GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

ISSN 1612-6793
Nr. ZAI-BSC-2015-11

Bachelor's Thesis

submitted in partial fulfilment of the
requirements for the course "Applied Computer Science"

Analysis and Prediction of User's Happiness in Online Social Networks

Stefan Peters

Institute of Computer Science

Bachelor's and Master's Theses
of the Center for Computational Sciences
at the Georg-August-Universität Göttingen

19 May 2015

Georg-August-Universität Göttingen
Institute of Computer Science

Goldschmidtstraße 7
37077 Göttingen
Germany

☎ +49 (551) 39-172000
FAX +49 (551) 39-14403
✉ office@informatik.uni-goettingen.de
🌍 www.informatik.uni-goettingen.de

First Supervisor:	Prof. Dr. Xiaoming Fu
Second Supervisor:	Dr. Xu Chen

I hereby declare that I have written this thesis independently without any help from others and without the use of documents or aids other than those stated. I have mentioned all used sources and cited them correctly according to established academic citation rules.

Göttingen, 19 May 2015

Acknowledgments

First of all, I want to thank Prof. Dr. Xiaoming Fu for introducing me to this interesting topic and supervising this thesis.

I am also very grateful to Dr. Xu Chen and Hong Huang who have supported me continuously throughout working on this thesis and offered ideas and input to improve my work.

Most of all, I am thankful for the unbroken support of my parents which made this possible in the first place.

I also want to thank my dear friends, fellow students, LSG buddies and all the other people who know me and stood by my side. Special thanks to our cats, Hanni and Loki, who sweetened my life ever since we got them.

Abstract

In today's society, it is hard to find a person that is not using some kind of online social network. Starting early in the history of the World Wide Web, online social networks quickly gained a large popularity among internet users. Facebook, introduced in 2004 and opened to the public in 2006, is now the largest social network with about 1.4 billion monthly active users.

Microblogging is a certain kind of online social network in which users post short messages that appear on their followers feeds. Twitter is one of the most popular microblogging platforms, and with its public access, Twitter is a great source for collecting a large amount of data for research and studies.

Using a data set of posts from Twitter, so-called "Tweets", the sentiment of those posts was analyzed. Each Tweet either got a happiness score between 1.0 and 9.0, or proved not to be eligible for further analysis. Additionally, Tweets were also assigned one of the categories "positive", "neutral" or "negative" based on the calculated happiness score.

With the scored Tweets as a basis, a regression analysis was performed on individual users to predict the happiness of their future Tweets based on available data about their past Tweets. Different features of the Tweets, like the time of the day or the day of the week on which they were posted, were used as predictors for the regression.

Some features proved to be better suited than others, and while the resulting correlation between features of a Tweet and their sentiment does not seem strong enough to accurately predict that Tweet's sentiment, it could be used to get fairly good predictions on average over a large number of Tweets.

Contents

Chapter 1

Introduction

1.1 Online Social Networks

Since Facebook was introduced in February 2004 [1], online social networks (OSNs) have become a consistent part of most people's lives [2]. In 2014, Facebook had 1.393 billion monthly active users [3], the microblogging platform Twitter has 302 million monthly active users [4]. People spend 1.72 hours and 0.81 hours daily on social networking and microblogging respectively [2]. Spending this much time on those platforms, a vast amount of data is created and can be harnessed for useful knowledge.

Especially public networks like Twitter are useful for this kind of analysis, thus in this thesis, public Twitter data is used. On Twitter, the users "tweet" short messages with up to 140 characters to the rest of the world [5]. These posts are therefore called "Tweets" and this expression will be used troughout this thesis.

As most of people's activities on networks like Twitter are personal, sentiment plays a huge part in the data created on these networks. Sentiment analysis provides the necessary ways to analyze this data in respect to the emotions expressed by the users. In this thesis, the focus lies on the happiness of users; is a user's Tweet rather happy or rather unhappy? A way to measure this is to look at the individual words of a Tweet posted by a user. The aggregate of the emotions of the words in a Tweet can represent the emotions of the whole Tweet. Similarly, the entirety of a user's Tweet's emotions can be used to make a statement about that user's overall happiness.

With the knowledge about the sentiment of users' Tweets, this data can be used to find a correlation between a Tweet's features and the sentiment of that Tweet, in regard to one specific user. The goal of this thesis is to find such a correlation and to predict a user's future Tweets' sentiments using this possible correlation.

1.2 Thesis

This thesis consists of three main topics:

Accumulation and mining of Twitter data.

Using Twitter's extensive application programming interface (API) it is possible to obtain a large number of Tweets within a short timeframe. This can be further distinguished into the use of Twitter's streaming API and search API.

Analysis of the emotions in Tweets (sentiment analysis).

One approach of sentiment analysis, namely using linguistic features by analysing the words of a Tweet, is used to determine either a "happiness score" (on a scale from 1.0 to 9.0) or a "happiness category" (positive, neutral or negative).

Prediction of a Tweet's sentiment using other features.

Using the previously calculated happiness scores or categories, regression analysis is used to find a link between the sentiment and other features of a Tweet. The results of the regressions are used to predict the sentiment of future Tweets and the quality of the prediction is determined by comparing the predicted values with the calculated sentiment values.

The preparation, implementation and results of these three parts will be explained in the respective chapters of the thesis (see section 1.4 Organization).

1.3 Contribution

The result of this thesis are methods to

- collect a large number of Tweets from the Twitter API,
- preprocess the texts of those Tweets,
- use the preprocessed texts to calculate a happiness score for each individual Tweet,
- find a correlation between other features and the calculated happiness score (or the resulting happiness category) and use this correlation to predict a Tweet's happiness from those other features.

This methods are documented and the process can be reproduced and adjusted by anyone. It can also serve as a basis for further analysis.

1.4 Organization

This thesis is organized into five parts. Chapter 1 was a short introduction and is finished after this section. Chapter 2 (Foundation and Background) will cover necessary background knowledge to establish a foundation to understand this work. Chapter 3 (Methods and Implementation) will describe the methods used for mining Tweets and analyzing and predicting the sentiment of Tweets, and will also cover parts of the implementation of these. Chapter 4 (Results) will present the data and the results of the described methods, and chapter 5 (Summary and Conclusion) will summarize this thesis and draw conclusions out of the results.

Chapter 2

Foundation and Background

2.1 History of Online Social Networks

Ever since the introduction of the World Wide Web in 1991 [6], the internet has become a part of most people's lives. People spend an average of 6.15 hours on the internet every day [2]. A lot of people in today's society claim they "could not live without the internet" [7]. This shows how important the internet has become for our society and humanity as a whole.

OSNs started early in the history of the internet. The need to communicate with other people led to the creation of the email in 1971 [8]. This was continued by the Bulletin Board System (1978) [9], Usenet (1979) [10] and Internet Relay Chat (IRC) as the first ever instant messaging system (1988) [11]. The World Wide Web then led to the creation of more sophisticated services that were also available for a much larger number of people.

danah m. boyd lists SixDegrees.com as "the first recognizable social network site" [12] because it combined several features of previously existing websites into one platform. As with most pioneers in their field, the site did not become profitable and closed down in 2000. Many other services that followed SixDegrees' idea spawned in the following years with varying success, most of which concentrated on specific groups of people. Facebook, which was launched in the beginning of 2004 [13], was one of those sites. It was only available for Harvard University students but was quickly expanded to other universities and eventually opened to the public in 2006 [14]. Facebook is now the most known and used OSN [15].

2.2 Microblogging

Microblogging can be defined as "the act or practice of posting brief entries on a blog or social-networking website" [16]. It started with actual blogs that were dedicated to very short entries (also called tumblelogs) [17], but was quickly adapted by major OSNs in the form of “status updates” and by websites made specifically for microblogging. The first microblogging site in the sense that we define it today was Twitter which launched in March 2006 [18]. Twitter is still one of the most popular microblogging platforms in the world and its users are sending about 500 million Tweets per day [4].

Tweets can be up to 140 unicode characters in length and can contain different media such as photos and videos. Users can mention other users by typing @username; those users will then be notified and can respond to that Tweet. Twitter allows the usage of #hashtags to tag a Tweet with a certain topic. Also, users have the option of retweeting another user’s Tweet, which means sharing that Tweet with their own followers. By following other users, those users’ Tweets will appear in one’s own feed. In general, Tweets are posted publicly, but there is the possibility to make an account private and only allow specific users to see the Tweets [19]. The huge number of public Tweets in combination with the public API Twitter offers makes Twitter a perfect platform for accumulating data for various types of studies and analyses. This is why Twitter was chosen as a data source for this thesis.

2.3 Sentiment Analysis in Online Social Networks

Bing Liu [20] defines sentiment analysis the following way:

> “Sentiment analysis, also called *opinion mining*, is the field of study that analyzes people’s opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes.”

This definition displays very well how suited OSNs are for this field of study. OSNs are the main place where people of today’s society express their sentiments towards those entities listed in the quote. So it’s not surprising that OSNs have become one of the main sources for sentiment analysis studies.

Go et al. [21] provided one of the first studies using natural language processing (NLP) for sentiment analysis on data from microblogs (in this case Twitter) in 2009. They developed “a novel approach for automatically classifying the sentiment of Twitter messages” and used this to classify the public sentiment about specific products and brands. In the following years, sentiment analysis in OSNs has become a popular study field.

