



# Project Topic 4: Docker-based Service Composition

Group 1, Topic 4

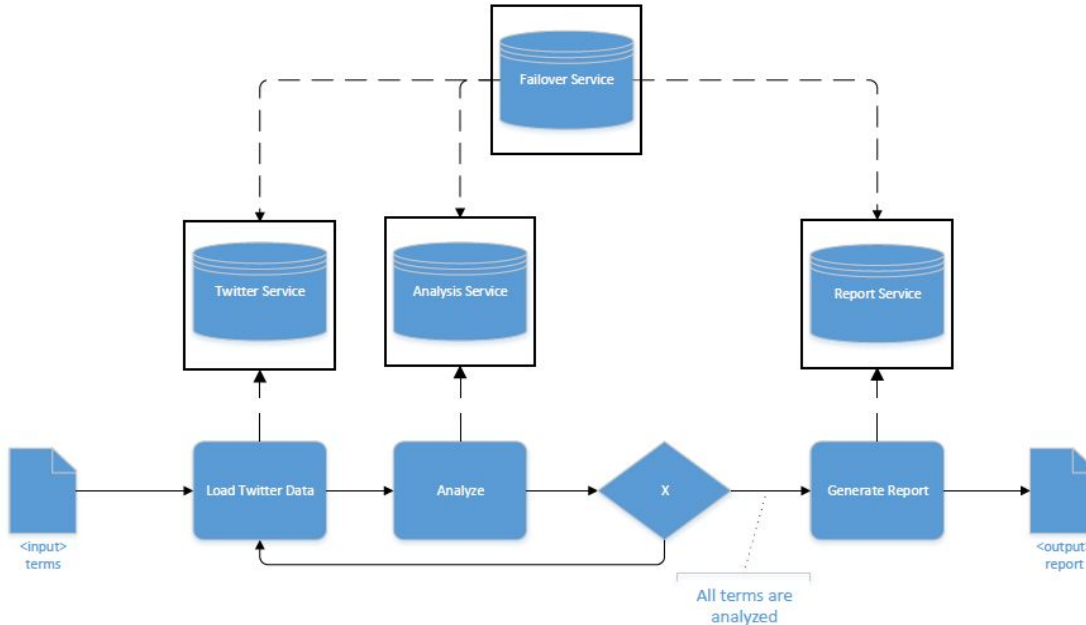
Gruzdev levgenii

Eitenberger Christoph

Nourizadeh Barabi Manouchehr

Alma Causevic

# Sentiment analysis for tweets



Sentiment analysis is the study of feelings by identifying attitudes, emotions and opinions in texts.

1. Twitter Service - obtains tweets for the provided search terms
2. Analysis Service - derives the sentiment for a given tweet
3. Report Service - generates a PDF report based on the results of the analysis

Each service is wrapped in a docker container.



# Tweets Collector

- Twitter4j\* API
- Obtain new API keys from twitter: <https://apps.twitter.com/>
- API keys stored in spring-boot application.properties file in src/main/resources
- Only “English” tweets

## Parameters(methods: GET/POST):

- |                               |  |
|-------------------------------|--|
| A. <b>url:</b>                | http://localhost:8080/submit                   |
| B. <b>“keywords”:</b>         | multiple keywords separated by “,”             |
| C. <b>“count”</b> [optional]: | limit the tweets(1 to 500) / default value: 10 |

Sample URL: <http://localhost:8080/submit?keywords=bitcoin,volkswagen&count=50>



# Analysis (I)

## Preprocessing

- aims to extract and normalize important textual information of tweet messages
- Output: feature vectors for classification
- Used tool: [Stanford NLP\\*](#) (Natural Language Processing)

## Preprocessing steps:

- |                               |  |
|-------------------------------|--|
| A. <b>Tokenization:</b>       | split input into related words also called tokens ( <a href="#">PTBTokenizer</a> ) |
| B. <b>Tagger:</b>             | adds word-class to each token ( <a href="#">MaxentTagger</a> )                     |
| C. <b>Create token lists:</b> | filter out irrelevant information based on tagged tokens                           |

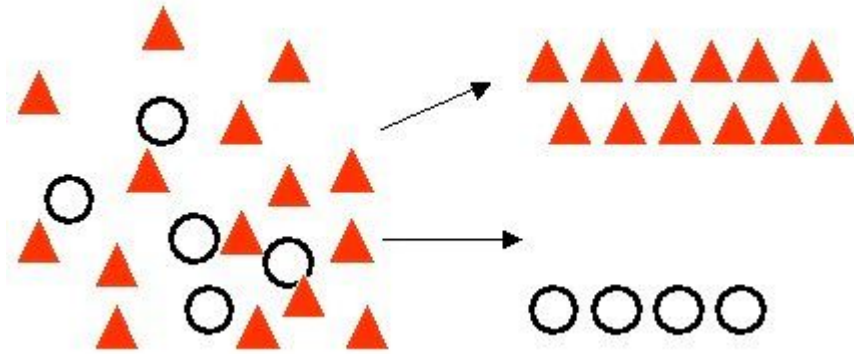
## Analysis (II)

### Classification

- assign sentiment to the textual input
- requires textual input in form of feature vectors
- used tool: **WEKA Classifier\***
- classification algorithm: **Naive Bayes Classifier**
- available sentiments: **positive** vs **negative**

### Aggregation

- collect sentiments for each tweet and calculate positive and negative ratios



\*<https://www.cs.waikato.ac.nz/ml/weka/>

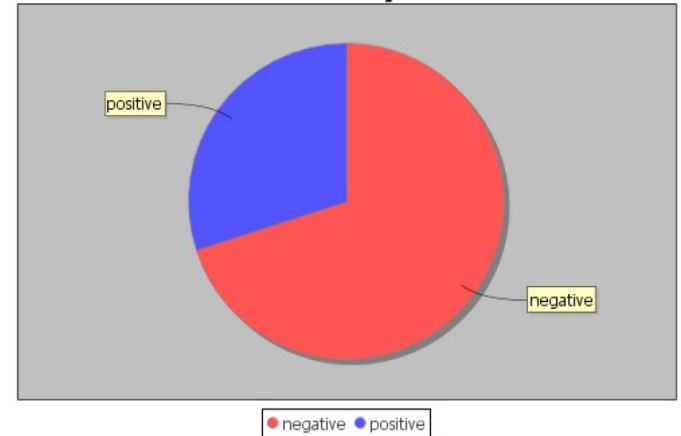
Figure source: <http://www.oracle.com/technetwork/database/enterprise-edition/odm-techniques-algorithms-097163.html>

# PDF Report

- PDFBox\* & JFreeChart\*\* API
- one side per search term
- pie chart ⇒ illustration of the analysis results (negative and positive values) per search term
- Report saved in the ReportService package

## Check if the analysis was successful

- If the search term was written incorrectly/no tweets could be found ⇒ show info in the PDF Report



Term: test

Term written incorrectly/no tweets could be found.

\*<https://pdfbox.apache.org/>

\*\* <http://www.jfree.org/jfreechart/>



# Demo of the implementation...