Event detection on a Twitter dataset

Albrici Tanguy
tanguy.albrici@epfl.ch

Di Dio Davide
davide.didio@epfl.ch

Bruno Wicht
bruno.wicht@epfl.ch

Abstract

Social networks now have a huge importance in our lives and many people use them to comment about events that are happening around the globe. With this project, we would like to see how the Swiss Twitter community reacts to important events happening in Switzerland or around the world. Our main goal is to determine to what extent and how well we can learn about what is happening in the world or in our country based on the Swiss tweets. The story we want to tell is the evolution of tweets during important events between 2010 and 2016 and discover what kind of events Swiss people are tweeting about the most. We are motivated to do this project and tell this story because none of us are active on Twitter and we're interested in understanding better how twitter is used in Switzerland.

1 Dataset Description and Pre-processing

At first, we extract the hashtags from each tweets. Each tweet is stored in a dataframe that only contains the informations we require, namely:

- The tweet id
- The user id
- The longitude and latitude
- The hashtags extracted from the text
- The day, month and year of the tweet creation

After the hashtag extraction, we only keep the tweets with at least one hashtag. This dataframe is called `df_tag` in our notebook.

2 Data Manipulation

2.1 Grouping by hashtag

In order to implement event detection efficiently, we need to compute a dictionary containing for each hashtags the ids of each post that contains it. This is done in the function `group_by_hashtag(...)`. This function iterates over every post in the database and adds the id of the post in the dictionary entry of the hashtag. In this method, we also compute the number of unique authors for this hashtag. With this value, we can easily filter out hashtags that were tweeted by only a few users.

3 Data analysis and visualization

This section covers how we visualize our data and allow us to get a feel on how to detect events.

3.1 Visualizing hashtag frequency

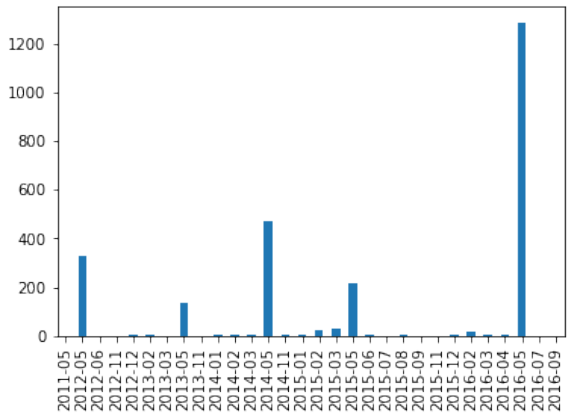

Figure 1: For the top months, plot the number of tweets with `#eurovision`

With frequency visualisation, we can visualize spikes in the number of tweets. The method `plot_frequency_tags(...)` computes for a given hashtag the numbers of tweets per frequency that contain this hashtag. Here we can

chose frequency as either day, month or year. Then it takes the `n` most tweeted dates and displays them in chronological order with a bar plot. We only take the `n` most tweeted dates because it provides the most compact visualisation. Note however that these plot are not homogenous in time.

Now, let us look a the frequency plot 1 for `#eurovision` . We can very clearly see an increase in the number of tweet during the month of may of each years. Since eurovision takes place in may, it is obvious that people will more likely talk about it during it's happening. We will be using this knowledge to develop an algorithm to detect those sudden spikes.

