Politecnico di Milano
Scuola di Ingegneria Industriale e dell'Informazione
Corso di Laurea Magistrale in Ingegneria Informatica
Dipartimento di Elettronica, Informazione e Bioingegneria

POLITECNICO
MILANO 1863

Working title

Relatore: Prof.ssa Letizia TANCA
Correlatori: Prof.ssa Maristella MATERA
Prof. Riccardo MEDANA

Tesi di laurea di:
Milica JOVANOVIĆ Matr. 000000
Mirjam ŠKARICA Matr. 836505

Academic Year 2015–2016

Some nice inspirational and aspirational quote. Some nice inspirational and aspirational quote.

Someone

Summary

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Abstract

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Contents

List of Figures

Chapter 1

Introduction

Some text [1] Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

[2] This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

1.1 Structure

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

- In the chapter 2 blahblah
- In the chapter 3 Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Chapter 2

State of the art

This chapter describes state of the art of sentiment analysis in social media. Chapter consists of three sections, each of them trying to bring closer the need of sentiment analysis in current market:

1. Need for sentiment analysis

2. Application of sentiment analysis in various companies and non-profit organizations

3. Most used tools for sentiment analysis

2.1 Need for sentiment analysis

With growth of people's interaction and company's advertisements through social media, we have come to the point of realizing that people sharing opinions could help us "predict" stock market and as well follow current trends by guiding the market according to the customers input. Customers nowadays have endless ways to interact with brands which could help increasing brand's awareness but if not properly analyzed could also lead to obtaining not quite accurate view of customer's satisfaction. The idea of analyzing customer opinion has driven companies to search for an automated way of understanding what message are customers sharing online. The main network of spreading opinions is social media. Almost every tweet, comment, re-share or review gives an information that could guide a company towards better planning , optimizing production and better stock managing. Reason for finding an automated way of analyzing customer's opinion comes from a problem of big data being generated each day which makes impractical of doing human analysis of each user input. Leaving the big data problem aside, brings us to another issue; being able to beat natural language processing challenge. Reason for making the task harder is that user input might be informal, "slang like content with emojis , hash tags, even full with sarcastic sentences which would lead to unreliable results of sentiment analysis.

2.2 Application of sentiment analysis in various companies and non-profit organizations

2.3 Most used tools for sentiment analysis

Commercial solutions

As every commercial product, basic goal is user satisfaction. Commercial solutions provide user with rich customizable, easy to use interfaces for a not so fair price. By paying for the service users, usually medium to large scale companies, receive a platform which contains algorithms for data analysis used as a black box and detailed colorful visualization tools for representing results of the analysis. One of important issues that users wouldn't deal with, as they would if building their own solutions, is that such platforms usually come with needed infrastructure to support such data intense analysis. Here we will mention few most widely used commercial tools.

Google Analytics

Google Analytics helps you know your audience, find your best content, and optimize ad inventory. Providing you with real-time reports of what is happening on your site right now so you can make adjustments fast. Engagement metrics help you see what is working, while integrations with Google and publisher tools like AdSense, DoubleClick AdExchange, and DoubleClick for Publishers (Analytics 360 only) make it easy to package and sell your ad inventory. Google has developed a solution which enables the user to gather data, preprocess it, and train a model using Google Prediction API like a black box.

Sales Force Marketing Cloud solution - Radian6

Most certainly that human sentiment analysis is the most accurate method even if you think how much human differ in their interpretation. Radian6 has introduced an automated sentiment analysis tool which has flexibility to allow users to change the perspective of analysis. If you do real sentiment evaluation manually, you will obtain more accurate results than any other automated tool could give you. Given a simple example, if a user compares different beverage brands, most likely he would rate better the beverage he prefers based on the prevailing taste of it. Radian6 solution will enable the user to do deeper analysis into specific topics via different types of ad hoc analysis Radian6 has given various solutions to fill the gap between marketing and customer satisfaction by using social insights to drive marketing campaigns. By listening, engaging and analyzing data on social media, users are able to create sales

