Universität Zürich^UZH

Bachelorarbeit
zur Erlangung des akademischen Grades
Bachelor of Arts
der Philosophischen Fakultät der Universität Zürich

Twitter Sentiment Analysis: the Use of Automatically Acquired Training Data for Machine Learning and Hybrid Methods

Verfasser: Simon Wegmüller
Matrikel-Nr: 10-724-474

Referent: Prof. Dr. Martin Volk
Betreuer: Dr. Manfred Klenner

Institut für Computerlinguistik

Abgabedatum: 22.06.2015

Abstract

This is the place to put the English version of the abstract.

Zusammenfassung

Und hier sollte die Zusammenfassung auf Deutsch erscheinen.

Contents

List of Figures

List of Tables

List of Acronyms

NLP	Natural Language Processing
OCR	Optical Character Recognition
POS	Part-Of-Speech
SA	Sentiment Analysis
UTF-8	Unicode Transformation Format (8-bit)

List of additional Files

The following files can be found on the CD that was submitted together with this Paper:

lex.py	Baseline leixcon based system
lex2.py	Extended baseline lexicon based system

1 Introduction

Since its foundation in 2006, Twitter has rapidly become one of the Internets biggest social networking services, with more than 100 million users who in 2012 posted 340 million tweets per day [1] With its simple format that allows the users to send and read basic, 140-character long messages, Twitter has become an important asset in the way people communicate and share their opinions in the 21st century, thus making Twitter an invaluable tool for analyzing public opinions on current debates, products or even social studies. For example, a business is able to analyze tweets mentioning certain products of theirs, deciding whether the public opinion is favorable or more critical towards this product. This process is called sentiment analysis or opinion mining, an approach that uses natural language processing, text analysis and other techniques from the field of computational linguistics in order to identify and extract this sort of information from the source material. One basic task in sentiment analysis is classifying the polarity of a text (e.g. a tweet). This can be done either at the document, sentence, or feature level and basically returns a decision whether this document, sentence or feature is positive, negative or neutral. There also exist more advanced techniques which go beyond polarity classification, for example deciding between different emotional states such as "happy", "sad" or "aggressive". Generally, one can distinguish between four separate, main categories of techniques that can be used to perform sentiment analysis: keyword spotting, lexical affinity, statistical methods, and concept-level techniques. Keyword spotting refers to a technique that relies on classifying text by spotting certain unambiguous affect words (e.g. happy, sad, afraid). Lexical affinity is a continuation of this approach, including arbitrary words which have been assigned an affinity value towards particular emotions or towards a polarity. Statistical methods make use of large data sets in order to perform statistical analyses by using a number of different machine learning algorithms. Finally, concept-level techniques make use of elements from the field of knowledge representation (e.g. ontologies, semantic networks) and are able to detect semantics in the text by analyzing concepts which are more subtle, by linking these to to other concepts which may relay relevant information.

[1] https://blog.twitter.com/2012/twitter-turns-six

The motivation of this paper is to create and compare different methods of performing sentiment analysis for German language tweets. One of the main problems of using statistical methods is the limited data that is available. In order to train a (supervised) machine learning classifier, a relatively large set of annotated data has to be at hand. Since there is no large enough corpora for German tweets available today, one of the goals of this paper is to automatically create/retrieve data which can then be used in order to train a number of classifiers. Furthermore, the results of these methods shall be compared to a lexicon-based approach, using a polarity lexicon and polarity values of individual words in order to classify tweets, as well as to an improved version of this system, including additional features, such as hashtags and emoticons, in order to improve the classification task.

The structure of this paper is as follows: chapter 2 will cover the theoretical concepts that are relevant to the topic as well as explain the research questions. In chapter 3, the acquisition process for the data will be explained as well as the methods and techniques used for the different approaches for analyzing the tweets. This includes a closer look at the python scripts that were created for these tasks. Chapter 4 will present the results of the analyses and chapter 5 will entail a close analysis of the results and a discussion thereof. Finally, chapter 6 will highlight the conclusions to be drawn from the results as well as entail a short outlook on further tasks and questions to be answered in the future.