In 2010, Cha et al. [22] compared different measures of influence between Twitter users and found “new insights for viral marketing”.

In 2012, Kim et al. [23] studied “Social Aspects of Emotions in Twitter Conversations” where they found out that usually, a conversation partner responds to a message with a similar emotion, which they called *Emotion accommodation*.

In 2013, Hogenboom et al. [24] used emoticons to improve the results of sentiment analyses in social networks.

There are many more examples of sentiment analysis in OSNs. Some more practical examples are discussed in the next section.

2.4 Applications of Sentiment Analysis

There are many practical applications of sentiment analysis in OSNs. Here are some examples:

Opinion Finding

Companies and other institutions can perform market evaluations through sentiment analysis in OSNs. For example, a company could evaluate the effectiveness of a change in their strategy by looking at the general sentiment about their brand or product before and after announcing the change.

Stock Prediction

Bollen et al. [25] found a correlation between the public Twitter feed and the Dow Jones Industrial Average ups and downs. They were able to predict the daily changes with an accuracy of 87.6%.

Comparing Living Standards

You et al. [26] used user-generated content from Flickr.com to compare the happiness of people in different cities in the US. Though they were only using a relatively small data set, they were able to find correlations between real world factors, like the criminal rate, and the happiness score of a city.

Measuring people's opinions in regard to certain things has become more and more important, and sentiment analysis offers a way to quantify this.

Chapter 3

Methods and Implementation

This chapter describes the methods used in this thesis and how they were implemented. First, the data from the Twitter API has to be collected and stored into a database. For the database backend, MySQL [27] was chosen because it is widely compatible, flexible and scales nicely even for large tables of data. During the collection of data, a sentiment analysis is performed for every Tweet that is stored in the database. The result of the sentiment analysis will either be a score between 1.0 and 9.0 (higher means happier), or 0.0 when the analysis could not determine a sentiment in the Tweet. This happiness score is also used to sort the Tweets into happiness categories (positive, negative or neutral). In the last part, the collected Tweet and corresponding sentiment data is used for a regression analysis and the results of the regression are used to predict the sentiment of Tweets.

All of the described methods are implemented in Python. Python is a simple, flexible, cross-platform programming language which has become much more popular these last years [28]. It has many available libraries that can easily be installed via pip [29] or easy_install [30], and many of those libraries like Pattern [31], pymysql [32] and statsmodels [33] are being used in this thesis.

3.1 Data Collection

In this section, it is described how the data, necessary to perform the planned analysis, was collected from the Twitter API and stored into a MySQL database.

3.1.1 Twitter API

Twitter offers a public API for accessing their service. To be able to use the API, one has to have a Twitter account and create an application to receive API-keys [34]. These keys are used to authenticate the application against the Twitter servers.

For the purpose of this thesis, two main API functions were used: streaming and getting the timeline of a user. Twitter gives access to real time Tweets through their streaming API which also includes filters like language and keywords [35]. Using the sample stream, the output rate of Tweets is very limited. More details on this can be found in section 3.1.3. In addition to the sample streaming with the language filter for English, the search API provides timeline methods that were used to receive a more complete set of Tweets for a specific user. These methods allow access to a user's most recent 3200 Tweets [36]. This was used for the top 5000 Twitter users by follower count present in the database. Only these users were considered for the subsequent analysis.

As a means for accessing the Twitter API, the *Tweepy* library for Python [37] was chosen as it is one of the recommended ways of accessing the Twitter API [38]. *Tweepy* has support for both the streaming and the search API. The results are transmitted via JavaScript Object Notation (JSON) and *Tweepy* packs them into an own container that can easily be used within Python.

3.1.2 Data Extraction and Storage

The data was collected by using Twitter's streaming and search APIs. The script collecting data from the streaming API was started on 17 February 2015 at 5:50 pm, the script collecting data from the search API (i.e. mining Tweets from the users' timelines) was started shortly after on 20 February 2015 at 11:31 am. Both scripts were stopped on 13 April 2015 at 10:52 am. Therefore, the total timespan in which the data was mined was 54 days, 17 hours and 2 minutes long.

Regardless of the API used to access Twitter, the resulting data has the same format and therefore processing and storage were handled the same for both ways.

A Tweet or *status* contains several general features of the Tweet, informations about the authors and several so called *entities* (e.g. URLs, hashtags, mentions, . . .). The following features of a Tweet were extracted and saved to the database:

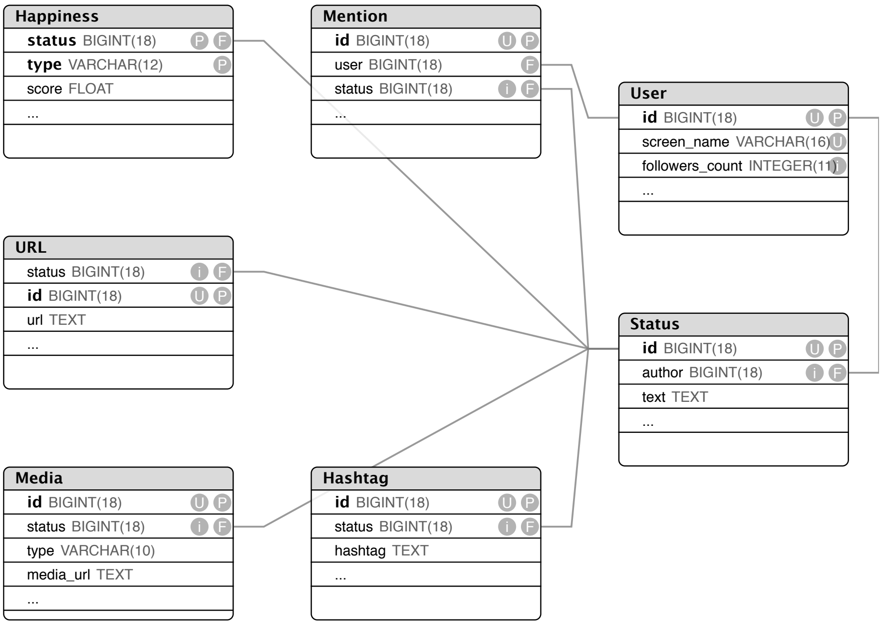
- ID
- author
- text
- time of creation in UTC
- UTC offset of author (timezone)
- if it is a reply: user and status id that was replied to
- favorite count
- retweet count
- language tag
- source (client used by the user, e.g. *Twitter for iPhone*)
- if it is a retweet: the status ID that was retweeted
- URLs contained in the Tweet

- hashtags contained in the Tweet
- user mentions
- media contained in the Tweet

The following features of a user were extracted and saved to the database:

- ID
- screen name (username)
- display name
- followers count
- friends count
- statuses count
- UTC offset
- location, if transmitted
- language tag
- favorites count
- verification status
- time of creation
- profile description

This data is stored into a MySQL database. Several database tables were created to accommodate the data. The following diagram shows the basic structure of the database. Note that only the most important columns are shown for each table. The complete database structure can be found in appendix A.



In addition to these tables, there are two additional tables. **Status_wl0_3** and **Status_filled** are intermediate tables used during the regression analysis. The former contains the data that is directly used for the regression analysis. This data is assembled once and can then be accessed quickly for subsequent analysis runs. The later contains information about which users' Tweets were already compiled into the **Status_wl0_3** table.

3.1.3 Limitations

Twitter has several rate limits when using their API [39]. Using the timeline method, 200 Tweets can be retrieved per API call [36]. Calls to the timeline are limited to 180 calls per 15-minute-block per API-key per method [39]. Therefore, using the timeline method to obtain users' histories of Tweets, 36,000 Tweets can be retrieved per 15-minute-block.

Twitter does not allow going back in time indefinitely when accessing a user's timeline. Only the latest 3,200 Tweets can be retrieved using this method [36]. Therefore, the data set used in this thesis did not contain more than 3,200 + x Tweets per user while x is the number of Tweets the user posted after the first time their timeline was dumped.

As mentioned in the previous section, the streaming API also has limitations. While the access to the API is not restricted by time or number of Tweets, the number of Tweets offered by the streaming endpoint is limited [35]. Twitter does not clarify exactly how many Tweets are offered through the streaming API, but by the usage of the API for this thesis, a rate of at least 1000 Tweets per minute can be estimated.

3.1.4 Implementation

In this section, the basic implementation of the data collection will be explained. For a full overview of all files, see appendix B. For the full source code, please consult the attached CD or use the download link under appendix B. For a guide on how to install and use the code, please see appendix C.

database/DatabaseCreds.py

In this file, the database credentials are stored. It also contains a method *getConnection* which can be used to get pointers to the database and a cursor.

database/TwitterTokens.py

In this file, the Twitter API tokens are stored. Please consult the comments within the file on how to obtain these API tokens.

database/SaveStatus.py

This file offers a class to save Tweets that were obtained through the Twitter API into the MySQL database. Upon saving, the sentiment score will already be calculated for convenience (as explained in the next section).

database/TweetStreaming.py

This file is executable. It uses the *tweepy* Python library [37] to access Twitter's streaming API. It watches the sample stream with the language filter "English" and saves all incoming Tweets into the database.

database/TweetDumping.py

This file is also executable. It uses the *user_timeline* method from the Twitter API to obtain the Tweet history of the top 5000 Twitter users in the database (by follower count) and saves those Tweets into the database. This file is essential for this thesis because a large number of Tweets per user is needed for performing the regression analysis described in a later section.

