Analysis and Classification of Twitter messages

Master's Thesis

at

Graz University of Technology

submitted by

Christopher Horn

Know Center
Graz University of Technology
A-8010 Graz

29 April 2010

Advisor: Dipl.-Ing. Dr.techn. Michael Granitzer

Analyse und Kategorisierung von Twitter Nachrichten

Masterarbeit

an der

Technischen Universität Graz

eingereicht von

Christopher Horn

Know Center
Technische Universität Graz
A-8010 Graz

29. April 2010

Diese Arbeit ist in englischer Sprache verfasst.

Betreuer: Dipl.-Ing. Dr.techn. Michael Granitzer

Abstract

Social networks like Twitter are the latest trend in the globalized world. Twitter is used in different scenarios by a broad set of different users. Mining their messages may reveal valuable information.

In this thesis, we propose a way to automatically classify Tweets (and thus the users) using a supervised machine-learning approach. Based on Support Vector Machines, we build an application to set up and train the classifier. Also, a sentiment detection module is implemented.

After constructing the data set, an evaluation is carried out to measure the accuracy of the classifier. The results are very promising: we achieve an accuracy of more than 80%. We also examine the impact of n-gram representation, the amount of training data and the usage of word-stemming and word-conversion. The empirical evaluation shows that word stemming and n-gram representations of the features does not improve the accuracy of the classifier, whereas word-conversion (using regular expression) does.

The output of this thesis are a web application that can be used to classify arbitrary Twitter users and the empirical finding that Support Vector Machines are well suited for classifying Twitter messages. The annotated dataset will be made available.

Kurzfassung

Soziale Netzwerke wie Twitter sind der letzte Trend in der globalisierten Welt. Twitter wird von vielen unterschiedlichen Benutzern in einer Vielzahl von Anwendungsszenarien verwendet. Eine Analyse der Twitter-Nachrichten (sogenannten Tweets) könnte wertvolle Erkenntnisse hervorbringen.

In dieser Arbeit werden Tweets (und dadurch auch deren AutorInnen) mittels eines maschinellen Lernverfahren kategorisiert. Hierfür wird eine auf Support Vector Machines-basierende Applikation erstellt, um den Klassifikator zu trainieren und zu testen. Ein Modul zur Sentiment Detection (Gefühlserkennung) von Tweets wird ebenfalls implementiert.

Die Genauigkeit des erstellten Klassifikators wird anhand eines dafür erstellen Datensets gemessen. Die Ergebnisse sind vielversprechend - es wird eine durchschnittliche Genauigkeit von über 80% erziehlt. Bei der Feature Selection werden die Auswirkungen einer n-gram-Repräsentation, die Anzahl der Trainingsdatensätze, sowie einer Vorverarbeitung der Tweets gemessen. Die Ergebnisse zeigen, dass eine Vorverarbeitung (mittels regulären Ausdrücken) der Nachrichten die Genauigkeit des Klassifikators verbessert, eine n-gram Repräsentation und Word Stemming hingegen nicht.

Das Resultat dieser Arbeit ist eine Web-Applikation, die verwendet werden kann um beliebige Twitter-Benutzer zu klassifizieren. Des Weiteren wird das annotierte Datenset zur Verfügung gestellt. Ein weiteres Resultat dieser Arbeit ist die Erkenntnis, dass Support Vector Machines gut für die Klassifikation von Twitter-Nachrichten geeignet sind.

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am …………………………… ………………………………………………..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…………………………… ………………………………………………..
date (signature)

Danksagung

An dieser Stelle möchte ich mich bei Dr. Michael Granitzer und DI Elisabeth Lex für ihr Engagement und ihre professionellen Ratschläge bedanken.

Besonderer Dank gebührt auch meinen Eltern Brigitte und Gerald und meiner Schwester Katrin für die liebevolle Unterstützung während der Studienzeit.

Das größte Dankeschön gilt meiner Freundin Eva, die mich beim Erstellen dieser Diplomarbeit stets mit hilfreichen Tipps und Verbesserungsvorschlägen sowie mit motivierenden Worten tatkräftig unterstützt hat. Danke!

Graz, im April 2010 Christopher Horn

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Nowadays, social networks such as Twitter are the latest trend in the globalized word. With more than 40 million users [Tec09] using the 140-characters microblogging platform, Twitter became a process with its own dynamic. It is used in different scenarios by a broad set of different users. Each of them has their own behavioral pattern, their own style of writing. Mining these platforms may extract valuable information.

Currently, there is heavy research going on in this area, with promising results. For example, Sitaram and Huberman [AH10] used Twitter to predict box-office revenues for movies and achieved a 97% accuracy.

Also, Twitter was used to monitor the U.S. presidential debate in 2008 [DS]. Tweeters tended to favour Obama over McCain, and Obama really won the election afterwards. This shows that Twitter can also be used to predict political election results.

Jansen et al. [JZSC09] showed that 19% of Tweets mention a certain brand, of which 20% contained a sentiment. Using sentiment detection, market researchers have a valuable tool to monitor how a product is accepted. There is no need to start a time- and cost-intensive survey anymore.

With this in mind, we want to apply state-of-the-art classification tools to extract information from Twitter. We want to answer the question if it is possible to automatically classify Tweets (and thus the users themselves). This may help to separate professionally created news messages from user-generated messages. News messages tend to have an informational character, while user-generated messages usually have a communicational purpose.

Also, the classifier can be used as a spam detector. With the growth of Twitter, spam (unsolicited advertisements) becomes more and more of a problem. The latest trend are ‘sponsored Tweets’. People can sign up at an advertisement platform, allowing it to post advertisement Tweets on their Twitter page. The created classifier can be used to track down such Tweets. In the evaluation section, we will answer the question how many Tweets are actually advertisements.

1.2 Objective of this thesis

In this thesis, we are creating and evaluating a supervised classifier, which will be able to automatically classify Tweets. This classifier should not rely on a rule-base or other external information, but should rather analyze only the Tweets itself using a supervised machine-learning approach.

In order to build the classifier a backend is necessary where the training and test data can be set up and evaluated. The aim is to create a high quality dataset, which will be used to classify the users with high accuracy.

Once the classifier has been built, it needs to be evaluated. The goal is to deliver good results (with an accuracy of more than 80%), while still performing well.

Last but not least, a prototype of a Frontend will be created. It should provide a convenient way to classify arbitrary Twitter users.

1.3 Structure of this thesis

Chapter 2 gives a small introduction to the microblogging platform Twitter. Different aspects will be highlighted, this includes the business model, the political impact and usage statistics. Also, it describes the categories used. Chapter 3 deals with the theoretical foundations of this thesis as well as related work. We will highlight the different approaches to text classification and explain why SVM is the classifier of choice for our problem. In chapter 4, the actual implementation is described, followed by a detailed evaluation in chapter 5. Finally, chapter 6 summarizes the work and gives a brief outlook on future work, which could be done.

1.4 Contribution

The output of this thesis consists of the following items:

1. **Data set**. We create a dataset which consists of of a list of Twitter users (120) and their Tweets (approx. 40 Tweets per User). This set of approximately 4800 Tweets was used as training set for the classifier, and is manually categorized. The database dump and the created SVM model file are offered for download.

2. **Platform**. The main task of this thesis was to create a platform which categorizes Twitter messages. We provide a frontend to conveniently categorize the Tweets of a given user, and a backend to setup and test the classifier. The frontend will be made available publicly afterwards.

3. **Scientific contribution**. We show that supervised machine learning systems such as Support Vector Machines can be used to categorize microblogging messages. This is not self-evident, since those messages heavily differ from 'normal' documents in regard to the length and used words. We investigate several approaches of feature selection (word conversion and word stemming), and determine the optimal cost-parameter setting for the SVMs.

Chapter 2

Twitter

Twitter is a popular social network site which only asks one question: *'What are you doing?'*. The answer is limited to 140 characters. Figure 2.1 shows a screenshot of the current Twitter User Interface. Status updates can be sent via a web browser, SMS, e-mail or third party applications and are displayed on the users' profile.

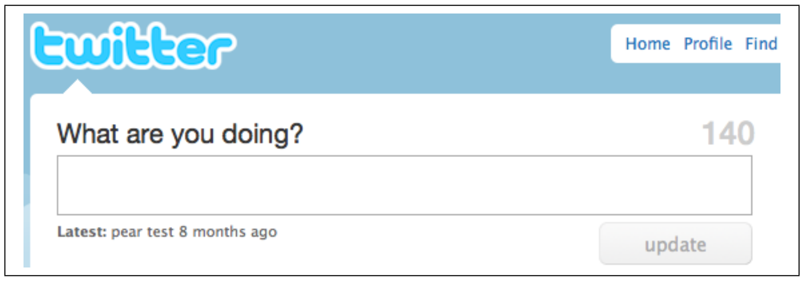
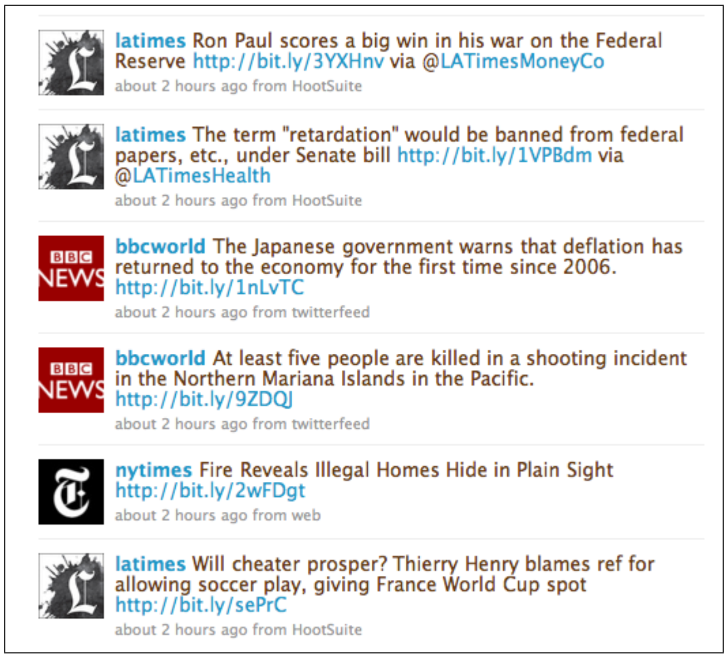


Figure 2.1: Screenshot of Twitter's user interface

2.1 Followers

Twitter implemented a concept of so-called followers. If a certain user updates his/her status, all followers are informed of the new status. This is achieved by adding the new entry to their personal Twitter overview page (Figure 2.2).



latimes Ron Paul scores a big win in his war on the Federal Reserve http://bit.ly/3YXHnv via @LATimesMoneyCo
about 2 hours ago from HootSuite

latimes The term "retardation" would be banned from federal papers, etc., under Senate bill http://bit.ly/1VPBdm via @LATimesHealth
about 2 hours ago from HootSuite

bbcworld The Japanese government warns that deflation has returned to the economy for the first time since 2006. http://bit.ly/1nLvTC
about 2 hours ago from twitterfeed

bbcworld At least five people are killed in a shooting incident in the Northern Mariana Islands in the Pacific. http://bit.ly/9ZDQJ
about 2 hours ago from twitterfeed

nytimes Fire Reveals Illegal Homes Hide in Plain Sight http://bit.ly/2wFDgt
about 2 hours ago from web

latimes Will cheater prosper? Thierry Henry blames ref for allowing soccer play, giving France World Cup spot http://bit.ly/sePrC
about 2 hours ago from HootSuite

Figure 2.2: Screenshot of the personal Twitter overview page

One can follow every other user unless this user has set his/her profile to ‘private’. In this case, an initial request for approval has to be send first.

2.2 Business Model

Like several other popular social network sites, Twitter struggles to find a valid business model which actually generates revenue. Twitter itself confirms this on their web page:

> Twitter has many appealing opportunities for generating revenue but we are holding off on implementation for now because we don’t want to distract ourselves from the more important work at hand which is to create a compelling service and great user experience for millions of people around the world. While our business model is in a research phase, we spend more money than we make.[1]

To finance the service Twitter relies heavily on investors and has thus generated a total funding of 155 Million dollars. According to the Financial Times, the investors currently valuate the site with 1 Billion dollars. [Tim09]

In April 2010, Twitter made a new attempt to find a business model. They introduced so-called ‘Promoted Tweets’[2]. According to their Company Blog, they

[1]http://twitter.com/about
[2]http://blog.twitter.com/2010/04/hello-world.html

want to display advertisement Tweets on the search result page. If the feedback is positive, they also plan to show advertisements in Tweets. This approach seems to be similar to the ‘Sponsored Tweets’ which we are using to classify spam accounts (see Section 5.1.1) .

2.3 Twitter slang

Due to the 140 characters limit, users have developed strategies to put as much information as possible in the messages. This includes the usage of the hash character (#) to tag a message with certain topics. For example, a message may look like this: ‘I’m currently testing a new Twitter feature. #test #twitter #graz’. The message is now tagged with ’test, twitter, graz’, and other people are able to search for that tags using the Twitter site.

Also, the usage of URL shortening services became very popular. Those service allow to shorten a certain URL so that it can be posted on Twitter. For example, `http://www.know-center.tugraz.at/forschung/` becomes `http://tinyurl.com/yh8lqhz` using the shortening-service ‘TinyURL.com’. TinyURL.com was the first well-established shorting services and thus was the default service in Twitter. But in early 2009 Twitter silently switched to bit.ly for unknown reasons. The New York Times reports that

> Bit.ly, which recently raised $2 million in venture financing, tracks real-time statistics on how many times links are clicked and where users are coming from information that could be valuable to companies and brands looking to measure the impact of an e-mail message, link, tweet or mention online. [NYT09]

2.4 Political usage

During the 2009 election in Iran, Twitter played an important role. While the newspapers and blogs in Iran were heavily censored by the government, Twitter users published news from the street in real-time. They either used the hashtags #iran or #iranelection. News cooperations from all over the world displayed the latest Twitter messages. After a while, the government tried to suppress those messages. Users reacted and asked all Twitter users to change their location to ‘Teheran, Iran’, making it impossible for the Irianian Ministry of Intelligence to locate those people.

The U.S. State Department knew about the importance of this communication tool and asked Twitter to delay a scheduled upgrade which took place 3 days after the election on June 15th, 2009. Twitter agreed and carried out the upgrade on 2pm U.S. time, which is 1.30am Teheran time. [TM09]

The leader of the opposition, MirHossein Mousavi, even twittered his arrest: *Dear Iranian People, Mousavi has not left you alone, he has been put under house arrest by Ministry of Intelligence #IranElection* [3]

Also, Twitter was used to raise funds for the victims of the Haitian earthquake[Pep10]. Shortly after the earthquake, the American Red Cross sent following Tweet: *'You can text 'HAITI' to 90999 to donate $10 to Red Cross relief efforts in #haiti.'*[4]. Soon, the famous Haitian singer Wycliff Jean[5] started to support this campaign, and several other celebrities followed him.

This shows that Twitter is playing an important role in the society these days and can be used for more than just ordinary status updates of what you're currently doing.

2.5 Spam

Like every successful communication platform, Twitter is prone to Spam. On their Company Blog[6], the Twitter operators define spam as *'as a variety of different behaviors that range from insidious to annoying'*. This includes aggressive following/unfollowing, links to phishing/malware sites and the classical unsolicited advertisements. Twitter fights hard to avoid spam as good as possible. For example, every user profile has a dedicated 'report for spam' button. According to the operators, they managed to bring down the Spam level to 1-2%. In Section 5.5.9, we will compare this number to our analysis.

2.6 Usage statistics

While Twitter itself refuses to publish their actual user numbers, they have released a geographical distribution of their web traffic. 40% of the traffic is generated inside the U.S. and 60% from international users. From those 60%, 39% are generated in Japan, followed by Spain and UK (11% and 10%)[7]. The actual user number is still unknown. Balachander et al. [KGA08] tried to estimate the user number based on the user ID which is assigned to new users. Taking several changes to the ID generator into account, they estimated a total of 1.4 million users in february 2008.

[3]http://twitter.com/mousavi1388/status/2159159988

[4]http://twitter.com/RedCross/status/7698390067

[5]http://twitter.com/wyclef

[6]http://blog.twitter.com/2010/03/state-of-twitter-spam.html

[7]http://blog.twitter.com/2008/02/twitter-web-traffic-around-world.html

The number of Tweets remains unknown as well. Java et al. [JSFT07] spotted that the number of Tweets doubles each month. Other sources report that Twitter reached 44.5 million users in june 2009 [Tec09]

According to the web-crawling company Alexa, Twitter.com is among the 20 most popular web sites in the world[8].

2.6.1 Famous Twitter users

Twitter is used by many famous people around the world. This includes politicans like the U.S. president Barack Obama[9] with 2.6 Million followers, the U.K. Prime Minister Gordon Brown[10], the President of the European Commission Jose Manuel Baroso[11] and the UN Secretary-General Ban Ki-moon[12]. Also, Twitter is used by sport stars like the basketball star Shaquille O'Neal[13] or the cyclist Lance Armstrong [14], and by famous broadcasters like ABC News' George Stephanopoulos[15] or the ORF's Armin Wolf[16]

But the group which takes the most advantage of Twitter are celebrities. For example, Britney Spears had 3.7 Million followers in November 2009.

In April 2009, Ashton Kutcher challenged CNN 'Breaking News' Account to a race to get 1 Million followers. Both contestants took the challenge seriously and Kutcher finally won by 2.000 followers on 17. April 2009. In an interview Kutcher said that

> he found it astonishing that one person can actually have as big of a voice online as what an entire media company can on Twitter.

2.7 Twitter API

Twitter provides an Application Programming Interface (API) to access its data[17]. Using the HTTP protocol, Twitter provides a method for every feature that can be used on the site. This includes status updates, search operations and accessing a

[8]http://www.alexa.com/siteinfo/twitter.com
[9]http://twitter.com/BARACKOBAMA
[10]http://twitter.com/DowningStreet
[11]http://twitter.com/JMDBarroso
[12]http://twitter.com/secGen
[13]http://twitter.com/THE_REAL_SHAQ
[14]http://twitter.com/lancearmstrong
[15]http://twitter.com/GStephanopoulos
[16]http://twitter.com/arminwolf
[17]http://apiwiki.twitter.com/Twitter-API-Documentation

users timeline. The usage of the API is free of charge, it only requires an active Twitter account and is limited to 20.000 requests per hour.

The API is used by a wide range of third-party twitter applications and so-called ‘mash-ups’, which we will discuss in the next section.

Chapter 3

State-of-the-Art

3.1 Twitter Research

Currently, there is a lot of research going on in the area of user classification and sentiment detection. Though, most of the research is heavily focused on sentiment detection and not so much on user classification.

3.1.1 User classification

Naaman et al. [NBL10] categorized Twitter messages based on their content. They created 9 categories, and manually assigned the latest 10 Tweets of 350 randomly selected users. The categories are as follows: IS (Information Sharing), SP (Self Promotion), OC (Opinions/Complaints), RT (Statements and Random Thoughts), ME (It's all about me), QF (Questions to followers), PM (Presence maintenance), AM (Anecdote - me), AO (Anecdote - other). The results show that more than 40% of the Tweets are categorized as ME (for example, *'I am hungry'*), followed by RT, OC and IS with approximately 20% each. In other words, this means that 20% of the Tweets have a news character, while 80% can be characterized as user-to-user communication.

Sankaranarayanan et al. ([SST$^+$09]) propose a system called 'TwitterStand' which captures breaking news Tweets. They split the Tweets into two categories ('news' and 'junk') and use a Naive Bayes classifier to categorize them.

Cheong and Lee [CL09] categorized the Twitter users into following groups: 'Personal', 'Group' (e.g. a fanclub), 'Aggregator' (e.g. news agencies), 'Satire' and 'Marketing'.

3.1.2 Sentiment detection

Sentiment detection (also known as sentiment analysis, sentiment classification or opinion mining) is the approach of detecting the sentiment (or feelings) of the author in regard to some topic. This is particularly interesting for companies interested in knowing how users feel about their products. For example, if the word iPad is afflicted with more positive of more negative sentiments. The same applies for movies, songs, cars, holiday destinations, political parties and so on.

Different approaches exist when it comes down to actually trying to determine the sentiment, ranging from lexicographical analysis to machine learning techniques using SVMs.

Pang et al. [PLV02] evaluated how well machine learning techniques performed on sentiment detection. Using a self-created dataset based on movie reviews, they achieved an accuracy of 80% using SVMs. Kamps et al. [KKM02] combined SVMs with Osgood semantic differentiation (using WordNet relationships) and lemmatization, and accomplished an accuracy of 89% on the same data set. A different approach was used by Agrawal et al. [AS09]. Using a heuristic scoring method, which is based on SentiWordNet [ES06], they achieved an accuracy of 85% on that data set.

3.1.2.1 Existing mash-ups

A ‘mash-up’ recombines existing work and creates something new out of it. In the Web 2.0 - era, a mash-up usually connects several services through their API with the intention to offer a new service. For example, Google Maps is heavily used in mash-ups[1], ranging from Wikimapia[2] (a fusion of Wikipedia and Google maps) to MapOfStrange[3], where people can pin obscure things they found in Google maps.

The same applies for Twitter. Since it offers an open API, the data can be easily accessed. In this section, we are presenting two existing Twitter mash-ups offering sentiment detection.

3.1.2.1.1 Tweetfeel TweetFeel is a real-time sentiment search for Twitter. You can enter person’s name, a product, an event etc. and TweetFeel searches the current Tweets for that terms. It then applies an algorithm to the Tweet trying to determine whether the sentence is positive or negative. The algorithm itself is proprietary, however, it is assumed that it takes words like ’good’, ’bad’, ’sucks’,

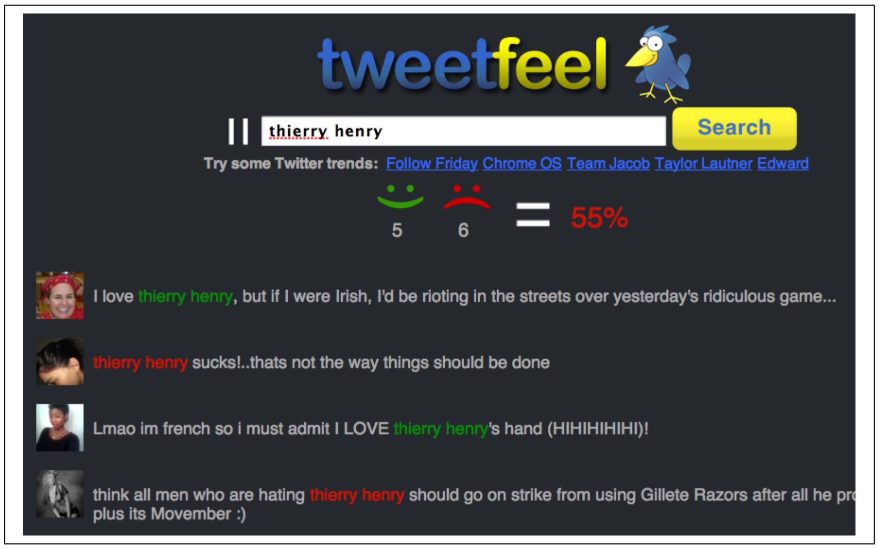
[1]According to http://www.programmableweb.com/api/google-maps/mashups, it is used in almost 2000 different mash-ups

[2]http://www.wikimapia.org/

[3]http://www.mapofstrange.com/

'love', 'hate' etc. into account. The vendor itself describes on the webpage roughly how the algorithm works:

> We compare your search result against a number of indicators to determine whether someone is generally positive or negative towards your term. We then apply some insanely complex algorithms to make sure your results fairly display the true feelings[4].


tweetfeel
thierry henry
Search
Try some Twitter trends: Follow Friday Chrome OS Team Jacob Taylor Lautner Edward
5
6
= 55%

I love thierry henry, but if I were Irish, I'd be rioting in the streets over yesterday's ridiculous game...

thierry henry sucks!..thats not the way things should be done

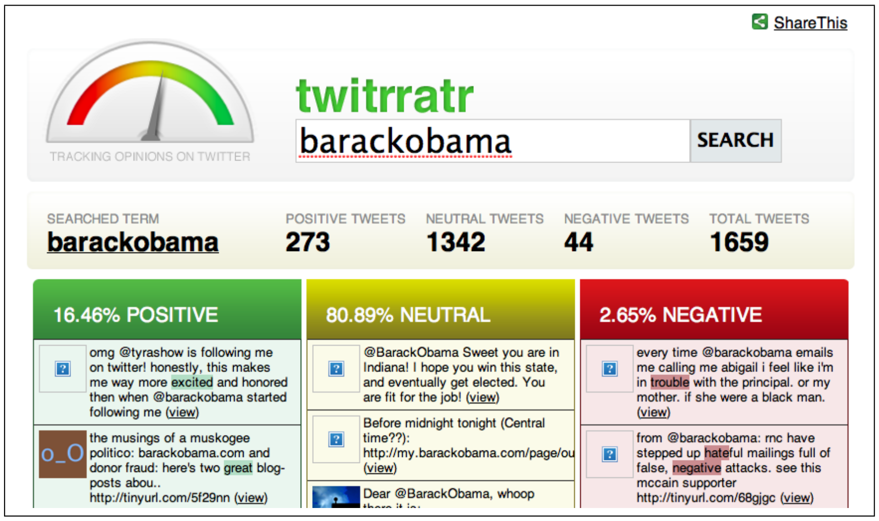
Lmao im french so i must admit I LOVE thierry henry's hand (HIHIHIHIHI)!

think all men who are hating thierry henry should go on strike from using Gillete Razors after all he pr
plus its Movember :)

Figure 3.1: Screenshot of TweetFeels.com interface

3.1.2.1.2 Twitrratr Twitrratr is another sentiment search based on Twitter. Like on TweetFeel, you can enter a search term and the service identifies if this term is rated positively or negatively. Details of the algorithm remain unknown, but the developers say that it uses a list of positive and negative keywords, which the adjectives of the Tweet are checked against[5]. Figure 3.2 show a screenshot of the application. After entering a search term ('barackobama' in this case), the application displays the results in three different areas: the positive Tweets on the left-hand side, followed by the neutral Tweets on the center, and the negative Tweets on the right-hand side.

