Syntactic Sugar

JARGON

SENTIMENT ANALYSIS

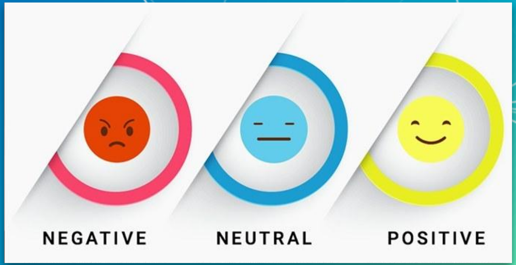
COMPIAX

SOFTWARE ENGINEERING

Jargon explained:

Jargon is an AI-driven social listening and sentiment analysis platform. Jargon provides users the ability to analyse the sentiments of tweets around a specific topic/topics.

This analysis is performed automatically by the system and presented to the user in a format that best allows them to interpret the data and make better business/personal decisions.

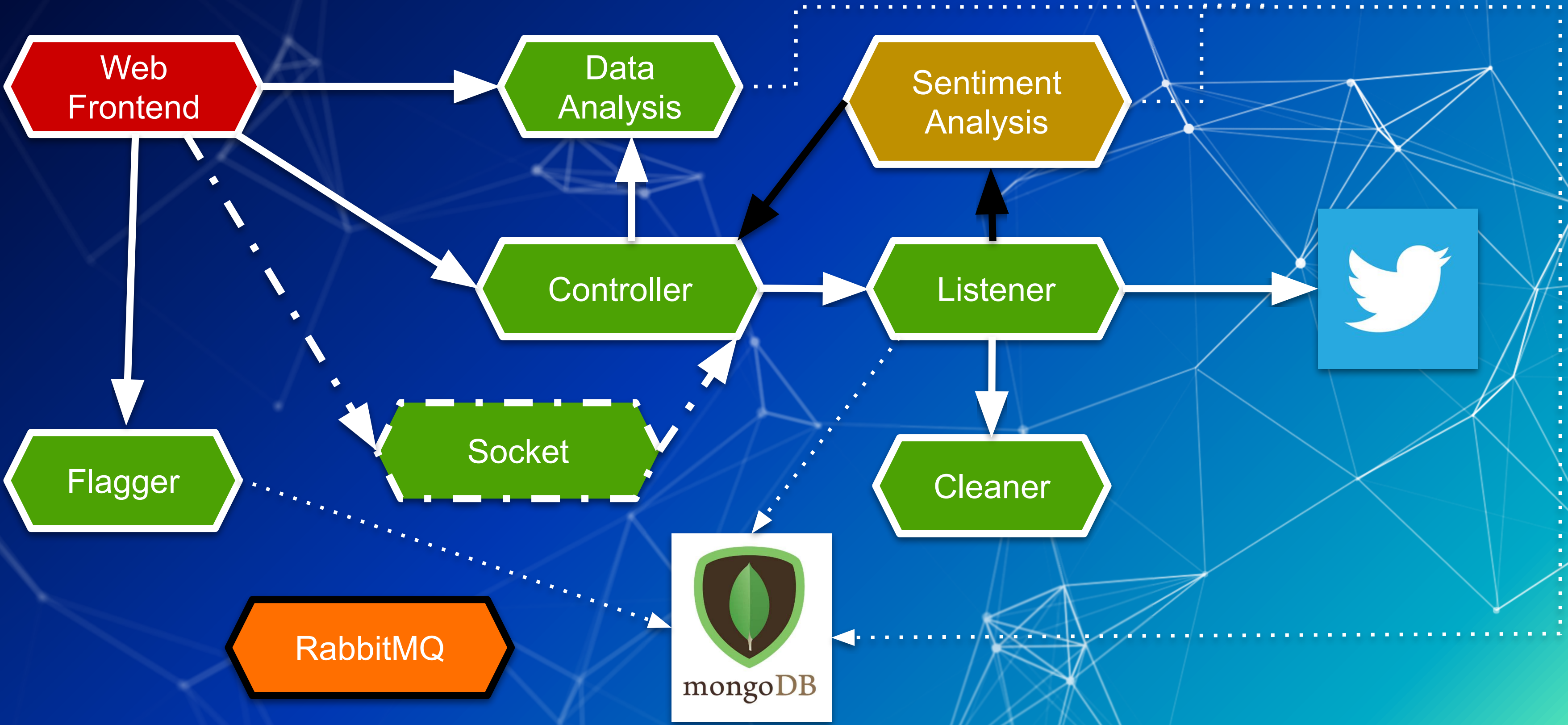


Core Functionality

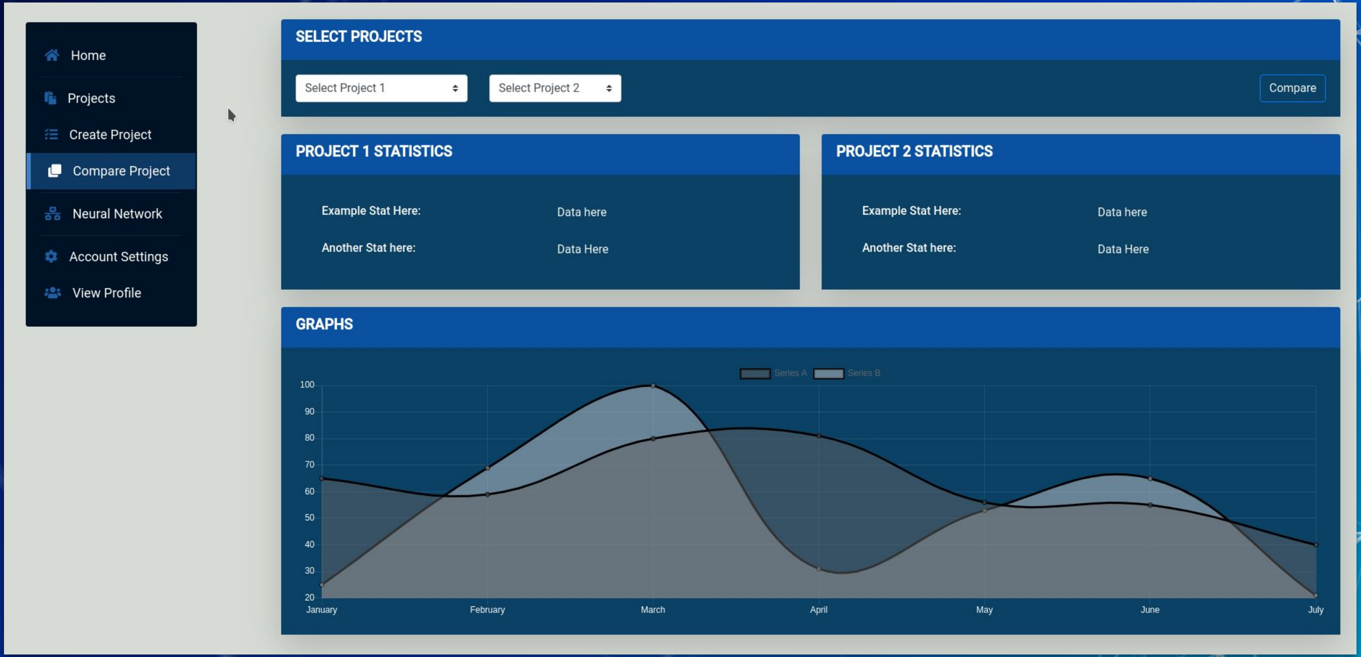
- Manage projects
- Stream realtime social media data
- Clean collected data
- Analyse and display sentiment in realtime
- Statistical Analysis of results
- Compare Projects
- Flag incorrect sentiments (for training and improvement)