2 Theoretical Background

According to the Handbook of Natural Language Processing, textual information can be categorized into two main types: facts and opinions Indurkhya and Damerau [2010, 7]. Whereas facts represent objective expressions about entities, events and their properties, opinions are subjective expressions, describing sentiments or feelings towards entities or events. Naturally, the concept of opinions is very broad, encompassing a variety of complex ways in which they can be expressed. In order to simplify things, the focus here lies on expressions that convey either positive or negative sentiments. The World Wide Web has led to an ever increasing amount of information that is available to the public and it has dramatically changed the way people express their views and opinions [Indurkhya and Damerau, 2010, 7]. A very large number of social media, for example news, forums, blogs or product reviews, contain sentiment-based sentences. The texts in these domains are in a large part created by the users themselves and analyzing these texts has become an important task in NLP in general and especially for sentiment analysis, also known as opinion mining, which is the automatic detection of opinions, or sentiments in these texts. One of the biggest sources of this sort of information is Twitter, which was introduced in 2006 and has quickly become the focus of a fairly large number of researchers, mainly due to the sheer amount of information that is available. In the following sections I will explain the theoretical concepts and approaches regarding sentiment analysis, as well as present a (limited) survey of current research in the field.

2.1 Sentiment Analysis

Sentiment analysis, sometimes also referred to as opinion mining, sentiment detection or sentiment extraction, is a technique using NLP methods in order to analyze texts concerning people's opinions, evaluations, appraisals and emotions towards certain entities, such as products, services, organizations, individuals, issues, events, topics and their properties. According to Liu, the term sentiment analysis first appeared in Nasukawa and Yi, 2003 [Liu, 2012, 7]. Although Research in the field had already appeared earlier (e.g. Das and Chen [2001], ?). Before the year 2000, little research had been done concerning sentiment and opinion analysis, but in recent years the area has become very active, mainly due to the wide range of applications it provides, as well as the challenging nature of the problems it imposes, which are well worth studying.

The applications of sentiment analysis are extremely wide, ranging from "consumer products, services, healthcare and financial services to social events and political elections" [Liu, 2012, 8]. Many big corporations today have built their own in-house capabilities, e.g. Microsoft, Google, HewlettPackard [Liu, 2012, 8]. Furthermore, sentiment analysis has been used to predict sales performance (Liu et al. [2007]), reviews were used to rank products and merchants (McGlohon et al. [2010]), Twitter sentiment was linked with public opinion in (O'Connor et al. [2010]) and to predict election results (Tumasjan et al. [2010]). In Asur and Huberman [2010], Twitter data, movie reviews and blogs were used in order to predict box-office revenues for movies. In Bollen et al. [2011], Twitter moods were used to predict the stock market and in Zhang and Skiena [2010], blog and news sources were analyzed in order to study trading strategies using sentiment analysis.

Sentiment analysis can be categorized into three main granularities, referring to the level at which the task is focused. These three levels are: Document level, Sentence level and Entity and Aspect level. At the document level, the task is to classify whole documents according to their sentiment polarity (negative, positive). An example would be to classify product reviews in regard to the expressed opinion towards the specific product. One problem with this granularity is the assumption that each document only discusses one entity, or expresses an opinion towards one single entity (e.g. a single product). Thus, if a document discusses or compares multiple entities, it is not applicable. At the sentence level, the task is defined as classifying individual sentences, deciding whether they express positive, negative or neutral opinions. As Liu [2012] states , this task is closely related to subjectivity classification, which is a method to distinguish between sentences which express "factual information from sentences that express subjective information" [Liu, 2012, 11]. Finally, the entity and aspect level is focused on performing a finer-grained analysis, in order to detect "what exactly people liked or did not like" [Liu, 2012, 11]. Here, the focus does not lie on documents, paragraphs, sentences or clauses, but instead directly on opinion. The assumption is that opinions consist of a sentiment (positive or negative) and a target. The entity and aspect level of sentiment analysis is clearly the most challenging one, even though both document and sentence level sentiment analysis pose to be difficult tasks in their own right.

Sentiment analysis approaches can be further classified into two main categories: machine-learning based approaches and lexicon bases approaches.

Concerning machine-learning based approaches, a further distinction is made between supervised and unsupervised methods. Supervised learning requires training data in order to be trained to classify new instances into predefined classes. In contrast, unsupervised learning is the task of finding a hidden structure in unlabeled data. Since the data is unlabeled, there is no error or reward signal to evaluate a potential solution. Machine learning algorithms require so-called features, e.g. n-grams, part-of-speech tags or patterns in order to be able to classify instances.