3.2 Visualizing event localisation

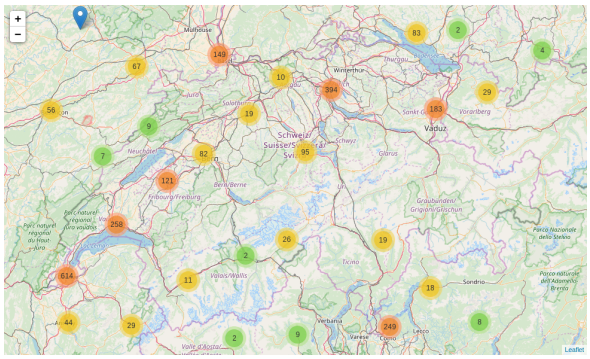

Figure 2: Map with every tweet for `#eurovision`

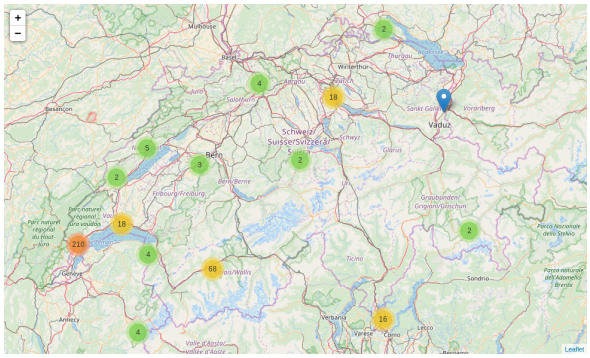

Figure 3: Map with every tweet for `#paleo`

The map are displayed in order to get an intuition on the geographic repartitions of tweets. The map are created using folium. We use a MarkerCluster to group every tweet close in space. Other than that, it is a pretty straightforward folium map. Now, look at the first map2. We see that the distribution is regrouped in the cities but overall it is quite uniform if we consider the population density. On the other side, in the second map 3, we see that most of the tweets are located in Nyon, which is indeed where this music festival takes place. The logic here people close to the actual event are much more likely to talk about it. Whereas international event are known in switzerland as a whole. Hence a local events lead to tweet localisation around the position of the event and international events have a spread tweet geographic distribution. We will use this knowledge to find the type of an event.

4 Event detection

In this part of the project we focus on automatic event detection.

4.1 Filtering out irrelevant hashtags

In the data manipulation, we constructed a dictionary containing each hashtag and their respective number of post and unique authors. Since hashtags that do not have enough posts and that are posted by only a few users are not likely to be considered events by our detection algorithm. We will be filtering out every hashtag that have less than `100` overal posts and less than `50` total authors.

4.2 Main Challenges

They were two main challenges we faced when trying to come up with an accurate event detection algorithm.

The first one is that our dataset spans almost 7 years (from early 2010 to late 2016) and thus the frequency of tweets is obviously much higher at the end of our dataset than at the beginning, as shown in Figure 4. Therefore, we had to compare the popularity of a hashtag at a certain date to some kind of baseline, so that our results can adapt to the increasing number of tweets.

The second challenge is that they are multiple types of events, and they can have a very different characteristics. For example, for an event that is scheduled to happen a a certain date, the number of tweets about it gradually increases up to that date, whereas for a sudden unpredictable event, nobody talks about it and then a big peak happens at the time of the event. Moreover, some events might last multiple days or even weeks, as opposed to one day. Therefore this makes it harder to find an algorithm that detects well all types of events.

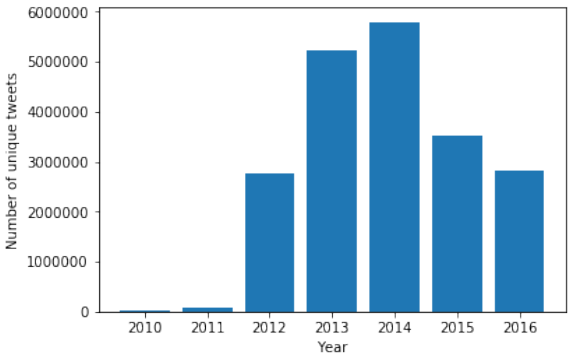

Figure 4: Number of tweets per years in our dataset. Note that the years 2010 and 2016 are not included in full in the dataset.