[4]http://www.tweetfeel.com/faq.php
[5]http://twitrratr.com/about/



barackobama

SEARCHED TERM	POSITIVE TWEETS	NEUTRAL TWEETS	NEGATIVE TWEETS	TOTAL TWEETS
barackobama	273	1342	44	1659

16.46% POSITIVE	80.89% NEUTRAL	2.65% NEGATIVE
omg @tyrashow is following me on twitter! honestly, this makes me way more excited and honored then when @barackobama started following me (view)	@BarackObama Sweet you are in Indiana! I hope you win this state, and eventually get elected. You are fit for the job! (view)	every time @barackobama emails me calling me abigail i feel like i'm in trouble with the principal. or my mother. if she were a black man. (view)
the musings of a muskogee politico: barackobama.com and donor fraud: here's two great blog-posts abou.. http://tinyurl.com/5f29nn (view)	Before midnight tonight (Central time??): http://my.barackobama.com/page/ou (view)	from @barackobama: rnc have stepped up hateful mailings full of false, negative attacks. see this mccain supporter http://tinyurl.com/68gjgc (view)
	Dear @BarackObama, whoop	

Figure 3.2: Screenshot of Twitrratr.com interface

3.1.2.1.3 Twitter Sentiment TwitterSentiment[6] is the only platform which uses a machine learning approach for sentiment detection. The classifier, developed by Go et al. at the Stanford University in an academic project, is very well described in a paper [GBH09]. Basically, Go et al. import a training set by searching for ':)' and ':('. The first result is treated as a positive sample and the latter one as negative sample. Using different machine learning algorithms like Naive Bayes, Maximum Entropy and SVM, they achieved an accuracy of more than 80%.

Figure 3.3 shows a screenshot of the interface, and Figure 3.4 a screenshot of the result list. The salmon-colored entries indicate negative sentiments, while the green ones indicate positive sentiments.

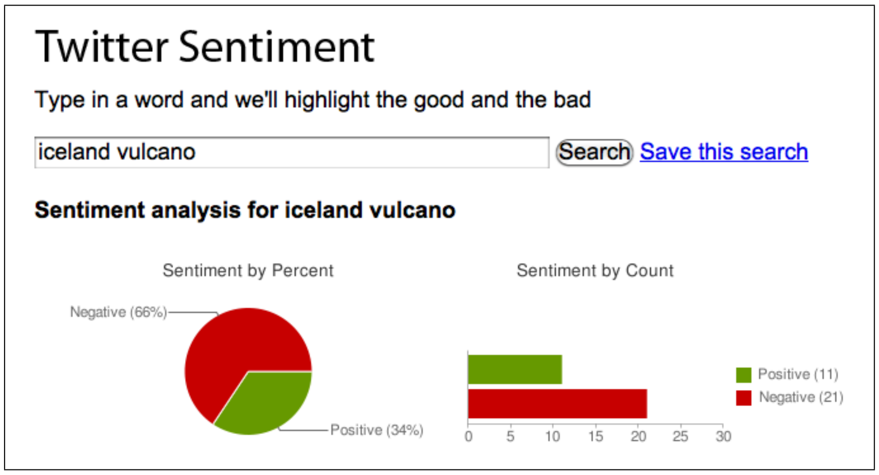
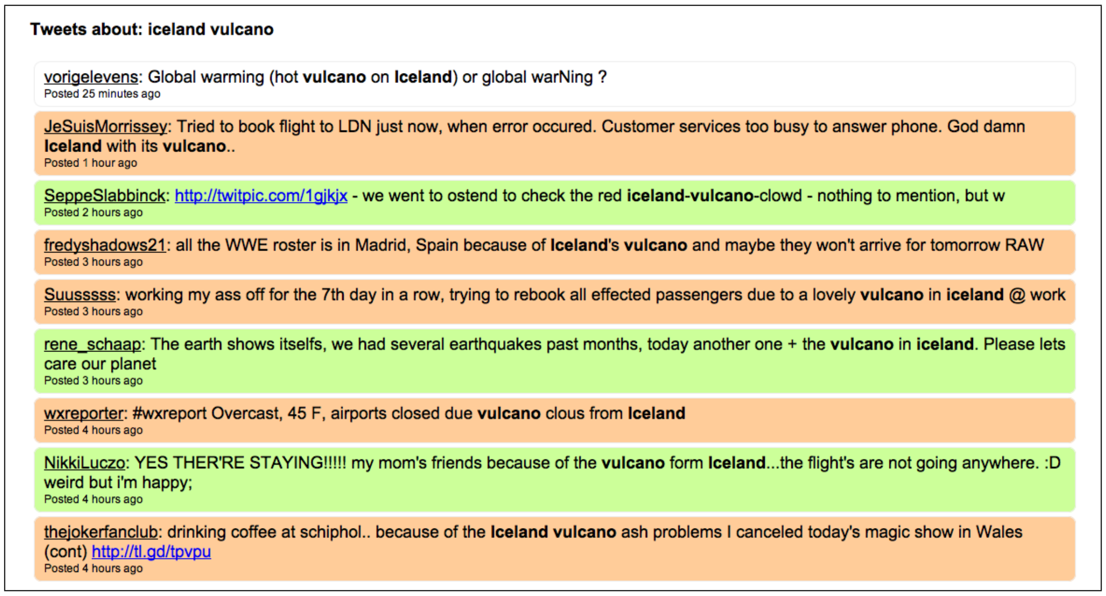


Figure 3.3: Screenshot of Twitter Sentiment interface

[6] http://twittersentiment.appspot.com/



Tweets about: iceland vulcano

vorigelevens: Global warming (hot **vulcano** on **Iceland**) or global warNing ?
Posted 25 minutes ago

JeSuisMorrissey: Tried to book flight to LDN just now, when error occured. Customer services too busy to answer phone. God damn **Iceland** with its **vulcano**..
Posted 1 hour ago

SeppeSlabbinck: http://twitpic.com/1gjkjx - we went to ostend to check the red **iceland-vulcano**-clowd - nothing to mention, but w
Posted 2 hours ago

fredyshadows21: all the WWE roster is in Madrid, Spain because of **Iceland's vulcano** and maybe they won't arrive for tomorrow RAW
Posted 3 hours ago

Suusssss: working my ass off for the 7th day in a row, trying to rebook all effected passengers due to a lovely **vulcano** in **iceland** @ work
Posted 3 hours ago

rene_schaap: The earth shows itselfs, we had several earthquakes past months, today another one + the **vulcano** in **iceland**. Please lets care our planet
Posted 3 hours ago

wxreporter: #wxreport Overcast, 45 F, airports closed due **vulcano** clous from **Iceland**
Posted 4 hours ago

NikkiLuczo: YES THER'RE STAYING!!!!! my mom's friends because of the **vulcano** form **Iceland**...the flight's are not going anywhere. :D weird but i'm happy;
Posted 4 hours ago

thejokerfanclub: drinking coffee at schiphol.. because of the **Iceland vulcano** ash problems I canceled today's magic show in Wales (cont) http://tl.gd/tpvpu
Posted 4 hours ago

Figure 3.4: Result list of Twitter Sentiment application

In addition to the related work, we want to give an overview of the current state-of-the-art in microblogging research.

3.1.3 Social Network Analysis

3.1.3.1 User categories

Every user can be quantified by 3 entities: the number of status updates, the number of friends and the number of followers. Based on those entities, Krishnamurthy et al. [KGA08] categorized the Twitter users in 3 categories: 'broadcasters', 'acquaintances' and 'miscreants'. Broadcasters are defined as *having a much larger number of followers than they themselves are following*. This includes users such as broadcasting stations, news papers or famous twitter users. The next group, acquaintances, have a balanced amount of friends and followers (they *tend to exhibit reciprocity in their relationships*). The last group, the miscreants, are defined by *following a much larger number of people than they have followers*. Spammers or stalkers fall into this type of group.

Java et al. [JSFT07] applied the HITS algorithm [Kle99], which was initially developed for page ranking in the WWW, to locate hubs and authorities within Twitter. In this case, a hub can be defined as someone who has a lot of followers and a lot of friends. An authority, on the other hand, is someone who has a lot of followers, but less friends. Someone with almost no friends and followers is neither a hub nor an authority. Based on this categorization, they split the user intention into 3 parts: 'information sharing', 'information seeking' and 'friendwise-relationships'.

3.1.3.2 Friends

Every Twitter user can follow other users, becoming a *followee* of them. Huberman et al. [HRW08] investigated with how many of those *followees* the users are actually communicating with. They define a *friend* as a person whom the user has sent at least two directed messages. The results show that people communicate directly with only 13% of their followees. This reveals two different networks in Twitter: The first one consists of the relationships between followers and followees, and the second one of the connections between the actual friends.

3.1.3.3 Anatomy of status updates

Krishnamurthy et al. [KGA08] analyzed status updates. 60% of the updates were sent over the Twitter website, while the other 40% were sent with mobiles and custom applications respectively. In regard to the daytime of status update, they found out the the amount increases in the morning hours, levels off during the day, and drops off during the night hours. Another interesting finding is that users who have more followers post more status updates than users with fewer followers.

Morris et al. [MTP10] investigated what types of questions are asked in an online social network like Twitter. They carried out a survey among 624 people. More than 50% answered that they have used their status message to get useful information by asking explicit questions. According to the survey, the participants preferred this way rather than search engines because they have more trust in the responses from their friends, and by the fact that they believed that search engines simply are not able to answer their question. The question itself are 17% factual (asking for a objective answer), while more than 50% are opinionated or recommendation questions. In regard to the topic, the majority asked for technical advice (29%), followed by entertainment (17%) and family questions (12%).

3.1.3.4 Geographical distribution

Java et al. [JSFT07] carried out a detailed analysis of Twitter in the year 2007, being one of the first scientific papers to deal with this topic. They performed a detailed geographical analysis. The results show that Twitter is mostly used in the United States (especially East Coast), in Europe and Asia (mainly Japan). Twitter is adopted the most in the cities Tokyo, New York and San Francisco. Figure 3.5 visualizes the geographical distribution.

Also, Java et al. [JSFT07] found out that the social network ties are closer in Europe and Asia than in North America.

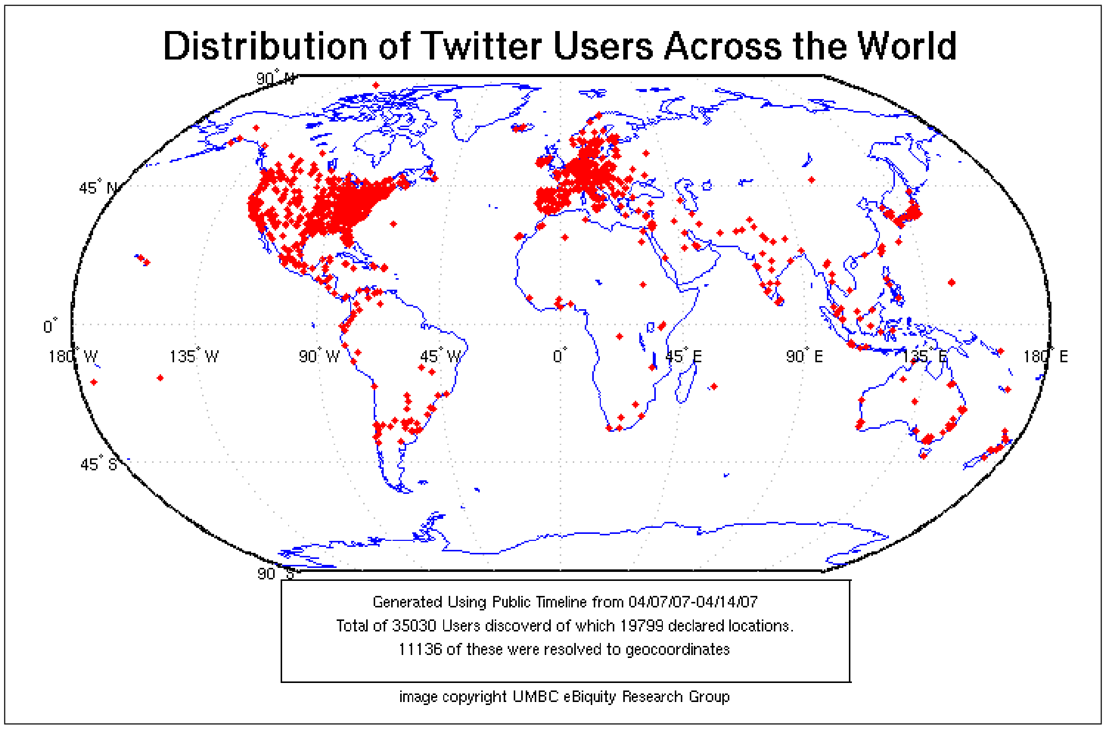

Distribution of Twitter Users Across the World
90° N
45° N
0°
45° S
90° S
180° W
135° W
90° W
45° W
0°
45° E
90° E
135° E
180° E
Generated Using Public Timeline from 04/07/07-04/14/07
Total of 35030 Users discoverd of which 19799 declared locations.
11136 of these were resolved to geocoordinates
image copyright UMBC eBiquity Research Group

Figure 3.5: Geographical distribution of Twitter users. Image from [JSFT07]

3.1.4 Recommender systems

Phelan et al. [PMS09] built a recommender system called 'Buzzer' which proposes 'topical news stories' to users. Given a list of news stories (in form of RSS feeds), Twitter can be used to rank this list. The system builds a term vector of both the RSS list and the user's Tweets, and uses the TF-IDF weighting scheme (see section 3.2.4) to locate the RSS feeds which correspond the most with the Tweets. The Tweets can either be public Tweets (public timeline), or Tweets solely from friends. The evaluation revealed that using the public timeline produces the best results. 67% of the users state that they preferred the results produced by the public timeline, while 22% stated a preference for friend-based produced results (11% didn't prefer any strategy).

3.1.4.1 zerozero88

Another recommending system called 'zerozero88' was developed by Chen et al. [CNN+10]. It consists of three different algorithms: 'Candidate URL', 'topic relevance' and 'social voting process'. They can be linked together as required.

'Candidate URL' deals with the task of selecting the most relevant URLs which are posted on Twitter. According to the authors, it is not possible to gather all URLs which are posted on Twitter, so they need to limit their set of URLs (Candidate URLs). They used two approaches to chose the URLs: only select the URLs that are posted in the user's Twitter neighborhood (posted by followees, or followees-of-followees), and to select the most popular URLs being posted on Twitter (using the API from *tweetmeme.com*).

The 'topic relevance' algorithm recommends URLs that might interest a user by comparing the vector space representation of a URL set to the vector space representation of a given user using cosine similarity. For a given user, the system first creates two vectors: one contains all the words that the user uses in his/her Tweets (bag-of-word), and the other vector consists of the 'high-interest words' for that user. The terms in the first vector describe the interest in that particular word. The latter vector is created by scanning the Tweets of the user's followee an selecting the top-20% of identical words. One of those two vectors can now be compared to the URL vector. This URL vector is created by scanning the Tweet list as well, and adding the words which are used to describe the URL, to the vector.

'Social voting process' is based on the 'one person, one vote' approach developed by Hill and Terveen [HT96]. If a certain URL is mentioned in the user's Twitter neighborhood (followees, or followees-of-followees), a vote is added to that URL. In the end, the URLs with the most votes are recommended. Also, they weight the votes, depending on which person mentioned that Tweet. If that person posts a link every hour, the weight is decreased. On the other hand, if the person posts a link twice a months, the weight is increased.

In total, the recommender can be configured in 12 different ways. The evaluation shows that the 'Social voting process' significantly improves the performance, resulting in approximately 70% useful recommendations. A configuration which only recommends random links from the popular Twitter URLs results in only 30% useful recommendations.

3.1.5 e-learning

Ebner and Schiefner [ES08] investigated if microblogging can be useful for e-learning using mobile devices. They set up a group on the microblogging platform Jaiku[7], asking users to join and discuss about e-learning. 23 users joined the group and participated in the survey. The main finding of this survey is that users mainly use microblogging platforms in order to connect with each others and to share news. This applies especially to the scientific community, which use it, for example, to report live from conferences, or to inform the community of new papers. They see microblogging platforms as a community where less commitment is expected, thus

[7]http://www.jaiku.com

being a good alternative to ‘classic’ e-learning systems like Blackboard or Moodle.

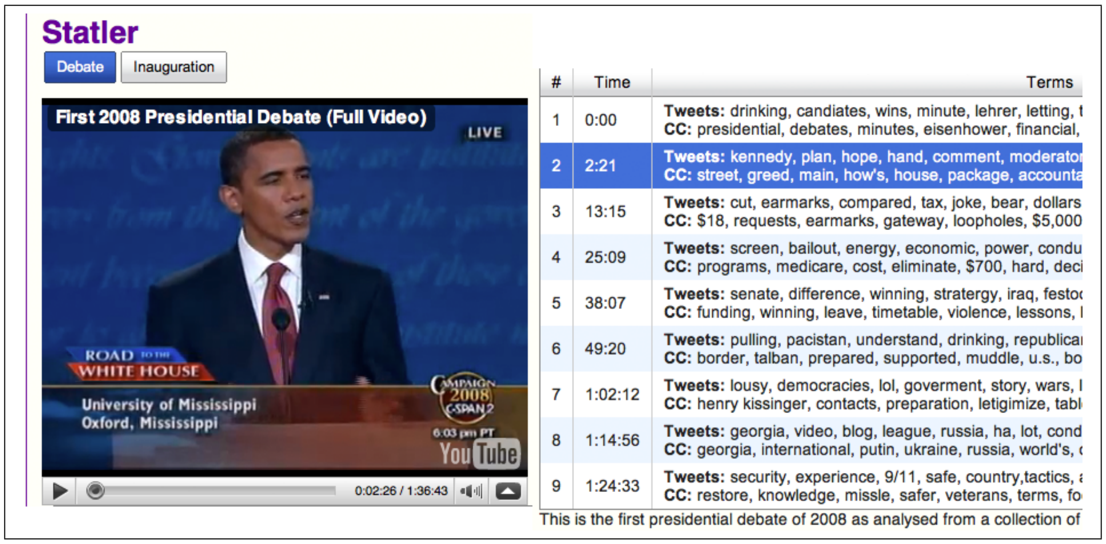
3.1.6 Trend analysis

Cheong and Lee [CL09] researched the anatomy of ‘trending topics’. A trending topic is a certain topic which is posted heavily on Twitter (for example, ‘haiti’ in January 2010). First, they have split the trending topics into 3 categories: long-term-, medium-term- and short-term topics. Long-term topics occur infrequently, but over a long amount of time in the public time-line, while medium-term topics occur more frequently. However, the medium-term topics are limited to a time range of a few days. Short-term topics are heavily discussed topics, and often refer to current events. Also, they categorize the users into 3 major groups: ‘Personal’, ‘Aggregator’ and ‘Marketing’. Those groups correspond to our categorization (C1). The results show that mostly users who talk about their personal life (‘Users’) contribute to emerging trending topics. An interesting finding is that spammers (‘Marketing’) burst upon trending topics to generate attention.

3.1.7 Tweetgeist

Shamma et al. [SKC10] used Twitter to discover the semantics and structure of live media events. They built a system called ’Statler’[8] which is able to cross-link a live media event with community annotations. Figure 3.6 shows a screenshot of the system. The system analyzed the U.S. presidential debate in 2008. As one can see, the video is segmented into different parts, and each part is annotated with several tags. This tag list is generated by tapping into the public Twitter timeline while the media is broadcasted.

[8]http://bit.ly/statler



#	Time	Terms
1	0:00	**Tweets:** drinking, candiates, wins, minute, lehrer, letting, t **CC:** presidential, debates, minutes, eisenhower, financial,
2	2:21	**Tweets:** kennedy, plan, hope, hand, comment, moderato **CC:** street, greed, main, how's, house, package, accounta
3	13:15	**Tweets:** cut, earmarks, compared, tax, joke, bear, dollars **CC:** $18, requests, earmarks, gateway, loopholes, $5,000
4	25:09	**Tweets:** screen, bailout, energy, economic, power, condu **CC:** programs, medicare, cost, eliminate, $700, hard, deci
5	38:07	**Tweets:** senate, difference, winning, stratergy, iraq, festoc **CC:** funding, winning, leave, timetable, violence, lessons, l
6	49:20	**Tweets:** pulling, pacistan, understand, drinking, republica **CC:** border, talban, prepared, supported, muddle, u.s., bo
7	1:02:12	**Tweets:** lousy, democracies, lol, goverment, story, wars, l **CC:** henry kissinger, contacts, preparation, letigimize, tabl
8	1:14:56	**Tweets:** georgia, video, blog, league, russia, ha, lot, cond **CC:** georgia, international, putin, ukraine, russia, world's, c
9	1:24:33	**Tweets:** security, experience, 9/11, safe, country,tactics, a **CC:** restore, knowledge, missle, safer, veterans, terms, fo

This is the first presidential debate of 2008 as analysed from a collection of

Figure 3.6: Screenshot of the Statler application.

3.1.8 Knowledge Acquisition

Weng et al. ([WLJH10]) developed an approach to locate influential Twitter users by calculating a score which they call 'TwitterRank' (influenced by Google's PageRank). This score takes the so-called 'homophily' - a term from social network studies - into account. In this context homophily describes the phenomenon that people who are interested in the same topics are more likely to bond with each other than with other people. That way it is possible to identify users which are interested in the same topics. The topics itself are distilled using an unsupervised machine learning technique (Latent Dirichlet Allocation). Using this approach, they are able to generate a list of top topics and cross-link this list with the most influential Twitter users for that topics.

Wagner and Strohmaier [WS10] analyzed social awareness streams (such as Twitter) by developing a network-theoretic tri-partite model called 'Tweetonomies'. The model consists of messages, users and resources, and extends the existing model of folksonomies. It allows to distinguish different aggregations of social awareness stream. Consider the topic 'support vector machines'. In order to aggregate a stream which deals with this topic, all messages which contain the hashtag '#svm' could be added. The stream can as well be aggregated by searching for the key words 'support vector machines' in the messages. Also, it's possible to browse through a user directory which deals with support vector machines. As one can see, for the same topic, there are different ways to aggregate streams. In order to compare those streams, they defined several measures:

- 'Social diversity' measures the number of distinct users in a stream

- ‘Conversational Diversity’ counts the users being mentioned
- ‘Conversational Coverage’ counts how many of the messages are of conversational purpose
- ‘Lexical Diversity’ counts the numbers of unique keywords
- ‘Topical Diversity’ represents the average number of topics per message
- ‘Information Diversity’ measures how many messages have an informational character

The results indicate that different aggregations reveal different semantic models. Especially hashtag-generated streams seem to be a promising way to produce meaningful semantic models.