The files *TweetStreaming.py* and *TweetDumping.py* can be run simultaneously to collect Tweets from the Twitter API and save them into the database. It is recommended to let *TweetStreaming.py* run for a while before running *TweetDumping.py* so that an accurate list of the top 5000 Twitter users can be built.

3.2 Sentiment Analysis

For analyzing the sentiment of the collected Tweets, a linguistic approach was chosen. By using word lists collected from the biggest online content providers and letting humans score those words in regard to happiness, Tweets are given a score between 1.0 and 9.0, higher means happier. These scores therefore represent the sentiment in a Tweet and are used for further analysis.

3.2.1 labMT 1.0

In 2011, Dodds et al. [40] compiled a set of the most frequently used 10,222 English words from different sources (Twitter, Google Books, music lyrics and the New York Times) and let humans evaluate the sentiment of these words on a scale from 1 to 9 (while 1 is the least happy and 9 is the happiest). The average of these values resulted in one happiness score for each word, and the total word list with all happiness scores is publicly available [41] and can be used for sentiment analysis and similar purposes, like this thesis. Here are some examples for word scores:

- laughter: 8.50
- love: 8.42
- dog: 6.70
- you: 6.24

- spending: 5.24
- these: 5.10
- scary: 2.58
- killed: 1.56

The happiness score of a phrase (or Tweet) could be calculated by taking the average of all the words in that phrase which have a happiness score assigned to them. Some examples:

- "I love you." Happiness score: 7.33
- "My dog was just killed..." Happiness score: 4.13

Neutral words or *stop words*, like "these" or "spending" should be excluded because they don't express any sentiment and only skew the score to the average.

3.2.2 Modifications

In comparison to the approach of Dodds et al. [40], there were a few things that were added or changed for this thesis:

- The exclusion of stop words, i.e. words that are neutral and should not be included in the sentiment analysis, was changed in that words that have a score between 4.5 and 6.2 were excluded instead words with a score between 5.0 and 6.0. Testing revealed that these boundaries work better for the score calculation because more neutral words are being excluded.
- The Tweets were more thoroughly preprocessed. This is described in detail in the next section.
- Only Tweets which text contains at least two words with a score in the labMT word list were given a sentiment score and were therefore used in the subsequent analysis. This was done to exclude Tweets that don't express enough sentiment to qualify for such an analysis.
- Emoticons like :) or :-(were taken into account. They are replaced with placeholders (e.g. *SMILEY_HAPPY*) that were manually assigned a sentiment score by choosing the score of a word from the existing word list that can represent that smiley. This change was inspired by Hogenboom et al.'s paper about using emoticons for sentiment analysis [24].
- Negating words, like *don't*, *didn't*, *shouldn't* and *not*, were taken into the calculation by flipping the score of the following word.

  The following equation was used for flipping a score: $new\_score = 10 - \frac{(old\_score+5)}{2}$. Because of the fact that negating a very positive word will not result in a very negative sentiment, a more restrained equation was used that flips the score to the other side of the neutral point (5.0), but shifts it in direction of this neutral point (examples: 7.0 -> 4.0, 4.0 -> 5.5).

3.2.3 Preprocessing

Preprocessing means taking apart the Tweet text and preparing it for the actual sentiment analysis. The result of the preprocessing process is a list of tokens (words and emoticons in this case) which are then used for determining the sentiment of the Tweet. Preprocessing is mostly about removing unnecessary data and bringing the rest of the text into a standardized form. To support this process, a Python library named *Pattern* is used [31]. *Pattern's* submodule *Pattern.en* has methods for parsing written text [42]. For this process, the *parsetree* method is used to split the Tweet text apart. The resulting parsed words are further processed in the following way:

- All mentions are removed.
- All URLs are removed.
- The # of hashtags is removed.
- All retweeted text is removed.
- Stop words and other unnecessary words are removed.
- Punctuation is removed.
- Emoticons are categorized and replaced by tokens.
- When encountering a negating word (*not* or a word ending on *n't*), the following word gets a prefix *NOT_* which tells the sentiment scorer to flip the score for this word.

Some of these steps were inspired by other works using preprocessing, namely [43] and [44].

3.2.4 Implementation

In this section, the basic implementation of the sentiment analysis will be explained. For a full overview of all files, see appendix B. For the full source code, please consult the attached CD or use the download link under appendix B. For a guide on how to install and use the code, please see appendix C.

preprocessing/Preproc.py

This file offers a class for the preprocessing part of the sentiment analysis. It contains only one method, called *process*, which takes the Tweet text as an argument and returns a list of lemmas and tokens that are used for the actual sentiment analysis. The used *pattern* library [31] and its method *parsetree* distinguish between “strings” and “lemmas”. A “string” is the actual string as found in the original text. A "lemma" is the base form of that word, as it is found in a dictionary. Both are used in this thesis.

scoring/SentimentScorer.py

This file offers a class for the actual sentiment analysis. It offers a method *score* which expects a phrase (in this case, the Tweet text) as an argument and returns either a float value between 1.0 and 9.0 (1.0 is the least happy, 9.0 is the most happy) or "None" of no sentiment could be evaluated.

3.2.5 Other Efforts

Besides the approach of using the labMT word list for the sentiment analysis, other options were considered. One of them was using the already used *Pattern* Python module developed by the Computational Linguistics & Psycholinguistics Research Center [31]. This module offers a variety of different methods in regard to data mining, including NLP. A simple *sentiment* method is offered by the submodule *pattern.en* which expects a string and returns a (polarity, subjectivity)-tuple representing the sentiment of said string [42].

This module was not used, mainly due to two reasons. First, it was not clear where exactly the scores used by the Pattern module come from and how the scores are calculated. Second, the offered *sentiment* method is said to be working best for product reviews [42]. Therefore, using smaller texts may not be optimal with the *Pattern* library.

3.3 Sentiment Prediction

After collecting enough data from the Twitter API and performing sentiment analysis on the collected Tweet texts, the next endeavor was to predict the sentiment of a Tweet through other features relating to that Tweet (e.g. time of the day, day of the week). For this, regression analysis was used to find a correlation between a set of features and the sentiment of a Tweet using Tweets from one specific user. Using the resulting model of the regression analysis the sentiment of the Tweet was predicted, and this prediction was then evaluated by comparing the predicted sentiment to the calculated sentiment using the labMT word list [40] described in the previous section.

3.3.1 Regression Analysis

There are two types of regressions used in this thesis. Both of these are linear regression models.

Ordinary least squares (OLS) regression

OLS is one of the simplest regression models. With only one independent variable, it tries to find a line that best fits the relation between the dependent and the independent variable. This can be transferred for multiple independent variables, as it will be for this thesis.

Multinominal logistic (MNLogit) regression

MNLogit can be used for a nominal dependent variable with more than two states (in a regular logistic regression, the dependent variable can only have two states, i.e. true or false). It tries to find a linear fit of the independent variables to predict the state of the dependent variable.

To convert the calculated happiness scores into happiness categories for the use of MNLogit, the complete set of calculated happiness scores were split into three roughly equal parts. This results in the following: Sentiment score < 5.8 → negative. Sentiment score > 6.675 → positive. 5.8 <= sentiment score <= 6.675 → neutral.

As described at the start of this chapter, Python was used in every part of this thesis, including the regression analysis. For the regression analysis, the *statsmodels* Python module [33] was used. It offers a variety of different statistical methods, including various types of regression analysis. The submodule *statsmodels.formula.api* offers methods for these regressions. For this thesis, *OLS* [45] and *MNLogit* [46] were used. Statsmodels depends, among others, on the well-known *numpy* module [47].

In addition to *statsmodels*, another Python module called *pandas* [48] was used as a supplementary module for the regression analysis. It offers the creation of so-called *DataFrames* [49] that can hold the data used for the regression analysis with *statsmodels*. It also allows the creation of dummy-variables. Dummy-variables are needed because nominal features of a Tweet are used as independent variables (e.g. "day of the week").

3.3.2 Features Used

The following features of a Tweet were used for the regression analysis:

Time of the day

The day was divided into five distinct parts: morning (6 am to 11 am), midday (11 am to 3 pm), afternoon (3 pm to 7 pm), evening (7 pm to 11 pm) and night (11 pm to 6 am). This was used as a categorical variable[1] for the regression.

Day of the week

The day of the week (Sunday, Monday, . . .) was also used as a categorical variable[1].

Relative number of favorites

The number of favorites of a Tweet in relation to the number of followers a user has:

[1] A categorical or nominal variable can take on different values or categories that are unordered. Other examples would be gender and hair color of a person. [50]

$\frac{favorites\_of\_Tweet}{followers\_of\_author}$

Number of hashtags
The number of hashtags used in the text of a Tweet.

Number of mentions
The number of other users mentioned in a Tweet.

Number of media
The number of media a Tweet contains (photos, videos, . . .).

Time since last Tweet
The time since the previous Tweet of that user in seconds.

Happiness score of last Tweet
The happiness score of the previous Tweet of that user.

A larger number of features were used because it was suspected that most features would be a bad predictor of the sentiment by itself, but that in combination they could prove to be a better predictor.

