**Report**

**Dangerous events forecasting**

1. **Description of the problem**

Forecasting events like terrorist attacks, conflicts, and any other mass violence can help to assess a risk and take a measure if it is necessary. Nowadays we have datasets with description of events from around the world, automatically collecting information from the Internet mass media sources. Can we predict the future events with this given information? Let’s consider a certain country – France and time period 2013-04-01 – 2016-09-03.

Note: the problem is not new, so you can check:

* <http://data.gdeltproject.org/documentation/ISA.2013.GDELT.pdf>
* <http://foreignpolicy.com/2014/01/03/half-a-billion-clicks-cant-be-wrong/>

1. **Approach to the problem**

The main idea to forecast dangerous events is to build time series from their characteristic – tone of event. In this context, tone is the score ranges from -100 (extremely negative) to +100 (extremely positive). Common values range between -10 and +10, with 0 indicating neutral. This score, calculated automatically, can be used for measuring of the importance of an event. For example, an event like terrorist attack has a tone less than -15.

When we have a time sequence of tone, we can apply appropriate methods for forecasting this set of data: ARMA/ARIMA model, neural network or genetic algorithm.

Time sequence has a day as a time unit. However, there a lot of events during a certain day, so before applying method, tone of events, happened in the same day, should be transformed according the following rules:

* If there are no a dangerous events during a day, tone is average of tones these events.
* If there are dangerous events during a day, tone is equal tone of dangerous event with minimum of tones.

The whole algorithm for approach includes the following steps:

* Loading and merging datasets
* Cleaning and transforming data
* Visualizing data
* Applying neural network

Let’s consider these steps with more details.

* 1. **Loading and merging datasets**

Datasets source is [GDELT event files](http://data.gdeltproject.org/events/index.html). In this webpage, we can see zipped csv files representing a certain date. There are about 1200 files for period 2013-04-01 – 2016-09-03. Each file consists of about 160 000 – 170 000 records. Therefore, finally, the total amount of records is about 200 000 000.

The first step is loading datasets on a local machine and during the loading to merge datasets representing the same month. After this step there are 42 files with the total size 68 GB.

Each record in these files describes one event with the following format:

* event represented as «Actor1 performed an action upon Actor2», where Actor can be a certain person, organization, group, country and so on. Note: everything described by codes;
* location of an event;
* tone and other characteristic;
* hyperlink;
* a date an event was added to a database.

For more details about structure of datasets see <http://data.gdeltproject.org/documentation/GDELT-Data_Format_Codebook.pdf>.

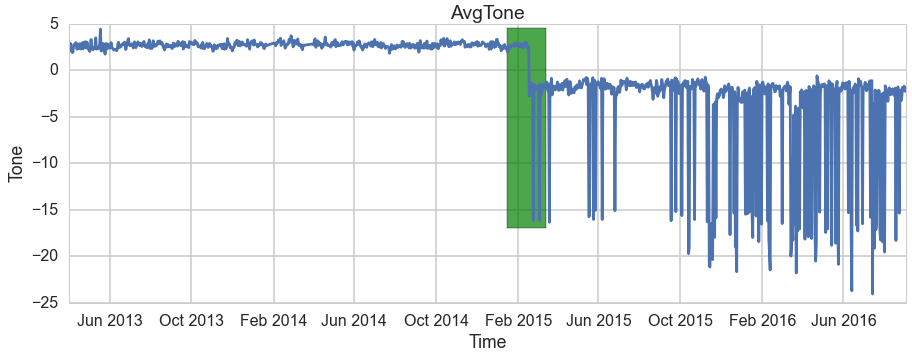
* 1. **Cleaning and transforming data**

First of all we need to extract information concerning events in France. After this step we delete duplicates applying the rules:

* Delete duplicates with the same URL
* Delete duplicates with the same characteristic:
* FractionDateN',
* 'QuadClass',
* 'GoldsteinScale',
* 'AvgToneN',
* 'Actor1CountryCode',
* 'Actor2CountryCode' .

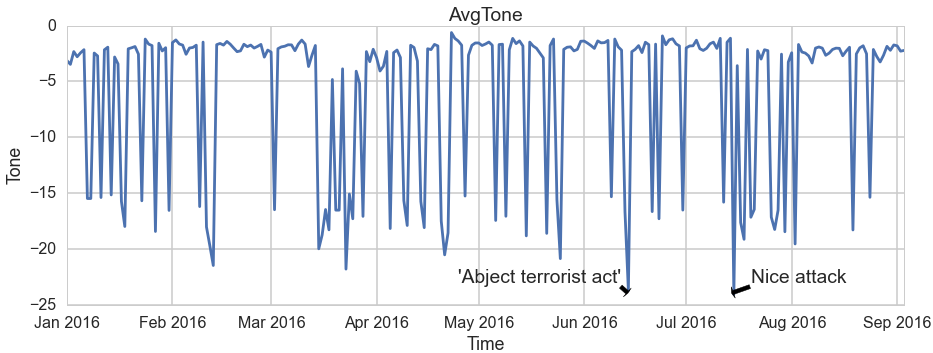
Note: Description of events in given datasets are not always correct because of machine processing.