plans which could lead to better stock planning.
Brandwatch
Brandwatch Analytics is a web-based platform with monthly subscription basis with different range of packages meeting needs of various scaled companies. They search and store date based on users query on the market. Quite accurately guarantees spam free and duplicate free data. With the gathered data they assure you of optimizing marketing in social media. The platform offers various customizations that could accommodate to the needs of the user. By acquiring data every day and providing users with tools to analyze and visualize them, they have convinced a lot of famous brands that Brandwatch is a good tool to help them make data-driven market decisions such as Cisco, British Airways and Dell. Good thing about Brandwatch as a commercial solution is that it provides coverage of various data sources, independent of language barrier or data quality. Besides of the coverage advantage, it provides stable analytic tools, as well as visualization tools. It is mostly used by large companies that could afford the platform.

Open source solutions

NLTK
Stanford's CoreNLP
Text-Processing

Chapter 3

Sentiment analysis workflow

This chapter describes the workflow used to analyze the sentiment of social media comments and their corresponding posts. In order to outline the workflow, a top down approach was taken where each subsequent section provides an ever more detailed insight into a particular step of the workflow. The big picture is shown in Figure 3.1 and consists of four parts:

1. Obtaining data
2. Sentiment prediction using an API
3. Determining real sentiment of data
4. Evaluation of that API's performance

First part is the simplest one and as such doesn't merit a more detailed recounting other than mentioning that we were provided with a small sample dataset which, most relevantly, contained about 6000 comments.

In the sections that follow, each of the three remaining parts are broken down into conceptual steps describing the methodology used whilst not cluttering it with too many implementation details. Additionally, it is interesting to note that the first and third steps are done only once. This means that, for each new API we want to use, the workflow for sentiment analysis effectively consists of only steps 2 and 4, namely sentiment prediction and performance evaluation.

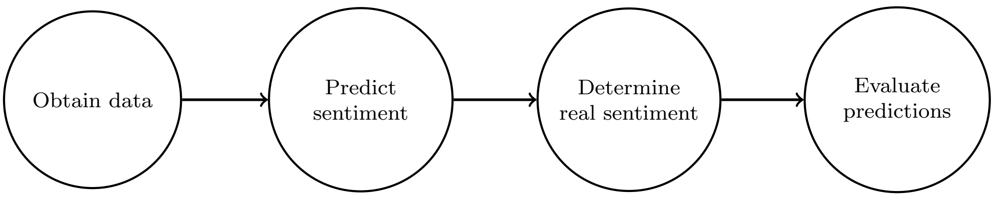

Figure 3.1: Sentiment analysis workflow

3.1 Sentiment prediction workflow

Let's assume we have access to an API for sentiment prediction. And by having access we mean being able to programmatically call the API with a text payload and have it return a prediction in some data format. The end goal is to analyze sentiment of all the comments in our sample dataset and aggregate the obtained data on a per post basis in order to infer whether it is was positively or negatively received, or even if it had no emotional impact whatsoever. And we want this to be done automatically, practically with a push of a proverbial button. By automatizing the process, it is easy to see how it can derive value for possible future ventures that extend far beyond our modest 6000 comment database.