Every run of predictions was done with four different subsets of these features:

- time of the day, day of the week
- time of the day, day of the week, relative number of favorites, number of hashtags, number of mentions, number of media
- time of the day, day of the week, time since last Tweet, happiness score of last Tweet
- all features listed above

The different results of using these subsets were then compared and evaluated.

3.3.3 Implementation

In this section, the basic implementation of the regression analysis and sentiment prediction will be explained. For a full overview of all files, see appendix B. For the full source code, please consult the attached CD or use the download link under appendix B. For a guide on how to install and use the code, please see appendix C.

prediction/Prediction.py
This file offers a general class for performing a regression analysis and using the results for prediction. It manages the testing and training data sets, allows categorical (nominal) variables with dummy variables, can perform the regression analysis and make predictions for testing data based on the regression results for the training data. It supports the regression

methods needed for this thesis, but could also be used for different purposes than analyzing the happiness of Tweets.

prediction/PredictionOLS.py
This file performs the OLS regression for the four different feature sets described above for the top 100 users by follower count in the database. The top 100 users were chosen because they can be expected to use Twitter regularly and not use too much slang in their Tweets. For each of these users, their available Tweets are split into a training set and a testing set. Their first two-thirds of their Tweets (ordered by ascending post date) are chosen as the training set to train the predictor. Based on these Tweets and their calculated sentiment scores, the regression analysis is performed. The rest of the Tweets are used as the testing set. For each of the Tweets in the testing set, a sentiment score based on the regression analysis is predicted and compared with the calculated sentiment score.

prediction/PredictionMNLogit.py
This file is very similar to the previous file, only that instead of OLS, MNLogit is used. This requires two changes: 1. The sentiment scores have to be converted to happiness categories (as described in section 3.3.1). 2. The result have to be handled differently, so that they can be presented as described in the following section.

3.3.4 Quality Evaluation of the Predictions

For the evaluation of the two different regression methods and the four different feature sets, a measurement of prediction quality has to be established. This is necessary to *a*) make a general statement about those predictions; *b*) compare the different feature sets with each other.

It is hard, if not impossible, to properly compare the results of the two different regression methods. Therefore, in this thesis, the two methods will be evaluated separately.

OLS regression

A prediction based on the result of a OLS regression with the happiness score as a dependent variable will be one distinct happiness score which can be compared to the calculated happiness score for that Tweet. Several values will be used to compare the different feature sets within this regression:

- Absolute difference between predicted happiness score and calculated happiness score (average and standard deviation).
- Number of predictions that were too low (more than 0.5 lower than the calculated happiness score).

- Number of predictions that were too high (more than 0.5 higher than the calculated happiness score).
- Number of predictions within +/- 0.1 of the calculated happiness score.
- Number of predictions within +/- 0.5 of the calculated happiness score.

The last measurement will be the main factor to evaluate the results. For a total range from 1.0 to 9.0, a prediction with a maximum difference of 0.5 can be considered a good prediction.

MNLogit regression

A prediction based on the result of a MNLogit regression will be a dictionary in the following form:

```
{"positive": x, "neutral": y, "negative": z}
with 0 <= x, y, z <= 1 and x + y + z = 1
```

x is the predicted probability that the Tweet of interest is positive, y is the predicted probability that the Tweet of interest is neutral and z is the predicted probability that the Tweet of interest is negative. With this as a result of each prediction, a table like table 3.1 can be compiled.

calculated ↓ predicted →	positive	neutral	negative
positive	pos_{pos}	pos_{neu}	pos_{neg}
neutral	neu_{pos}	neu_{neu}	neu_{neg}
negative	neg_{pos}	neg_{neu}	neg_{neg}

Table 3.1: Explaination for the prediction with MNLogit regression.

Each cell within the table represents an average prediction probability. For example, pos_{pos} is the average probability for a Tweet with a calculated positive sentiment to be predicted as positive, neg_{neu} is the average probability for a Tweet with a calculated negative sentiment to be predicted as neutral, and so on.

Also, during the MNLogit regression, the regressions failed or did not converge for some users with some feature sets. These values can also be compared to evaluate the usability of the features.

Chapter 4

Results

In this chapter, the results of the methods described in the previous chapter are presented and discussed. First, an overview and description of the data collected from the Twitter API is given. Then, the results from the sentiment analysis are presented and evaluated. And last, the results of the regression analysis and sentiment prediction are showed and discussed.

4.1 Data Collection

Here are some basic numbers about the resulting Tweet database:

- Number of Tweets: 91,823,701
- Number of users: 14,656,020
- Number of hashtags: 39,374,462
- Number of mentions: 85,057,345
- Number of URLs: 28,730,770
- Number of media: 22,119,804
- Tweets collected per hour (both APIs): 69,934
- Earliest Tweet collected (local time): 13 March 2007 at 01:43:39
- Latest Tweet collected (local time): 13 April 2015 at 21:49:41
- Averages follower count of users in the database: 1525

With about 500 million daily Tweets [4] and English as the most popular language on Twitter [51], this database represents only a small fraction of the activity on Twitter. The restrictions Twitter imposes on their APIs prevent a more complete dataset. Twitter offers the *firehose* streaming API [52] that returns all public Tweets, but the access requires special permission and the hardware requirements (mainly storage space, computing power and bandwidth) for working with such a large dataset would have been much higher.

In figure 4.1 it can be seen that more Tweets in the dataset were posted in recent time. This can be explained by the API limit that Twitter has on accessing the timeline of users (only the most recent 3200 Tweets [36]). In 2015, there is a much higher number of Tweets due to the use of the streaming API between February and April 2015.

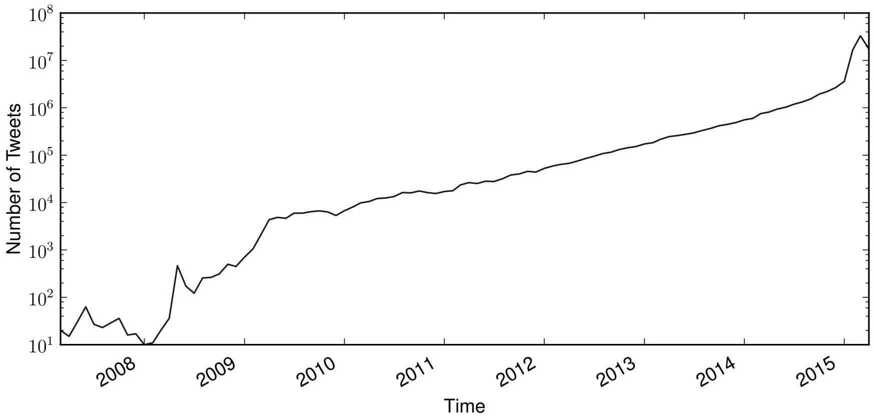


Figure 4.1: Number of Tweets over time (logarithmic scale, counted per month).

In figure 4.2, the number of Tweets by weekday is displayed. There is a slight downfall of the number of Tweets posted on the weekend, this might be due to concentration on family or other social activities.

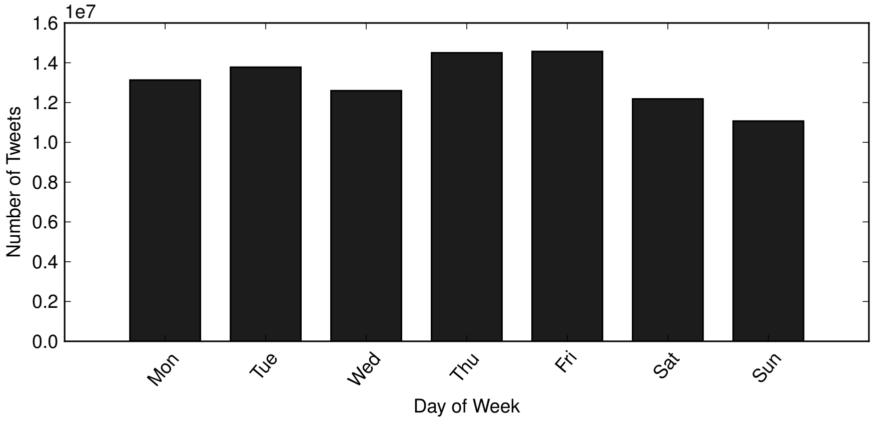


Figure 4.2: Number of Tweets by day of week.

In figure 4.3, the number of Tweets by time of the day is shown. The fewest Tweets are posted around 5 am, though the number of Tweets rises quickly during the morning hours. It rises more slowly during the afternoon hours and reaches its peak around 9 pm. After 10 pm, the number of

Tweets quickly falls as the night progresses.

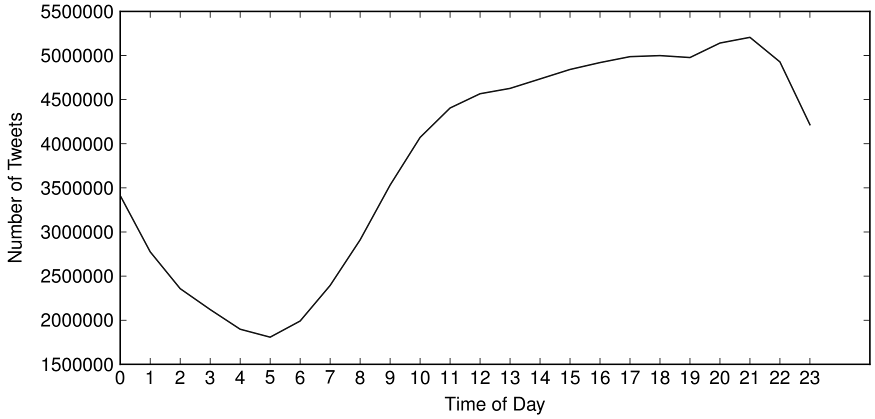


Figure 4.3: Number of Tweets by time of day.