Finally, we build time series of events tone:



**Figure 1**

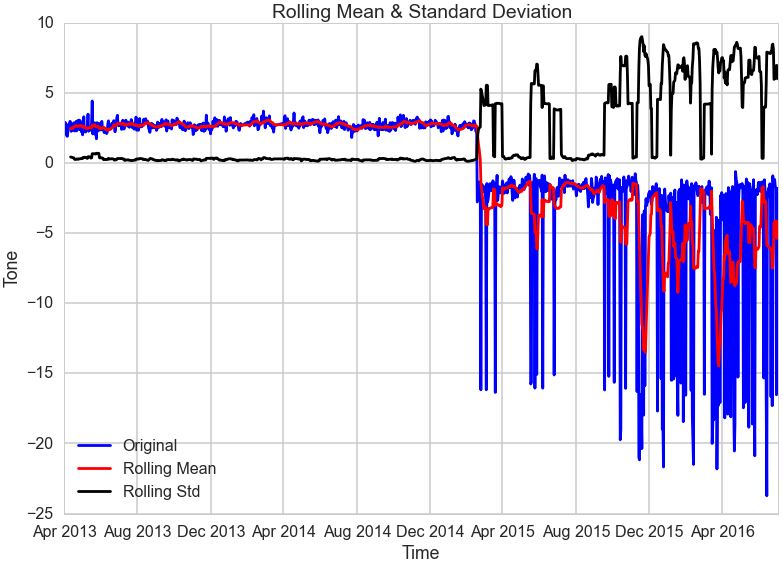
There was a significant change in behavour of Tone ('green' area in the picture above), but it's difficult to find reasons for that. Let's have a close look at events in 2016.



**Figure 2**

* 1. **Applying neural network**

Before applying neural network (NN), I tested stationarity of the time series:



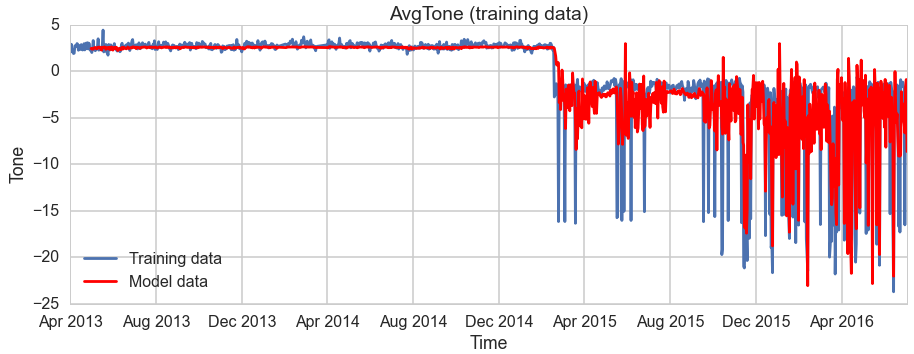
**Figure 3**

Standard deviation and mean are both changing with time so it's not the stationary time series. In this case it may be better to implement NN.

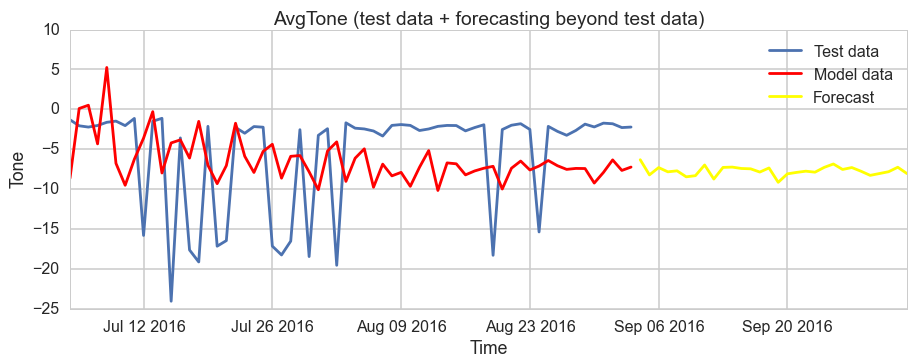
In this project, I used so-called LSTM network - the Long Short-Term Memory Networks. As input variables I took t-lag-1, t-lag-2, …, t values to calculate t+1 value.

1. **Results**

There are results below after implementation of NN (lag = 30).



**Figure 4**



**Figure 5**

We can see significant differences between test and model data in the last picture. Moreover, forecast line (yellow color) collapses, so decision needs some improvements, say, to play with lag and adjustments of NN. There is the further possible steps in conclusion part.

1. **Conclusion**

Approach with representing events as a time series of tone and implementing NN didn’t give good results. Nevertheless, some improvements for this model are still possible:

* find appropriate transformation for input data;
* play with parameters of NN.

Other approaches:

* Build a model for forecasting based on not only time, but also some independent variables.
* Use genetic algorithm instead of NN.