3.2 Text classification

The next sections gives an overview of the current state-of-the-art in text classification. This include text preprocessing, the vector space model and various classification techniques.

Text categorization (or text classification) is the task of labeling natural language texts with thematic categories [Seb02].

3.2.1 Preprocessing

For this thesis, we define ‘preprocessing’ as all tasks which take place before transforming the Tweets into the vector space representation. More specifically, we use regular expressions to process the Tweets (word conversion), and word stemming afterwards.

3.2.1.1 Regular Expressions

Regular Expression is a formal language that allows us to process texts. Jeffrey Friedl, the author of ‘Mastering Regular Expressions’, describes them as *‘with a general pattern notation almost like a mini programming language, Regular Expressions allow you to describe and parse text.’* [Fri06]

A pattern is defined which adheres to the formal Regular Expression language. This pattern is then used to process a certain text. It can be used to extract strings from a text, modify the text or just check if the pattern is contained in the text. Let us illustrate this with a small example: Consider an address field which contains the postal code and the city, for example '8043 Graz'. If we want to extract only the postal code, the following Regex can be used: **/([0-9]{4}) .*/**. Basically, it means that we are searching for four digits ([0-9]{4}) followed by a space and any other characters (.*). The parenthesis indicates that we want to extract this peace of text. If we want to extract only the city name, the Regex would look like this: **/[0-9]{4} (.*)/**.

Regular Expressions are available for almost any programming language. For our application, PHP's native regex library is used.

3.2.1.2 Stemming

Stemming is a mechanism for reducing English words to their stem (or root) form. For example, the stem form of the words 'connections' and 'connected' is 'connect'. This mechanism is particularly useful in the field of information retrieval and indexing. An algorithm for determining the stem form of a given word was initially present by Julie B. Lovins in 1968 [Lov68].

M.F Porter developed another algorithm in 1980 [Por80], which became the de-facto standard for word stemming. This algorithm works by applying a set of different rules, yielding to the word stem after 5 iterations (so-called 'steps'). Porter has developed 62 rules which may or may not apply to a given word.

Consider the word 'oszillators'. The first step removes the trailing s (Rule '$S \rightarrow$ '), and the second step transforms the word 'oszillator' to 'oscillate' (Rule '*ATOR* $\rightarrow$ *ATE'*). The third step is ommited as no rule matches. Step 4 applies the rule '*ATE* $\rightarrow$ ', which reduces our word to 'oszill'. Finally, step 5 produces the word stem 'oszil' by removing any L or D ligatures at the end of the word.

M.F. Porter has released a reference implementation of his algorithm ('Snowball'). For this project, the PHP implementation of this algorithm is used, which is based on Porter's reference implementation[9].

3.2.2 Term-Document Matrix

A Document-Term matrix is a matrix representing every document and word in a collection of documents. The rows represent the documents and the columns the words within these documents. Consider following 3 documents:

- Document A: 'Hello world'
- Document B: 'Good bye world'
- Document C: 'Hello everbody'

[9]http://code.google.com/p/php-stemmer/

Now, we build a global dictionary, where all words from that documents are contained. Then, a matrix is created which contains a row for each document. The value indicates if the word is present (1) or not (0).

Table 3.1: Document-Term matrix

	Hello	World	Good	bye	everbody
A	1	1	0	0	0
B	0	1	1	1	0
C	1	0	0	0	1

The next section explains how this matrix is transformed into a Vector Space representation. This representation is needed as input for the Support Vector Machine.

3.2.3 Vector Space Model

The Vector Space Model is a vector representation of a certain document set. Each vector corresponds to a single document, and each term in the vector to a term in the document.

Consider the document set D which contains an amount of documents D_i:

$D_i = (d_{i1}, d_{i2}, ..., d_{in})$

Each dimension in this vector corresponds to an index term d_i. The value is 1 if the term occurs in the document, and 0 otherwise. Also, it is possible not to use a binary encoding (0,1), but to weight the terms according to their importance. There exist several approaches to calculate the weight - amongst other, TF-IDF (Term Frequency - Inverse Document Frequency) is widely used [SWY75]. Nevertheless, in this thesis, we're using a binary weighting in our vector representation.

The Vector Space representation of the above example (Section 3.2.2) now looks as follows, and can be used as input for the Support Vector Machine.

$D_1 = (1, 1, 0, 0, 0)$

$D_2 = (0, 1, 1, 1, 0)$

$D_3 = (1, 0, 0, 0, 1)$

3.2.4 Term Frequency - Inverse Document Frequency

In a given document set, each term has a different importance in a certain document. The Term Frequency - Inverse Document Frequency calculates the weight for every term in a document, taking all document sets into account. The more a word occurs in a document, and the less it occurs in the other documents of the set, the higher the weight is.

To assign a weight to a term t in a document d, following formula is used [MRS08]:

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t .$$

The score is highest when the term occurs often within a small subset of the document set, and the lower the more times it occurs in other documents. TF-IDF is widely used to compare the similarity between documents. Also, it is used for search queries by calculating the similarity of a query q with a document d, providing a sorted list of the most relevant documents.

3.2.5 Feature selection

Being one of the most important tasks in text classification, feature selection is the task of selecting the terms which are to be used in the training set. Selecting the terms has two advantages: A reduced term size speeds up the training process. Also, it reduces the noise by removing irrelevant features, thus increasing the classification accuracy [MRS08]. There exists a vast amount of algorithms aiming to select only the relevant features (see [BL97] and [DDH$^+$07]).

Dumais et al. [DPHS98] carried out a test using the Reuters-21578 dataset and found out that using the simplest document representation (words delimited by whitespaces, no stemming) performed as good as more sophisticated representations. As we will later in section 5.5.4.4, we can second the theory that stemming brings no boost in accuracy. Also, Dumais et al. state that feature selection does not increase the speed to train the classifier. It even takes significantly longer to carry out the additional step of feature extraction than the actual training process. The same can be said for classifying new documents - far more time is being spent on pre-processing the text than on actually classifying it.

3.2.6 n-grams

Character n-grams are n adjacent characters from a given input string. For example, a 3-gram of the word 'TERM' would be '__T', '_TE', 'TER', 'ERM', 'RM_', 'M__'. n-grams of size 1 are called 'unigrams', of size 2 'bigrams', of size 3 'trigrams', and with a size of more than 3 characters, they are simply referred to as 'n-grams'. N-Grams are particulary useful for language detection ([CT94]) and speech recognition (see [LBS04], [WTSW02]). Cavnar and Trenkle [CT94] achieved a 99.8% correct language classification rate on Newsgroup articles. Also, n-grams are used to create string kernels in SVMs [LSST$^+$02].

3.2.7 Classification methods

There are various theoretical approaches how document classification can be achieved. Until the late 80s, the most common approach was ‘knowledge engineering’. Expert knowledge was encoded into a set of rules which were used to categorize the texts. It worked very well for large text corpora - for example, the CONSTRUE system which is based on TCS classifies documents with an accuracy of more than 90% [HANS90]. Despite the good results, the disadvantages of this approach can not be neglected. The main disadvantage is that the rule base needs to be entered and maintained by domain experts, which is a very time-consuming task.

To address this disadvantages, the community started to focus on machine learning techniques in the early 90s, eliminating the need to enter and maintain a rule set. The quintessence of those approaches is to have a domain expert create an initial training set. Based on this data, the process builds an automatic classifier which contains the information needed to categorize a new document. This approach is called a ‘supervised machine learning technique’ [Seb02].

In general, a text classification system can be divided into two categories: supervised classification and unsupervised classification. A supervised classifier uses external information (such as user input) in order to carry out the process, while an unsupervised classifier has no such information available.

As the theoretical foundation of this thesis is a supervised classifier (SVM), the focus is put on those systems. Let us have a look at the most well-known supervised classification techniques:

3.2.7.1 Naive Bayes classifiers

Developed in the 1970s by C.T. Yu and G. Salton [YS76], and S. Robertson and K. Spark [RJ76] respectively, the Binary Independence Model was one of the first models used in probabilistic information retrieval.

Basically, each document is represented by a vector $\vec{x} = (x_1, ..., x_m)$, where $x_t = 1$ if a certain term t is present in the document, and $x_t = 0$ otherwise (hence the word Binary in the name). Queries are built the same way. ‘Independence’ indicates that the model assumes that the terms are not associated with each other. Using the Bayes rule, the probability of relevance of a certain document can be calculated. Despite the simplifying assumptions this model uses, it achieves good results in practice [MRS08].

The Naive Bayes classifier is closely related to the Binary Indepence Model[10]. In general, the Naive-Bayes classifier assumes that the features are not associated with each other (identical to the BIM). A feature can be defined as an attribute which

[10]The multivariate Bernoulli NB Classifier actually equals the Binary Independence Model

helps to identify that object. For example, the features of a car are, amongst others, 'vehicle', '4 tires', 'engine', 'moves on ground'. Now, the Naive Bayes classifier assumes that none of those features are related to each other. While this is a rather simplifying assumption (hence the name Naive), it turned out that this algorithm performs remarkabely well in practice, even when strong feature dependencies are present [DP97].

Also, it is worth mentioning that the more state-of-the-art classification algorithm Okapi BM25 is based on the Binary Independence Model, but uses a more sophisticated weighting scheme (BM25) [MRS08].

3.2.7.2 kNN

The 'k Nearest Neighbor' approach assigns a class to each training document. Then it classifies a new document by assigning it to the same class as the k nearest neighbors. For example, if k is set to 1, the new document will have the same class as the immediate neighbor. If k is 5, the algorithm will chose the class which occurs most often in the surrounding 5 neighbors.

kNN (with $k > 1$) performs quite well. On a Reuters-21578 data set, kNN performed better than NB (82.6% accuracy vs. 72.3%), but not as good as SVMs (86.4%) [Joa02]

3.2.7.3 Random Forests

Developed in the mid 90s, the Random Forest uses a set of single classification trees. Each tree depends on a randomly sampled vector. If a new object is to be classified, the object is passed to each tree in the forest, asking it to classify the object. The result is treated as a vote. The forest choses the classification with received the most votes. According to Breiman [Bre01], Random Forests are considered to be one of the most accurate classifiers available to date. Rios and Zha [RZ04] have shown that Random Forests and SVMs perform equally well at spam classification, and both outperform a Naive-Bayes classifier significantly.

3.2.7.4 Support Vector Machines

Support Vector Machines (SVMs) are a technique used in supervised machine learning. Given a set of categories, which contain an arbitrary number of items, SVMs predict which category a new items belongs to. The theoretical background of SVMs is explained detailed in [Vap95] and [Bur98].

Figure 3.7 illustrates this: Given 2 categories (red items and blue items), SVM creates a hyperplane which separates the two categories with the highest possible margin (H1 and H2 in the figure). The items which determine this margin are called the Support Vectors (2 blue and 1 red item in this example).

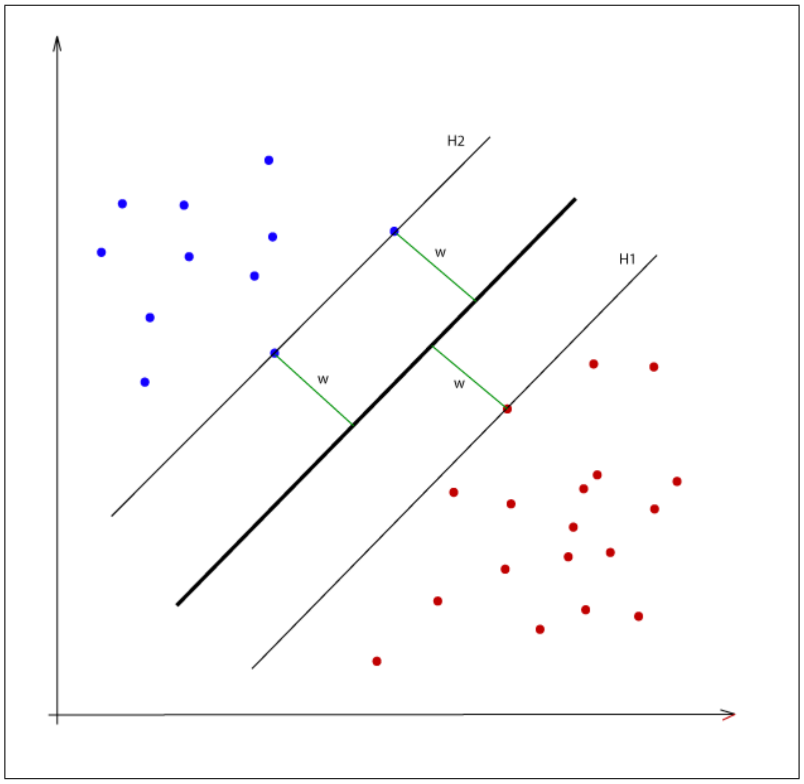


Figure 3.7: Separating hyperplane in SVM.

Following [Bur98], consider the training data

$$(x_1, y_1), ..., (x_l, y_l), x \in R^n, y \in \{+1, -1\},$$

which can be separated by the hyperplane

$$(w \cdot x) - b = 0$$

The values of y indicate two classes. $+1$, if a pair (x, y) is part of the class and -1, if it's not.

The vector w is a normal vector on the separating hyperplane. The aim is to maximize the length of w. To achieve this, let us introduce another two hyperplanes:

$$(w \cdot x_i) - b \geq 1 \qquad \text{if } y_i = 1,$$

and

$$(w \cdot x_i) - b \leq 1 \qquad \text{if } y_i = -1,$$

This two formulas can be rewritten using the following inequality:

$$y_i[(w\cdot x) - b] \geq 1, \qquad i = 1, ..., l.$$

The points which satisfy this inequality are called the *Support Vectors*. The linear classifier satisfies this inequality, while minimizing

$$\phi(w) = ||w||^2$$

This quadratic optimization problem can now be solved using Lagrange multipliers. The reason for this is that the training data will be in the form of dot products between vectors. For every inequality constraint, let us introduce a Lagrange multiplier a.

$$L_p = \tfrac{1}{2}||w||^2 - \sum_{i=1}^{n} a_i y_i (x_i\cdot w + b) + \sum_{i=1}^{l} a_i$$

Now, L_p needs to be minimized with respect to w and b, and maximized with respect to a. This results in the conditions

$$w = \sum_i a_i y_i x_i$$

and

$$\sum_i a_i y_i = 0$$

This two equality constraints are in the dual formation, and can thus be substituted into L_p, which gives:

$$L_D = \sum_i a_i - \tfrac{1}{2} \sum_{i,j} a_i a_j y_i y_j x_i\cdot x_j$$

Since L_p (primal) and L_D (dual) derive from the same function, but with different constraints, the solution can be found by either minimizing L_p or by maximizing L_D. The Support Vector Machine selects the support vectors by maximizing L_D. If the Lagrange multiplier a for a given training point is greater than 0, the point lies on one of the separating hyperplanes, and is therefore added as a support vector .

This applies for the case where the training data is linear separable. If the training data is not linear separable, a slack variable ξ needs to be introduced. This allows the classifier to make mistakes in order to maximize the margin. The constraints from the minimization problem can therefore be rewritten as:

$$\begin{array}{ll} (w\cdot x_i) + b \geq +1 - \xi_i & \text{if } y_i = +1, \\ (w\cdot x_i) + b \leq -1 + \xi_i & \text{if } y_i = -1, \end{array}$$

subject to

$$\xi_i \geq 0 \forall i.$$

For an error to occur, the corresponding ξ_i must exceed unity, resulting in $\sum_i \xi_i$ being the upper bound on number of training errors. On order to assign an extra cost for errors, we change the objective function to $\frac{||w||^2}{2} + C(\sum_i \xi_i)$, where C is used to set the penalty for classification errors.

The solution is to maximize

$$L_D = \sum_i a_i - \frac{1}{2} \sum_{i,j} a_i a_j y_i y_j x_i \cdot x_j$$

subject to

$$0 \leq a_i \leq C, \text{ and } \sum_i a_i y_i = 0.$$

The only difference to the linear separable case is that the Lagrangian multiplier a now has an upper bound ξ.

This is a useful protection to 'overfitting': the classifier can ignore a certain amount of outliers when trying to maximize the margin of the separating hyperplane - turning it to a soft margin.

3.2.8 Optimal classifier for text categorization

The main decision of this thesis is the selection of the optimal classifier for this task.

Dumais et al. [DPHS98] and Yang et al. [YL99] both claim that SVM (and kNN) classifier perform best for text classification tasks. Rios and Zha [RZ04] found out that SVMs compare equally well to Random Forests when classifying e-mail spam, and Joachims [Joa98] argues that SVMs are very well suited for text classification.

According to Joachims [Joa98], the significant advantages of SVMs in regard to text classification are

- the high dimensional input space
- few irrelevant features
- sparse document vectors
- text categoration problems are linearly separable

In regard to the kernel function, it is considered to be best practice to use the linear kernel in SVM [GM04].

Chapter 4

Implementation

4.1 Overview

The practical part of this master's thesis is the implementation of a platform. This platform is used to setup and train the classifier (backend) and a frontend which actually classifies a given user.

4.1.1 ER diagram

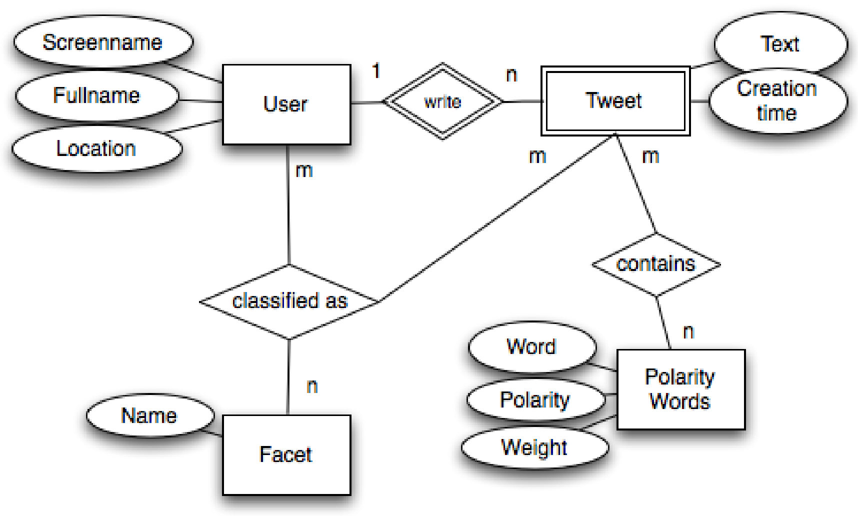


Figure 4.1: ER diagram of application

Figure 4.1 visualizes the application using an ER diagram. The application consists of four entities: User, Tweet, Facet and Polarity Word. A user writes Tweets, and a Tweet is always associated to one particular user. Each user has the attributes ‘screenname’ (Twitter username), ‘fullname’ and ‘location’ assigned.

User and Tweets are classified using facets. Facets can be either News, Users, Company, or Factual or Opinionated. If a certain user is classified as facet ’News’, all his/her Tweets are considered to be ’News’ as well. Nevertheless, it is possible that a Tweet overrides the facet. This is particularly useful for the facets Opinionated and Factual, where we cannot just bulk-assign users to that facet.

Tweets can contain Polarity Words. This words have a certain polarity, which is used for sentiment detection. The weight indicates how strong a word is in regard to the sentiment. For example, the words ‘kill’ and ‘abuse’ have both a negative polarity, but ‘kill’ has a higher weight.

4.1.2 Application diagram

Figure 4.2 visualizes how the application works internally. The client communicates with the Web Server using the HTTP protocol. The web server forwards the request to the PHP interpreter, which in turn processes the request. Using different frameworks (see Section 4.2), the interpreter handles the complete communication with all involved systems. This includes the communication with the Twitter API and the MySQL database (where the Tweets are stored). The interpreter also creates the vector files which are being picked up and transformed into SVM vector models by libSVM. Once the request is processed, the results are sent back to the client.

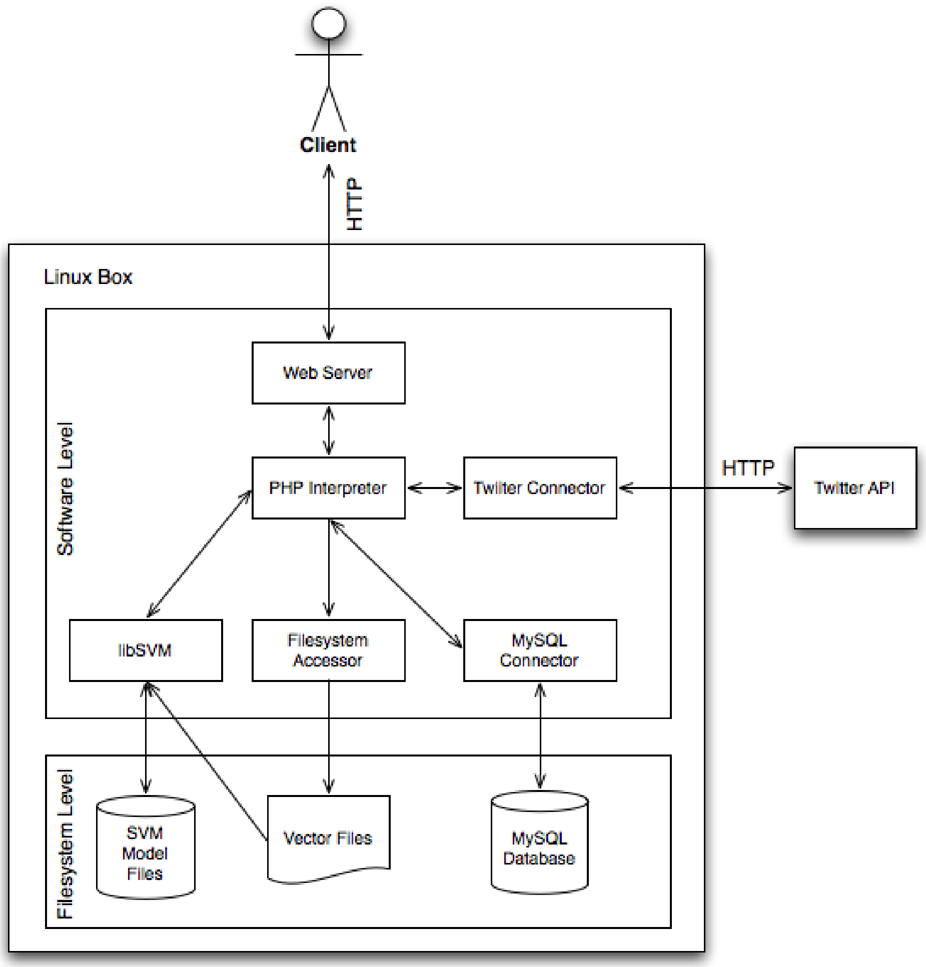


Figure 4.2: Application diagram

4.2 Technology

4.2.1 PHP

PHP is a hugely popular programming language for building web applications. It runs server-side (usually as a module on a web server), and the code is executed in the PHP runtime. The current version is 5.2 and is released as open-source software. PHP 5 offers all aspects of a modern web programming language, like object orientation, built-in access to various databases and a powerful library. PHP is used in 30% of all internet web sites[1]. The largest web site which is driven by

[1]http://www.nexen.net/chiffres_cles/phpversion/18284-php_statistics_for_march_2008.php

PHP is Facebook.

4.2.1.1 History

PHP was initially developed by Rasmus Lerdorf as a set of perl scripts for personal usage (hence the name ‘Personal Home Page Tools’)[2]. After rewriting the engine in C, Lerdorf released the second version in 1997 as open-source. At this time, PHP/FI was reported to be installed on 20.000 domains worldwide. The language still was very limited. In 1998, PHP 3.0 was released, which was a complete rewrite by Andi Gutmans and Zeev Suraski. Gutmans and Suraski used PHP/FI for an University project, but found out that it was too limited for their needs. Lerdorf, Gutmans and Suraski decided to announce PHP 3.0 as the official successor of PHP/FI. PHP 3.0 already had some of today’s features like object orientation.

The current version, PHP 5, was released in July 2004 and offers a complete and modern web programming language.