4.2 Sentiment Analysis

Already during the collection of data, the happiness score for each Tweet was calculated and stored into the database. Besides using these scores for the regression analysis and happiness prediction, there are some interesting properties to these scores that will be explained in this section.

4.2.1 Description of Sentiment Results

Of the total of 91,823,701 Tweets stored in the database, 35,178,029 (38.3%) were scored with a happiness score. There are three main reasons for Tweets not to be scored:

- The Tweet is too short (not enough cues for the sentiment analysis).
- The language in the Tweet is too colloquial (words in the Tweet could not be matched with the word list).
- The content of the Tweet is too neutral (words are filtered out as stop words).

In figure 4.4, the average happiness score over time (calculated monthly) of the available data is shown. The large fluctuations in the first years can be explained by the low density of Tweets available during that time frame. Between 2010 and 2015, the average happiness score remains relatively constant around 6.2, though it crashes slightly in 2015 for unknown reasons.

In figure 4.5, the distribution of happiness scores can be seen. There are two high points, one between 5.4 and 6.0 and a higher one around 6.6. This shows that there seems to be a large number

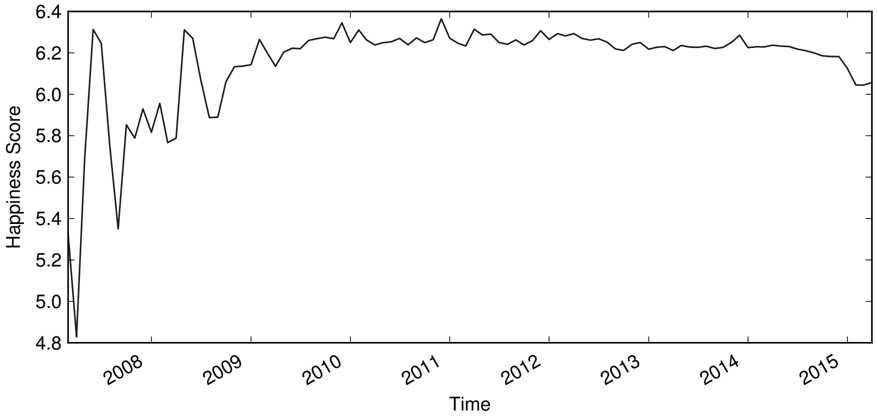


Figure 4.4: Average happiness score over time (calculated per month).

of relatively positive Tweets, but also a large number of more distributed, rather neutral Tweets. There are not many really positive or really negative Tweets, meaning that people do not tend to express extreme sentiments.

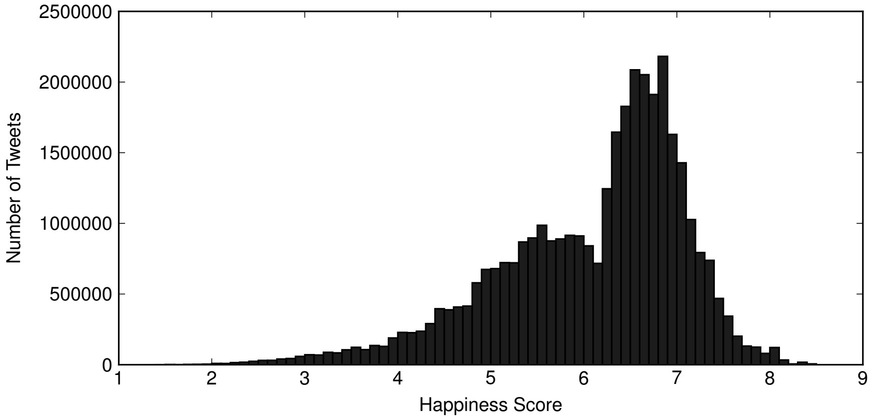


Figure 4.5: Happiness score distribution (steps of 0.1).

In figure 4.6, the average happiness score by the day of the week is shown. The fluctuations that can be seen are very small, but there seems to be a slight correlation in that the happiness seems to be lower on weekdays than on weekends. People seem to be the happiest on Saturdays, the high point of the weekend, and they seem to be the least happy on Thursdays where they are getting tired of their workweek.

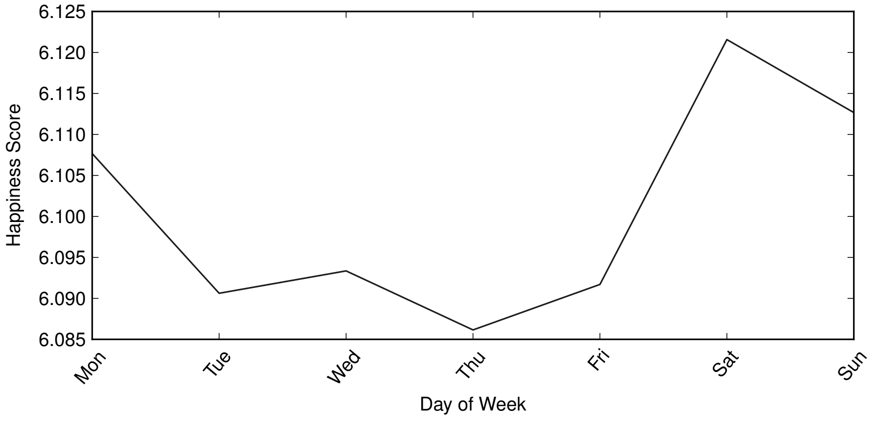


Figure 4.6: Average happiness score by day of week.

In figure 4.7, the average happiness score by the hour of the day is shown. The high point is around 10 am, declining slightly during the afternoon and evening and having its low point between 2 am and 3 am. After that, during the morning hours, the happiness rises heavily until 10 am. Note that these fluctuations are between average happiness scores of 6.04 and 6.15, so the fluctuations are not extreme. But this still indicates a slight correlation between the time of the day and the happiness score.

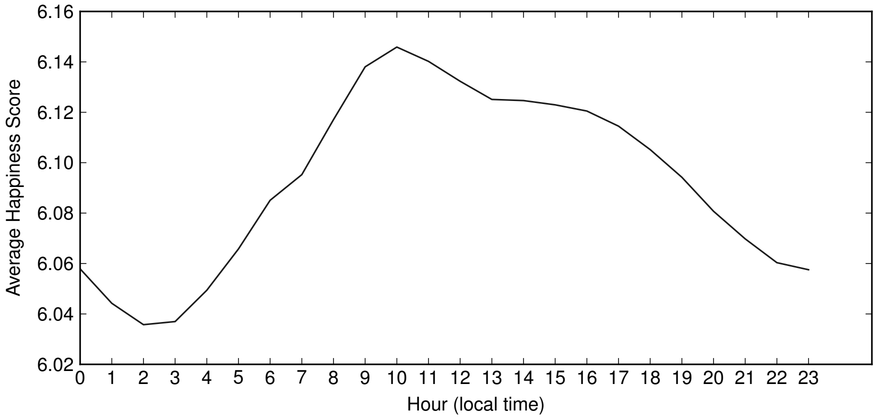


Figure 4.7: Average happiness score by time of day (hourly).

4.2.2 Comparison with similar works

In the foundational paper of Dodds et al. [40], they also analyzed some of the things that were discussed in the last section. For example, their results about the happiness score by the day of the week were similar to the results in this thesis. Their results about the happiness by time of the day were slightly different in that the low point of the curve in their result would be around 11 pm and the high point in the early morning hours, while those are shifted in the data obtained for this thesis. These differences could be explained by two main things: 1. They were using a different data set (much bigger, different time frame). 2. They did not include the modifications done in this thesis. Both of these could have influenced the results in this thesis.

Overall, the resulting sentiment data seems plausible and can mostly be explained by a real world phenomenon.

4.2.3 Effect on Prediction

The quality of the calculated sentiment scores have a huge impact on the expected quality of the sentiment predictions that were subsequently done. The less accurately the scores represent the actual sentiment of a Tweet, the less likely it is that the resulting predictions will be accurate. Therefore, the additional efforts during the sentiment analysis, for example the preprocessing, the inclusion of negatives and the consideration of emoticons, were very important. There is still a lot of room for improvement though. More sophisticated linguistic techniques could be used to analyze a sentence more deeply. This would have required much more knowledge, computing power and time than was available for the creating of this thesis.

4.3 Prediction

Both the OLS and the MNLogit regressions were executed with the four different feature sets listed in the previous chapter. Here is the list of those feature sets again. They will be referred to with their respective number in the following result tables.

1. time of the day, day of the week
2. time of the day, day of the week, relative number of favorites, number of hashtags, number of mentions, number of media
3. time of the day, day of the week, time since last Tweet, happiness score of last Tweet
4. time of the day, day of the week, relative number of favorites, number of hashtags, number of mentions, number of media, time since last Tweet, happiness score of last Tweet

Detailed explanations for the individual features can be found in section 3.3.2.