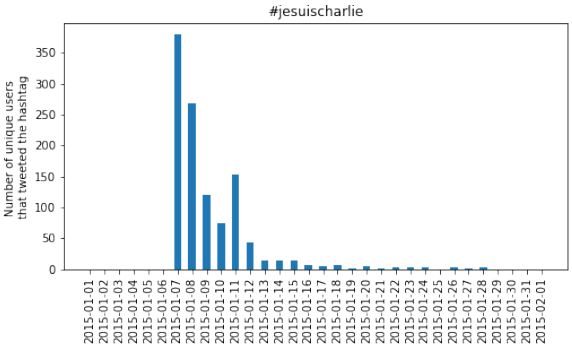

Figure 5: Example of an unpredictable event

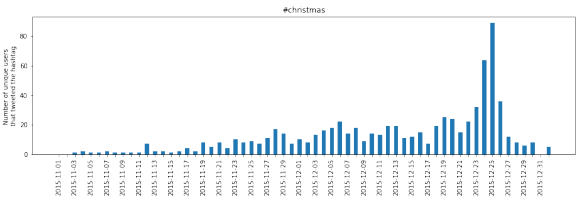

Figure 6: Example of a predictable event

4.3 Event Detection Algorithm

4.3.1 Algorithm setup

First of all, we decided to use as a metric of hashtag popularity the number of unique users that tweeted that hashtag. We think this is more relevant than using the total number of tweets, because it prevents us from detecting events based on tweets that were posted by a few users only (e.g. twitter bots or extremely active users). For simplicity, we also decided to detect only the day when an event happens, and not the time, as it would more complicated. Thus, our algorithm needs as input only the number of unique authors that tweeted a given hashtag, for each day spanned by our dataset, and for each hashtag.

4.3.2 Algorithm description

The main idea of our event detection algorithm is to compute what we will call an *event score*, for each day spanned by our dataset and for each hashtag. Then, we apply a threshold on that event score to get dates that are detected as an event. The event score is obtained by dividing some absolute measure of popularity, by a baseline. That baseline is obtained by computing an average of the number of unique users that tweeted the hashtag over some odd number of days N around the day for which we compute the event score. Dividing by this baseline solves the first problem we talked about. The absolute measure of popularity is obtained by a local weighted average over some odd number of days M around the day for which we compute the event score, where $M << N$. More specifically, if the number of unique users that tweeted a hashtag at a certain day d is given by U_d, and that the kernel for the local weighted average is given by

$$K = \left[w_{\lfloor -\frac{M}{2} \rfloor} \dots w_0 \dots w_{\lfloor \frac{M}{2} \rfloor}\right]$$

then, the event score at day d is obtained by :

$$event\_score(d) = \frac{\sum_{i=-\frac{M}{2}}^{\frac{M}{2}} w_i * U_{d+i}}{\frac{1}{N}\sum_{j=-\frac{N}{2}}^{\frac{N}{2}} U_{d+j}}$$

The kernel for the weighted local average should be some kind of gaussian, so that the maximum weight is for day d. The aim of this local weighted average is to better detect anticipated

events, or events lasting a few days. This was implemented to address the second challenge mentioned earlier.

4.3.3 Grouping events in time

After running the algorithm described above, we have a list of hashtags, and the corresponding days where the event score was above the threshold. What we do next is group this list of days for each hashtag so that the dates that are very close are represented as one event over multiple days. More specifically, we grouped two dates together if they were at most 2 days apart.

4.3.4 Parameters used

To detect events on the Swiss tweets dataset, we used the following parameters:

$$N = 35$$

$$M = 3$$

$$K = \begin{bmatrix} 0.1 & 0.8 & 0.1 \end{bmatrix}$$

$$threshold = 4$$

5 Event Localization

Now that we have a list of events, we want to determine the location of an event based on the latitude/longitude data from the tweets that caused this event to be detected. To do so, we chose to compute the median of the latitude and longitude of the tweets.

But for event that are not necessarily physically happening at some place or that are just trends, a location might not be relevant. Thus in order to determine if the event location is any good, we also compute the mean standard deviation between the computed event location and the location of the tweets. Then we decide that an event is local if the mean standard deviation is below a threshold. If it is not, the event location computed is most likely not relevant and thus we drop it.

6 Results

Here are some statistics to quantify our results:

- Out of 6197 different hashtag on which we run the event detection algorithm, we found events for 2108 of them.
- By grouping by similar dates, we then found 7028 different events.
- Out of these 7028 events, we found a relevant location for 440 of them.

We correctly detected the events we took as example previously, mainly #jesuischarlie and #christmas, as we can see in Figure . Actually, for #christmas, we detected events on the 24th and 25th of every year between 2012-2015, which is quite good (Note that december 2016 was not included in our dataset).

7 Conclusion

References

[Gusfield1997] Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.