4.2.2 Symfony Framework

With the gaining popularitity of PHP, some interesting frameworks for PHP emerged as well. Currently, one of the most popular framework is Symfony[3]. It provides a full MVC framework and various features for simplifying the developer’s life. Those features include a simple project configuration using YAML files, a database abstraction layer using Doctrine and tools to test and deploy the project. Both backend and frontend of the platform is implemented in Symfony.

4.2.3 PEAR

PEAR (‘PHP Extension and Application Repository’) is a community-driven framework. Unlike Symfony, PEAR is not considered to be a full MVC framework, but offers numerous extensions and applications (so called ‘packages’). Those extensions cover, among others, tools for image processing (‘Image_Transform’), web service implementations (‘Services_Amazon_S3’), data structures (‘DataSource_Excel’) and encryption implementations (‘Crypt_GPG’). For this implementation, the web service implementation ‘Services_Twitter’[4] is used. This package provides an easy way to access Twitter’s API.

[2]http://www.php.net/manual/en/history.php.php

[3]http://www.symfony-project.org/

[4]http://pear.php.net/package/Services_Twitter

4.3 MVC Application Design

The used framework for this application is MVC - based. MVC stands for Model-View-Controller and is an architectural design pattern. It separates the application into three different parts: the model, the view and the controller. The 'model' represents the used data. It abstracts the data from the used database and offers a simple interface to load, store and update operations. The 'controller' contains the application logic, and is the link between model and view. The 'view' is in charge of rendering the model according to the requested output format - which is HTML in our case.

4.3.1 Models

The application uses three Models: Users, Tweets, and Words. 'Users' represent the Twitter users which have been added to the backend, while 'Tweets' contain the actual Tweets written by the users. 'Words' represent the polarity words used for sentiment detection. If you compare the used models to the ER diagram, you will notice that there is no model for the facets. While we are aware that the resulting database model is not in Third Normal Form, we decided to add the facets as a property directly to the User and Tweet model. This becomes handy when a Tweet overrides the Users' facet. It's now just a matter of checking the Tweet first, and if no facet is set, use the Users' selected facet. Of course, this model is based on the assumption that there are always 3 categories (C1, C2 and C3). This is a valid assumption for this thesis, but the model needs to be re-factored if more categories are to be supported.

The resulting model is visualized in Figure 4.3.

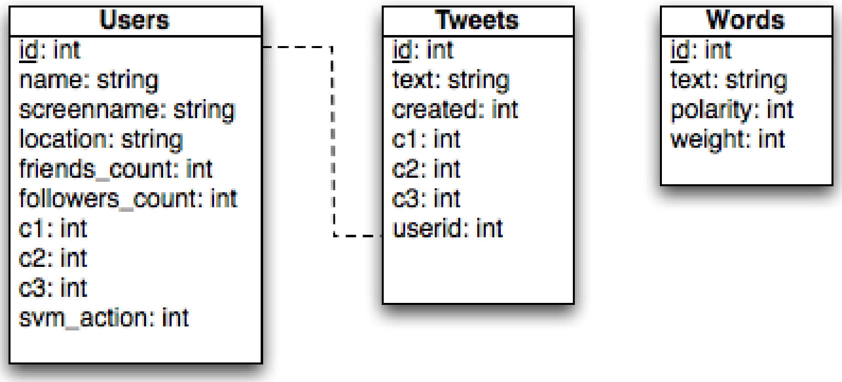


Figure 4.3: Relational Database Model

The table 'Words' is not connected to other tables, its sole purpose is to store the imported words which are used for sentiment detection (see Section 5.6).

4.3.2 Views

Views are used to render the output. For this application, the output is displayed as HTML.

Table 4.1: Most important Views used by the application

Module	View	Description
backend	**setup**	Renders the main Backend page. This page is used to setup the training- and test set
backend	**test**	Renders the Unit test results.
backend	**classify**	Renders the classification result. This view is used by the Frontend and the detailed classification results in the backend.
frontend	**index**	View for the static Frontend page. Loads AJAX scripts, and loads the classification result.

4.3.3 Controllers

The application uses two different controllers: a backend controller and a frontend controller. The controllers are responsible for handling the user input and generating the requested output. The offered actions are listed in Table 4.2 and Table 4.3, respectively.

Table 4.2: Backend Controller - offered methods

Action	Description
createUser	Creates a new User - also imports the latest 40 Tweets.
updateUsers	Updates the Tweets of all stored users in the backend.
classifyUser	Classifies a given Twitter user. Uses the SVM library.
saveUsers	Saves the new settings of the users (training or test set, categories)
importTimeline	Imports the latest 50 Tweets from the public timeline.
rateTweet	Updates the category of a given Tweet.
test	Runs the Unit test
crossValidation	Performs a k-folded cross-validation test. k can be passed as a parameter.

Table 4.3: Frontend Controller - offered actions

Action	Description
show	Renders the frontend.

The frontend itself offers only one single action ('show') - this results from the fact that the frontend just displays a static HTML page, forwards the user's input to the backend and displays the result afterwards (via AJAX). Hence, no further actions are required in the frontend - the classification process is completely handled by the backend via the 'classifyUser' action.

4.3.4 Libraries

Since we wanted to separate the classification-related code from the backend, that code has been outsourced to library files. This libraries are then used by the Backend to carry out the actual classification. The following Tables (4.4, 4.5 and 4.6) describe the libraries.

Table 4.4: SVM library (svm.class.php) - offered methods

Method	Description
createVectors($tweets, $globalDict, $type)	Creates a vector representation of the given input Tweets. The global dictionary is passed as parameter as well. The $type parameter is used as the file extension for the stored file (i.e. vector.c1, when type = c1)
createModelFile($type)	Creates the SVM model file for a given type (i.e. C1). Relies on a valid vector file which was created by **createVectors**. For example, if $type is set to 'c1', it will transform the file 'vector.c1' to 'vector.c1.model'
predictTweets($tweets, $globalDict, $type)	Classifies the given input Tweets. Runs the 'svm-predict' command. Returns an array with the classification results
deleteFiles()	Deletes all files in the SVM data directory

Table 4.5: Word conversion library (preprocessing.class.php) - offered methods

Method	Description
preprocessText($text, $useRegexs, $useStemming)	Preprocesses a given input text. If $useRegexs is set to true, the regular expression rules will be applied (see Section 3.2.1.1). If $useStemming is true, then the words will also be stemmed (see Section 3.2.1.2 for details). Returns the processed text as string.

Table 4.6: Sentiment detection library (lex.class.php) - offered methods

Method	Description
classify($text, $positiveWords, $negativeWords)	Performs a sentiment detection on a given input string based on a lexigraphical analysis. The parameters $positiveWords and $negativeWords contain the polarity words (with the associated weight). Returns an object which contains the score (< 0 if negative, > 0 if positive, and 0 if neutral) and the color-highlighted text. Red indicates a negatively identified word and green a positively identified word.

4.4 Access to Twitter API

Twitter provides a convenient API for accessing all major features of the platform[5]. The API can be accessed via HTTP GET or POST calls and returns the data either in 'json' or 'xml' format.

The API provides methods for fetching information such as the public timeline, popular topics ('trends') and status updates, and also allows the user to send new Tweets or perform other actions like adding new friends. In order to use most of the features, a valid Twitter account is required. Twitter allows 100 calls per hour from one user, but this value can be increased to 20.000 by filling out a 'white-listing request'[6].

For example, if we want to fetch the account information for the user 'nytimes', we send a HTTP call to *http://twitter.com/users/show/nytimes.xml* and will receive following XML response:

Listing 4.1: XML response from Twitter's users/show method

```
<?xml version="1.0" encoding="UTF-8"?>
<user>
  <id>807095</id>
  <name>The New York Times</name>
  <screen_name>nytimes</screen_name>
  <location>New York, NY</location>
  <description>Where the Conversation Begins.  Home page stories
     from NYTimes.com, special features and more.</description>
  <profile_image_url>http://a3.twimg.com/profile_images/57465005/
     twitter_avatar.nyt_normal.jpg</profile_image_url>
```

[5]http://apiwiki.twitter.com/

[6]http://apiwiki.twitter.com/Rate-limiting

```
    <url>http://www.nytimes.com/</url>
    <protected>false</protected>
    <followers_count>2333631</followers_count>
     [...]
    <friends_count>193</friends_count>
    <created_at>Fri Mar 02 20:41:42 +0000 2007</created_at>
    <favourites_count>0</favourites_count>
    <utc_offset>-18000</utc_offset>
    <time_zone>Eastern Time (US &amp; Canada)</time_zone>
    <verified>false</verified>
    <following></following>
     [...]
    <statuses_count>125248</statuses_count>
    <lang>en</lang>
    <status>
      <created_at>Sun Feb 07 17:45:01 +0000 2010</created_at>
      <id>8772119841</id>
      <text>Editorial - &quot;The Truth About The Deficit&quot; -
          http://nyti.ms/bFFsB3 (via @nytimesopinion)</text>
      <source>&lt;a href=&quot;http://cotweet.com/?utm_source=sp1&quot
          ; rel=&quot;nofollow&quot;&gt;CoTweet&lt;/a&gt;</source>
      <truncated>false</truncated>
      <in_reply_to_status_id></in_reply_to_status_id>
      <in_reply_to_user_id></in_reply_to_user_id>
      <favorited>false</favorited>
      <in_reply_to_screen_name></in_reply_to_screen_name>
      <geo/>
      <contributors/>
    </status>
  </user>
```

As we can see, the complete account information is returned. This includes the name, the description and even the number of friends. If we now want to access this user's last Tweet, we can simply achieve this by calling 'http://twitter.com/statuses/user_timeline/nytimes.xml?count=1'.

Listing 4.2: XML response from Twitter's 'statuses/user_timeline' method

```
  <?xml version="1.0" encoding="UTF-8"?>
  <statuses type="array">
  <status>
    <created_at>Sun Feb 07 18:03:25 +0000 2010</created_at>
    <id>8772724298</id>
    <text>Is Google Running A Super Bowl Ad? http://bit.ly/ahKJIC #
        sb44 (via @nytimesbits, @nytimes5thdown)</text>
    <source>TweetDeck</source>
    <truncated>false</truncated>
    <in_reply_to_status_id></in_reply_to_status_id>
    <in_reply_to_user_id></in_reply_to_user_id>
    <favorited>false</favorited>
    <in_reply_to_screen_name></in_reply_to_screen_name>
    <user>
      <id>807095</id>
      <name>The New York Times</name>
      <screen_name>nytimes</screen_name>
      <location>New York, NY</location>
       [...]
       </user>
    <geo/>
    <contributors/>
  </status>
  </statuses>
```

As we can see in Listing 4.2, all available information for a single Tweet is returned. It contains, among others, the creation date, the text and the application used for creating this Tweet (‘source’).

Basically, those two calls are all we need in order to build both the frontend and the backend of the classifier. To access the data within our classfier, the PEAR package ‘Services_Twitter’ is used. It provides a simple class for the communication with the Twitter API.

Listing 4.3: XML response from Twitter’s ‘statuses/user_timeline’ method

```
$twitter = new Services_Twitter('api_username', 'api_password');
$user = $twitter->users->show('nytimes');
$statuses = $twitter->statuses->user_timeline(Array('id'=>$user->id,
     'count'=>1);

foreach($statuses as $entry) {
  $tweet = new Tweet();

}
```

The imported data (user metadata and associated Tweets) is then stored persistently to the MySQL database using Doctrine[7]. The usage is straight-forward - after defining the model in a YAML config file, a new object is created, filled with data and stored by calling the ‘save’ method. Doctrine handles all SQL-related issues (choose if UPDATE or INSERT is required, escaping quotes etc.). Listing 4.4 shows a code-snippet which creates a new object and saves it to the database:

Listing 4.4: Storing an object using Doctrine

[7]http://www.doctrine-project.org/

```
// $twitterApi contains the Tweet information loaded from Twitter
$tweet = new Tweet();
$tweet->text = $twitterApi->getText();
$tweet->created = $twitterApi->getCreationDate();
$tweet->userid = $twitterApi->getUserId();
$tweet->save();
```

4.5 Global dictionary

This section covers the creation of the global dictionary. The global dictionary is a term-document matrix (see Section 3.2.2) representing all features of our training set. This matrix is needed in order to create the vectors used by the SVM. The creation itself is rather simple: we iterate through all Tweets in the training set, extract the feature representation, and add it to an array. Once finished, this array is treated as the global dictionary that will be used by the test sets as well. For performance purposes, the global dictionary is cached using APCCache[8] and only re-created if the training set changes.

4.6 SVM Integration

This section covers the integration of a Support Vector Machine (SVM) into the application. Since PHP doesn't provide a native SVM library, we decided to use libSVM [CL01]. LibSVM is a software for support vector classification, regression and distribution estimation[9]. It offers interfaces to many programming languages and applications like Python, Java, Haskell, LISP, R and Matlab, but unfortunately, not to PHP.

To bypass this limitation, we used the Linux command-line program 'libSVM tools[10]' and manually created an interface to PHP. LibSVM Tools provides two separate programs, one for creating the SVM model file and one for performing the actual prediction.

The first program 'svm-train' requires an input file which contains the vector representation of the selected features. The input format is as follows:

<label> <index1>:<value1> <index2>:<value2> ...

<label> corresponds to the categories which are used. In our case, label is either 1 (=News), 2 (=User) or 3 (=Company) for C1, and 1 (=Factual) or 2 (=Opinionated)

[8]http://php.net/manual/en/book.apc.php

[9]http://www.csie.ntu.edu.tw/ cjlin/libsvm/

[10]http://www.csie.ntu.edu.tw/ cjlin/libsvmtools/

for C2 respectively. *<index1>:<value1>* is a pair, where <index1> conforms to the index of word j in the global dictionary. <value> is a boolean value, which is always set to 1.

An excerpt from the input file is listed in listing 4.5.

Listing 4.5: Sample input file for training the SVM

```
1 754:1 787:1 798:1 807:1 822:1 838:1 840:1 [...]
1 17484:1 17510:1 17520:1 17537:1 17540:1 [...]
2 7:1 16:1 28:1 29:1 30:1 31:1 32:1 33:1 34:1 [...]
2 117:1 118:1 119:1 120:1 121:1 122:1 123:1 [...]
3 265:1 1857:1 3534:1 3727:1 3819:1  5487:1 [...]
3 157:1 3819:1 4014:1 7109:1 8713:1 8754:1 [...]
```

Each line corresponds to a single Tweet. Only the words which are actually included in the Tweet are represented in the vector. libSVM allows to omit features which are set to zero.

Now, we can use the program 'svm-train' to create the SVM model file. 'svm-train' is included in the libSVM tools package. The created model file begins with following content:

Listing 4.6: Head of created model file

```
svm_type c_svc
kernel_type linear
nr_class 3
total_sv 1945
rho −0.00721537 −0.933927 −0.845183
label 2 1 3
nr_sv 705 754 486
```

The first 2 values indicate the type of the used SVM (linear C-CSV in this case). *nr_class* shows the number of used categories and *total_sv* the amount of used support vectors. *nr_sv* displays the used support vectors for each category: News has 754, User 705 and Companies 486 support vectors. *rho* is the bias term in the decision function.

In order to predict the category of a new Tweet, a test file needs to be created as well. It has the same layout as the training file, only the category is set to 0:

Listing 4.7: Sample test file for SVM

```
0 3350:1 3817:1 4034:1 4131:1 5122:1 [...]
0 23:1 31:1 86:1 114:1 149:1 166:1 177:1 [...]
0 15539:1 15585:1 15727:1 15761:1 15767:1 [...]
[...]
```

Now, 'svm-predict' can be used to predict the category of each vector in the test file: *svm-predict test.txt training.model output.txt*

It uses the created test and training file as input and writes the result to output.txt, which only contains the classification result for each vector:

Listing 4.8: Output file containing the classification results

```
1
2
1
3
[...]
```

Each line corresponds to the vector in the test file. In this case, the first vector (which is an actual Tweet) has been classified as News (1), while the second vector has been classified as User (2). It is an easy task to parse the output file to know how a certain Tweet has been classified and to use the results in the backend (unit testing) or in the frontend (classification task):

4.7 Word conversion

Before creating the vector representation, each Tweet is pre-processed. This includes word conversion (using regular expressions) and word stemming. The word conversion is accomplished with the following regular expressions:

Listing 4.9: Regular Expressions used to preprocess the Tweets

```
// #1, Replace @username with @
$text = preg_replace("/(@[a-zA-Z0-9]*[ :\.!])/", '@ ', $text);

// #2, Add whitespace character before and after ! ? : . " ; ,
$text = preg_replace("/([!\?\"\.;,])/", ' $1 ', $text);

// #3, Replace dollar values with variable ($14.99 => $XX)
$text = preg_replace("/(\\$ ?\d[\.\d\d]* )/", '$XXX ', $text);

// #4, Replace percentage values with variable (25\% => XX\%)
$text = preg_replace("/([0-9]{1,2}\% )/", 'XX\% ', $text);
```

In Twitter, it is possible to mention other Twitter users using the '@' character. For example, if a user wants to mention 'johndoe' in the Tweet, the Tweet could look like this: 'got some news for @johndoe, call me!'. Although this is a strong indicator for a user-to-user communication, we face the problem that the usernames will differ most likely. Therefore we are removing the username to make use of this indicator. In our example the Tweet will become ‘got some news for @, call me!’. The importance of this rule can be emphasized by the fact that ‘@’ is now the top-used feature in the user category (see Table 5.3 in Section 5.4).

Rule #2 adds an additional whitespace after the special characters *! ? : . ; , and ¨*. This is required as the features are created by separating the Tweets with the whitespace character. Let us use the example ‘got some news for @, call me!’ again. If this rule wouldn’t exist, a feature called *me!* would be created. The same is true for the ‘@’ character. Applying this rule results in ‘got some news for @ , call me !’. Now, instead of the feature ‘me!’, two features labeled ‘me’ and ‘!’ are created.

Rule #3 replaces all dollar values with ‘XX’. This is especially useful for advertisement. Almost every bargain advertisement contains a different dollar value. The rule now transforms the Tweet ‘Save $ 14.99 on Windows 7’ to ‘Save $XXX on Windows 7’.

Finally, Rule #4 acts like Rule #3, but transforms percentages into ‘XX%’. For example, the Tweet ‘Save 5% on this awesome deal’ will become ‘Save XX% on this awesome deal’.

4.8 Backend

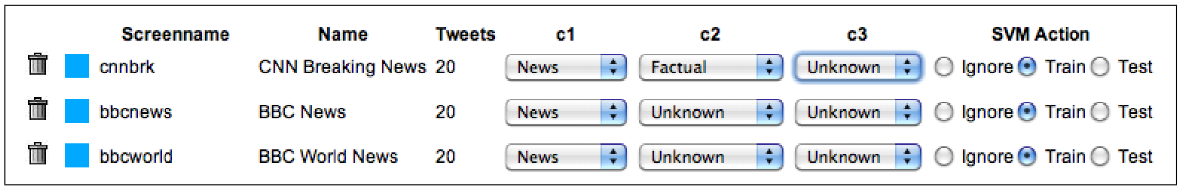
The backend consists of two main modules: the classifier module and the sentiment detection module. Please note that the backend still contains the category C3 (objective vs. subjective), though this category has been omitted (see Section 5.1.2 for details). The reason for this is that after the implementation of the the code we found out that this category correlates too strong with C2. Therefore we decided not to use it anymore.

4.8.1 Classifier module

The classifier module itself consists of two main parts: a GUI for setting up users, and a unit test facility to verify the performance of the classificator.

4.8.1.1 Setup users

Using this module, Twitter users can be added to act as the training (or test) set. Figure 4.4 shows a detail of the GUI for setting up users.



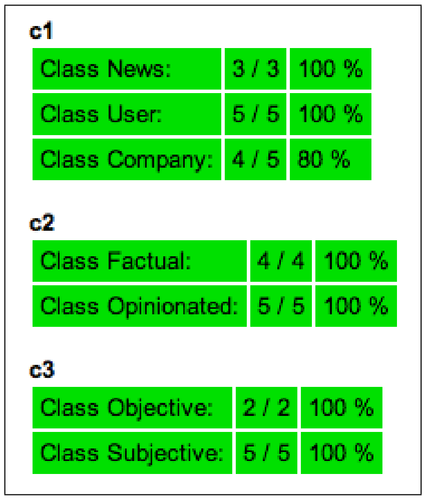
		Screenname	Name	Tweets	c1	c2	c3	SVM Action
🗑	🟦	cnnbrk	CNN Breaking News	20	News	Factual	Unknown	Ignore [✓ Train] Test
🗑	🟦	bbcnews	BBC News	20	News	Unknown	Unknown	Ignore [✓ Train] Test
🗑	🟦	bbcworld	BBC World News	20	News	Unknown	Unknown	Ignore [✓ Train] Test

Figure 4.4: Screenshot of 'Setup users' GUI

Looking from left to right, the first column shows an icon for deleting the user. Column 2 is a colored square, which corresponds to the selected category for that user (blue = News). The only reason for this field is to allow the maintainer to quickly identify the category of this user (usability). Column 3 displays the Twitter username, followed by the Twitter's real name in column 4. In columns 5, 6 and 7, the 3 categories of the user can be selected (for example, c1 consists of the 3 main categories 'News', 'User' and 'Company'). And last but not least, columns 9 - 11 allow the maintainer to select the SVM actions. If the action is set to 'train', all Tweets of that user will be used in the training set. If it is set to 'test', the Tweets will not be used in the training, but in the test set. 'Ignore' will neither add them to the training set, nor to the test set.

4.8.1.2 Run Unit tests

This module allows to test the set of added users. Based on the training and test set, it displays how many of the test examples have passed the test (= have been classified correctly). Figure 4.5 shows the overall classification result.



c1

Class News:	3 / 3	100 %
Class User:	5 / 5	100 %
Class Company:	4 / 5	80 %

c2

Class Factual:	4 / 4	100 %
Class Opinionated:	5 / 5	100 %

c3

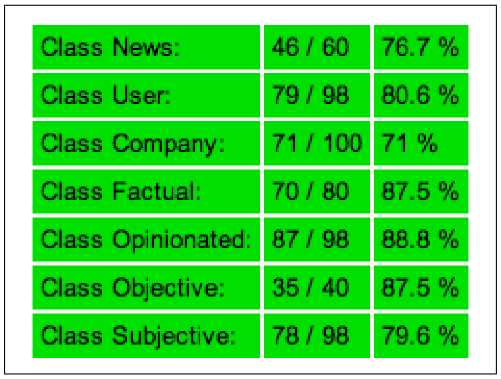
Class Objective:	2 / 2	100 %
Class Subjective:	5 / 5	100 %

Figure 4.5: Unit test, overall user classification

A test user has passed the test if the predicted category corresponds to the category the maintainer has set for this user. The actual category is determined by classifying the single Tweets of the user, and subsequently calculating which category has been predicted the most for those Tweets.

Let's clarify this with a simple example: Consider three categories (A, B and C). User 1 has 10 Tweets, of which 6 have been classified with category A, 3 with category B and 1 with category C. The user would be qualified as member of category A, as this category has the most Tweets. (6 / 10 Tweets).

As we can see in figure 4.5, 3 of 3 test users in category 'News' have been predicted correctly (hence 100%). For the category 'Company', the classifier predicted only 4 of 5 users correctly.