Architecture

Microservices



Web Frontend



SELECT PROJECTS

PROJECT 1 STATISTICS

Example Stat Here:	Data here
Another Stat here:	Data Here

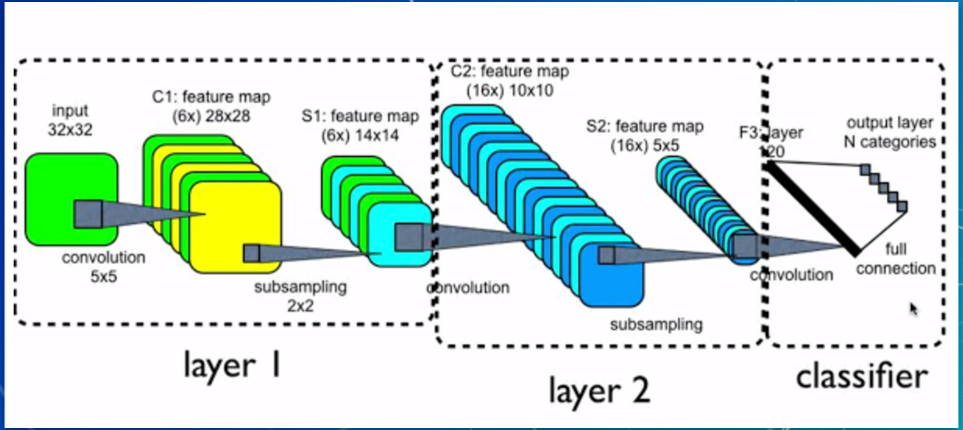
PROJECT 2 STATISTICS

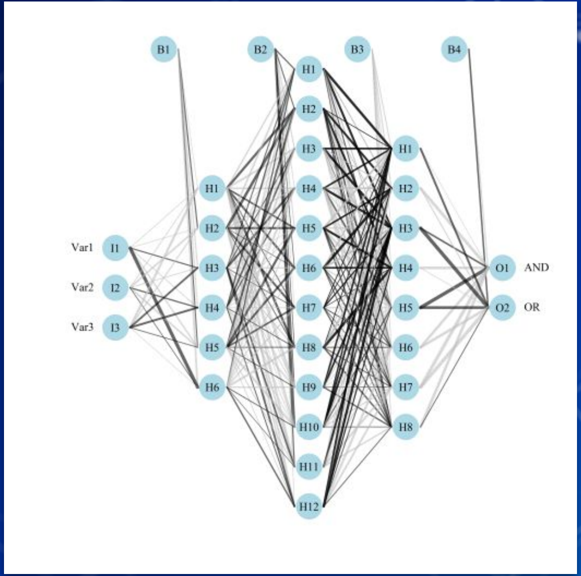
Example Stat Here:	Data here
Another Stat here:	Data Here

GRAPHS

Sentiment Analysis

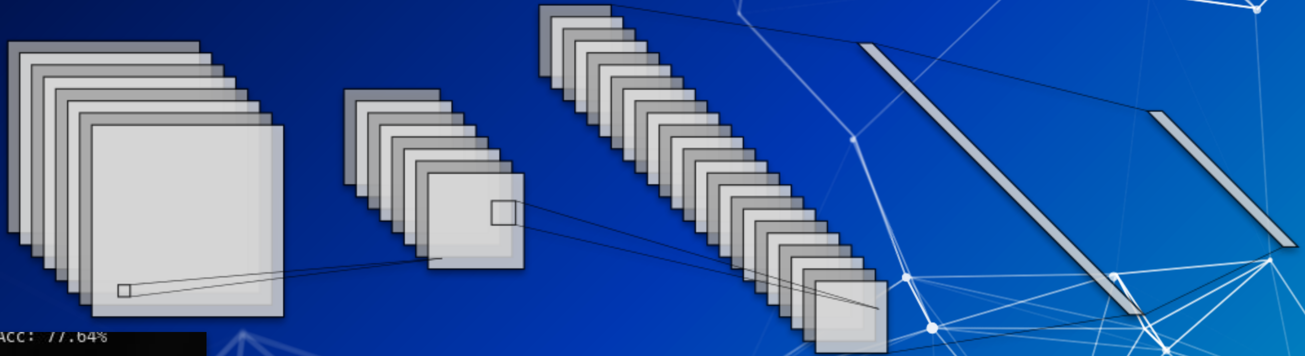
Convolutional Neural Network

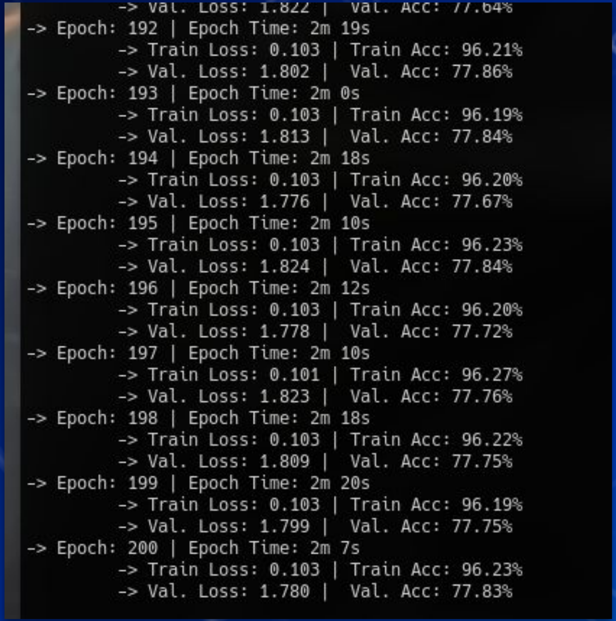




The Convolutional Neural Network used for analysis was built using PyTorch and an open source vector mapping model.