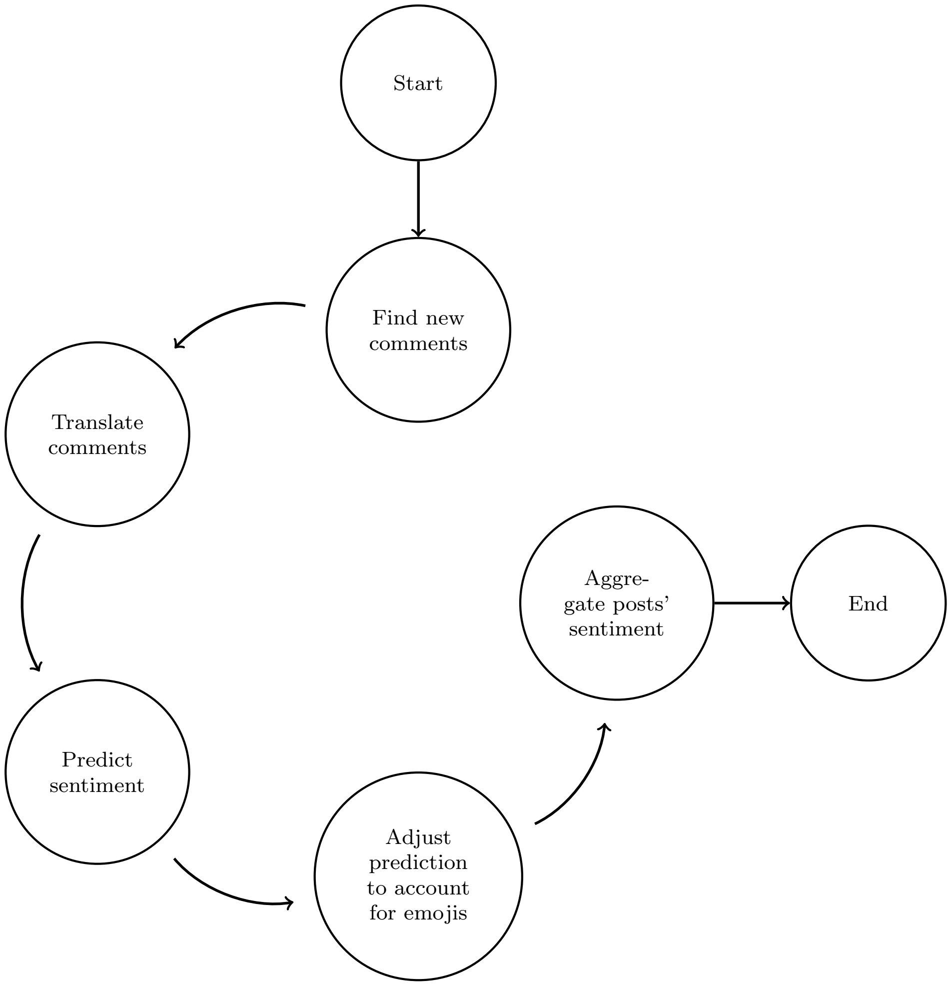

Figure 3.2: Sentiment prediction workflow

Figure 3.2 shows the main concepts that build up the workflow of our sentiment analysis. Since the term *workflow* can be a bit ambiguous, let us clarify exactly what we mean by it. In our case it is simply a python script named named *au-*

tomated_sentiment_analysis.py that can be run manually, or scheduled to run on a server at desired times/intervals. Sections that follow will explain each step in more detail and will also provide motivation for some, perhaps not so obvious, choices.

Find new comments

This part is quite straight forward. Once run, the script scans the database looking for comments that don't have a sentiment record attached to it and inserts one. The inserted rows' sentiment columns default to a json shown in Listing 3.1. The reason for this particular choice of json and for using the json format in the first place is discussed at length in Section 4.1. Also, notice the use of the plural form- sentiment columns. This way we are able to store sentiment predictions from each API we planned on using in their own columns.

```
{
  "sentiment_label": "",
  "sentiment_stats": {
      "positive": 0,
      "negative": 0
      "neutral" : 0
  }
}
```

Listing 3.1: Default sentiment json

Translate comments

To reiterate, our dataset consists of real comments to posts published by actual fashion brands. Since fashion truly is a global industry, the posted comments are in a myriad of different languages. In our case the number of different languages is somewhere north of 70. This provided us with a challenge because most sentiment analysis related APIs handle (well) only content written in English. And the very few that offer support for other languages do so just for a handful of them. This is especially true for the open source variety of APIs that were used for the purposes of this thesis.

Even thought the rationale for using comments' English translations seems to hold, we wanted numbers to back up our claims. In other words, we wanted to quantify just how much worse the APIs would perform if we fed them comments in their original language as opposed to English. So for two out of four APIs used, we analyzed both, the content in original language and the English language. The results are examined in Chapter 5, but in short, they are in accordance to what we expected.

