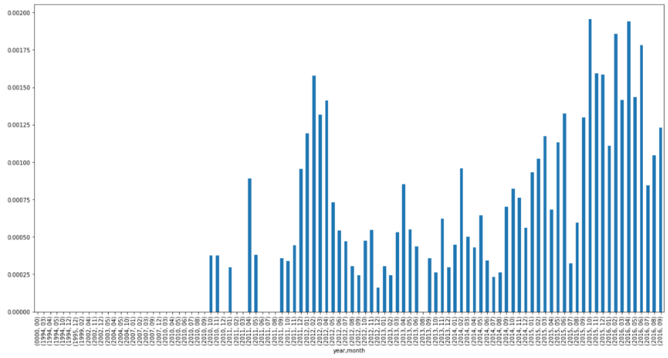

Figure 7. Fraction of political tweets per month in case 2: Bootstrapping with hand-made initial set, without use of the political user set

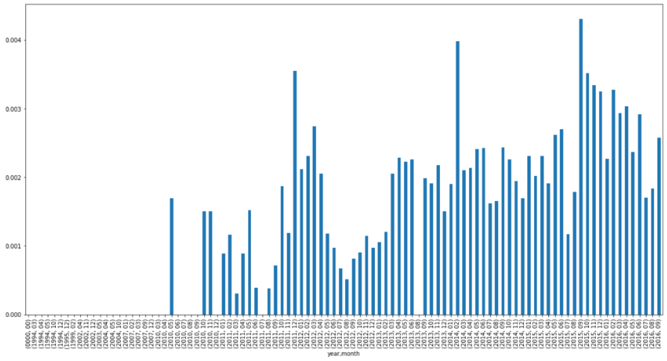

Figure 8. Fraction of political tweets per month in case 3: Hand-made initial set + set of words extracted from political user tweets, no bootstrap

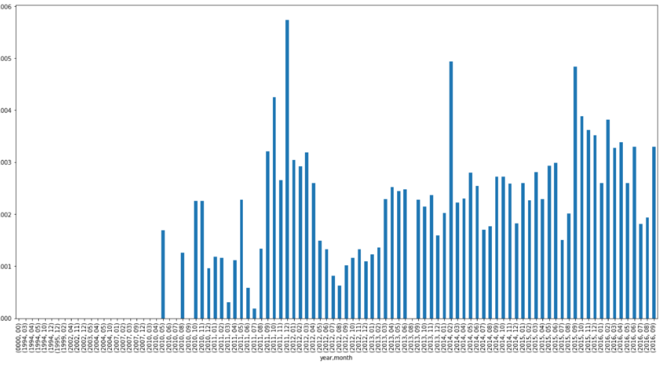

Figure 9. Fraction of political tweets per month in case 4: Handmade initial set + words extracted from political user tweets + bootstrapping

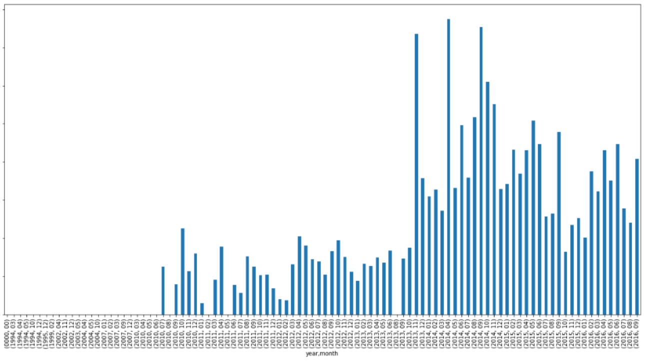

Figure 10. Fraction of political tweets per month for the political user tweet set

2014, which corresponds to a highly controversial referendum regarding migration. The original political user tweets plot shows that our algorithm is clearly superior in terms of detecting political trends, compared to solely using accounts.

Looking at the set of hashtags collected using our KL-divergence method (which may be found in our notebook and has been omitted here for the sake of brevity) on the political user tweets, we can see that hashtags such as #chvote and #ef2015 (federal elections) appear at the top of the list. In addition, the bootstrapping method adds hashtags such as #migration to the French word set and #abst14 (the trending hashtag for the Feb. 2014 referendum) to the German set. Our bootstrapping increases the number of French language tweets from case 3's 15,747 to case 4's 16,595, and the number of German language tweets from case 3's 6,040 to case 4's 8,019. The more significant increase however, comes with the words extracted from the political user tweets, which shows an increase from case 1's 3,522 French and 313 German tweets to 15,747 and 6,040 respectively.

IV. Summary

We have presented a visualisation and analysis of the geographical and linguistic distribution of Swiss tweets, recreating the Rstigraben, and we have also proposed a bootstrapping method to extract political tweets. Our plots of the political tweets show that our method can detect significant political events by detecting increased Twitter activity around them. Our method shows a significant improvement over a completely supervised method using known politician accounts.