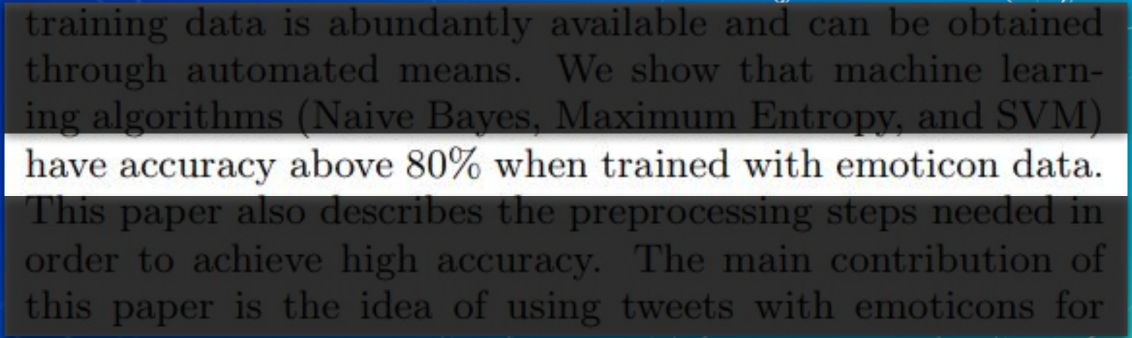
More on The Training





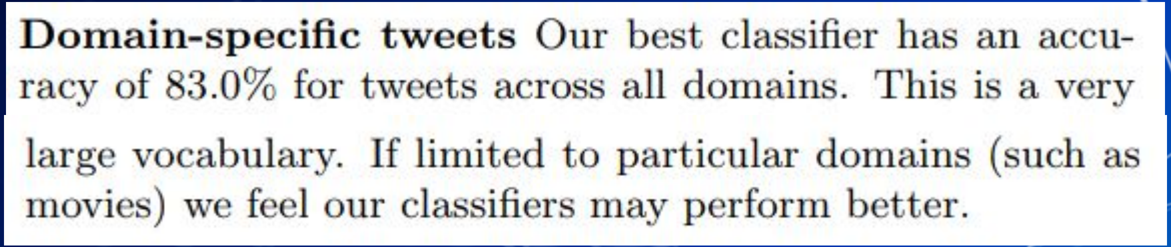
```
          -> Val. Loss: 1.822 |  Val. Acc: 77.64%
-> Epoch: 192 | Epoch Time: 2m 19s
          -> Train Loss: 0.103 | Train Acc: 96.21%
          -> Val. Loss: 1.802 |  Val. Acc: 77.86%
-> Epoch: 193 | Epoch Time: 2m 0s
          -> Train Loss: 0.103 | Train Acc: 96.19%
          -> Val. Loss: 1.813 |  Val. Acc: 77.84%
-> Epoch: 194 | Epoch Time: 2m 18s
          -> Train Loss: 0.103 | Train Acc: 96.20%
          -> Val. Loss: 1.776 |  Val. Acc: 77.67%
-> Epoch: 195 | Epoch Time: 2m 10s
          -> Train Loss: 0.103 | Train Acc: 96.23%
          -> Val. Loss: 1.824 |  Val. Acc: 77.84%
-> Epoch: 196 | Epoch Time: 2m 12s
          -> Train Loss: 0.103 | Train Acc: 96.20%
          -> Val. Loss: 1.778 |  Val. Acc: 77.72%
-> Epoch: 197 | Epoch Time: 2m 10s
          -> Train Loss: 0.101 | Train Acc: 96.27%
          -> Val. Loss: 1.823 |  Val. Acc: 77.76%
-> Epoch: 198 | Epoch Time: 2m 18s
          -> Train Loss: 0.103 | Train Acc: 96.22%
          -> Val. Loss: 1.809 |  Val. Acc: 77.75%
-> Epoch: 199 | Epoch Time: 2m 20s
          -> Train Loss: 0.103 | Train Acc: 96.19%
          -> Val. Loss: 1.799 |  Val. Acc: 77.75%
-> Epoch: 200 | Epoch Time: 2m 7s
          -> Train Loss: 0.103 | Train Acc: 96.23%
          -> Val. Loss: 1.780 |  Val. Acc: 77.83%
```