This brings us to another caveat. We've just coupled the quality of sentiment predictions with the quality of the translations. After all, the prediction can only be as good as the translation. Since we were trying to evaluate performance across multiple open source APIs, we wanted the best translations possible to try to mitigate this problem. Hence we opted for what we felt was the current industry standard, Google's Translate API[1]. It is worth noting that this is the only step we hadn't taken the open source option but used a free trial period instead to do a one-off translation of our entire dataset.

Predict sentiment

For each unanalyzed comment a we call a specific API requesting a sentiment prediction of the comment's translated content[2]. If no API is specified the script sequentially makes requests to all defined. Since each API's response is in a slightly different format, the response is parsed to adhere to the json definition shown in Listing 3.1. After which, the API's sentiment column for that particular comment is updated with the received (and parsed) values.

Adjust prediction to account for emojis

In this day and age everybody uses emojis and emoticons, and a lot of it. To disambiguate the two terms, here are the definitions offered by the Oxford dictionary:

emoji / ɪˈməʊdʒi /

> ***origin*** (1990s) Japanese, from e=picture + moji=letter, character
>
> ***noun*** a small digital image or icon used to express an idea or emotion

emoticon / ɪˈməʊtɪkɒn /

> ***origin*** (1990s) blend of words emotion + icon
>
> ***noun*** a representation of a facial expression such as a smile or frown, formed by various combinations of keyboard characters and used to convey the writer's feelings or intended tone

To put it simpler, the difference is between symbols ❤️ and <3. The former being an emoji and the latter being an emoticon. But we digress, the point was to emphasize the very emotional nature and motivation behind using these symbols in a text, comment or post. Having an emoji or an emoticon mixed with text can drastically change our perception of the sentiment behind it. Take these three simple comments:

[1] https://cloud.google.com/translate/v2/translating-text-with-rest

[2] As mentioned in the previous section, there are two API's for which we requested sentiment predictions in both, their original language and the English translation

```
I read that book
I read that book <3
I read that book ❤️
```

Unless we happen to know the person that wrote the the first comment, its content in plain text doesn't really codify enough information for us to make a judgment call weather or not this person liked or disliked that book. On the other hand, the other two comments are quite unambiguously positive. That one little symbol made all the difference in how we perceive the text that preceded it. Unfortunately, all APIs that we tested would ignore these descriptive symbols, so we decided to write up a very simple algorithm based on the *Emoji Sentiment Ranking*[3] which came to be as a part of the Sentiment of emojis study[3]. The algorithm will be described in more detail in Section 4.2. But in short, the algorithm tweaks the sentiment of comments which contain emojis or emoticons. Then it stores the recalculated result in a separate database table so it doesn't clobber the original data. This allows us to both fine tune our algorithm and to compare the predictions that took the sentimental value of emojis into account to those that didn't.

Aggregate posts' sentiment

Everything leading up to and including this point was done automatically by running the *automated_sentiment_analysis.py* script. Finally, all that is left for the script to do is to aggregate the sentiment data for each post. This boils down to counting how many sentimentally negative, neutral or positive comments does a post have. The results of this data aggregation are stored in a json format as shown in Listing 3.2. Perhaps the most informative field there is the *sentiment_label*. It is essentially one API's appraisal of how well (or badly) had the public received a published post. Of course, this aggregation is done for each post and API separately. So, for example, according to one API a post might have been overall positively received, while data coming from another API might yield a different conclusion. Sections 3.2 and 3.3lay out workflows for assessing API's reliability.

```json
{
  "sentiment_label": "positive",
  "sentiment_stats": {
    "positive": 38,
    "negative": 2,
    "neutral": 9,
    "total": 49
} }
```