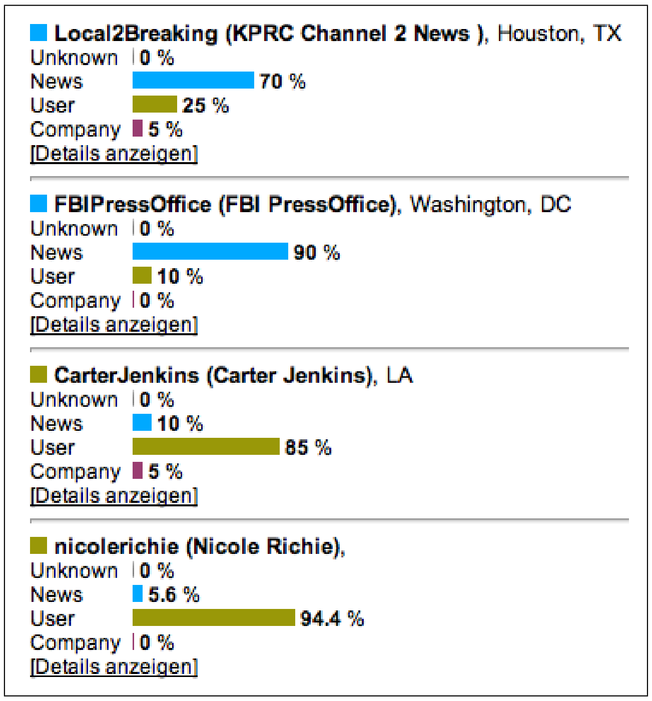
Class News:	46 / 60	76.7 %
Class User:	79 / 98	80.6 %
Class Company:	71 / 100	71 %
Class Factual:	70 / 80	87.5 %
Class Opinionated:	87 / 98	88.8 %
Class Objective:	35 / 40	87.5 %
Class Subjective:	78 / 98	79.6 %

Figure 4.6: Unit test, single Tweet classification

Figure 4.6 shows the single Tweet classification result. If a user is added to the test set, all of his/her Tweets are considered to be part of this category. Of course, this is not always the case. For example, a user could Tweet some 'breaking news', which would be classified as 'News'.

The screen shows all categories, the amount of correctly predicted Tweets and the corresponding percentage. For example, 46 of 60 Tweets of the category 'News' have been predicted correctly, which is 76.6 percent.

The next figure (4.7) shows the detailed user classification result for the first category ('user', 'news', 'company'). As we can see in this figure, 70% of the Tweets from the user 'KPRC Channel 2 News' have been classified as 'News'.



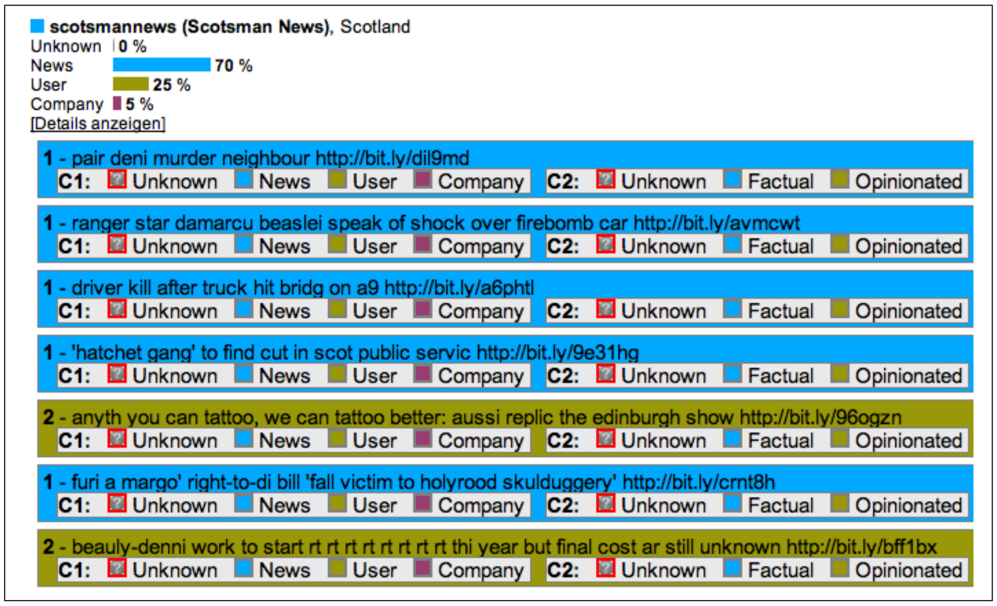
Local2Breaking (KPRC Channel 2 News), Houston, TX
Unknown 0 %
News 70 %
User 25 %
Company 5 %
[Details anzeigen]

FBIPressOffice (FBI PressOffice), Washington, DC
Unknown 0 %
News 90 %
User 10 %
Company 0 %
[Details anzeigen]

CarterJenkins (Carter Jenkins), LA
Unknown 0 %
News 10 %
User 85 %
Company 5 %
[Details anzeigen]

nicolerichie (Nicole Richie),
Unknown 0 %
News 5.6 %
User 94.4 %
Company 0 %
[Details anzeigen]

Figure 4.7: Unit test, detailed user classification result



scotsmannews (Scotsman News), Scotland
Unknown 0 %
News 70 %
User 25 %
Company 5 %
[Details anzeigen]

1 - pair deni murder neighbour http://bit.ly/dil9md
C1: Unknown News User Company C2: Unknown Factual Opinionated

1 - ranger star damarcu beaslei speak of shock over firebomb car http://bit.ly/avmcwt
C1: Unknown News User Company C2: Unknown Factual Opinionated

1 - driver kill after truck hit bridg on a9 http://bit.ly/a6phtl
C1: Unknown News User Company C2: Unknown Factual Opinionated

1 - 'hatchet gang' to find cut in scot public servic http://bit.ly/9e31hg
C1: Unknown News User Company C2: Unknown Factual Opinionated

2 - anyth you can tattoo, we can tattoo better: aussi replic the edinburgh show http://bit.ly/96ogzn
C1: Unknown News User Company C2: Unknown Factual Opinionated

1 - furi a margo' right-to-di bill 'fall victim to holyrood skulduggery' http://bit.ly/crnt8h
C1: Unknown News User Company C2: Unknown Factual Opinionated

2 - beauly-denni work to start rt rt rt rt rt rt rt rt thi year but final cost ar still unknown http://bit.ly/bff1bx
C1: Unknown News User Company C2: Unknown Factual Opinionated

Figure 4.8: Unit test, detailed Tweet classification result

Figure 4.8 shows the detailed Tweet classification result. The first four Tweets of the user 'Scotsmen News' have been classified as 'News', and the 5th Tweet as 'User'. Please note that those Tweets have already been pre-processed (word conversion and word stemming). The backend also allows the user to manually re-classify single Tweets. Those Tweets are then added to the training set as well and will be classified correctly in the next run with a probability bordering on certainty.

4.8.1.3 Sequence diagram

Figure 4.9 shows the sequence diagram of the maintenance process. Basically, the maintainer adds several users to the backend, runs the unit tests and refines the training/test sets until the classifier produces satisfying results.

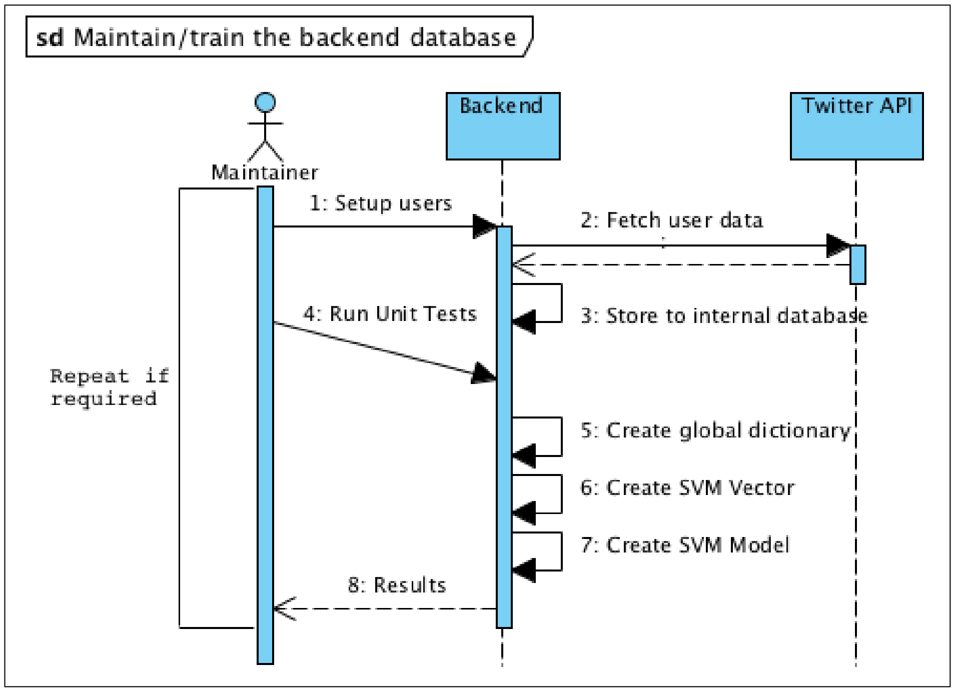


Figure 4.9: UML Sequence diagram of backend training

1. Maintainer adds several users to the backend and assigns them to the correct category. He/she also assigns them to either the training- or the test set

2. Backend loads all associated Tweets from the Twitter site

3. Tweets are stored in the internal MySQL database

4. Maintainer start now the unit test to test how accurate the classifier works with the provided training- and test sets

5. Now, the global dictionary is created using a term-document matrix).

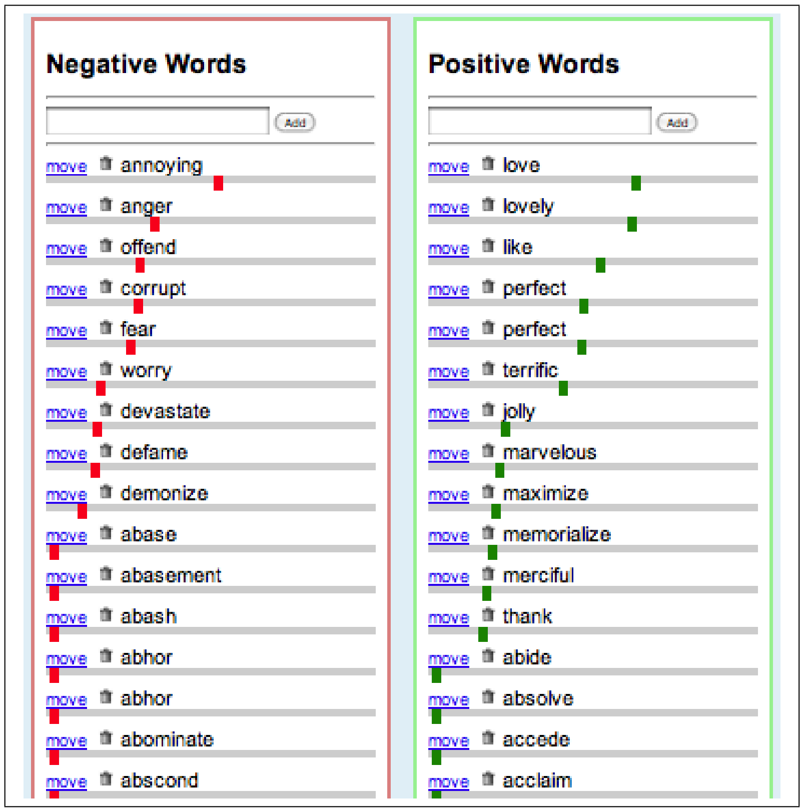
6. The backend now creates the SVM vectors for both the training and the testing set

7. Based on the SVM vectors, the SVM model file is created (using 'svm-create')

8. The results are displayed to the maintainer. He/she now decides to refine the training- and test set until the results are satisfying

4.8.2 Sentiment detection module

The second part of the backend is the sentiment detection module. Since we are using the scoring approach (see Section 5.6), a facility for adding and maintaining positive and negative words is needed. Figure 4.10 shows a screenshot of the actual implementation.



Negative Words	Positive Words
annoying	love
anger	lovely
offend	like
corrupt	perfect
fear	perfect
worry	terrific
devastate	jolly
defame	marvelous
demonize	maximize
abase	memorialize
abasement	merciful
abash	thank
abhor	abide
abhor	absolve
abominate	accede
abscond	acclaim

Figure 4.10: Sentiment module

Negative words can be added on the left-hand side, positive words on the right-hand side. It is also possible to delete words, or move them to the other side. One of the main features is the ability to weight certain words. For example, the word 'love' is weighted higher than 'like'. The weighting can be accomplished by moving the slider - the farther right, the higher is the rating. The slider itself is implemented using the Scriptaculous Slider Library[11]. Upon release, it sends a HTTP request

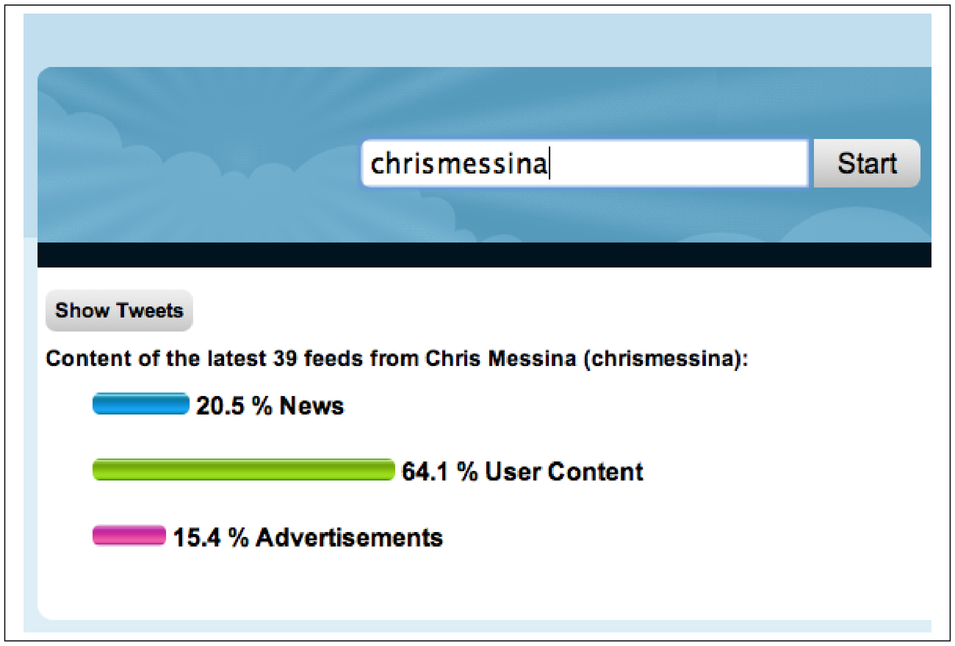
[11]http://wiki.github.com/madrobby/scriptaculous/slider

(AJAX) to the backend which stores the new weight.

4.9 Frontend

The frontend offers a convenient interface to classify a given user. There is just a single input field where the screenname of the user can be entered. Once submitting the field, the application fetches the latest Tweets from Twitter and analyzes them. Figure 4.11 shows a screenshot of the frontend.

Figure 4.11: Frontend - Main screen



Content of the latest 39 feeds from Chris Messina (chrismessina):

20.5 % News

64.1 % User Content

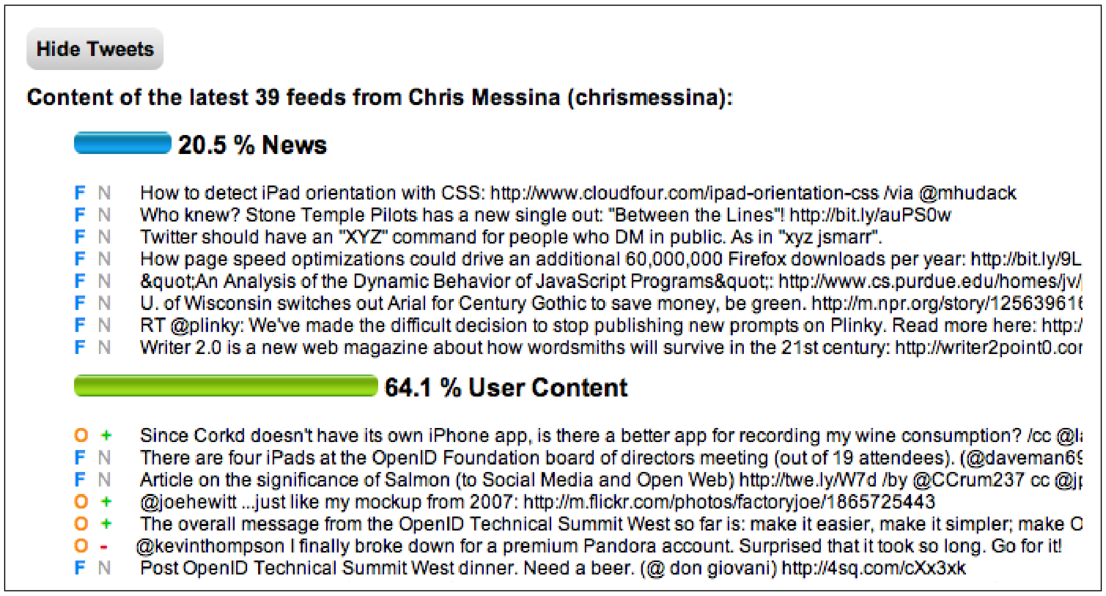
15.4 % Advertisements

The three bars displayed in Figure 4.11 show the classification result for the category C1. In this case, 20.5% of Chris Messina's[12] Tweets are classified as News, 64.1% as user content and 15.4% as advertisements.

A further click on 'Show Tweets' reveals the details (Figure 4.12). The first column indicates if a certain Tweet has been classified as factual (F) or opinionated (O). Then, the sentiment is displayed (+, - and N for neutral), followed by the actual Tweet.

[12]Chris Messina is a open web advocate who is considered to be the inventor of the Twitter hashtags

Figure 4.12: Frontend - Details


Hide Tweets

Content of the latest 39 feeds from Chris Messina (chrismessina):

20.5 % News

F N How to detect iPad orientation with CSS: http://www.cloudfour.com/ipad-orientation-css /via @mhudack
F N Who knew? Stone Temple Pilots has a new single out: "Between the Lines"! http://bit.ly/auPS0w
F N Twitter should have an "XYZ" command for people who DM in public. As in "xyz jsmarr".
F N How page speed optimizations could drive an additional 60,000,000 Firefox downloads per year: http://bit.ly/9L
F N &quot;An Analysis of the Dynamic Behavior of JavaScript Programs&quot;: http://www.cs.purdue.edu/homes/jv/
F N U. of Wisconsin switches out Arial for Century Gothic to save money, be green. http://m.npr.org/story/12563961
F N RT @plinky: We've made the difficult decision to stop publishing new prompts on Plinky. Read more here: http:/
F N Writer 2.0 is a new web magazine about how wordsmiths will survive in the 21st century: http://writer2point0.co

64.1 % User Content

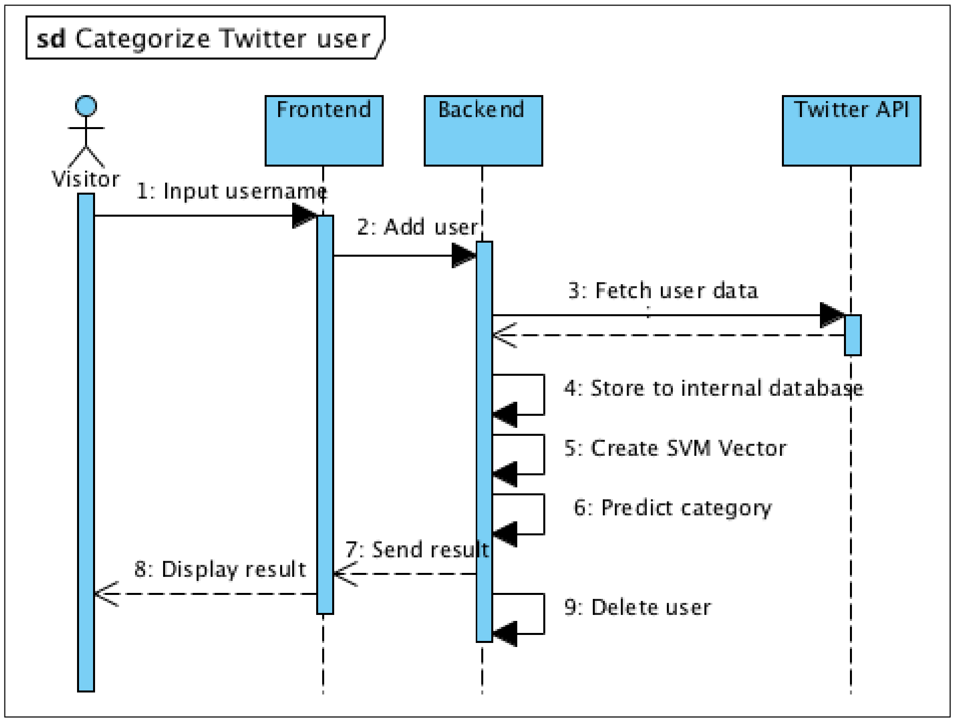
O + Since Corkd doesn't have its own iPhone app, is there a better app for recording my wine consumption? /cc @l
F N There are four iPads at the OpenID Foundation board of directors meeting (out of 19 attendees). (@daveman6
F N Article on the significance of Salmon (to Social Media and Open Web) http://twe.ly/W7d /by @CCrum237 cc @j
O + @joehewitt ...just like my mockup from 2007: http://m.flickr.com/photos/factoryjoe/1865725443
O + The overall message from the OpenID Technical Summit West so far is: make it easier, make it simpler; make C
O - @kevinthompson I finally broke down for a premium Pandora account. Surprised that it took so long. Go for it!
F N Post OpenID Technical Summit West dinner. Need a beer. (@ don giovani) http://4sq.com/cXx3xk

The interface itself makes heavy usage of dynamic JavaScript. Once the visitor enters a username, the application sends a HTTP request in the background to the classifier, while displaying a ‘loading’ icon. Once the classifier is done, it sends the response to the client, which replaces the icon with the actual results.

4.9.1 Sequence diagram

Figure 4.13 shows the sequence diagram of the classification process for frontend users.

Figure 4.13: UML Sequence diagram of frontend training



1. The visitor enters a Twitter user's name into the input field
2. The frontend adds the username to the backend
3. The backend tries to load the Tweets from Twitter. If the user's data cannon be loaded (for example, if the username is invalid or if the user has protected his/her Tweets), an error message is raised
4. Then, the Tweets are copied to the internal database
5. The system creates the SVM vectors based on the global dictionary
6. For every Tweet, the category is predicted
7. The results are sent back to the frontend
8. The frontend displays the result to the user
9. The backend deletes the newly created users and all associated Tweets again

Chapter 5

Evaluation

5.1 Twitter User classification model

This section describes how the Tweets have been categorized. There are 2 main categories: C1 and C2. The first one contains the facets 'News', 'User' and 'Company', and the latter one 'Factual' and 'Opinionated'.

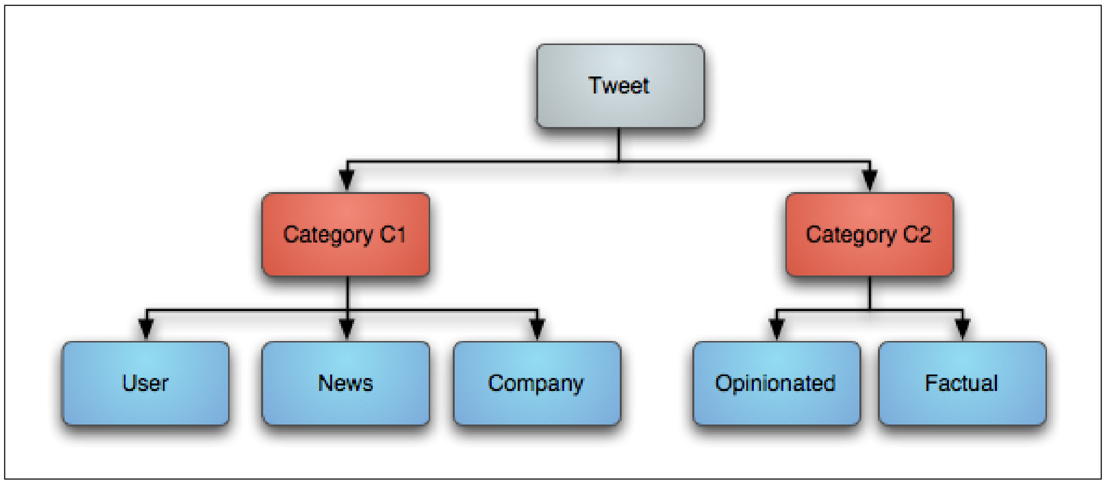


Figure 5.1: UML diagram

5.1.1 C1 - News, User, Companies

The first category distinguishes between 'News', 'User' and 'Company'. 'News' contains all Tweets which deal with all kind of news. This can be world news, sport results or other current events. 'User' are Tweets which deal with personal information. For example, when someone writes about his/her daily activities or user-to-user communication. For the 'Company' category, the focus was put on company advertisements. This includes classical spam accounts ('make money fast') as well as 'sponsored tweets', which gained popularity over the last couple of months. Sponsored tweets are company advertisements which show up on Twitter pages of ordinary users. Of course, those users need to sign up for the program using a special advertisement platform (i.e. 'sponsoredtweets.com' and 'ad.ly'). Such a platform acts as a link between the advertiser and the user by placing the advertisement on the user's Twitter page. While it can be doubted that anybody actually gets rich with this system (besides the advertisement platform), it might be interesting to know how many of today's tweets are 'sponsored tweets'.