Our training results

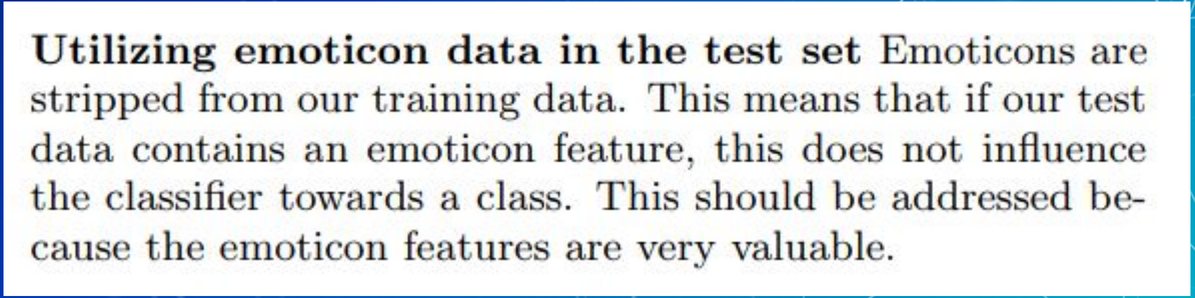


training data is abundantly available and can be obtained through automated means. We show that machine learning algorithms (Naive Bayes, Maximum Entropy, and SVM) have accuracy above 80% when trained with emoticon data. This paper also describes the preprocessing steps needed in order to achieve high accuracy. The main contribution of this paper is the idea of using tweets with emoticons for

Citation: Go, A., Bhayani, R. and Huang, L., 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford, 1(2009), p.12*.



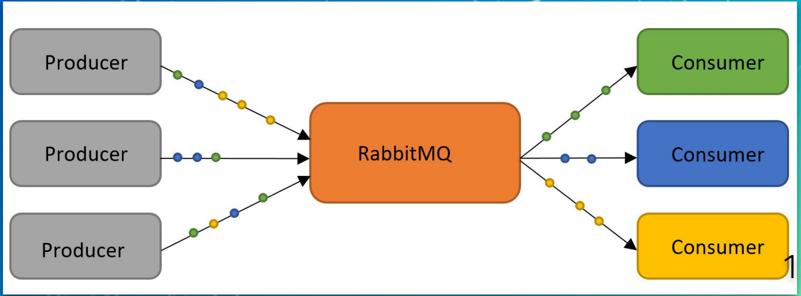
Domain-specific tweets Our best classifier has an accuracy of 83.0% for tweets across all domains. This is a very large vocabulary. If limited to particular domains (such as movies) we feel our classifiers may perform better.