Listing 3.2: Example of a post sentiment json

[3]http://kt.ijs.si/data/Emoji_sentiment_ranking

3.2 Determining real sentiment workflow

In order to answer the question weather or not the obtained sentiment predictions are any good and to determine if any one API outperforms all others- we need sentiment data that we hold true and we need it for each comment. That way we have a real (true) sentiment record to compare against. Since the only state of the art sentiment analyzing machines at our disposal were the two humans writing this thesis, we decided to read all the comments one by one and input our sentiment predictions by hand. Thus, from this point on, when ever we refer to *real sentiment* we mean our own judgment of the sentiment behind the comment. To make this manual process a bit easier for ourselves, we've made it possible to input or modify sentiment for each comment in multiple ways. It can by done via the command line, e.g by doing a curl call to the framework's REST API or via its the graphical user interface. But easiest and most efficient way is to run the *update_real_sentiment.py* script. The script allows you to specify a range of comments which you want to analyze using command line arguments. The script then sequentially fetches specified comments, prints out their ids, content and English translations and asks for 3 pieces of information as shown in Figure 3.3. It requests a sentiment prediction to be input, weather or not one assesses this comment to be spam and if there was a mention of another user in the comment in question. We were interested to have the two last pieces of intelligence mainly out of curiosity to see how API's would have performed if the dataset was clean from these types of comments, however, they are also a good basis for future extensions of our work.

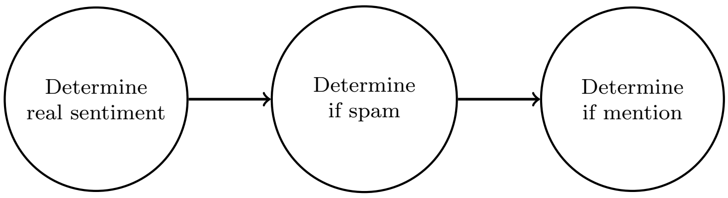

Figure 3.3: Determine real sentiment workflow

3.3 Evaluation workflow

Now that we have real sentiment of each comment as well as sentiment predictions, we can evaluate performance of each individual API. Running the *evaluate_api_performance.py* will calculate accuracy, precision and recall for each API unless a specific one is specified as an argument.

Chapter 4

Framework

4.1 Design

Why json when it violated the 1NN rule? already in mysql, will eventually support json, and easily movable to nosql db, or even elastic search.

4.2 Implementation

Emoji analysis describe the simple alg https://github.com/mirjamsk/sentiment-analysis/wiki/Emoji-analysis

