Social sentiment analyser- draft 0.1

Large scale social networks content analysis framework for sentiment research

MFF UK Software Project

Project name: *TBA*

Team: *Jan Pavlovský*

Jaroslav Knotek

…

Intro

As a mandatory component of masters' studies at MFF UK, students take a software project class, in which a team of 4-8 student should implement an extensive piece of software work during the time frame of 9 months. This document briefly introduces the idea of social sentiment analysis tools, a topic we would like to take on in our Software Project.

Motivation and Goals

For more than a decade already, there has been an enormous growth of social networks and their audiences. People connect together on these networks, post their life stories and opinions as well as pictures and videos, chat and form societies interested in common topics. Since the foundation of Facebook in 2004, there has been an uncountable number of social networks of all kinds and purposes founded, some surviving until present day, some not. And if you try to agree on a single generally accepted fact about them, it would likely read that social networks are an indispensable part of everyday life for majority of people nowadays, and these networks affect us all.

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. If we were to enumerate all this data, the numbers would be astronomical at the very least. And many have already figured out that all this data could be utilised in pursuit of commercial, public or personal goals.

As we already know from the example of Cambridge Analytica, networks 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 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 most of the data published online, especially on social networks, are publicly accessible and mostly general in character. If gathered in very large amounts and analysed statistically and, naturally, anonymously, there could be very useful patterns discoverable in all that data. And this is the path we would like to venture in this project.

We are certainly not the first to experiment in this field. There has been a number of projects already created in both, academic and commercial sectors. However most of what can by openly found are quite narrow-purpose tools designed for a single specific task. What we would like to achieve is to

generalise whatever is possible to generalise in these tools and create a multi-purpose framework for gathering and analysing social networks' content.

Use-case 1: sentiment analysis of elections (eg. for political marketing)

The task of sentiment analysis aims to determine the people's attitude and feelings about a certain entity or topic, and model them in time. That way, we can find out which event affect the public sentiment the most and in which matter, and act accordingly.

Utilizing this approach, a politician could, for example, reflect on his actions and statements and see which have raised positive and which negative reaction among the crowd. From marketing point of view, he could learn useful lessons from that and assess his future direction accordingly.

Use-case 2: 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.

Realtime prediction

Processing historical and present data are 2 quite different fields though. Modelling historical sentiment development can be done with batch-processing, using data gathered over long period of time. However, with processing present data, it is quite a different story.

Following on previous section to give an example, if we want to use sentiment analysis for trading decisions, we need to react quickly to current event. For that, we need the system to be able to process the data in real-time, as streams. Besides primary technical challenges of developing such a system, there are several additional issues, among those: limited network bandwidth, social networks limiting access for single access token and computational power. Nevertheless, the benefits of being able to process the social networks content in real-time and predict events development based on that could show to be quite far-reaching.

Overview

A system of this kind would generally consist of 4 primary modules:

1. Data mining
2. Pre-processing
3. Storage
4. Analysis

1. Data Mining

Data mining is the process of gathering the data for future use. There are a few distinct tasks necessary to carry out in this phase. From high level view, these would be:

- Social network access (access tokens, accounts, access limits…)
- Data crawling
- Data downloading

2. Pre-processing

Appropriate data needs to be firstly selected and sorted, before storing in the database, so that we do not store unnecessary data. Also, we want to store only relevant data to their specific topics and, in ideal scenario, store only the minimal necessary amount of data.

The other important task to do here is the data transformation. Especially when we want to aggregate data from multiple sources, and possibly also different forms of data (text, images, sound…), we need to transform the data into common form, so that it can be stored, accessed and read in a generalised way later.

3. Storage

There are many options for storing structured big data available on the market and utilised today. The decision about specific technologies and approaches to use here are mainly affected by the specific use-cases they are aimed for.

Important factor to consider here is correct categorisation and deduplication of data. As a single post or image could easily relate to hundreds of topics and entities, we need to store the data in a way such that they do not get duplicated in the database, but also are kept linked to all entities they should.

4. Data analysis

When we have all the relevant data stored in a structured manner, we can finally advance to actually analyse it. There have been many methods proposed and used for statistical analysis of big data, both conventional and machine learning.

The choice of specific technology and method here is again very use-case dependant. That is why we would like to tackle this module in the most general way. The plan is to design an efficient and easy to use interface from the data storage, so that various data analysis tools may be used, and possibly even aggregated to solve a single task as effectively as possible.

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.

2. 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.
3. Implementation (4-5 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.
4. Finalisation, documentation (1-2 months)
   In the terminal phase, we will put together the final product, assess what has been accomplished and what hasn't. Also, we will create the necessary user and development documentation.