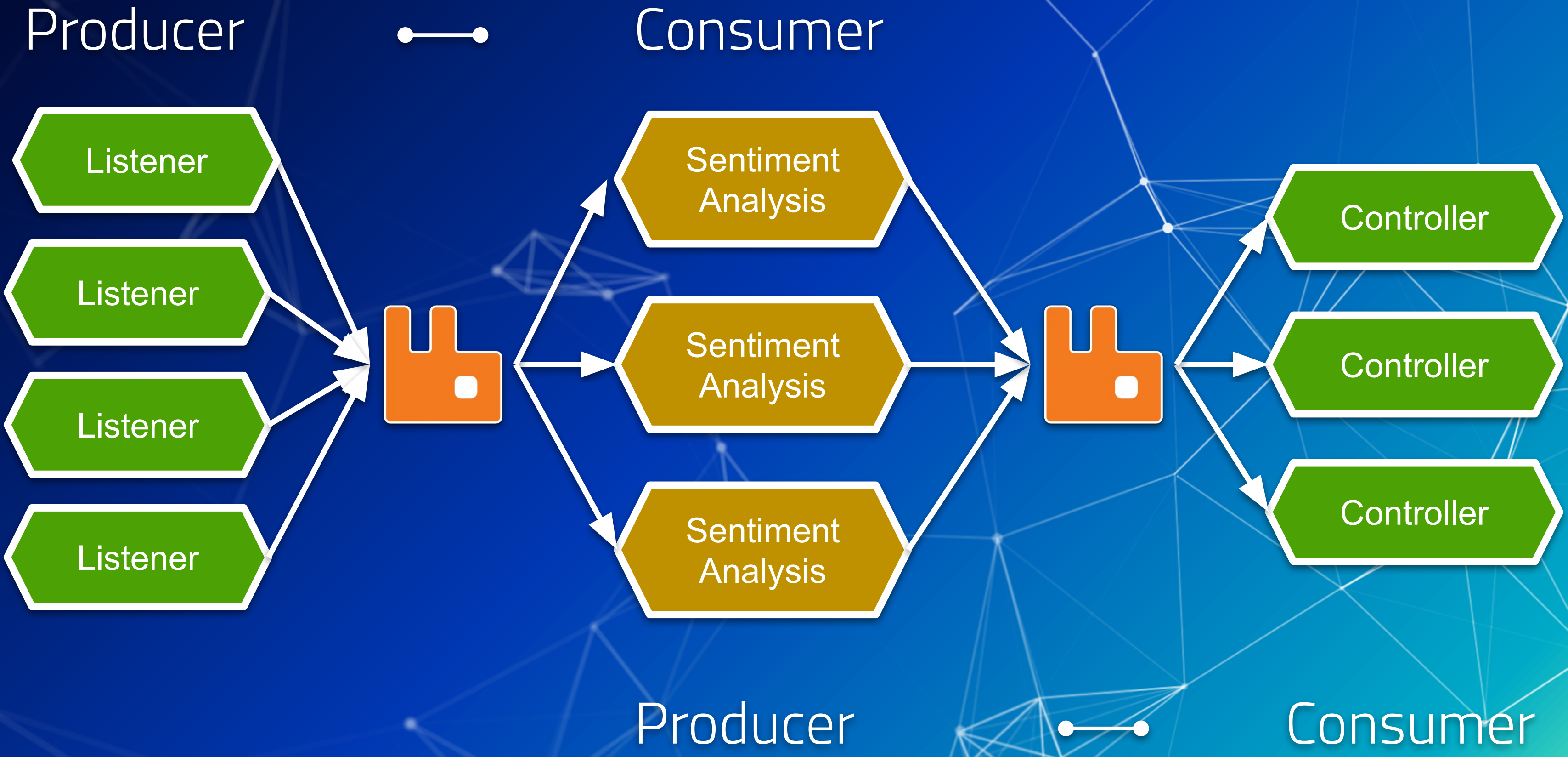


Utilizing emoticon data in the test set Emoticons are stripped from our training data. This means that if our test data contains an emoticon feature, this does not influence the classifier towards a class. This should be addressed because the emoticon features are very valuable.

RabbitMQ

RabbitMQ was utilized to ensure safe, concurrent messaging throughout our Microservices architecture. Scalability