4.3 User interface

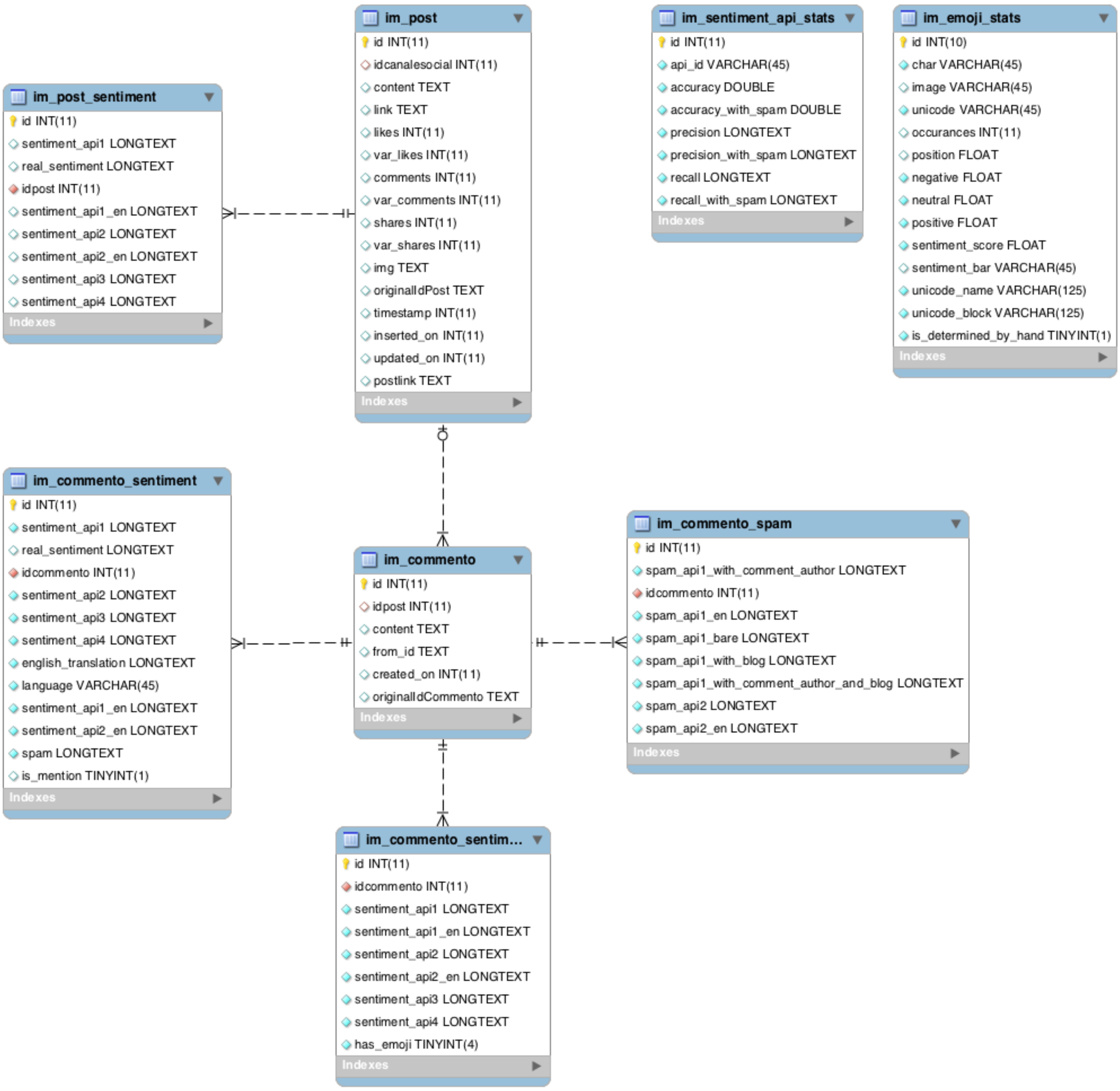

Figure 4.1: TTest caption

Chapter 5

Results

For the sake of completeness, below are the definitions used to calculate those 3 metrics. Accuracy is the simplest of all metrics as it is just the fraction of correctly classified comment sentiments.

$$Accuracy = \frac{number\ of\ correct\ prediction}{total\ number\ of\ comments}$$

Precision and recall, on the other hand, are a bit more complex to understand and are calculated separately for each sentiment label $\in \{positive, negative, neutral\}$. They use concepts such as $True\ Positive$, $False\ Positive$ and $False\ Negative$. In the context of out framework, these concepts are calculated as:

True Positive correctly predicted labels correctly identified like pairs real sentiment, predicted sentiment positive,positive, negative negative

One first item

Two second item

Three third item

False Positive number of times it didn't predict the label when it should have incorrectly identified negative, neutral

False negative umber of times it predicted the label when it shouldn't have

Recall is the proportion of positive, negative, neutral comments were actually predicted as positive, negative, neutral. In other words, out of all the positive, negative, neutral examples, what fraction did the classifier pick up?

$$Recall = \frac{TP}{TP + FN}$$

Precision is the proportion of labels that were predicted as positive,negative, neutral actually are positive,negative, neutral. In other words, out of all the examples the classifier labeled as positive,negative, neutral, what fraction were correct?

$$Precision = \frac{TP}{TP + FP}$$

Chapter 6

Conclusion

Chapter 7

Future work

Bibliography

[1] M. Masse, *REST API design rulebook.* O'Reilly Media, Inc., 2011.

[2] F. Corvetta and V. Rizzo, "Progettazione e sviluppo di mashup mobile context aware: il progetto camus," Master's thesis, Politecnico di Milano, 2015.

[3] P. Kralj Novak, J. Smailović, B. Sluban, and I. Mozetič, "Sentiment of emojis," *PLoS ONE*, vol. 10, no. 12, p. e0144296, 2015.