5.1.2 C2 - Factual, Opinionated

The second distinction is 'factual' vs. 'opinionated'. The idea for this two facets is taken from the Faceted Blog Distillation Task at the TREC 2009 conference[1].

A Tweet is considered to be 'factual', if it solely contains facts and 'opinionated' if it contains opinions. But let us define 'facts' and 'opinions' first.

According to WordNet[2], a **fact** is

- *a piece of information about circumstances that exist or events that have occurred*
- *a statement or assertion of verified information about something that is the case or has happened*
- *an event known to have happened or something known to have existed*
- *a concept whose truth can be proved*

An **opinion** is defined as

- *sentiment, persuasion, view, thought (a personal belief or judgment that is not founded on proof or certainty)*
- *view (a message expressing a belief about something; the expression of a belief that is held with confidence but not substantiated by positive knowledge or proof)*
- *a belief or sentiment shared by most people; the voice of the people (vox populi)*
- *impression, feeling, belief, notion, opinion (a vague idea in which some confidence is placed)*

[1]http://ir.dcs.gla.ac.uk/wiki/TREC-BLOG
[2]http://wordnetweb.princeton.edu

It is obvious that users of the category 'News' mostly write about facts, while this generalization does not hold for the 'User' and 'Company' categories. We can bypass this problem by rating each tweet individually in the latter two categories. But this leads to the next problem: Sometimes facts and opinions are mixed up in a single tweet. Let us illustrate this with some examples. All examples are taken from the evaluation data used in section 5.3.1.

'Potential jury candidate fell asleep in the gallery. selection is slow...'. That someone fell asleep is undoubtably a fact. But is 'selection is slow' a fact as well, or more an opinion of that user? Can a 'slow selection' be defined?

'I have to work today. Ten hours of overtime, but not worth it.'. That's a good example of a fact ('have to work today'), followed by an opinion ('not worth it'). How should this tweet be categorized?

'I'm working in the med room today. We'll see how that goes...'. The first part is a fact, followed by an opinion.

'My phone just had some dust under the screen that was beginning to annoy me.'. Another example of a fact followed by an opinion.

Another issue is that Tweets look like facts, but can be opinions as well:

'At&t edge is way faster in Socal than in the bay area.'. While this sentence looks like a fact, it could also be an opinion of that user. It depends if the user actually performed tests which prove that the connection is faster.

'Jimmy Wales is smaller than I pictured him.' While this sentence looks like a fact, it's actually an opinion.

Also, how should quotations be classified? Technically the Tweet *'The man who does not read good books has no advantage over the man who can't read them, Mark Twain once said'* is definitely a fact, but wouldn't it be better to classify it as an opinion?
The following examples show that the distinction betweed factual and opinionated is not easy at all.

'yup - I was the king :)'

'I'm having a series of really shitty days all in a row.'

'Ewwww. I turn 22 on my next birthday.'

'Watching Charlie, Rejected, Banana Phone, What What, Magical Trevor, and more with Brittany, Marshall, and Mike. Life is awesome.'

'Listening to Dr. Dre - The Next Episode. Love this song!!'

'I can't feel my arms!!!'

'I've been with the Rotary people in India during Polio immunization days and they do incredible work.'

'I just saw something i could never imagine. a man in siberia who has me tatooed on his arm'

'The Bureaucrat of the Rings #boringmovies'

As you can see, it's not always obvious if a certain Tweet is either a fact or an opinion, sometimes there is room for interpretation.

Also, we wanted to distinct between 'objective' and 'subjective'. Let us again define the words using WordNet [3]:
objectivity is defined as: *judgment based on observable phenomena and uninfluenced by emotions or personal prejudices*
subjectivity is defined as: *judgment based on individual personal impressions and feelings and opinions rather than external facts*

Obviously, objectivity seems to correspond strongly with 'factual' and subjectivity with 'opinionated'. After manually labeling a few dozen Tweets we found out that the correlation is too strong - Tweets labeled as opinions are most likely subjective as well. Also vice versa - factual Tweets are always objective, otherwise they wouldn't be factual. Taking this into consideration, we decided to drop this distinction.

5.2 Hardware setup

All tests were carried out on a virtual XEN instance[4] running on a Linux box. The technical specifications are as displayed in Table 5.1.

[3]http://wordnetweb.princeton.edu
[4]http://www.xen.org/

Table 5.1: Server setup

CPU	Intel i7 Core 920 @ 2.67GHz (one core dedicated to this instance)
Memory	2GB RAM (12 GB RAM in total)
Linux Version	2.6.26-2-xen-amd64 #1 SMP
Debian Version	5.0.3
XEN Version	3.2.1-2
LibSVM Version	2.85.0-1 (Debian package)
Apache2 Version	2.2.9-10+lenny6
PHP5 Version	5.2.6.dfsg.1-1+lenny4

5.3 Data set

5.3.1 Data set construction

After designing and implementing the classifier, it needs to be evaluated. The main task of this evaluation is to setup the user database. It sounds simple, but this is the crucial step in order to obtain good results. As mentioned earlier, we have split the Twitter users into three categories (facets): ‘News’, ‘Users’ and ‘Companies’. We now need to choose a set of Twitter users for each category. Let us answer this basic question first: How do we select representative members for each category?

In ‘Categorization of Natural Objects’ [MR81], Mervis and Rosch introduced six salient problems of categorization. Two of them affect our own categorization. First, are all category members equally representative for this category? For our selection, this means that we must only choose equally representative Twitter users for each category, and need to avoid over-representative and under-representative users. The second problem is the ‘determinacy of category membership and representation’. This basically comes down to the question if the category boundaries are well defined or not. In our case, we need to choose Twitter users which only have Tweets belonging to their very own category and do not trespass category boundaries. For example, if a certain Twitter user deals with world news half the time, he is not considered to be an optimal representative for the category ‘User’ as the tweets do not stay inside the ‘User’ category.

For each category 40 representative users have been selected. And for every user the latest 40 tweets were imported. This resulted in approximately 1.200 tweets per category[5]. On average, each tweet contained 14 words or 95 characters.

[5]Not all users had exactly 40 Tweets, but all had more than 35

The next four sub-sections describe how the users have been selected.

5.3.1.1 C1 - Choosing News

The first step was to add the most well-known news coperations, like BBC, CBS, CNN, NY Times, Reuters News and Wall Street Journal. After that, the web was searched for newspapers in major world cities (e.g. Tokio, Shanghai, Mumbai, Moscow, Istanbul). Most of those newspapers have a Twitter account where current events are posted. Those accounts have been added to the database as well. The last step was filling up the list with accounts not associated with newspapers or news cooperations. We achieved that by searching for the term 'Breaking News' in the Twitter search panel, this revealed several useful accounts (like 'Paul Fisher', 'Elizabeth Benjamin' and 'Steve Wagner').

The complete list of accounts is listed in table A.1.

5.3.1.2 C1 - Choosing Users

The aim was to find accounts that represent the 'ordinary user' as best as possible. But what defines such users? In this case, a user is considered to be 'ordinary', if he/she the fulfills following requirements:

- Writes about his/her exciting daily activities ('I went shopping', 'I'm watching TV')
- Communicates a lot with other users ('RT', '@xyz')
- Is an active Twitter user
- Has a reasonable amount of followers and friends (more than 50)
- Writes in correct English language (no slang)
- Does neither write about current events , nor uses sponsored Tweets

As there are plenty of users which fulfill those requirements, we started by choosing some of the most well-known celebrity accounts, i.e. Bill Gates, Ashton Kutcher, Jessica Alba and Paulo Coelho. After that, we scanned the list of followers and chose random accounts. If those chosen accounts fulfilled the requirements, they were added to the backend. The last step was to browse through the 'Trending Topics' on the Twitter page and chose random accounts from that list.

The complete list of used accounts is listed in Table A.2.

5.3.1.3 C1 - Choosing Companies

As said in Section 5.1.1, the focus for this category is put on company advertisement (sponsored Tweets). To identify the accounts, the twitter site was searched for Tweets which contain terms like 'make money' and 'increase your'. On the other hand, accounts using sponsored Tweets can be easily identified by searching, among others, for terms like '#ad' , '#deal' or 'sponsored'. Another way of identifying those accounts is to go to the advertisement companies' website and browse through the list of signed-up users. Of course, we must pay attention that those accounts are solely 'sponsored', hence not tweeting about anything else. This refers to the second categorization problem mentioned earlier ('determinacy of category membership and representation').

Table A.3 shows the complete list of accounts used for the category 'Company'.

5.3.1.4 C2 - Choosing Factual and Opinionated Tweets

After adding the users to the database and assigning them to one of the three main categories, the next step is to determine if the users either write about facts ('factual') or write opinionated. As discussed in Section 5.1.2, this task is not straight-forward at all. While all Tweets in the 'News' category can be rated as factual, we have to differentiate within the 'User' category. The first thought might be that the majority of those Tweets are opinionated, but still, there might be a lot of factual tweets as well. Let us quickly summarize the categorization problems for this category as listed in Section 5.1.2. There might be Tweets where facts are followed by opinions (and vice-versa), Tweets that use quotations and Tweets that contain questions.

5.3.1.4.1 Facts and opinions mixed up When facts and opinions are mixed up in a single Tweet, it is the **statement** of the whole Tweet which sets the category. Consider the Tweet *'Potential jury candidate fell asleep in the gallery. selection is slow...'*. Although the first part is a fact, the whole Tweet is considered to be an opinion. The user wants to notify the world that the jury selection process is too slow (which is an opinion). On the other hand, the Tweet *'NatWest only have a 2003 version of Excel, jokers!'* is treated as a fact, although the word 'jokers' indicates that the user is not happy with that version (hence it might be an opinion as well). But the statement of the tweet is that there is only the 2003 version of Excel installed.

5.3.1.4.2 Quotations Quotations are treated as opinions. While technically the tweet *'The man who does not read good books has no advantage over the man who* *can't read them, Mark Twain once said'* is a fact, we consider it to be an opinion. It is the opinion of Mark Twain, and it is safe to say that the user agrees with that opinion - otherwise he/she would not have posted this quotation.

5.3.1.4.3 Questions Another problem arises at the classification of questions. Questions are neither facts nor opinions per se. We deal with this issue by classifying the possible answer to this question, which may be a fact or an opinion. For example, an answer to the question *'How are you today?'* might be *'I'm fine, thanks'*, and that is an opinion. On the other hand, a possible answer to the question *'What was the score of the game yesterday?'* is *'Bayern won by 3 to 1.'*, which is a fact.

5.4 Word analysis

Let us start the evaluation by analyzing the Tweets used in each of the three categories of C1. Table 5.2 shows the Tweet analysis, and Table 5.3 the top-30 words of the category C1.

Table 5.2: Tweet analysis of category C1

	News	User	Company
Number of Tweets	1,636	1,539	1,560
Number of distinct words	7,378	5,277	6,035
Average amount of chars per Tweet	98	79	105
Average time between 2 Tweets	1h 20min	13h 0min	1h 37min

Table 5.3: Top-30 used words in each category

News		**User**		**Company**	
count	word	count	word	count	word
470	in	804	@	814	$XXX
456	to	500	the	749	-
396	of	462	to	323	to
360	the	449	I	278	and
235	for	388	a	217	on
219	a	295	and	213	SAVE
207	on	263	you	200	#deal
166	and	255	in	200	for
121	-	226	of	186	a
114	at	213	my	167	at

Table 5.3: Top-50 used words in each category

News		User		Company	
count	word	count	word	count	word
106	The	208	is	154	the
94	with	180	for	143	you
91	is	168	on	134	from
78	US	161	it	130	money
69	says	125	at	122	XX%
68	as	122	-	120	with
65	from	122	with	119	of
65	has	118	that	117	your
62	after	118	me	111	#ad
59	A	105	RT	108	Free
55	his	102	be	102	Make
53	over	91	have	90	in
52	by	88	i	89	Twitter
49	Tiger	88	are	88	Save
49	News	88	I'm	87	Your
45	will	83	not	80	The
43	times	80	so	79	#Deal
43	not	80	this	73	Money
42	was	78	was	73	free
42	BREAKING	77	like	71	make

Table 5.4 shows the Tweet analysis for the category C2, followed by the top-30 words for that category in Table 5.5. As one can imagine, possesive and personal pronouns ('I', 'you', 'my', 'me') are among the top words in the the class of of opinionated Tweets.

Table 5.4: Tweet analysis of category C2

	Factual	Opinionated
Number of Tweets	2892	1843
Number of distinct words	11856	5710
Average amount of chars per Tweet	82	79
Average time between 2 Tweets	1h 25min	12h 17min

Table 5.5: Top-30 used words in each category

Factual		**Opinionated**	
count	word	count	word
138	the	446	i
137	to	412	the
100	in	330	to
88	a	311	a
86	i	235	you
83	and	225	and
75	my	174	my
65	of	173	is
59	on	162	in
53	-	159	of
49	at	141	for
44	for	114	it
37	with	114	on
32	is	99	that
28	rt	87	with
28	you	85	at
27	from	83	me
25	i'm	80	have
24	have	79	i'm
24	just	78	be
22	are	77	not
21	it	76	so
20	this	76	just
20	be	74	like
19	that	72	rt
19	we	70	was
18	out	69	are
18	&	68	this
16	new	67	-

5.5 k-fold cross validation

Once the users had been set up, the classifier was evaluated with a k-fold cross validation. For k, a value of 5 was chosen. Given 40 records per category, this resulted in 8 runs.

For the first run, the first 5 records of each category where used as test set, and the remaining 35 records as training set. For the second run, the records 5-10 of each category where used as test set, and the records 1-5 and 11-40 as training set.

5.5.1 Cross validation of C1

Table 5.6 shows the percentage of correctly predicted Tweets in each category. For example, a value of 88.90 means that 88.9% of the Tweets in this category have been classified as being a member of this category, hence classified correctly. The other 11.1% have been classified with the wrong category.

Table 5.6: 8-fold cross validation of category C1. Values show the percent of correctly predicted Tweets in each fold.

k	News	User	Company
1	75.50	83.51	85.19
2	90.45	67.96	76.00
3	83.50	83.76	72.50
4	86.00	83.77	35.50
5	69.04	89.39	95.50
6	77.00	85.94	88.00
7	87.00	83.42	76.50
8	81.50	82.35	79.00

Those values have been analyzed statistically using the 'R' program[6]. The results are shown in Table 5.7. It can be said that the classification of all three categories worked remarkably good. The mean values are 81.25% (News), 82.51% (User) and 76.02% (Company). The second observation is that 'News' and 'User' got a lower scatter than Tweets of the class 'Company'. (69.04% to 90.45% for 'News', 67.96% to 89.39% for 'Users', and 35.50% to 95.50% for 'Companies'). After taking a detailed look at fold 4 in which only 35.50% of the Company Tweets have been classified correctly, it can be said that this bad result was mainly caused by a single representative of the test set in class 'Company'. All the Tweets of this user started with '@username', and besides that, the text was difficult to identify as a 'Company' Tweet (example: '@afbfreelancers buy fantastic products ranging from CDs to cloths from phones to kitchen items on this site http://www.kellyfonky.com'). With hindsight, it might have been better not to choose this user as a 'Company' representative.

[6]http://www.r-project.org/

Table 5.7: Statistical analysis of cross validation for category C1

News		User		Company	
Min.	:69.04	Min.	:67.96	Min.	:35.50
1st Qu.	:76.62	1st Qu.	:83.15	1st Qu.	:75.12
Median	:82.50	Median	:83.63	Median	:77.75
Mean	:81.25	Mean	:82.51	Mean	:76.02
3rd Qu.	:86.25	3rd Qu.	:84.31	3rd Qu.	:85.89
Max.	:90.45	Max.	:89.39	Max.	:95.50

Figure 5.2 visualizes the result using a box plot graph.

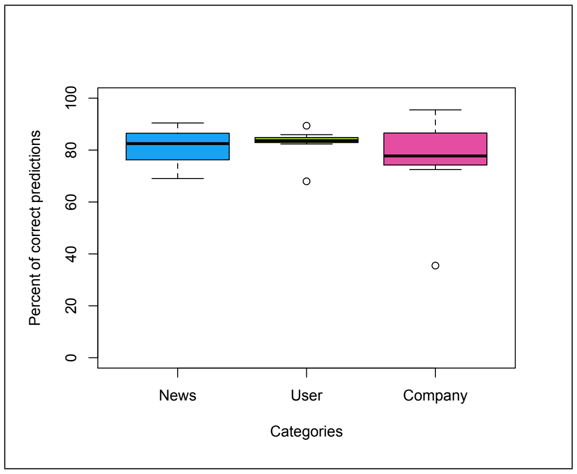


Figure 5.2: Boxplot of cross validation results for category C1

5.5.2 Cross validation of C2

The cross validation was also performed for the second category (Factual vs. Opinionated), using the same data and the same value for k. Table 5.8 shows the detailed result, and Table 5.9 the statistical analysis.

Table 5.8: 8-fold cross validation of category C2. Values show the percent of correctly predicted Tweets in each fold.

k	Factual	Opinionated
1	80.00	66.46
2	84.29	61.72
3	88.40	62.50
4	85.12	60.96
5	91.01	83.85
6	86.44	64.25
7	90.86	70.62
8	91.90	57.29

Table 5.9: Statistical analysis of cross validation for category C2

Factual		Opinionated	
Min.	:80.00	Min.	:57.29
1st Qu.	:84.92	1st Qu.	:61.53
Median	:87.42	Median	:63.37
Mean	:87.25	Mean	:65.96
3rd Qu.	:90.90	3rd Qu.	:67.50
Max.	:91.90	Max.	:83.85

It can be said the the classification of the class 'Factual' worked better than the classification of the class 'Opinionated', indicated by the mean value of 86.58% vs. a mean value of 67.30%. This can most likely be explained by the fact that all Tweets of the class 'News' have been assigned to the 'Factual' class, hence those Tweets have been classified correctly with a high propability. On the other hand, the Tweets in the other two main classes ('User' and 'Company') had to be assigned manually to one of the two C2 categories, which left a reasonable amount of room for interpretation (see Section 5.1.2).

Figure 5.3 shows the box plot diagram of the results.

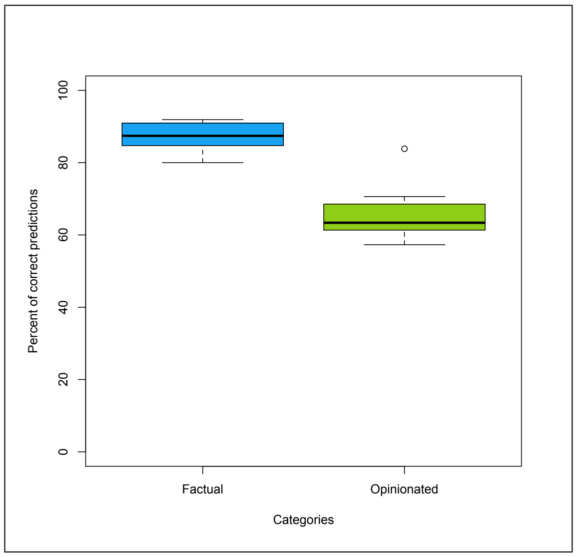


Figure 5.3: Boxplot of cross validation results for category C2

5.5.3 Confusion matrix of C1 results

The classification results from the previous section have now been analyzed using a confusion matrix (Table 5.10 and Figure 5.4). A confusion matrix is a visualization method used in machine learning, typically when the number of categories exceeds 2. Given an amount of possible categories, the rows display the predicted category, while the columns display the actual categories. Thus, it is possible to check if the classifier is confusing two categories (hence the name 'confusion matrix').

Table 5.10: Confusion matrix of C1 results

	Predicted		
	News	User	Company
News	1297	190	109
User	173	1272	94
Company	172	210	1207
Actual			

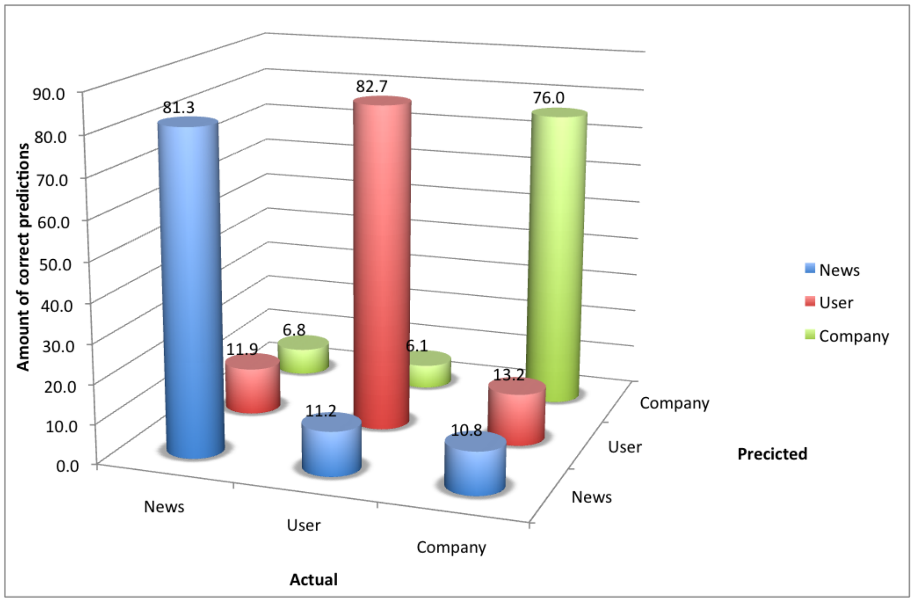


Figure 5.4: Confusion Matrix of C1

Looking at the results, it can be said that there is a stronger correlation between News and User than between News / Company and User / Company respectively. News agencies have been mis-classified as Users almost twice as often (190 vs. 109) - and also vice-versa: Users have been mis-classified as News agencies twice as often as well (173 vs. 94). On the other hand, the mis-classifications of Companies balances between News (172) and Users (210).

5.5.4 Impact of word conversion and word stemming

Now, we want to measure the impact of word conversion and word stemming to the classification results. To do so, the k-folded cross validation tests have been repeated with the following setting:

1. Word conversion / No word stemming (encoded as 10)
2. No Word conversion / Word stemming (encoded as 01)
3. No Word conversion / No word stemming (encoded as 00)

5.5.4.1 Word conversion / No Stemming

Table 5.11 shows the result of the k-fold cross validation using text word conversion, but no word stemming.

Table 5.11: Statistical analysis of cross validation, word conversion, no stemming

News		User		Company	
Min.	:73.60	Min.	:71.82	Min.	:45.50
1st Qu.	:81.00	1st Qu.	:84.51	1st Qu.	:66.00
Median	:81.50	Median	:86.32	Median	:68.00
Mean	:82.88	Mean	:85.59	Mean	:72.09
3rd Qu.	:84.38	3rd Qu.	:88.83	3rd Qu.	:85.51
Max.	:94.97	Max.	:92.42	Max.	:92.50

5.5.4.2 No Word conversion / Stemming

Table 5.12 shows the result of the k-fold cross validation using no word conversion, but word stemming.

Table 5.12: Statistical analysis of cross validation, no word conversion, stemming

News		User		Company	
Min.	:63.45	Min.	:64.09	Min.	:50.50
1st Qu.	:73.00	1st Qu.	:76.88	1st Qu.	:66.50
Median	:76.50	Median	:78.82	Median	:71.75
Mean	:75.99	Mean	:77.37	Mean	:73.00
3rd Qu.	:78.12	3rd Qu.	:80.98	3rd Qu.	:82.99
Max.	:89.95	Max.	:83.42	Max.	:89.00

5.5.4.3 No Word conversion / No Stemming

Table 5.13 shows the result of the k-fold cross validation using no word conversion and no word stemming.