was a key requirement we wanted to achieve and RabbitMQ is one of the ways in which we ensured that data transfer balancing would scale with our system.




Producer
Consumer
Listener
Listener
Listener
Listener
Sentiment Analysis
Sentiment Analysis
Sentiment Analysis
Controller
Controller
Controller
Producer
Consumer

Most Impressive Aspects

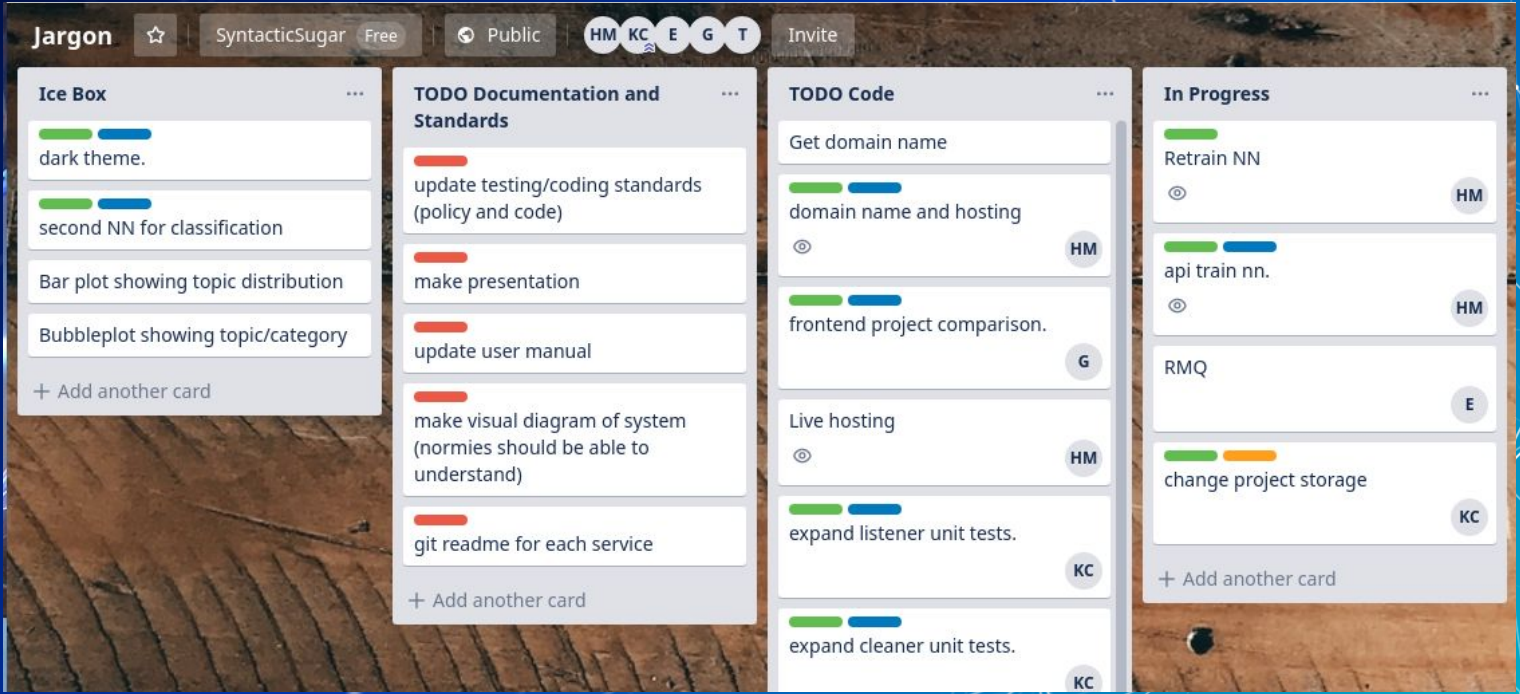
Architecture and use of modern technology stacks.