Lexicon based approaches rely on sentiment words, also called opinion words. These are words that are used to express positive or negative sentiments, e.g. good, wonderful, bad, or poor. These words are compiled into a lexicon, called sentiment lexicon or opinion lexicon, which can then be accessed by the sentiment analysis algorithm and are used to aggregate a score, assigning either a positive, negative or neutral value to a phrase, sentence or document. Lexicons can also consist of phrases and idioms apart from words. As Liu states, these lexicons alone are not powerful enough to perform high-quality sentiment analysis [Liu, 2012, 12]. Some of the problems with using lexicon-based approaches include:

- positive or negative sentiment words may have opposite orientations in different application domain, e.g. “such” ususally indicates negative sentiment, but it can also imply positive sentiment (“This vacuum cleaner really sucks”)
- A sentence containing sentiment words may not express any sentiment
- Sarcastic sentences with or without sentiment words are hard to deal with
- Many sentences without sentiment words can also imply opinions

Liu [2012, 12-13]

The focus of this paper lies in document-level classification. Each tweet is considered to be a document. Liu [2012] defines the main problem of document-level classification as follows: “given an opinion document d (. . .), determine the overall sentiment s of the opinion holder”. The implicit assumption made here is that each document expresses an opinion on a single entity and contains opinions from a single opinion holder. Since tweets are of exceptionally short range (limited to 140 characters), one can assume that the focus usually lies on a single point of interest and it is therefore assumed that only one topic, or entity is discussed per tweet.

2.2 Twitter

Twitter is a platform on which users share short message, links images or videos and can be defined as a microblog. Users may write short, 140-characters long message which they can then share with their followers. Topic-wise, twitter encompasses everything from personal messages to large companies using the site as a way to communicate with their customers. Similar to Facebook, Tumblr, Google+ or FourSquare, Twitter relies on user-generated content. It was first introduced on July 16 2006 and since then has seen an enormous rate of growth. Today, Twitter has become an invaluable source of information for many researchers in a large variety of fields of sciencs. Twitter is currently the nineh most popular website in the world, with an average of nearly eleven million hits per day [1]. Figure 1 shows the growth rate of twitter:

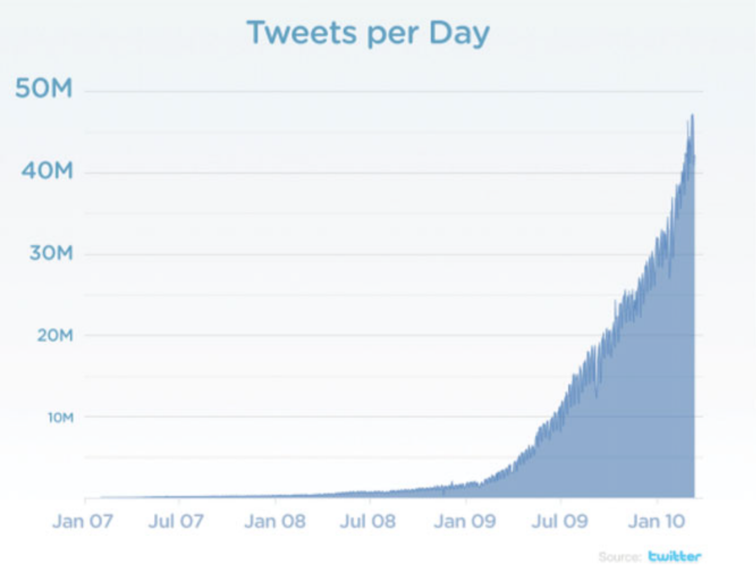


Figure 1: Twitter Growth Rate [Martínez-Cámara et al., 2012, 4]

Tweets may also sometimes reference other users. This is done by preceding the username with an @ (@username), this is called a mention. “Originally, Twitter was created in order for users to write small messages about their daily lives, hence the question ’What are you doing?’ [Martínez-Cámara et al., 2012, 4]. Over time however, Twitter has become much more diverse, representing a powerful tool for spreading information quickly and globally. This has led to Twitter changing the question to ’What’s happening?’ “in order to encourage users to comment on all that is happening around them” [Martínez-Cámara et al., 2012, 4]. Another development towards this is the ability to retweet (RT). This allows users to retweet tweets they

[1] http://www.alexa.com/siteinfo/twitter.com

find worth spreading, making them visible to their own followers. Another important feature of Twitter is the ability to label information, using hashtags. This is expressed by preceding the name of a topic by the # character (e.g. #sentiment-analyis). This then allows users to find other tweets mentioning the same hashtag by clicking on it. The most popular topics or hashtags are aggregated by Twitter under the branch trending topics (TT). Due to this enormous amount of information, Twitter has quickly become the focus of attention of scientists working in NLP and especially sentiment analysis. The applications of such tools, allowing the automatic extraction of opinions and emotions is of high interest to politics, religion, economics, businesses and so on. For example, politicians may use such tools to learn how they are perceived by the public, businesses learn about their customers and how they feel about their products. Tweets have become a recognizable type of text and can be defined by the following characteristics [Martínez-Cámara et al., 2012, 7]:

- The linguistic style of tweets is usually informal, using lots of abbreviations, idioms. The usage of jargon is very common
- Users do not care about the correct use of grammar, which increases the difficulty of carrying out a linguistic analysis
- Because the maximum length of a tweet is 140 characters, ther users usually refer to the same concept with a large variety of short and irregular forms. This problem is known as data sparsity and is a challenge for sentiment-topic task
- The lack of context is a very difficult problem that the sentiment analysis systems have to deal with

Polarity classification (i.e. deciding wheter a tweet expresses a positive or negative sentiment) has become the center of a number of research projects in the last few years. One of the first studies in this area (Go, Bhayani and Huang, 2009) attempted to classify tweets by using supervised machine learning. Due to the difficulty of manually creating a training set (i.e manually annotating labels to individual tweets), they used emoticons that usually appear in tweets in order to differentiate between positive and negative tweets. Read later demonstrated the validity of this approach. The algorithms they used to classify the tweets were support vector machine, naive bayes and maximum entropy. The results were generally positive, drawing some interesting conclusions such as “that the use of part-of-speech tags does not provide valuable information for the classification task for tweets, that the use of unigrams to represent tweets provides very good results and that the results obtained with unigrams can be slightly improved by the combination of unigrams and bigrams" [Martínez-Cámara et al., 2012, 8]. Pak and Paroubek [2010] later used a similar approach by generating a corpus of positive tweets with positive emoticons and negative with negative emoticons. They also conducted a frequency analysis of the different syntactic categories in their corpus over the individual classes. They then used support vector machine, naïve bayes and conditional random fields classifiers to classify the polarity of the tweets. Their conclusion was that the best algorithm to use was naïve bayes, in combination with using n-grams. Another interesting study was done by Zhang et al. in which they propose a hybrid method, combining a lexicon based approach with machine learning. Wheras lexicon based systems have the problem of low recall values, since they rely on the presence of opinionated words in the tweets, machine learning based approaches require large, labeled data sets, which are not easy to acquire or to generate. "To overcome these problems, the authors propose a hybrid system for the analysis of sentence-level opinions on Twitter" [Martínez-Cámara et al., 2012, 10]. They used such techniques as removing retweets, translation of abbreviations into original terms and deleting links, tokenization and morphosyntactic labelling. In order to solve the problem of low recall, they "attempted to identify a greater number of words indicative of subjective content" [Martínez-Cámara et al., 2012, 10]. To do this, they applied the X2 test, the basic idea being that "if a term is more likely to appear in a positive or negative judgment, it is more likely to be a subjective content identifier" [Martínez-Cámara et al., 2012, 10]. Using this method, they were able to increase the number of labeled tweets. They then applied the support vector machine classification algorithm to classify new tweets.

2.3 Machine Learning

Machine learning is a technique that is used in wide range of fields of research. Generally speaking, it is a subfield of computer science closely related to computational statistics, pattern recognition and artificial intelligence. The basic idea is to train an algorithm in such a way that it becomes able to solve a given problem without having to be explicitly programmed. The applications of machine learning range from spam filtering, OCR (optical character recognition), self-driving cars, search engines, speech recognition to classifying DNA sequences. In the past decade or so, it has become immensely popular, partly due to the advancement in processing power of modern computers, making it possible to process larger numbers of data. A basic machine learning system adheres to following structure shown in Figure 2