4.3.1 Results

OLS regression

Table 4.1 shows the results for the predictions based on the OLS regression for all four feature sets used.

measure ↓ feature set →	1	2	3	4
abs. difference average	1.53	1.52	0.90	0.93
(standard deviation)	(1.489)	(1.41)	(0.77)	(2.76)
total predictions	40063	40063	40063	40063
prediction too low	15366	15512	13985	13663
(percentage)	(38.4%)	(38.7%)	(34.9%)	(34.1%)
prediction too high	12912	14034	10836	11277
(percentage)	(32.2%)	(35.0%)	(27.0%)	(28.1%)
within +/- 0.1 range	2565	2275	3360	3254
(percentage)	(6.4%)	(5.7%)	(8.4%)	(8.1%)
within +/- 0.5 range	11785	10517	15242	15123
(percentage)	(29.4%)	(26.3%)	(38.0%)	(37.7%)

Table 4.1: Results for the predictions with OLS regression.

Feature set 3 performed best out of the four sets with 38.0% of the predictions within a +/- 0.5 range of the calculated happiness score and an average absolute difference of 0.90. Feature set 4 performed slightly worse even though it contained more predictor variables. This indicates that some of the features added in feature set 4 are bad predictors. This is confirmed by the results for feature sets 1 and 2 where feature set 2 performed slightly worse with the same added features. A look at the "prediction too high" row also shows that those features (which are relative number of favorites, number of hashtags, number of mentions and number of media) tend to overestimate the sentiment of Tweets.

The features that were added from feature set 1 to feature set 3 (namely, time since last Tweet and happiness score of last Tweet) improved the results a lot (8.6% more predictions fell into the +/- 0.5 range). This shows that considering Tweets of the recent past (like the previous Tweet) has a positive impact on the quality of the predictions.

MNLogit regression

In table 4.2, table 4.3, table 4.4 and table 4.5 the results for the MNLogit regression for the four different feature sets respectively are shown.

In this case, the feature sets had little effect on the results. Feature set 3 performed slightly better than the other feature sets, which is in unison with the results for the predictions based on the

OLS regression. Another similarity can be seen in that the added features for set 2 and 4 slightly increased the probabilities for a positive prediction.

In these results it can be seen that the average predictions have a correct tendency, but the values are still very similar and an accurate prediction for a single Tweet will be hard to make.

calculated ↓ predicted →	positive	neutral	negative
positive	0.403	0.341	0.256
neutral	0.355	0.357	0.288
negative	0.281	0.307	0.412

Table 4.2: Results for the prediction with MNLogit regression (feature set 1). Number of total predictions: 39,837.

calculated ↓ predicted →	positive	neutral	negative
positive	0.421	0.338	0.241
neutral	0.369	0.352	0.279
negative	0.30	0.316	0.384

Table 4.3: Results for the prediction with MNLogit regression (feature set 2). Number of total predictions: 36,563.

calculated ↓ predicted →	positive	neutral	negative
positive	0.414	0.341	0.245
neutral	0.359	0.36	0.281
negative	0.281	0.307	0.412

Table 4.4: Results for the prediction with MNLogit regression (feature set 3). Number of total predictions: 39,981.

calculated ↓ predicted →	positive	neutral	negative
positive	0.422	0.341	0.237
neutral	0.367	0.349	0.284
negative	0.285	0.31	0.405

Table 4.5: Results for the prediction with MNLogit regression (feature set 4). Number of total predictions: 36,262.

In table 4.6, some basic information about the MNLogit regressions for the different feature sets is shown. It can be seen that feature sets 2 and 4 have much higher rates of failed regressions or regressions that did not converge. This again confirms that the features added in those sets are not very suitable for the sentiment prediction.

regression ↓ feature set →	1	2	3	4
total number	100	100	100	100
skipped (not enough data)	1	1	1	1
did not converge	9	55	11	52
failed	6	13	4	16

Table 4.6: Basic information about the MNLogit regressions for all four feature sets.

4.3.2 Reflection

The results of the predictions show that there is at least some weak correlation between the calculated happiness score (or the consequent happiness category) and the features chosen as predictors for the happiness. This correlation does not seem strong enough to make definite predictions about the happiness of a single Tweet, but on average, the predictions have the right tendency and do a fair job in predicting the happiness.

Chapter 5

Summary and Conclusion

This thesis had two goals:

1. Analysis of the happiness in Tweets (sentiment analysis).
2. Prediction of the happiness of future Tweets based on a user's past Tweets.

As a prerequisite for achieving these goals, some data had to be collected first. With the use of Twitter's public APIs, over 90 million Tweets could be collected within a timespan of about two months. These include real time Tweets posted in that time frame as well as past Tweets of the most followed Twitter users. A variety of features about the Tweets and about their authors were saved within a database and used for further analysis.

During the collection of the data, every Tweet was already analyzed for its sentiment. For the sentiment analysis, an established method of using word lists of happiness scores was implemented. Tweets that do not express a certain level of sentiment at all were skipped during the scoring process. In total, about 35 million Tweets were assigned a sentiment score and thus could be used for further analysis.

For the sentiment prediction, a regression analysis was performed. The first two-third of a user's Tweets (ordered by posting time) were used as a training set for the regression, while the remaining third was used as a testing set for the prediction. Four different feature sets and two different regression methods were used for this process and compared. Some features (like day of the week or time of the day) proved to be a better predictor than some other features, which when used even worsened the prediction results. While the results indicate a correlation that might not be strong enough to accurately predict the happiness of a single Tweet, on average the predictions had the right tendency and did a fairly good job in predicting the happiness of Tweets.

Outlook

With the work of this thesis as a foundation, a variety of ideas for future works come to mind. The work could be extended to other public social networks like Reddit or Tumblr. The sentiment analysis could be further improved by using more sophisticated linguistic methods or machine learning algorithms. Different aspects could be considered, like the interactions between users within an online social network.

Even though OSNs already have a very large user base, there is still a lot of room for improvement and innovation. For example, decentralized social networking platforms like GEMSTONE [53] could become more interesting as users want to protect their privacy and take their data into their own hands. The research field of online social networks will certainly not become less interesting in the future.

List of Figures

List of Acronyms

API application programming interface. 2, 5, 7, 8, 10, 11, 14, 19, 20, 28, 33, 34, 36, 37

IRC Internet Relay Chat. 4

JSON JavaScript Object Notation. 8

MNLogit multinominal logistic. 15, 17, 18, 24–27, 34, 37

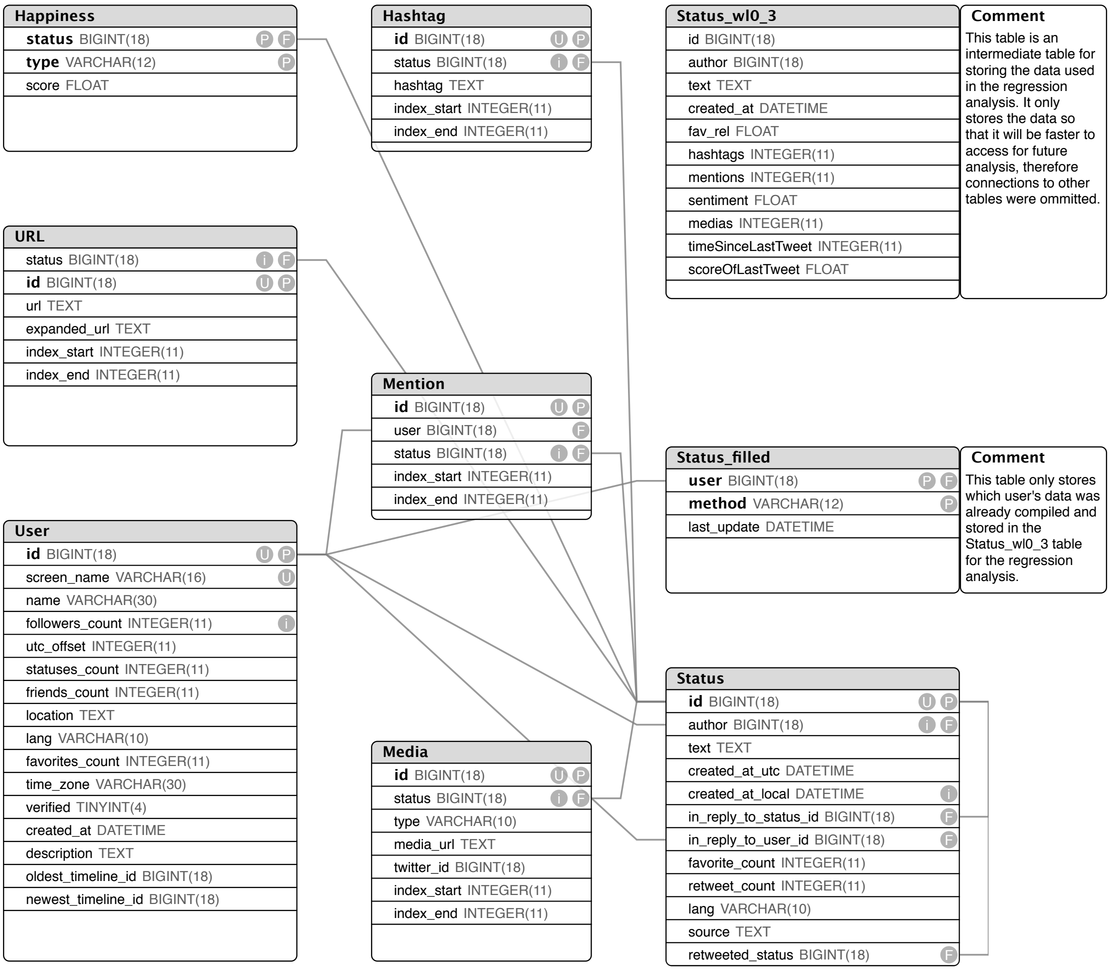
NLP natural language processing. 5, 14

OLS ordinary least squares. 14, 17, 24–26, 34, 37

OSN online social network. 1, 4–6, 29

Appendix A

Appendix: Database Structure



This table is an intermediate table for storing the data used in the regression analysis. It only stores the data so that it will be faster to access for future analysis, therefore connections to other tables were ommitted.

