Queen Mary, University of London

Final Year Project

Initial Specification

Femi-Balogun, Fayimora

fayi@fayimora.com

-

October 24, 2013

Contents

1 Project Description

The aim of this project is to detect themes/topics in social media traffic, profile people in terms of their interest and run sentiment analysis on these topics in order to investigate how people's sentiments correlates with different topics they are interested in. If time permits, I will also investigate how topics and sentiments correlate with location.

1.1 Functional requirements

The data will be sent to a system for **filtration**. This is where the data will be split into two groups. These groups include:

1. Relevant: This is data that is relevant to the general topic we are interested in.
2. Irrelevant: This is data that we are definitely not interested in as they have nothing to do with the general topic we are interested in. For instance, if we decide to use "apple" as a general topic, looking at data talking about the fruit or an apple drink/pie will be completely irrelevant.

After filtration, the data is passed on to another system for **Topic Modelling**. This is where the data is classified into different topics. After the data has been classified into different topics, each topic will be passed to another system for **sentiment analysis**. Sentiment Analysis will be simulated on a per topic basis.

1.2 Non-functional requirements

Besides the functional requirements, the final product should have the following non-functional requirements.

1. **Ease of use:** The system should be extremely user-friendly. As of now, the plan is to write a front end web application for users to interact with the system. The web application will be an admin-style interface with charts and configurable options for the systems.
2. **Ease of installation:** Should be able to install it easily on any system. A cross-platform system with as few systems requirements as possible would be the ultimate goal.

3. **Reliability:** The system has to be very reliable. If one component goes down, it should not affect other components. Also, another should be booted up to replace the dead one.

4. **Maintainability:** Since the final product will comprise of multiple sub services/components, it has to be built with maintainability in mind. One way to achieve this is by logging everything that happens so bugs/failures can be easily traced having good logging.

2 Techniques to be applied

2.1 Algorithms

In order to achieve the above tasks, I will be looking into different techniques and algorithms. Some of which include:

- Naive Bayes classifier for data filtration.
- k-nearest neighbour algorithm and Latent Dirichlet Allocation model for topic modelling

2.2 Data

The data to be analysed will be gathered mostly from Twitter. Twitter's data is public and they provide a streaming API from which we can fetch all tweets that include at least one of a set of preconfigured words. Although Twitter caps this to 1% of their firehose, the limit falls in the range of millions of tweets per day which should be adequate.

A fraction of this data will be manually labelled and used as test data. The test data will be used to see how accurate my models are. The rest of the data will be used as train data.

When the data filtration algorithm starts producing results with high accuracy, the train data can be swapped for fresh data periodically.

3 Tools

Implementation of every involved will be done in Scala. Scala is a programming language that runs on the Java Virtual machine. It brings the best of the Functional and Object Oriented worlds together. Scala will allow me to focus on the task at hand rather than battling with the verbosity of Java.

Scala does not ship with tools for Natural Language Processing or Machine Learning so I will be employing the help of a few libraries. Some of them include:

- **MALLET:** As its webpage says, MALLET is a Java-based package for statistical natural language processing, document classification, clustering, topic modelling, information extractions and other machine learning applications to text.
- **Breeze:** Breeze is a library for numerical processing, machine learning and natural language processing. It is well known for it numerical processing and data visualization modules.
- **LIBSVM:** A library for Support Vector Machines.
- **LIBLINEAR:** A library for large linear classification.

Some of the libraries listed above have overlapping functionality so I might not end up using them all. For instance, I might use MALLET for clustering and topic modelling but turn to liblinear for any sizable linear classification.

Data collected from Twitter will be stored in a MongoDB database. MongoDB is the an open-source NOSQL document database. Data in MongoDB is stored in JSON-like documents with dynamic schemas. Twitter's streaming data is provided in JSON format so storing and fetching this data in MongoDB will be faster and easier compared to a relational database.

4 Plan

4.1 Semester 1

Please note that this section will be updated with a more detailed plan within the next two weeks.

1. **Data collection:** By Monday 4th of November, I plan to have collected the necessary data and I should have started labelling them.
2. **Data filtration:** At the very least, by the end of December, I should have a good system that filters my data into the right groups. I will try to achieve as much accuracy of possible.

4.2 Semester 2

A more detailed plan for Semester 2 will be drawn up in Semester 2 but generally, I expect to be done with the whole project by the first week of April.