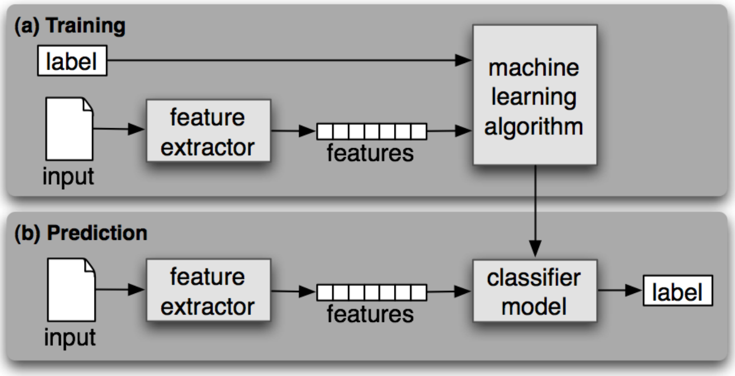


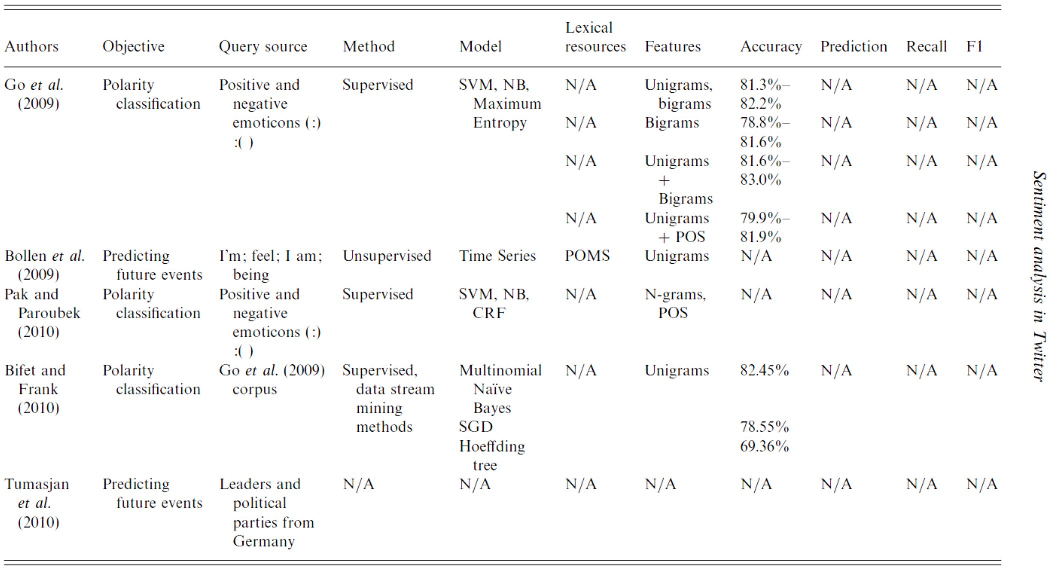
Figure 2: Supervised Classification Bird et al. [2009]

As already mentioned, machine learning can be classified into a number of distinct categories, the two main ones being supervised learning and unsupervised learning. Whereas in supervised learning, the algorithm is presented with training data (or example input according to the problem to be solved) as well as their desired output, in unsupervised learning no labels are given to the algorithm. One further distinction can be made between those two categories, so-called semi-supervised learning, where the algorithm receives an incomplete set of training data, with some of the targeted output missing.
Another way to classify machine learning tasks is according to the sort of problem one wants to solve. Here, one can distinguish between the following cases:

- Classification: categorize the input into two or more classes, e.g. spam filter
- Regression: outputs are continuous rather than discrete
- Clustering: similar to classification, but the categories/groups are not known beforehand
- Density estimation: finding the distribution of inputs in some space
- Dimensionality reduction: simplifying input by mapping it into a lower- dimensional space

Bishop [2010, 32]

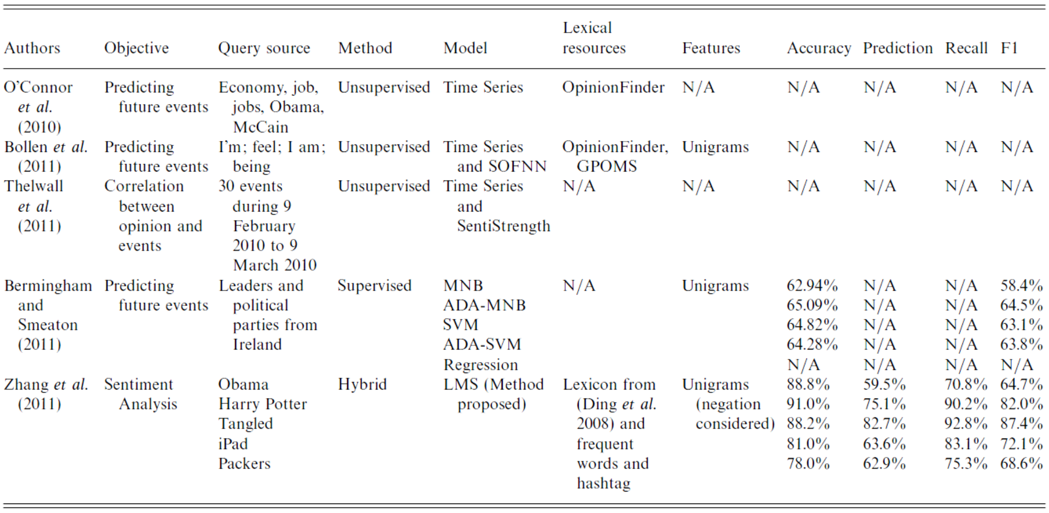
Today, there exist a large number of algorithms and methods one can use in order to perform machine learning. These include for example decision tree learning, artificial neural networks, support vector machines, clustering, Bayesian networks or hidden Markov models. For the purposes of this paper, two different machine learning algorithms shall be presented in closer detail, although to explain them fully would be too extensive, since they are fairly complicated in nature. These two models are Naive Bayes and Maximum Entropy. As mentioned, Twitter has become one of the primary areas of research in the field of sentiment analysis, an overview over existing work concerning sentiment analysis on Twitter can be found in figures 3, 4 and 5.