This table only stores which user's data was already compiled and stored in the Status_wl0_3 table for the regression analysis.

Appendix B

Appendix: Overview and Description of Files

All Python files contain additional information on their usages in a comment within the files. These files can be found on the attached CD or as a download from `http://dl.exo.pm/ZAI-BSC-2015-11-peters.zip`. The password for this zip archive can be inquired from the author of this thesis, Stefan Peters (speters@exo.pm).

README
A readme file containing basic information and references of used code snippets.

ZAI-BSC-2015-11-peters.pdf
The PDF version of this thesis.

__init__.py (in every folder and subfolder)
Needed for Python so that it can treat the folder as a package.

database/DatabaseStructure.sql
The basic database structure used in this thesis.

database/DatabaseCreds.py
Contains the database credentials and offers a method *getConnection* which returns a tuple of a database pointer and a cursor pointer.

database/SaveStatus.py
Offers a class *SaveStatus* to save Tweets obtained from the Twitter API into the database.

database/GetStatus.py
Offers a class *GetStatus* to get Tweets for a certain user from the database, containing all necessary information for the regression analysis.

database/TwitterTokens.py
Contains the Twitter API tokens and a guide on how to obtain them. The presetting of the tokens are the author's private tokens and should be replaced if possible.

database/TweetStreaming.py

An executable file that connects to the Twitter streaming endpoint, collects English Tweets and saves them into the database.

database/TweetDumping.py

Executable file. Uses Twitters timeline API methods to obtain the recent 3200 Tweets for the top 5000 Twitter users by follower count currently in the database. Use this file after having already obtained a reasonable number of Tweets through the streaming API.

preprocessing/Preproc.py

Offers a class *Preproc* with a method *process* that performs the preprocessing steps described in the thesis and returns a list of tokens that can be used for the sentiment analysis.

scoring/SentimentScorer.py

Offers a class *SentimentScorer* with a method *score* that calculates a happiness score for a Tweet.

scoring/SentimentWordlist.txt

Contains the labMT 1.0 word list [41]. Some sentiment scores for emoticons were added afterwards for this thesis.

prediction/Prediction.py

Offers a class *Prediction* for managing training and testing data, performing the regression (OLS or MNLogit) and predicting the happiness for a Tweet from the testing data.

prediction/PredictionOLS.py

Executable file that performs an OLS regression to predict happiness scores for Tweets of the top 100 users by follower count in the database. It uses all four feature sets as described in the thesis, and returns some results in a similar form as presented in this thesis.

prediction/PredictionMNLogit.py

Executable file that performs a MNLogit regression to predict happiness categories for Tweets of the top 100 users by follower count in the database. It also uses all four feature sets as described in the thesis, and returns some results in a similar form as presented in this thesis.

Appendix C

Appendix: Guide to Installation and Usage of Files

The following is a guide on how to install all dependencies and prerequisites for running the code provided by this thesis and building a Tweet database.

This guide was tested on a clean Ubuntu 14.04 LTS installation with the latest updates installed. It should work similarly on other Linux distributions, though detailed explanations for every possible system or configuration cannot be provided in this thesis.

1. Use the download link provided in appendix B or the attached CD and extract or copy the provided files to your computer. The folders have to be readable and writable.
2. Open a command line interface (all the following steps are done from a command line).
3. Switch to the *database* folder of the provided files.
4. Install Python for your system:

```
sudo apt-get install python2.7 python-dev
sudo apt-get install python-setuptools build-essential
```

5. Use easy_install to install pip (see [54] for the issue with the pip version that could be obtain via apt-get):

```
sudo easy_install -U pip
```

6. This step is only needed for Ubuntu 14.04 specifically, see [55]:

```
sudo apt-get install libffi-dev libssl-dev
sudo -H pip install requests[security] --upgrade
```

7. Install dependencies for the subsequent Python modules:

```
sudo apt-get install libfreetype6-dev libxft-dev libpng-dev
sudo apt-get install gfortran libopenblas-dev liblapack-dev
sudo apt-get install python-numpy
```

8. Use pip to install all necessary Python modules:

```
sudo -H pip install pymysql tweepy
sudo -H pip install pandas
sudo -H pip install matplotlib
sudo -H pip install statsmodels
sudo -H pip install pattern
```

9. The following step is not necessary if a MySQL server that can be used for this purpose is already installed or available.

   Install the MariaDB server (an alternate MySQL server):

```
sudo apt-get install mariadb-server
```

   During the installation, you have to set the password for the root database user. We need this in the next step.

10. Open a MySQL promt. You can another existing user that has the rights to create a database and a user to access that database.

```
mysql -uroot -p (type password here)
```

11. In the MySQL promt, create a database and a user for accessing that database (from [56]):

```
CREATE DATABASE `ba-twitter`
      CHARACTER SET utf8 COLLATE utf8_general_ci;
GRANT ALL ON `ba-twitter`.* TO `ba-user`@localhost
      IDENTIFIED BY 'ba-password';
FLUSH PRIVILEGES;
```

12. Still in the MySQL promt, import the database structure from the provided file:

```
USE `ba-twitter`;
SOURCE DatabaseStructure.sql;
QUIT;
```

13. Open DatabaseCreds.py with any editor and change the database credentials (if all the steps above were followed and the user credentials not replaced, no change is needed here).

14. Open TwitterTokens.py. In there is a guide on how to obtain the necessary API tokens for accessing the Twitter API. Use that guide and replace the tokens with your newly obtained tokens.

15. Now everything is prepared. For running the script that obtains Tweets from the streaming API, run:

    ```
    python TweetStreaming.py
    ```

16. After some time of running the streaming script, another terminal can be opened and the script for the search API can be started as well:

    ```
    python TweetDumping.py
    ```

Running these two scripts, the database will be filled with Tweets and their associated sentiment scores that can be used for the further analysis explained in the thesis.

For running the OLS regression and happiness score prediction on the top 100 users by follower count in the database, switch to the *prediction* folder of the provided files and type into a terminal:

```
python PredictionOLS.py > PredictionOLS-results.txt
```

This might take some time. After the script is finished, the results can be found in the file *PredictionOLS-results.txt*.

For running the MNLogit regression and happiness category prediction on the top 100 users by follower count in the database, switch to the *prediction* folder of the provided files and type into a terminal:

```
python PredictionMNLogit.py > PredictionMNLogit-results.txt
```

This might take some time. After the script is finished, the results can be found in the file *PredictionMNLogit-results.txt*. Note that the file will probably contain additional output from the regression algorithm that could not be omitted.

Bibliography

[1] Nicholas Carlson, "At last – the full story of how Facebook was founded," 2010, http://www.businessinsider.com/how-facebook-was-founded-2010-3?IR=T [accessed: 2015-05-13].

[2] GlobalWebIndex, "GWI Social Q4 2014," 2015, http://www.globalwebindex.net/blog/daily-time-spent-on-social-networks-rises-to-1-72-hours [accessed: 2015-05-13].

[3] Facebook, Inc., "Facebook Q4 2014 Results," 2015, http://investor.fb.com/common/download/download.cfm?companyid=AMDA-NJ5DZ&fileid=805520&filekey=2D74EDCA-E02A-420B-A262-BC096264BB93&filename=FB_Q414EarningsSlides20150128.pdf [accessed: 2015-04-22].

[4] Twitter, Inc., "Company," 2015, https://about.twitter.com/company [accessed: 2015-05-09].

[5] Twitter, Inc., "Character Counting," 2015, https://dev.twitter.com/overview/api/counting-characters [accessed: 2015-05-13].

[6] Barry M. Leiner, Vinton G. Cerf, David D. Clark, Robert E. Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Larry G. Roberts, Stephen Wolff, "Brief History of the Internet," 2012, http://www.internetsociety.org/sites/default/files/Brief_History_of_the_Internet.pdf [accessed: 2015-05-13].

[7] Carl Johnson, "The Internet: Can't Live Without It," 2011, http://www.forbes.com/sites/carljohnson/2011/11/02/the-internet-cant-live-without-it/ [accessed: 2015-05-13].

[8] Ray Tomlinson, "The First Network Email," http://openmap.bbn.com/~tomlinso/ray/firstemailframe.html [accessed: 2015-05-19].