Use and implementation of sophisticated AI.

Relevance of data in the ever-evolving modern world.

DevOps and Project Management

Trello

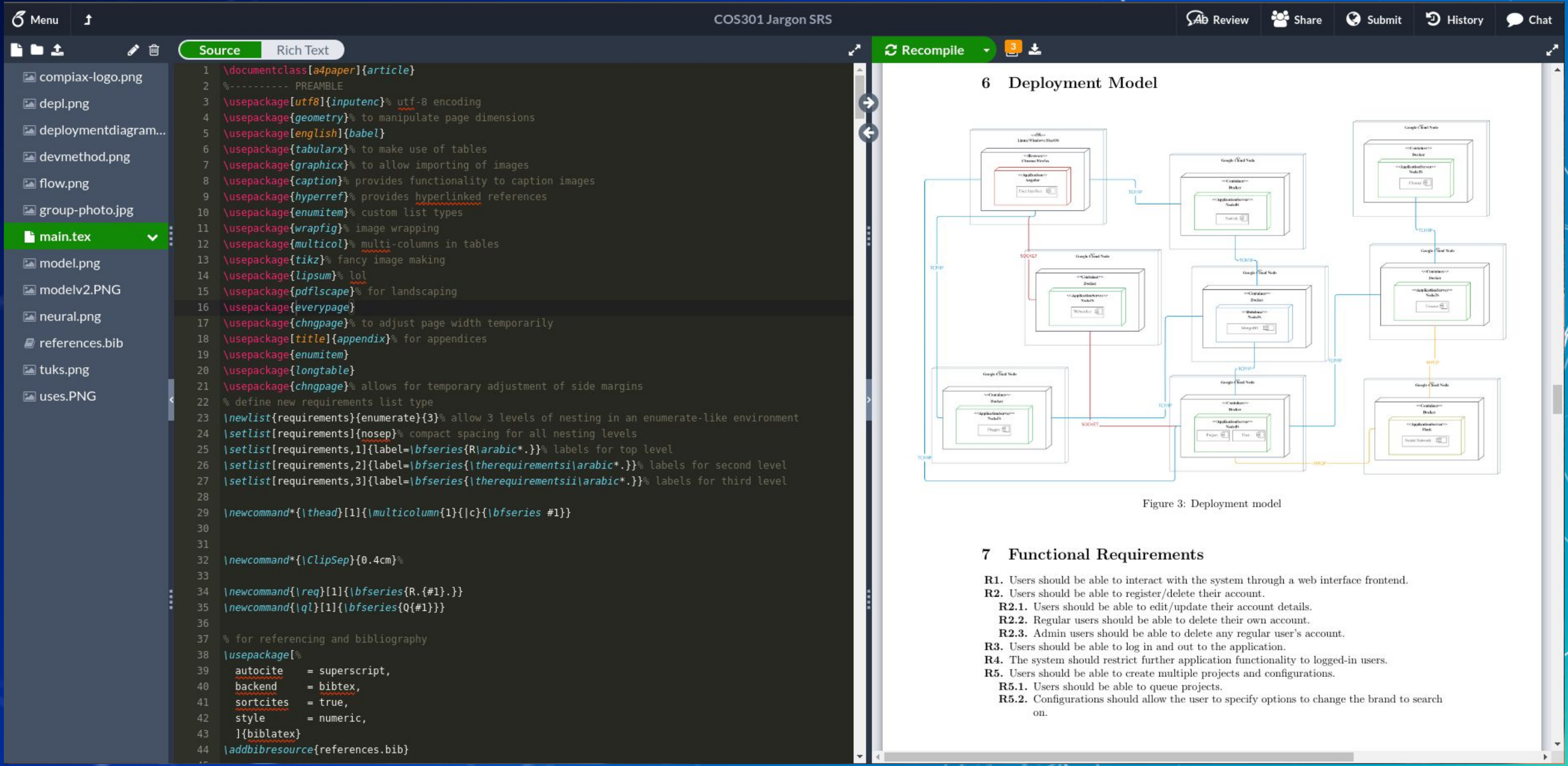

Jargon | SyntacticSugar Free | Public | HM KC E G T | Invite

Ice Box
- dark theme.
- second NN for classification
- Bar plot showing topic distribution
- Bubbleplot showing topic/category
- + Add another card

TODO Documentation and Standards
- update testing/coding standards (policy and code)
- make presentation
- update user manual
- make visual diagram of system (normies should be able to understand)
- git readme for each service
- + Add another card

TODO Code
- Get domain name
- domain name and hosting — HM
- frontend project comparison. — G
- Live hosting — HM
- expand listener unit tests. — KC
- expand cleaner unit tests. — KC

In Progress
- Retrain NN — HM
- api train nn. — HM
- RMQ — E
- change project storage — KC
- + Add another card



```
\documentclass[a4paper]{article}
%---------- PREAMBLE
\usepackage[utf8]{inputenc}% utf-8 encoding
\usepackage{geometry}% to manipulate page dimensions
\usepackage[english]{babel}
\usepackage{tabularx}% to make use of tables
\usepackage{graphicx}% to allow importing of images
\usepackage{caption}% provides functionality to caption images
\usepackage{hyperref}% provides hyperlinked references
\usepackage{enumitem}% custom list types
\usepackage{wrapfig}% image wrapping
\usepackage{multicol}% multi-columns in tables
\usepackage{tikz}% fancy image making
\usepackage{lipsum}% lol
\usepackage{pdflscape}% for landscaping
\usepackage{everypage}
\usepackage{chngpage}% to adjust page width temporarily
\usepackage[title]{appendix}% for appendices
\usepackage{enumitem}
\usepackage{longtable}
\usepackage{chngpage}% allows for temporary adjustment of side margins
% define new requirements list type
\newlist{requirements}{enumerate}{3}% allow 3 levels of nesting in an enumerate-like environment
\setlist[requirements]{nosep}% compact spacing for all nesting levels
\setlist[requirements,1]{label=\bfseries{R\arabic*.}}% labels for top level
\setlist[requirements,2]{label=\bfseries{\therequirementsi\arabic*.}}% labels for second level
\setlist[requirements,3]{label=\bfseries{\therequirementsii\arabic*.}}% labels for third level

\newcommand*{\thead}[1]{\multicolumn{1}{|c}{\bfseries #1}}


\newcommand*{\ClipSep}{0.4cm}%

\newcommand{\req}[1]{\bfseries{R.{#1}.}}
\newcommand{\ql}[1]{\bfseries{Q{#1}}}

% for referencing and bibliography
\usepackage[%
  autocite    = superscript,
  backend     = bibtex,
  sortcites   = true,
  style       = numeric,
  ]{biblatex}
\addbibresource{references.bib}
```

6 Deployment Model

Figure 3: Deployment model

7 Functional Requirements

R1. Users should be able to interact with the system through a web interface frontend.
R2. Users should be able to register/delete their account.
  R2.1. Users should be able to edit/update their account details.
  R2.2. Regular users should be able to delete their own account.
  R2.3. Admin users should be able to delete any regular user's account.
R3. Users should be able to log in and out to the application.
R4. The system should restrict further application functionality to logged-in users.
R5. Users should be able to create multiple projects and configurations.
  R5.1. Users should be able to queue projects.
  R5.2. Configurations should allow the user to specify options to change the brand to search on.

slack

MOCHA

chai

git

Travis CI

Thank You!