Authors	Objective	Query source	Method	Model	Lexical resources	Features	Accuracy	Prediction	Recall	F1
Go *et al.* (2009)	Polarity classification	Positive and negative emoticons (:) :( )	Supervised	SVM, NB, Maximum Entropy	N/A	Unigrams, bigrams	81.3%–82.2%	N/A	N/A	N/A
					N/A	Bigrams	78.8%–81.6%	N/A	N/A	N/A
					N/A	Unigrams + Bigrams	81.6%–83.0%	N/A	N/A	N/A
					N/A	Unigrams + POS	79.9%–81.9%	N/A	N/A	N/A
Bollen *et al.* (2009)	Predicting future events	I'm; feel; I am; being	Unsupervised	Time Series	POMS	Unigrams	N/A	N/A	N/A	N/A
Pak and Paroubek (2010)	Polarity classification	Positive and negative emoticons (:) :( )	Supervised	SVM, NB, CRF	N/A	N-grams, POS	N/A	N/A	N/A	N/A
Bifet and Frank (2010)	Polarity classification	Go *et al.* (2009) corpus	Supervised, data stream mining methods	Multinomial Naïve Bayes SGD Hoeffding tree	N/A	Unigrams	82.45% 78.55% 69.36%	N/A	N/A	N/A
Tumasjan *et al.* (2010)	Predicting future events	Leaders and political parties from Germany	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

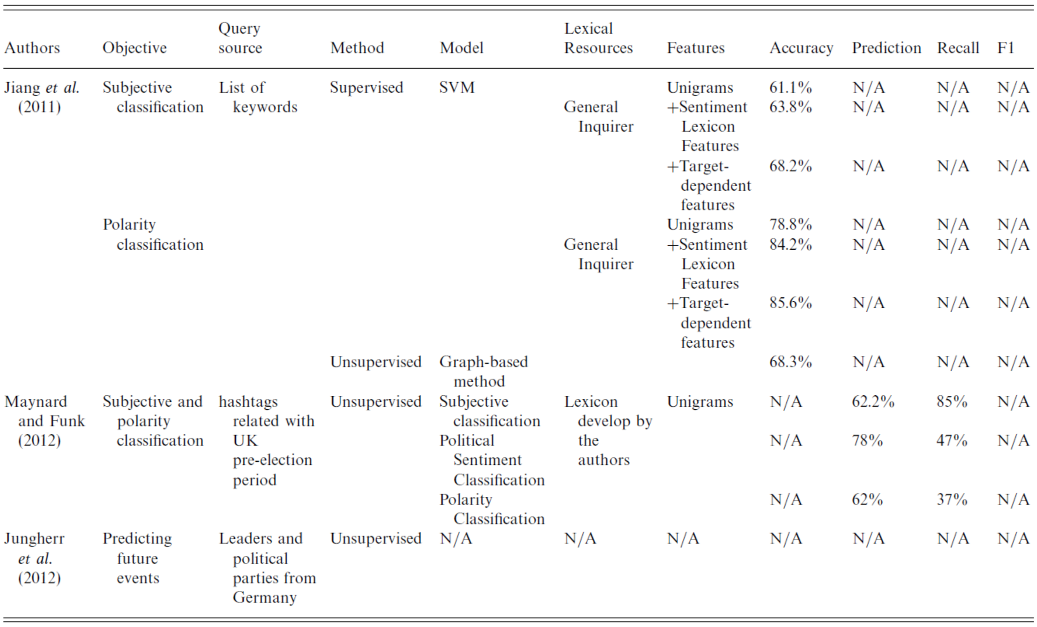
Sentiment analysis in Twitter

Figure 3: Overview of Research [Martínez-Cámara et al., 2012, 21]



