SocNet Analyser - draft 2.0

Large scale social networks content analysis framework for sentiment and network research

### MFF UK Software Project

*Project name: “We really need something…”*

*Team: Jan Pavlovský*

*Jaroslav Knotek*

*Petra Doubravová‎*

*Lukáš Kolek*

*Supervisor: Doc. RNDr. Irena Holubová, Ph.D.*

## Motivation

For more than a decade already, there has been an enormous growth of social networks and their audiences. As people post about their life and experiences, comment on other people’s posts and discuss all sorts of topics, they generate a tremendous amount of data that are stored on these networks.

As we already know from the example of Cambridge Analytica, the users’ personal data can be used in, if not exactly harmful, very immoral way. The main issue with CA was that they were using personal data that were not anonymized as marketing tools primarily in the 2016 US presidential elections. They were targeting people as individuals and assessing data specific only to them, that those people may not have agreed to share publicly.

However there are lots of the data shared on various social networks, especially other than Facebook, that are open. If gathered in very large amounts and analysed statistically and, naturally, anonymously, there could be very useful patterns discoverable in all that data.

## Problem

In general, social networks provide abundance of information concerning various topics in the form of posts, comments, and other. The amount of posts and comments is so high that it is virtually impossible for a human being to read through any significant fraction about even a very specific topic and make some opinion about the public sentiment. And most social networks do not offer tools supporting quick orientation in the overflow of posts.

The goal of this project is to create a framework for analysing public social networks content and computing sentiment over the data published on supported social network under given conditions (time period, topic…). The result of the analysis will be presented to the user in a structured format consisting of

* sentiment analysis
  + of a specific topic
  + in a given time frame
* significant keywords
* clustered opinions
  + positive, neutral, negative

## Supported networks

Social network planned to be supported are:

* Twitter
* Reddit
* Facebook

These platforms contains huge amounts of valuable data, however they are not available in the full extent. Social networks does not allow for querying extensive data. Data acquisition is limited (for a free user) by both, requests per time period (e.g. currently 15 minutes for Twitter, 1 hour for Facebook) and total amount of records retrieved. These limits make it impossible to support extensive online analysis.

# Data retrieval

There are several approaches to overcome the limited access, e.g.:

* Analysing only data spanning over short periods of time
* Watchdog mode - Data are analysed as they are acquired regardless of its extent allowing user to have up to date analysis of the most recent data (especially useful for a long time user interested in his organization/business profile)

Tohle celý chce přepsat. Nedává to smysl.

Short period analysis works well for very low amount of post which is given by social network api limits. When amount of posts reach the limits, user needs to wait very long time (e.g. several hours) for his results.

The large amount post acquisition may be done efficiently using watchdog mode that does not require any interaction. When selecting a specific start date, the application will start fetching newer post until all posts until current time are retrieved and then will continue to periodically watch for new ones.

In both modes, data come in batches. In case of short analysis, data are processed after all of them are loaded and analysed afterwards. Watchdog is designed to perform analysis after each batch (size might vary) and update overall result.

# Use cases

In general, user selects data sources for the data to be acquired from, and selects what type of analysis to be performed.

In case of social network, user selects for example twitter as a source of analysis. Then he supplies topic in which he is interested in and selects sentiment analysis. Optionally, he will be also able to supply credentials (access token…) for a site/profile, whose posts he is interested in and has access to (a organization manager with administration access to their profile for example).

This job definition is then translated to a command chain beginning from data acquisition, transformation and storage, to analysis and selected form of presentation.

In case of long-running jobs, user is given a special url which will store up-to-date result and current progress. When the job finished, the user will be notified and given access to the results.

#### *An example: market sentiment analysis*

When referring to stock or financial market, we can analyse and model sentiment about a specific company stocks, or a currency. That way we can research behaviour of traders in reaction to certain stimuli. We can compare patterns in historical sentiment model with the graphs of actual market of stocks, as well as with events and news in the outside of the market. If we would be able to find correlation between these, we could possibly predict the development of market in future.

This can be especially useful with markets with high levels of volatility and manipulability (and therefore practically very manipulated). A first-hand example of such fast-paced and highly manipulated market are the cryptocurrencies, and their trading platforms.

# Architecture and technology

Software consists of four primary cooperating modules

* Data acquisition - a module acquiring data from selected data sources.
* Analyser - a module analysing stored data according to selected analysis type.
* Storage - a wrapper encapsulation existing database solution
* Coordinator - a web application coordinating work of all previous modules from job definition to data acquisition and transformation to analysis to visualisation.

The system will be implemented as a web application. Various architectural patterns may be used. Simplest plausible model is to implement four Web services. It is easy to create and easy to manage. The most significant disadvantage is, that it does not scale as well as some more complex solutions.

Better performance will be achieved by creating finer grained modules encapsulated in containers and then orchestrated according to defined rules and performance requirements. This approach is much more complex, but if designed well, almost no runtime management is later required. This approach also makes it possible to scale easily according to current according to data load.

Scaling is expected to make sence for long-running analysis only. It won’t be required by *Data acquisition* module since it can’t download that much data because of networks’ limitation.

Further analysis will be performed for proper technology selection.

# Data storage

In general, all analysis tools working with social network data requires a unified format to work with. Each social network has a unique structure of data which has to be unified into single form and then stored in a common storage.

The unified form has to reflect metadata about posts and anonymized data about social network users. Important for the analysis are:

* Text
* User Id
* Keywords
* Tags
* Relations

To prevent storage of obsolete or otherwise not necessary data, user will be able to set a data retention period, while older data will get automatically deleted.

# Extensibility

The application is designed to obtain data from social network in small batches and then store them. The stored data can be then loaded and analysed. Nevertheless, additional functionality can be added using user-made modules.

For example, the *Data acquisition* module can be extended by adapters to various data sources: files, other social networks, news media… The only requirement is that data must follow given schema.

The *Analyser* module can be extended by a custom made analysis. This alternation must be accompanied with appropriate visualisation tools.

## Time and tasks estimation

1. Analysis (cca. 1 month)

In the beginning, we need to state more specifically what should be the output of our work, assess our ability to carry it out, and carefully specify what if achievable in the time frame given. We also need to study existing materials about the topic in this phase.

1. Design, work division (1 – 2 months)

Following the specification, we as a team should make fundamental design and architectural decisions about the software. Based on that, we can divide work among team members.

1. Implementation (3-4 months)

Individual team members will implement assigned modules of the system. We would like to adapt an iterative development approach, in which we would iteratively integrate and test individual system modules.

1. Testing, Experiments (1-2 month)

The system as a whole will be extensively test to ensure all functionality if working. Also there is space for various experiments with possible software extensions.

1. Finalisation (1 month)

In the terminal phase, the framework will be prepared to be presented which requires acquisition of data reflecting various use-cases, ensuring connectivity. The team will also need to prepare for a defense of this project.