[9] W. Christensen and R. Suess, "Hobbyist Computerized Bulletin Board," *Byte Magazine*, vol. 03, no. 11, pp. 150–157, Nov. 1978. [Online]. Available: https://ia801609.us.archive.org/15/items/byte-magazine-1978-11-rescan/1978_11_BYTE_03-11_The_Sky_is_the_Limit.pdf

[10] Tom Truscott, "Invitation to a General Access UNIX* Network," http://www.newsdemon.com/first-official-announcement-usenet.php [accessed: 2015-05-19].

[11] Jarkko Oikarinen, "Founding IRC," http://www.mirc.com/jarkko.html [accessed: 2015-05-19].

[12] d. boyd and N. Ellison, "Social Network Sites: Definition, History, and Scholarship." *Journal of Computer-Mediated Communication*, vol. 13, no. 1, Oct. 2007, article 11. [Online]. Available: http://www.danah.org/papers/JCMCIntro.pdf

[13] Facebook, Inc., "About Facebook," https://www.facebook.com/facebook/info?tab=page_info [accessed: 2015-05-19].

[14] Facebook, Inc., "Welcome to Facebook, everyone." 2006, https://www.facebook.com/notes/facebook/welcome-to-facebook-everyone/2210227130 [accessed: 2015-05-19].

[15] Maeve Duggan, Nicole B. Ellison, Cliff Lampe, Amanda Lenhart, Mary Madden, "Social Media Update 2014," 2015, http://www.pewinternet.org/2015/01/09/social-media-update-2014/ [accessed: 2015-05-19].

[16] Dictionary.com, "Definition microblogging," http://dictionary.reference.com/browse/microblogging [accessed: 2015-05-19].

[17] Jason Kottke, "Tumblelogs," 2005, http://kottke.org/05/10/tumblelogs [accessed: 2015-05-19].

[18] Twitter, Inc., "Company Milestones," https://about.twitter.com/milestones [accessed: 2015-05-19].

[19] Twitter, Inc., "About public and protected Tweets," https://support.twitter.com/articles/14016-about-public-and-protected-tweets [accessed: 2015-05-19].

[20] B. Liu, "Sentiment Analysis and Opinion Mining," in *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, May 2012, p. 7.

[21] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification using Distant Supervision," Tech. Rep., 2009. [Online]. Available: https://sites.google.com/site/twittersentimenthelp/home

[22] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi, "Measuring User Influence in Twitter: The Million Follower Fallacy," *ICWSM 2010*, 2010.

[23] S. Kim, J. Bak, and A. H. Oh, "Do You Feel What I Feel? Social Aspects of Emotions in Twitter Conversations," in *Sixth International AAAI Conference on Weblogs and Social Media*, May 2012. [Online]. Available: https://www.aaai.org/ocs/index.php/ICWSM/ICWSM12/paper/view/4630

[24] A. Hogenboom, D. Bal, F. Frasincar, M. Bal, F. de Jong, and U. Kaymak, "Exploiting Emoticons in Sentiment Analysis," in *Proceedings of the 28th Annual ACM Symposium on*

Applied Computing, ser. SAC '13. New York, NY, USA: ACM, 2013, pp. 703–710. [Online]. Available: http://doi.acm.org/10.1145/2480362.2480498

[25] J. Bollen, H. Mao, and X.-J. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science*, vol. 2, no. 1, pp. 1–8, Mar. 2011, arXiv: 1010.3003. [Online]. Available: http://arxiv.org/abs/1010.3003

[26] S. You, J. DesArmo, and S. Joo, "Measuring happiness of US cities by mining user-generated text in Flickr.com: A pilot analysis," *Proceedings of the American Society for Information Science and Technology*, vol. 50, no. 1, pp. 1–4, Jan. 2013. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/meet.14505001167/abstract

[27] Oracle Corporation, "MySQL :: The world's most popular open source database," https://www.mysql.com [accessed: 2015-05-19].

[28] CodeEval, "Most Popular Coding Languages of 2015," 2015, http://blog.codeeval.com/codeevalblog/2015 [accessed: 2015-05-19].

[29] PyPA, "pip," https://pip.pypa.io/en/stable/ [accessed: 2015-05-19].

[30] PyPA, "Installing and Using Setuptools," https://bitbucket.org/pypa/setuptools [accessed: 2015-05-19].

[31] CLiPS Research Center, "Pattern," http://www.clips.ua.ac.be/pattern [accessed: 2015-05-19].

[32] PyMySQL, "PyMySQL: Pure-Python MySQL Client," https://github.com/PyMySQL/PyMySQL [accessed: 2015-05-19].

[33] the statsmodels development team, "Statsmodels," http://statsmodels.sourceforge.net [accessed: 2015-05-19].

[34] Twitter, Inc., "Obtaining access tokens," https://dev.twitter.com/oauth/overview [accessed: 2015-05-19].

[35] Twitter, Inc., "GET statuses/sample," https://dev.twitter.com/streaming/reference/get/statuses/sample [accessed: 2015-05-19].

[36] Twitter, Inc., "GET statuses/user_timeline," https://dev.twitter.com/rest/reference/get/statuses/user_timeline [accessed: 2015-05-19].

[37] Tweepy, "Tweepy," http://www.tweepy.org [accessed: 2015-05-19].

[38] Twitter, Inc., "Twitter Libraries," https://dev.twitter.com/overview/api/twitter-libraries [accessed: 2015-05-19].

[39] Twitter, Inc., "API Rate Limits," https://dev.twitter.com/rest/public/rate-limiting [accessed: 2015-05-19].

[40] P. S. Dodds, K. D. Harris, I. M. Kloumann, C. A. Bliss, and C. M. Danforth, "Temporal patterns of happiness and information in a global social network: Hedonometrics and Twitter," *PLoS ONE*, vol. 6, no. 12, p. e26752, Dec. 2011, arXiv: 1101.5120. [Online]. Available: http://arxiv.org/abs/1101.5120

[41] Dodds, Peter Sheridan and Harris, Kameron Decker and Kloumann, Isabel M. and Bliss, Catherine A. and Danforth, Christopher M., "labMT 1.0 word list," 2011, http://arxiv.org/src/1101.5120v5/anc/labMT-1.0.txt [accessed: 2015-05-19].

[42] CLiPS Research Center, "pattern.en," http://www.clips.ua.ac.be/pages/pattern-en [accessed: 2015-05-19].

[43] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," in *Proceedings of EMNLP*, 2002, pp. 79–86.

[44] I. Hemalatha, G. P. S. Varma, and A. Govardhan, "Preprocessing the Informal Text for efficient Sentiment Analysis," *International Journal of Emerging Trends & Technology in Computer Science*, vol. 1, no. 2, Aug. 2012. [Online]. Available: http://www.ijettcs.org/Volume1Issue2/IJETTCS-2012-08-14-047.pdf

[45] Josef Perktold, Skipper Seabold, Jonathan Taylor, statsmodels-developers, "statsmodels.regression.linear_model.OLS," http://statsmodels.sourceforge.net/0.5.0/generated/statsmodels.regression.linear_model.OLS.html [accessed: 2015-05-19].

[46] Josef Perktold, Skipper Seabold, Jonathan Taylor, statsmodels-developers, "statsmodels.discrete.discrete_model.MNLogit," http://statsmodels.sourceforge.net/devel/generated/statsmodels.discrete.discrete_model.MNLogit.html [accessed: 2015-05-19].

[47] Numpy developers, "NumPy," http://www.numpy.org [accessed: 2015-05-19].

[48] Lambda Foundry, Inc., PyData Development Team, "Python Data Analysis Library," http://pandas.pydata.org [accessed: 2015-05-19].

[49] the pandas development team, "pandas.DataFrame," http://pandas.pydata.org/pandas-docs/dev/generated/pandas.DataFrame.html [accessed: 2015-05-19].

[50] UCLA Institute for Digital Research and Education, "What is the difference between categorical, ordinal and interval variables?" http://www.ats.ucla.edu/stat/mult_pkg/whatstat/nominal_ordinal_interval.htm [accessed: 2015-05-19].

[51] Semiocast, "Arabic highest growth on Twitter English expression stabilizes below 40%," 2011, http://semiocast.com/en/publications/2011_11_24_Arabic_highest_growth_on_Twitter [accessed: 2015-05-19].

[52] Twitter, Inc., "GET statuses/firehose," https://dev.twitter.com/streaming/reference/get/statuses/firehose [accessed: 2015-05-19].

[53] D. Koll, F. Tegeler, and X. Fu, "GEMSTONE: A Generic Middleware for Social Networks," *ACM/USENIX MobiSys 2010 poster session*, Jun. 2010.

[54] thomas.mc.work on Stack Overflow, "How do I fix 'ImportError: cannot import name IncompleteRead'?" http://stackoverflow.com/questions/27341064/how-do-i-fix-importerror-cannot-import-name-incompleteread [accessed: 2015-05-19].

[55] Nathan M on Stack Overflow, "InsecurePlatformWarning: A true SSLContext object is not available. This prevents urllib3 from configuring SSL appropriately [duplicate]," http://stackoverflow.com/questions/29134512/insecureplatformwarning-a-true-sslcontext-object-is-not-available-this-prevent [accessed: 2015-05-19].

[56] Andrew McCombe, "MySQL Create Database with UTF8 Character Set Syntax," 2013, https://www.euperia.com/development/mysql-create-database-with-utf8-character-set-syntax/1064 [accessed: 2015-05-19].