Authors	Objective	Query source	Method	Model	Lexical resources	Features	Accuracy	Prediction	Recall	F1
O'Connor *et al.* (2010)	Predicting future events	Economy, job, jobs, Obama, McCain	Unsupervised	Time Series	OpinionFinder	N/A	N/A	N/A	N/A	N/A
Bollen *et al.* (2011)	Predicting future events	I'm; feel; I am; being	Unsupervised	Time Series and SOFNN	OpinionFinder, GPOMS	Unigrams	N/A	N/A	N/A	N/A
Thelwall *et al.* (2011)	Correlation between opinion and events	30 events during 9 February 2010 to 9 March 2010	Unsupervised	Time Series and SentiStrength	N/A	N/A	N/A	N/A	N/A	N/A
Bermingham and Smeaton (2011)	Predicting future events	Leaders and political parties from Ireland	Supervised	MNB ADA-MNB SVM ADA-SVM Regression	N/A	Unigrams	62.94% 65.09% 64.82% 64.28% N/A	N/A N/A N/A N/A N/A	N/A N/A N/A N/A N/A	58.4% 64.5% 63.1% 63.8% N/A
Zhang *et al.* (2011)	Sentiment Analysis	Obama Harry Potter Tangled iPad Packers	Hybrid	LMS (Method proposed)	Lexicon from (Ding *et al.* 2008) and frequent words and hashtag	Unigrams (negation considered)	88.8% 91.0% 88.2% 81.0% 78.0%	59.5% 75.1% 82.7% 63.6% 62.9%	70.8% 90.2% 92.8% 83.1% 75.3%	64.7% 82.0% 87.4% 72.1% 68.6%

Figure 4: Overview of Research [Martínez-Cámara et al., 2012, 22]



Authors	Objective	Query source	Method	Model	Lexical Resources	Features	Accuracy	Prediction	Recall	F1
Jiang *et al.* (2011)	Subjective classification	List of keywords	Supervised	SVM		Unigrams	61.1%	N/A	N/A	N/A
					General Inquirer	+Sentiment Lexicon Features	63.8%	N/A	N/A	N/A
						+Target-dependent features	68.2%	N/A	N/A	N/A
	Polarity classification					Unigrams	78.8%	N/A	N/A	N/A
					General Inquirer	+Sentiment Lexicon Features	84.2%	N/A	N/A	N/A
						+Target-dependent features	85.6%	N/A	N/A	N/A
			Unsupervised	Graph-based method			68.3%	N/A	N/A	N/A
Maynard and Funk (2012)	Subjective and polarity classification	hashtags related with UK pre-election period	Unsupervised	Subjective classification	Lexicon develop by the authors	Unigrams	N/A	62.2%	85%	N/A
				Political Sentiment Classification			N/A	78%	47%	N/A
				Polarity Classification			N/A	62%	37%	N/A
Jungherr *et al.* (2012)	Predicting future events	Leaders and political parties from Germany	Unsupervised	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Figure 5: Overview of Research [Martínez-Cámara et al., 2012, 23]

2.3.1 Naive Bayes

Naive Bayes classifiers are a group of classifiers which are based on Bayes' theorem, operating under the strong (naive) independence assumption between features. It is one of the more simple techniques used in machine learning. They are used to assign class labels to problem instances and utilize feature values represented as vectors. The core principle of these classifiers is the assumption that the value of a feature in independent of the value of any other feature. In other words, each feature contributes independently to the probability of the class decision, that is, correlations between the features are disregarded.Figure 6 shows the procedure the algorithm uses to choose the label for a document.

As Bird et al. [2009] state: "In the training corpus, most documents are automotive, so the classifier starts out at a point closer to the "automotive" label. But it then considers the effect of each feature." In this example, the classifier tries to decide between the three categories 'sports documents', 'murder mysteries' and 'automotive'. The input document contains 'dark', which is a (weak) indicator for murder mysteries, but also the word 'football', which is a stronger indicator for sports documents. Finally, after all the features have been applied, the classifier assigns a label to the input.

The naïve Bayes classifier is commonly used in text categorization (deciding whether

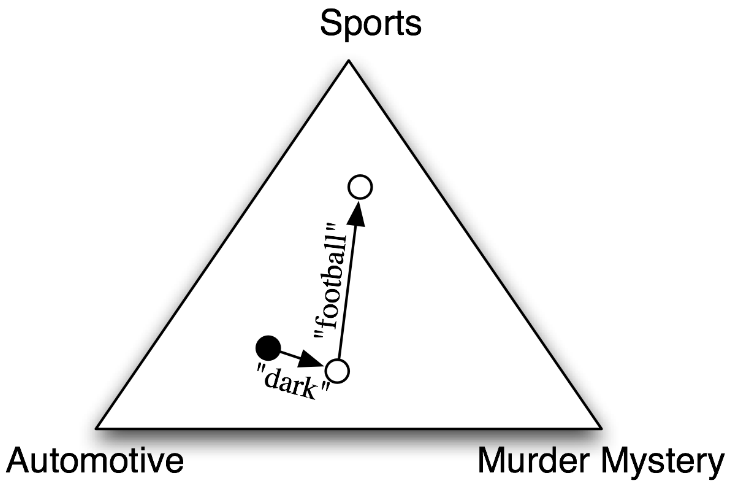


Figure 6: Bayes Classifier Bird et al. [2009]

a given document belongs to a certain category or the other), using word frequencies as the features. According to Rennie et al. [2003], "it is competitive in this domain with more advanced methods including support vector machines."