Table 5.13: Statistical analysis of cross validation, no word conversion, no stemming

News		User		Company	
Min.	:63.96	Min.	:76.80	Min.	:48.50
1st Qu.	:71.88	1st Qu.	:83.70	1st Qu.	:65.75
Median	:77.25	Median	:85.48	Median	:70.50
Mean	:75.92	Mean	:85.02	Mean	:72.49
3rd Qu.	:79.38	3rd Qu.	:87.09	3rd Qu.	:82.21
Max.	:87.94	Max.	:90.40	Max.	:92.50

5.5.4.4 Overall result

Figure 5.5 shows the overall result of the measurement. The first set of 4 boxes refer to the 'News' category (blue), the second set to the 'User' category (green) and the last set to the 'Company' category (magenta). The X-Axis shows whether word conversion and word stemming had been enabled or not. The encoding is as follows:

- 11: Word conversion enabled / Word stemming enabled
- 10: Word conversion enabled / Word stemming disabled
- 01: Word conversion disabled / Word stemming enabled
- 00: Word conversion disabled / Word stemming disabled

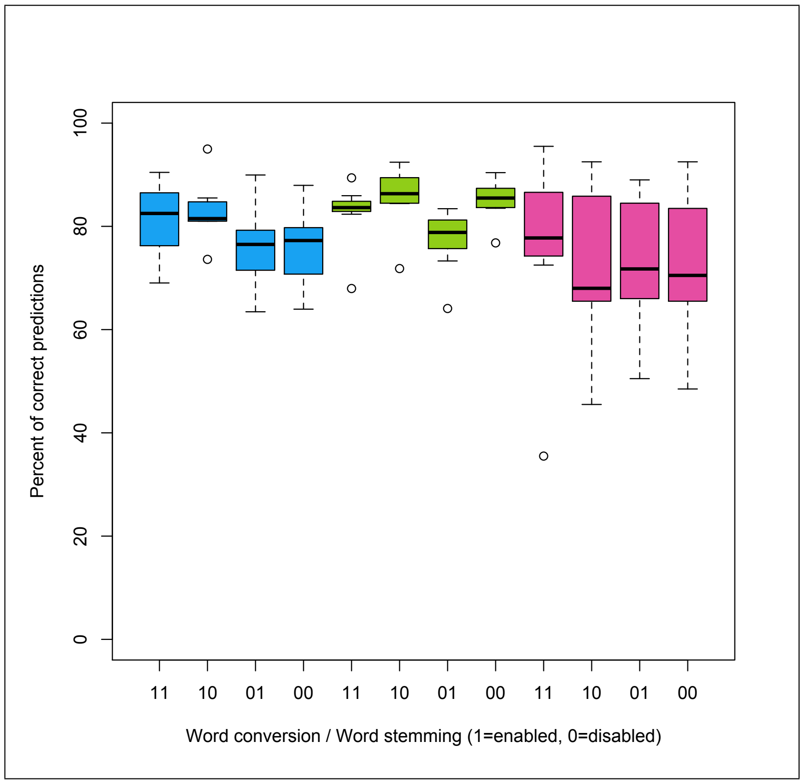


Figure 5.5: Boxplot of overall cross validation results for category C1

All four methods reach an average percentage of about 80% (indicated by the bold lines in the diagram). One explanation would be that word conversion and word stemming helps to obtain an accuracy-level of 80% with fewer training samples. If we add more and more training samples, the accuracy levels off at 80% with all four methods. The next section evaluates this theory.

5.5.5 Impact of amount of training samples

In the previous section, the results have been evaluated with 40 users for each category. Using a 8-fold cross validation, this resulted in 35 training samples and 5 test samples per evaluation run. Now, we measure the impact of the amount of training samples on the accuracy of the classfier. At the end of the last section, we postulated the theory that word conversion and word stemming helps the classifier to obtain good results with fewer training samples.

To second the theory, the data set of each category has been split into the following groups:

1. 4 training samples, 16 test samples
2. 8 training samples, 16 test samples

3. 16 training samples, 16 test samples
4. 24 training samples, 16 test samples

The test runs have been repeated 10 times with randomly chosen training and test samples. The first step was to select 16 test samples. Then, 4 training samples have been chosen. After running the tests with those 4 training samples, another 4 samples have been added to the training set, making it 8 training samples and 16 test samples in total. The test samples were left unchanged. In the next run, 8 randomly chosen samples were added to the training set (=16 training samples, 16 test samples). In the last step, another 8 samples were added to the training set, resulting in 24 training samples and 16 test samples. This process has been repeated 10 times to takeadvantage of the law of large numbers, giving us a good approximation of the actual results.

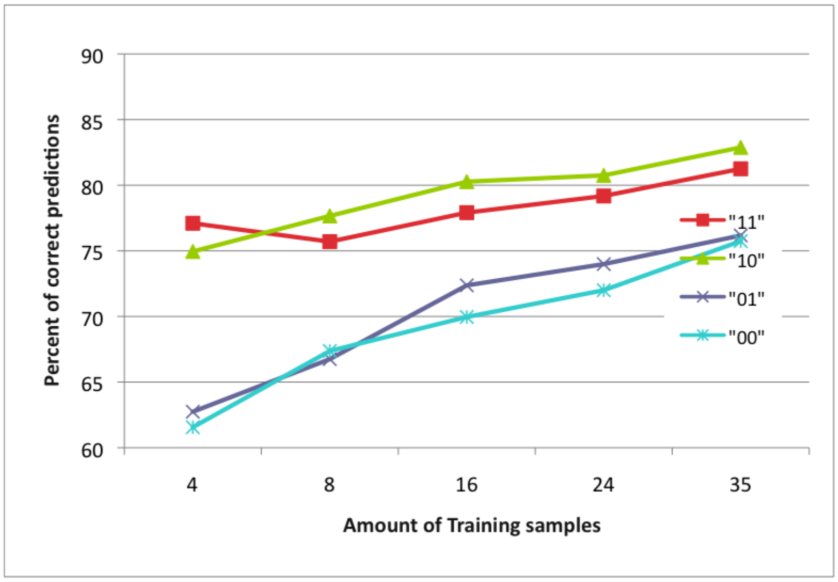


Figure 5.6: News - Prediction accuracy

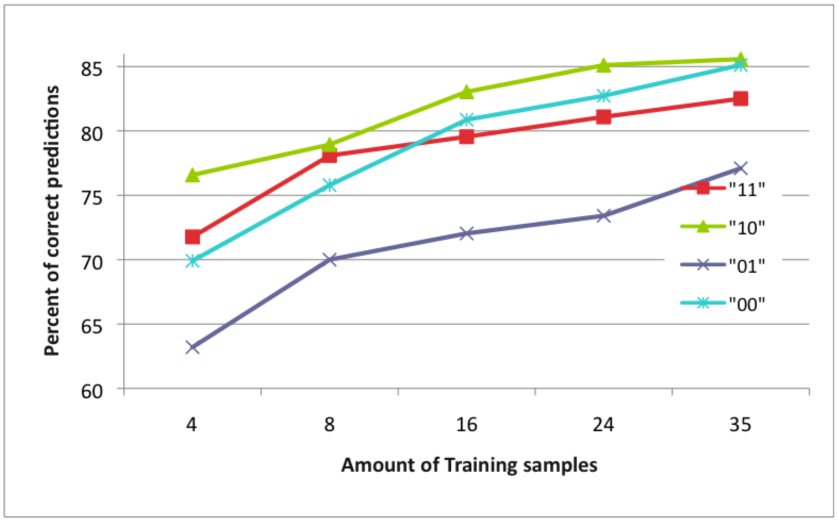


Figure 5.7: User - Prediction accuracy

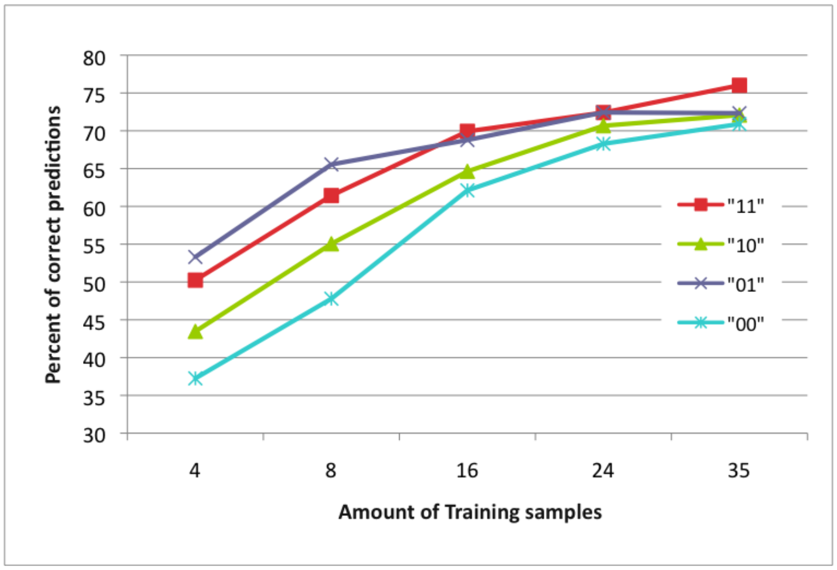


Figure 5.8: Company - Prediction accuracy

Figure 5.9 visualizes the results.

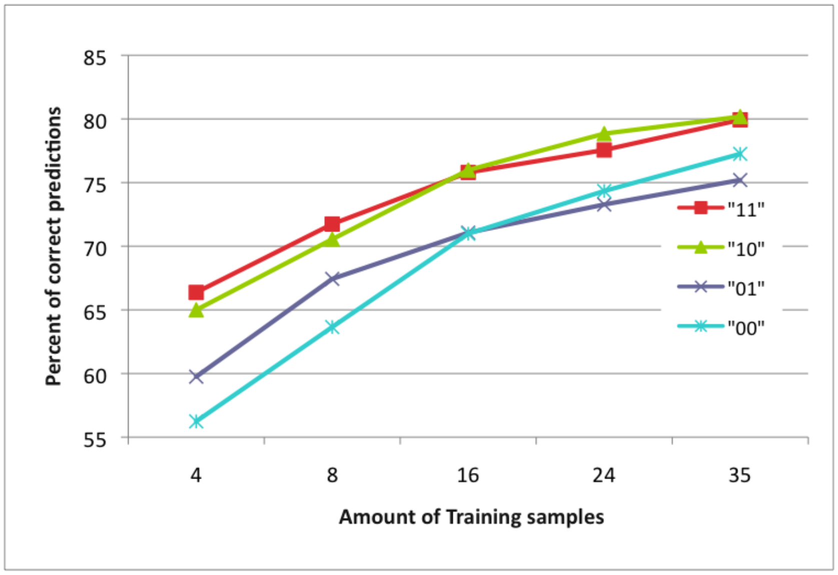


Figure 5.9: Overall - Prediction accuracy

Apparently, only word conversion increases the accuracy - more than 3 percentage points in this case. Also, word conversion helps to achieve good results with fewer training samples. Using 4 training samples, the level of accuracy is more than 65% with word conversion enabled and below 60% if not enabled. On the other hand, it is safe to say that word stemming does not increase the level of accuracy - no matter how much training samples are used.

5.5.6 Tuning the Cost parameter

Now, we measure the impact of the cost parameter to the classification results. As explained in Section 3.2.7.4, a cost value of 1 means that the SVM is not allowed to make any classification errors, while a small value allows it to do so, but penalizes it (soft margin).

The tests have been carried out on the same data set, only for C1 though. Table 5.16 lists the results. C indicates the value of the Cost parameter, followed by the classification accuracy (in percent). Overall is the mean value of News, User and Company. N SV is the number of used support vectors, and the execution time shows how long the machine needed to create the model file and to carry out the test using 8-fold cross validation.

Table 5.14: Cost

C	News	User	Company	Overall	N SV	Execution time
0.001	85.37	79.15	75.76	80.09	2087	39m6s
0.01	84.93	82.42	79.61	82.47	1568	33m47s
0.1	84.69	82.73	77.91	81.78	1415	34m18s
0.2	83.32	82.86	76.33	80.84	1381	35m5s
0.3	82.00.88	82.85	76.15	80.63	1390	35m35s
0.4	82	82.33	76.33	80.22	1386	35m37s
0.5	81.25	82.51	76.02	79.93	1384	35m50s
0.6	80.50	81.86	75.65	79.34	1369	35m53s
0.7	80.13	81.47	75.58	79.06	1372	36m3s
0.8	80.06	81.08	75.20	78.78	1374	36m5s
0.9	79.62	80.69	75.13	78.48	1365	36m4s
1	79.37	80.49	75.31	78.39	1362	36m12s

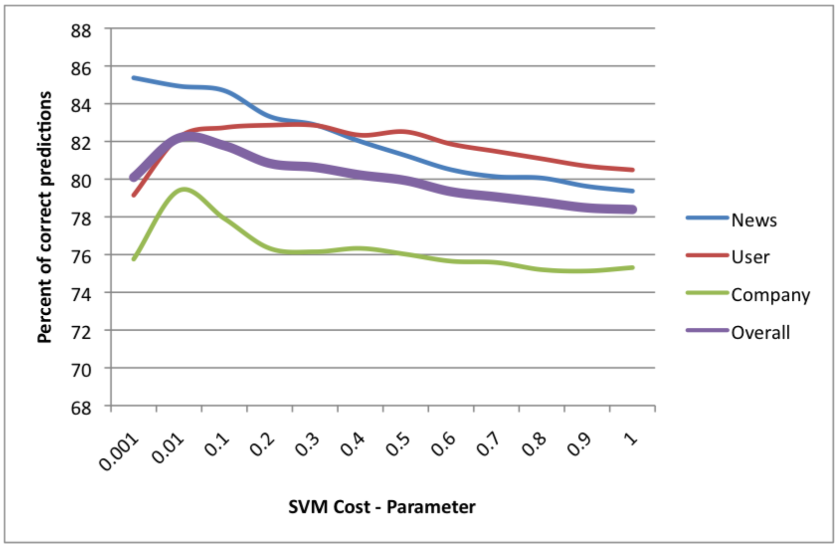


Figure 5.10: Impact of the SVM Cost parameter to the classification results in C1

The results show that a smaller value of C increases the accuracy and decreases the execution time - both issues are favorable. The optimal value of C for our task is 0.01. Using our new knowledge, we repeated the tests from Section 5.5.4 which measured the impact of word conversion and word stemming. The results are summarized in table 5.15.

Table 5.15: Overall result of C1, with tuned C parameter

	News	User	Company	Overall
11	85.06	82.36	79.94	82.45
10	87.16	83.00	76.22	82.13
01	83.37	67.40	72.37	74.38
00	83.95	75.81	72.25	77.34

As we can see, the best results are still achieved with the combination 'word conversion' and 'word stemming' (11). But since the difference to 'word conversion, no word stemming' (10) is negligible, it might be better to disable word stemming due to performance issues. We will investigate this issue in section 5.5.10.

5.5.7 Impact of n-grams

The next question we want to answer is the impact of n-gram selection on the overall classification results. As presented in Section 3.2.6, n-grams are n adjacent characters from a given input string. Instead of treating each word of a Tweet as a feature, the n-gram representation is used. For example, the feature selection for the Tweet 'I am tired' would be 'I A', ' AM', 'AM ', 'M T', ' TI', 'TIR', 'IRE', 'RED', 'ED ', 'D ' using a 3-gram representation. Using a 5-gram representation, the same Tweet is represented as 'I AM ', ' AM T', 'AM TI', 'M TIR', ' TIRE', 'TIRED', 'IRED ', 'RED ', 'ED ', 'D '. We will use 2-, 3-, 4-, 5- and 6-gram representations.

The tests were carried out with the identical setup as in the previous section (8-fold cross validation of C1). For the C Parameter, 0.01 was used. This value performed best in the previous test (82.18% overall level of accuracy).

Table 5.16 and Figure 5.11 show the results. The first column indicates the length of the n-grams and the following four columns the level of accuracy in percent. N SV is the number of support vectors created by the model, the execution time is the time needed to carry out the 8-fold cross validation test.

Table 5.16: N-gram impact

n-gram	News	User	Company	Overall	N SV	Execution time
2	74.90	85.03	61.84	73.92	2394	43m06s
3	62.23	81.83	46.68	63.58	2688	22m10s
4	80.26	50.52	69.56	66.78	2671	14m09s
5	85.40	34.28	44.80	54.83	3420	13m10s

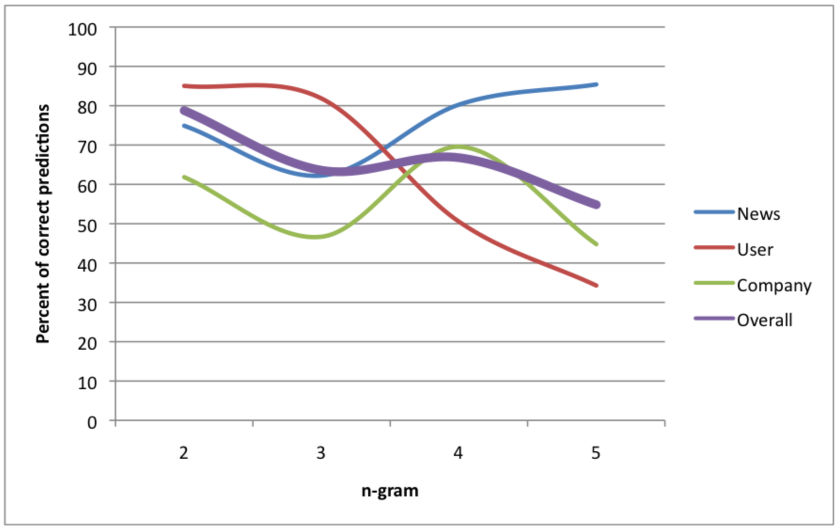


Figure 5.11: Impact of n-gram selection to the classification results in C1

As we can see, the accuracy declines if the length of the n-grams is increased. Another interesting observation is that the number of support vectors increases as well, though the execution time decreases. This means that the classifier starts to learn by rote, which is something we want to avoid when it comes to text classification.

In general, it can be said that using n-gram representations has a negative influence on the classification accuracy. The 'best' results are accomplished by the bigram representation, which nevertheless impairs the accuracy by 10%, while increasing the execution time by more than 30% (compared against not using a n-gram representation). It is obvious that using a n-gram representation is useless for this classification task.

5.5.8 Artificial feature inflation

During the evaluation of the n-gram impact, we made a serendipitous discovery. If the used features (=words) are artificially inflated (using n-grams), the accuracy of the classificator could be improved by more than 2%. However, the performance suffers significantly using this technique - the classification takes almost twice as long. Nevertheless, we briefly explain the approach as it might give some inspiration.

Consider the Tweet 'This is a test'. First, we start by creating a 6-gram representation of a Tweet. This results in the tokens 'This i', 'his is', 'is is_', 's is a', '_is a_', 'is a t', 's a te', '_a tes', 'a test'. Now, instead of using the tokens as features, we re-concatenate the tokens to a single string. This results in 'This i his is is is s is a is a is a t s a te a tes a test'. As we can see, the Tweet has been artificially inflated. Then, the Tweet is split into single words again (using the whitespace character as separator), which results in the following features: 'This','i','his','is','is','is','s','is','a','is','a','is','a','t','s','a','te','a','tes','a','test'. The amount of features has increased to 21. The original representation had 4 features, and the 6-gram representation 9 features.

Table 5.17 shows the impact of this technique to the classification results. Using a 6-gram representation, the level of accuracy increases by 1.74 percentage points (83.92% vs. 82.18%). The results were evaluated on the C1 dataset using 8-fold cross validation. The cost parameter (C) was set to 0.01.

The column 'n-gram' indicates the length of the n-grams, followed by the prediction accuracy for News, User and Companies. Overall is the mean value of the previous three columns. Diff shows the difference in percentage points to the reference score of 82.18%, which is the best score achieved by now (see Section 5.5.6).

Table 5.17: Impact of artificial feature inflation

n-gram	News	User	Company	Overall	Diff
3	83.96	84.52	73.60	80.69	-1.49
4	84.78	86.55	76.52	82.62	0.44
5	85.90	86.27	78.27	83.48	1.3
6	85.96	86.33	79.47	83.92	1.74
7	85.02	86.86	79.29	83.72	1.54
8	84.95	86.79	78.91	83.55	1.37
9	85.33	86.21	78.66	83.40	1.22
10	85.58	86.21	78.54	83.44	1.26
11	85.64	86.14	78.79	83.52	1.34

5.5.9 Large-scale Tweet analysis

Having our new classifier tested and well tuned, we are ready to do a large scale import of Tweets to analyze them. The aim is to know how many of the Tweets are classified as News, User and Company, and as Factual and Opinionated. To import the Tweets, a script was written which imports the latest 50 Tweets of the current public timeline. This script was executed every minute for about 19 hours. This resulted in 19.320 distinct Tweets, imported between March 6, 2010, 3PM and Mach 7, 2010, 10AM.

After that, the Tweets have been classified. Tables 5.18 and 5.19 show the results:

Table 5.18: Results of large-scale Tweet analysis

Type	Count	Percent
News	2.948	15.3%
User	15.202	78.7%
Company	1.170	6.1%
Factual	7.120	36.85%
Opinionated	12.200	63.15%

Table 5.19: Factual/Opinionated breakdown

C1	Factual	Opinionated
News	98.10%	1.90%
User	21.09%	78.91%
Company	87.35%	12.65%

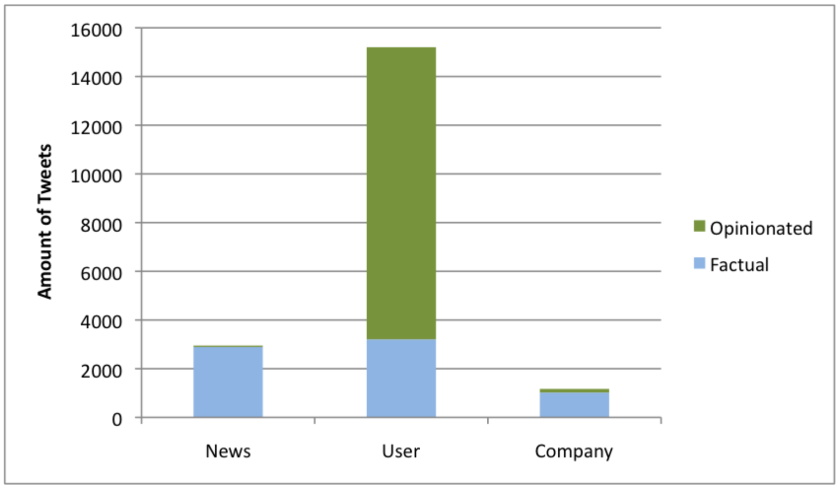


Figure 5.12: Result of large-scale Tweet analysis

The large-scale analysis shows that 79% of the imported Tweets are classified as User messages, 15% as News messages, and 6% as Company advertisements. Besides that, about two-thirds of the Tweets are factual. The most interesting observation is that the occurrence of ‘Company’ Tweets is relatively low. Due to the nature of those Tweets, they can also be seen as spam on a certain level. Twitter itself claims that the spam level is at around 1-2% (see section 2.5), while our analysis resulted in 6%. This difference to our result could be explained by the fact that Twitter itself might not consider ‘Sponsored Tweets’ (which we used to classify advertisements) as spam.

In section 3.1.1, we summarized the research from Naaman et al. [NBL10]. They estimated the number of tweets with an informational character (IS) with 20%, while 80% can be characterized as user-to-user communication. We can map the the first Tweets (IS) to our ‘News’ category, while the latter Tweets can be mapped to the ‘’User category. This results in 20% ‘News’ messages, and 80% ‘User’ messages. This corresponds very well to our result of the Tweet analysis, as our empirical evaluation shows that 15% of the Tweets are ‘News’ messages, while almost 80% are ‘User’ messages.

5.5.10 Performance

Now, we measure the performance of our algorithms. Are there differences between stemming and non-stemming? Does it make a difference if we use word conversion or not?

To answer those questions, the execution time used to create the training set and the time used to classify a large set of Tweets was measured. The training set consisted of the complete data set described in section 5.3.1 (4.842 Tweets). For the test set, the large-scale import from the last section was used (19.320 Tweets).

Following characteristics were measured:

- CTV: Time (in seconds) for creating the training vector
- CTM: Time (in seconds) for creating the training model (SVM)
- STR: Size (in MB) of the created training model
- TTV: Time (in seconds) for creating the test vector
- TTM: Time (in seconds) for creating the test model (SVM)
- STE: Size (in MB) of the created test model
- ACC: Accuracy (in percent), taken from Table 5.15

In regard to hardware specifications, the tests were carried out on the system as described in section 5.2.

Table 5.20 shows the results of these tests. The first column is the type (11 = word conversion / word stemming, 00 = no word conversion / no word stemming), the subsequent columns correspond to the encoding from the above listing. Also, we included the best result achieved in section 5.5.8 (artificial feature inflation), labeled as AFI 6.

Table 5.20: Performance measurement results

	Training				**Test**				
Type	CTV	CTM	Total	STR	TTV	TTM	Total	STE	ACC
11	114s	162s	276s	27mb	441s	420s	861s	185mb	82.45%
10	114s	153s	267s	25mb	417s	436s	853s	187mb	82.13%
01	**113**s	164s	277s	26mb	403s	376s	779s	**118**mb	74.38%
00	**113**s	**149**s	**262**s	**24**mb	**397**s	**358**s	**755**s	120mb	77.34%
AFI 6	121s	283s	404s	41mb	458s	767s	1225s	328mb	**83.92**%

Looking at the results, we can say that when creating the model file, it makes no significant difference if word conversion and/or word stemming is used. However, the results vary when performing the classification. Using no word conversion and no stemming speeds the classification process up by approximately 10%, but achieves 5 percentage points less accuracy. As always, it is a trade-off between performance and accuracy.

5.6 Sentiment detection

The last part of the evaluation process was to check if certain users prefer to write positive, negative or neutral Tweets. In this thesis, the approach of word scoring was used. It is based on the scoring approach used by Agrawal et al. (see Section 3.1.2, and [AS09] respectively), but without heuristic elements.

First, a list of positive and negative words was needed. The MPQA Opinion Corpus provides such a list. MPQA stands for Multi-Perspective Question Answering and is a workshop funded by the Northeast Regional Research Center (NRRC). In 2005 they compiled a subjectivity lexicon [WWH05] and offer it for download[7].

For this thesis, we imported all words to our internal database and used them to classify the Tweets in regard to the sentiment. Since the word list contains both stemmed and non-stemmed words, all non-stemmed words were ignored. This resulted in a total of 488 positive words and 998 negative words. Also, the words have a certain expression type assigned (strong subject and weak subject). We assigned the strong subjects a weighting score of 3, and a weighting score of 2 to the weak subjects. In order to classify a Tweet as positive or negative, following algorithm was used: Iterate through all words in the database and assign the weight of this word to the Tweet if it is contained within the Tweet. If the word is positive, add the weight, and if it's negative, subtract the weight. For example, the Tweet *'I loved that! Thanks!'* received a score of +6 since the words 'loved' and 'thanks' are both strong positive subjects. On the other hand, the Tweet *'Murder inquiry launched after Glasgow man killed in knife attack'* got classified as a negative sentiment with the score of -8, as 'murder', 'kill', 'knife' and 'attack' are weak negative subjects.

Nevertheless, there are Tweets that were classified incorrectly. For example, the Tweet *'i wish you'd want me for more than just your own person pleasure..'* received a score of +9 ('wish', 'want' and 'pleasure' are all strong positive subjects), but this Tweet has undoubtably a negative sentiment.

The tests were carried out with the same dataset used in the previous sections. The results are listed in tables 5.21 and 5.22, and visualized in figures 5.13 and 5.14.

[7]http://www.cs.pitt.edu/mpqa/

Table 5.21: Sentiment classification for C1

Category	Positive	Neutral	Negative
News	14.6%	64.7%	20.7%
User	25.6%	61.6%	12.8%
Company	13.9%	81.0%	5.1%

Table 5.22: Sentiment classification for C2

Category	Positive	Neutral	Negative
Factual	13.4%	73.4%	13.2%
Opinionated	28.3%	59.2%	12.4%

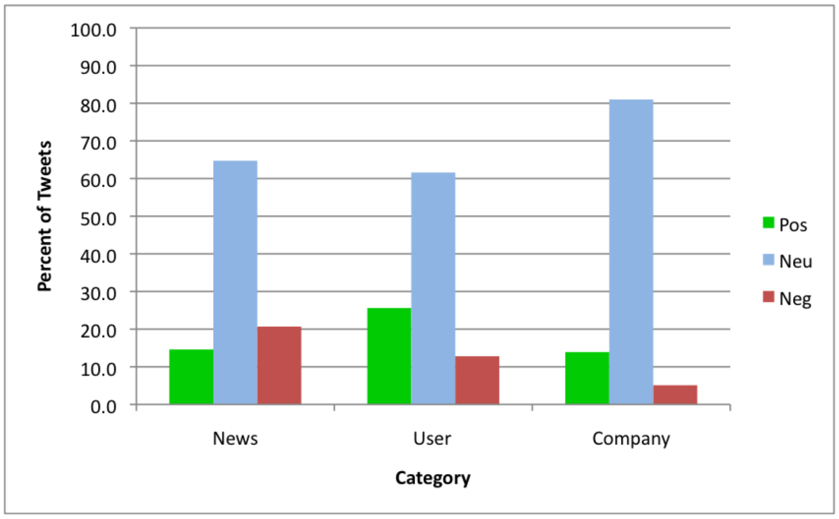


Figure 5.13: Sentiment classification for C1

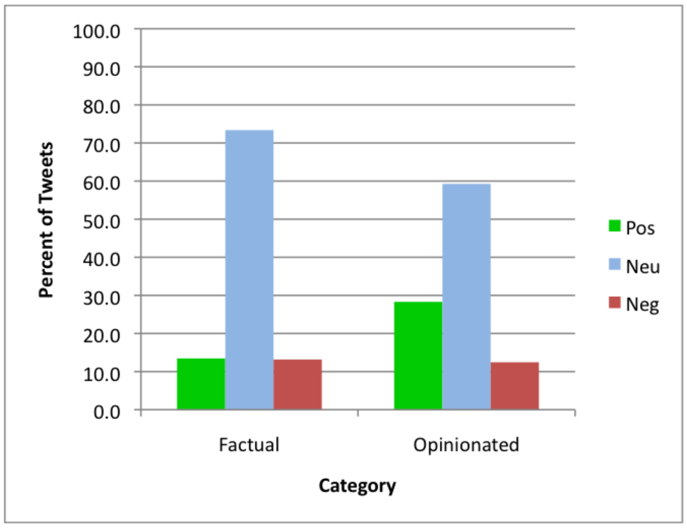


Figure 5.14: Sentiment classification for C2

Taking a closer look at the results in Figure 5.13, the first observation is that the ‘News’ category has the most negatively classified Tweets. (20.7%). On the other hand, the ‘Company’ category has the fewest negative Tweets. Based on this analysis, it can be said that advertisements tend to avoid negative words, while ‘News’ make more use of them. Another interesting observation is that Tweets of the ‘User’ are classified positively twice as often as negatively.

For the second category C2, it can be said that the positive and negative Tweets balance each other in the ‘factual’ category, while in the ‘opinionated’ category, positive Tweets occur twice as much as negative Tweets.

Due to missing test data, the level of accuracy of the technique could not be measured. Creating a test-set would have been outside the scope of this thesis.

Chapter 6

Conclusion

In this thesis, a way for analyzing Twitter messages (and as a result, the users) was proposed. Using SVMs, we were able to predict the category of a given Tweet with a level of accuracy of more than 80%.

The main challenge was to find a representative data set. It was necessary that the representatives do not overlap each other and they constitute a optimal representative for their category. After defining and specifying the categories, we have manually chosen 120 Twitter users and assigned them to a category.

In respect to the short and sparse information transported with a single Tweet, we showed that it is best not to use word stemming, but to pre-process Tweets with different regular expressions (Section 5.5.4.4). Using n-gram representation has no positive impact on the level of accuracy (even quite the opposite - the level decreases), but reduces the execution time. As always, it is a trade-off between performance and accuracy. Tuning the cost parameter for SVM achieved a boost in accuracy of almost 5% (see Section 5.5.6).

In regard to sentiment detection, a scoring approach using a list of positive and negative words was used. Our analysis showed that News Tweets use more negative than positive words and that User Tweets use positive words twice as often as negative ones.

A large-scale analysis showed that almost 80% of the Tweets are categorized as 'User' Tweets, 15% as News and 6% as Company advertisements. This indicates that the spam level is not very high, but actually higher than the 1-2% that Twitter claims (see Section 2.5).

The scientific output of this thesis is a dataset containing 120 users and approximately 4.800 Tweets, all categorized. Also, a platform was implemented which can be used to classify arbitrary Twitter users. And last but not least, our empirical evaluation showed that Support Vector Machines can be used very well for classifying Twitter messages (and thus users).

6.1 Future work

The most challenging task would be to use the proposed scoring algorithm to automatically bootstrap the training examples for the SVM - thus switching to a semi-supervised learning setting - and examine if the created system outperforms current sentiment detectors.

Also, it would be interesting to implement different classifiers (like kNN, or Random Forests) and compare the results.

For the application, a self-updating model file would be an interesting option. This is useful for an accurate prediction of new Tweets because of the simple reason that the used language changes rapidly. Users introduce new words or phrases (think of hashtags), news companies write about current events (Olympic games in 2008, Obama in 2009, Haiti in 2010) and companies constantly update their advertisements.

Appendix A

Chosen Twitter users

A.1 Chosen Twitter users for C1 - News

Table A.1: Chosen News accounts

Account name	entered location	Twitter screenname
CNN Breaking News	Everywhere	cnnbrk
BBC News	London	bbcnews
BBC World News	London, UK	bbcworld
CNN		CNN
The Guardian	London	guardiannews
The New York Times	New York, NY	nytimes
NYTimes World	New York, NY	nytimesworld
The Washington Post	Washington, D.C.	washingtonpost
msnbc.com		msnbc
Wall Street Journal	New York, NY	WSJ
Breaking News	Global	BreakingNews
Chicago Tribune	Chicago, IL	chicagotribune
The Japan Times	Japan	japantimes
Houston Chronicle	Houston, TX	HoustonChron
msnbc.com - World		msnbc_world
News Ticker	The Internets	newsticker
PhillyInquirer	Philadelphia, PA	PhillyInquirer
CBS News	New York, NY	CBSNews
Reuters Top News	Earth	Reuters_TopNews
USA TODAY Top News	USA TODAY	USATODAY
herald_tribune		herald_tribune

Table A.1: Chosen News accounts (continued)

Account name	entered location	Twitter screenname
The Daily Telegraph	London, England	TelegraphMG
Los Angeles Times	Los Angeles, CA	latimes
FBI PressOffice	Washington, DC	FBIPressOffice
Shanghai Daily	Shanghai	shanghaidaily
Elizabeth Benjamin	New York City	ebdailypolitics
The Moscow Times	Moscow, Russia	MoscowTimes
Scotsman News	Scotland	scotsmannews
Bangkok Post	Thailand	bpbreakingnews
Haaretz.com	Israel	haaretzonline
world news freedom		worldnewstweets
KPRC Channel 2 News	Houston, TX	Local2Breaking
DW — Europe	Bonn, Germany	dw_europe
Hurriyet Daily		HurriyetDaily
mumbai mirror	India	mumbaimirror
Steve Wagner	San Luis Obispo, Ca	USNPL
Sky News Breaking	London, UK	SkyNewsBreak
Times of india		timesofindia01
The Saturday Star		saturdaystar
Paul Fisher		ukPoliticsme

A.2 Chosen Twitter users for C1 - User

Table A.2: Chosen User accounts

Account name	entered location	Twitter screenname
Micah Alpern	Sunnyvale, ca	malpern
tony	san francisco, ca	tony
John Breslin	Galway, Ireland	johnbreslin
Sarah Richie	Lake Jackson, Texas	sarah
jayna	London	jayna
Paulo Coelho	Rio de Janeiro	paulocoelho
Billy Abbott	Ealing, London	cowfish
fredrock	Royal Oak, MI	fredrock
DaveJMatthews	Faubourg Marigny, Louisiana	davejmatthews
lilyroseallen	london	lilyroseallen
anjali	redondo beach, california.	anjalapeepee

Table A.2: Selected News accounts (continued)

Account name	entered location	Twitter screenname
newportironman	Newport Beach	newportironman
ashton kutcher	here	aplusk
Perez Hilton	Hollywood, California	PerezHilton
Lauren Barber	Kent, UK	GGeePR
Shola Gordon	Sheffield, United Kingdom	sholagordon
Amber Gillespie	michigan	ambergillespie
Blake Lively	NYC	blakeclively
Rob Anthony Letcher	Plymouth UK	AnthonyLetcher
Daniel Magee		magee21
Andrew Wertheim	Somewhere	awertheim
Devin Knox	Austin, Texas	dknox41
Paris Hilton	T: 25.222988,55.357721	ParisHilton
CharLene	Reading, Pennsylvania	MsCharLene
Nicole Richie		nicolerichie
Carter Jenkins	LA	CarterJenkins
Chris Schneider	Munich, Germany	upsidedownchris
Kevin Alambra	Rizal, Philippines	KevinAlambra
Sahil	Delhi, India	illsahil
Meriel Pamintuan	San Francisco/Karakura Town	mayomeriel
Daniel J. Garcia	T: 25.750587,-80.180493	daniel_j_garcia
Bill Gates	Seattle, WA	billgates
Matthew Babits	North East U.S.	gravitywave3
Jamie Harshman	Kingston	jamie0901
Lindsay Lohan	T: 40.764068,-73.976541	lindsaylohan
Brittany Farrell	Canada	BFarrell202
John Wood	Perpetual Global road trip	johnwoodRTR
Anthony Weeling	Los Angeles / Paris	anthonyweeling
Liam Ryan	Dublin	liamryanie
Jessica Alba		jessicaalba
ashlynn tyler	your closet.	ashlynnxtyler

A.3 Chosen Twitter users for C1 - Company

Table A.3: Chosen Company accounts

Account name	**entered location**	**Twitter screenname**
xemmy	Beverly Hills, CA	xemmy
Erwin Ensing	Amsterdam Netherlands	erwinensing
sudhir jain	Mumbai,India	jainsudhir
RWNB Tech Deals	Hampton Roads, VA	rwnbtechdeals
veronica fairfax	houston	ronnie_fairfax
Noor	India	adsenseideas
Larry Monroe	working from home	immakingmoney
Everyday Money	USA	myeverydaymoney
Beauty Cosmetic	San Francisco, CA	youarepretty
1or2kaday	Chicago	1or2kaday
The Coupon Club	shoprmall.com	TheCouponClub
Ken Leon	Suburban Chicago, IL	homealliance92
lowfat_recipes	USA	lowfat_recipes
kelvin Osondu	uk	kellyfonky
tony		freemoney76
Adam Terwinski	London, United Kingdom	adas4010
GetGDImoney	WorldWide	getgdimoney
Electronic Deals	San Francisco, CA	e_bargains
iPhone & iPod stuff	San Francisco, CA	3gsdude
Computer Bargains	San Francisco, CA	compudah
Deal Nay	San Francisco	dealnaydotcom
unlooses	San Francisco, CA	unlooses
Inner	San Francisco, CA	innerinner
Recession Proof	International	recessionproofp
Deals Rebates	USA	dealsrebates
Miracle Profit	Brussels	WonderfulProfit
UK Voucher Codes	United Kingdom	_ukvouchercodes
Kid Deals	Worldwide	kidDEALS
Jim S	At My Home Office	salemjsells
Wed For Less	UK	wedforless
Stefan	Romania/Galati	rhade24
w Ariel		arielgabriel
Steve	Sydney	earnaustralia
Deal Feeder Deals		dealfeeder
Laptop News		News_Laptop
Angler Alley Fishing		angleralley

Table A.3: Chosen Company accounts (continued)

Account name	**entered location**	**Twitter screenname**
free foru	earth	free4u2day
Super Discount Daily	New York	discount_daily
Gazaro: Latest Deals	USA	latestdealsus
michel joan	I'm Working At Home	marketing32

Bibliography

[AH10] Sitaram Asur and Bernardo A. Huberman. Predicting the future with social media. Mar 2010.

[AS09] Shaishav Agrawal and Tanveer j Siddiqui. Using syntactic and contextual information for sentiment polarity analysis. In *ICIS '09: Proceedings of the 2nd International Conference on Interaction Sciences*, pages 620–623, New York, NY, USA, 2009. ACM.

[BL97] Avrim L. Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

[Bre01] Leo Breiman. Random forests. In *Machine Learning*, volume 45, pages 5–32, 2001.

[Bur98] Christopher J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[CL01] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[CL09] Marc Cheong and Vincent Lee. Integrating web-based intelligence retrieval and decision-making from the twitter trends knowledge base. In *SWSM '09: Proceeding of the 2nd ACM workshop on Social web search and mining*, pages 1–8, New York, NY, USA, 2009. ACM.

[CNN$^+$10] Jilin Chen, Rowan Nairn, Les Nelson, Michael Bernstein, and Ed Chi. Short and tweet: experiments on recommending content from information streams. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, pages 1185–1194, New York, NY, USA, 2010. ACM.

[CT94] William B. Cavnar and John M. Trenkle. N-gram-based text categorization. In *In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1994.

[DDH$^+$07] Anirban Dasgupta, Petros Drineas, Boulos Harb, Vanja Josifovski, and Michael W. Mahoney. Feature selection methods for text classification. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 230–239, New York, NY, USA, 2007. ACM.

[DP97] Pedro Domingos and Michael J. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997.

[DPHS98] Susan Dumais, John Platt, David Heckerman, and Mehran Sahami. Inductive learning algorithms and representations for text categorization. In *CIKM '98: Proceedings of the seventh international conference on Information and knowledge management*, pages 148–155. ACM Press, 1998.

[DS] N.A. Diakopoulos and D.A. Shamma. Characterizing debate performance via aggregated twitter sentiment. *CHI 2010, ACM, Atlanta Georgia (2010).*

[ES06] Andrea Esuli and Fabrizio Sebastiani. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC-06, the 5th Conference on Language Resources*, Genova, IT, 2006.

[ES08] Martin Ebner and Mandy Schiefner. Microblogging - more than fun? *Proceedings of IADIS Mobile Learning Conference 2008,*, pages 155–159, 2008.

[Fri06] Jeffrey E. F. Friedl. *Mastering Regular Expressions.* O'Reilly Media, 3 edition, 2006.

[GBH09] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. 2009.

[GM04] Evgeniy Gabrilovich and Shaul Markovitch. Text categorization with many redundant features: using aggressive feature selection to make svms competitive with c4.5. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 41, New York, NY, USA, 2004. ACM.

[HANS90] P. J. Hayes, P. M. Andersen, I. B. Nirenburg, and L. M. Schmandt. Tcs: a shell for content-based text categorization. In *Proceedings of the sixth conference on Artificial intelligence applications*, page 325, Piscataway, NJ, USA, 1990. IEEE Press.

[HRW08] Bernardo A. Huberman, Daniel M. Romero, and Fang Wu. Social networks that matter: Twitter under the microscope. *CoRR*, abs/0812.1045, 2008.

[HT96] Will Hill and Loren Terveen. Using frequency-of-mention in public conversations for social filtering. In *CSCW '96: Proceedings of the 1996 ACM conference on Computer supported cooperative work*, pages 106–112, New York, NY, USA, 1996. ACM.

[Joa98] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Machine Learning: ECML-98*, pages 137–142. Springer, 1998.

[Joa02] T. Joachims. *Learning to Classify Text Using Support Vector Machines – Methods, Theory, and Algorithms.* Kluwer/Springer, 2002.

[JSFT07] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: understanding microblogging usage and communities. In *WebKDD/SNA-KDD '07: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65, New York, NY, USA, 2007. ACM.

[JZSC09] B.J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*, 2009.

[KGA08] Balachander Krishnamurthy, Phillipa Gill, and Martin Arlitt. A few chirps about twitter. In *WOSP '08: Proceedings of the first workshop on Online social networks*, pages 19–24, New York, NY, USA, 2008. ACM.

[KKM02] Jaap Kamps, , Jaap Kamps, and Maarten Marx. Words with attitude. In *In 1st International WordNet Conference*, pages 332–341, 2002.

[Kle99] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[LBS04] Diego Linares, José-Miguel Benedí, and Joan-Andreu Sánchez. A hybrid language model based on a combination of n-grams and stochastic context-free grammars. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(2):113–127, 2004.

[Lov68] Julie B. Lovins. Development of a stemming algorithm. 1968.

[LSST$^+$02] Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, Chris Watkins, and Bernhard Scholkopf. Text classification using string kernels. *Journal of Machine Learning Research*, 2:563–569, 2002.

[MR81] C. B. Mervis and E. Rosch. Categorization of natural objects. *Annual Review of Psychology*, 32(1):89–115, 1981.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval.* Cambridge University Press, 1 edition, 2008.

[MTP10] Meredith Ringel Morris, Jaime Teevan, and Katrina Panovich. What do people ask their social networks, and why?: a survey study of status message q&#38;a behavior. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, pages 1739–1748, New York, NY, USA, 2010. ACM.

[NBL10] Mor Naaman, Jeffrey Boase, and Chih-Hui Lai. Is it really about me?: message content in social awareness streams. In *CSCW '10: Proceedings of the 2010 ACM conference on Computer supported cooperative work*, pages 189–192, New York, NY, USA, 2010. ACM.

[NYT09] Jenna Wortham New York Times. Bit.ly eclipses tinyurl on twitter. Website, 2009. `http://bits.blogs.nytimes.com/2009/05/07/bitly-eclipses-tinyurl-on-twitter/`.

[Pep10] Julianne Pepitone. Twitter mobilizes haiti aid efforts. Website, 2010. `http://money.cnn.com/2010/01/13/technology/twitter_haiti_donations/index.htm`.

[PLV02] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, 2002.

[PMS09] Owen Phelan, Kevin McCarthy, and Barry Smyth. Using twitter to recommend real-time topical news. In *RecSys*, pages 385–388. ACM, 2009.

[Por80] M. F. Porter. An algorithm for suffix stripping. pages 130–137, 1980.

[RJ76] S. E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.

[RZ04] Gordon Rios and Hongyuan Zha. Exploring support vector machines and random forests for spam detection. In *CEAS*, 2004.

[Seb02] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, 2002.

[SKC10] David Shamma, Lyndon Kennedy, and Elizabeth Churchill. Tweetgeist: Can the twitter timeline reveal the structure of broadcast events? *CSCW (2010)*, 2010.

[SST$^+$09] Jagan Sankaranarayanan, Hanan Samet, Benjamin E. Teitler, Michael D. Lieberman, and Jon Sperling. Twitterstand: news in tweets. In *GIS*, pages 42–51. ACM, 2009.

[SWY75] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

[Tec09] TechCrunch. Twitter reaches 44.5 million people worldwide in june (comscore). Website, 2009. `http://www.techcrunch.com/2009/08/03/twitter-reaches-445-million-people-worldwide-in-june-comscore/`.

[Tim09] Financial Times. Twitter now worth $7.14m per character. Website, 2009. `http://www.ft.com/cms/s/0/641057ae-a933-11de-9b7f-00144feabdc0.html`.

[TM09] Lev Grossman Time Magazine. Iran protests: Twitter, the medium of the movement. Website, 2009. `http://www.time.com/time/world/article/0,8599,1905125,00.html`.

[Vap95] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, 1995.

[WLJH10] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: finding topic-sensitive influential twitterers. In Brian D. Davison, Torsten Suel, Nick Craswell, and Bing Liu, editors, *WSDM*, pages 261–270. ACM, 2010.

[WS10] Claudia Wagner and Markus Strohmaier. The wisdom in tweetonomies: Acquiring latent conceptual structures from social awareness streams. *Semantic Search 2010 Workshop (SemSearch2010), in conjunction with the 19th International World Wide Web Conference (WWW2010), Raleigh, NC, USA*, 2010.

[WTSW02] Zhirong Wang, Umut Topkara, Tanja Schultz, and Alex Waibel. Towards universal speech recognition. In *ICMI '02: Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, page 247, Washington, DC, USA, 2002. IEEE Computer Society.

[WWH05] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of HLT/EMNLP*, 2005.

[YL99] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49, New York, NY, USA, 1999. ACM.

[YS76] C. T. Yu and G. Salton. Precision weighting—an effective automatic indexing method. *J. ACM*, 23(1):76–88, 1976.