2.3.2 Maximum Entropy

Maximum entropy classifiers utilize a classification method called multinomial logistic regression, which generalizes logistic regression (a statistical model used to predict a binary response) to multiclass problems. It is commonly used in NLP, especially in information retrieval and speech retrieval problems. It is based on the principle of maximum entropy, which means that it selects the model, created from the training data which has the largest entropy. "Labeled training data is used to derive a set of constraints for the model that characterize the class-specific expectations for the distribution" [Nigam et al., 1999, 1]. In contrast to the naïve Bayes classifier, the maximum entropy classifier does not assume the independence of the features towards each other. This is the main advantage of the maximum entropy classifier over naive Bayes, since sometimes, features may correlate with each other and incorporating these potential relationships would help improve the efficiency of the classifier. For example, using sentence length as a feature for classifying some sort of text amongst other features, such as for example syntactic complexity of sentences. Since syntactic complexity has a correlation with sentence length, the maximum entropy classifier would perform better, due to its ability to include this information in the model building process, whereas naïve Bayes classifiers would assume these features to be independent of each other.

2.4 Research Questions

The main goal of this paper is to present a comparison of different methods for sentiment analysis of German tweets. In order to do so, a relatively large amount of data had to be acquired. Since there are not many German language resources available for Twitter, a corpus of tweets had to be created which could be used as training data for the two different classifiers that were used in order to classify tweets concerning their polarity. The main question here is whether it is possible to attain acceptable results doing sentiment analysis for German tweets by using training data that is automatically acquired, based on emoticons pertaining to either positive or negative emotions, an approach similar to Pak and Paroubek [2010]. Another question that shall be answered is how well does a lexicon based approach perform in comparison to machine learning based approaches. Furthermore, the possibility of combining a lexicon based approach with machine learning approaches and possible improvements thereby are explored.

3 Data and Methods

In order to build a sentiment analysis system, one either needs a lexicon consisting of sentiment words or opinion words, or a large set of labeled date which can be used to train the classifiers. In the following sections, both the lexicon that was used in this paper as well as the acquisition process of the data will be explained in detail. Furthermore, the individual systems that were created in order to perform sentiment analysis will be presented and explained. Finally, the evaluation process will be illustrated.

3.1 Lexicon-based Approach

The lexicon that was used was provided by Clematide and Klenner. It consists of about 8'000 words, labeled according to their polarity and polarity strength values. The word classes are nouns, verbs and adjectives. In Clematide and Klenner [2010], the authors automatically extended an existing lexicon by using synsets from GermaNet (which is a WordNet-like lexical database). They "carried out experiments to automatically learn new subjective adjectives together with their polarity orientation and polarity strength, by applying a corpus-based approach that works with pairs of coordinated adjectives extracted from a large German newspaper corpus"[Clematide and Klenner, 2010, 1]. The structure of the lexicon is as follows:

Word {NEG,POS,NEU,SHI,INT}=PolarityStrength PoS
SHI for Shifters, INT for Intensifiers
INT <1, e.g. 0.5 is a reduction factor, >1, e.g. 2 is a gain factor

Thus, an entry looks like this:

beeindrucken POS=0.7 verben

Which means that the word 'beeindrucken' has a positive value of 0.7, with values ranging from 0 to 1. For the purposes of the system created for this paper, shifters and intensifiers were neglected.

The lexicon was extended with a number of positive and negative emoticons, in order to measure if this yields better results for the lexicon based approach. As mentioned, emoticons seem to be good discriminators for deciding the positive or negative polarity of a sentence and since tweets mostly consist of only one sentence, this could potentially improve the lexicon based approach. In order to test the classifier, a labeled set of tweets, provided by Narr et al. was used. These tweets were human-annotated by three Mechanical Turk workers according to their polarity. The tweets are aggregated in a .tsv file. The files lex.py and lex2.py contains the methods handling the processing of the tweets and their classification. The basic structure of the script is as as shown in Figure 7.

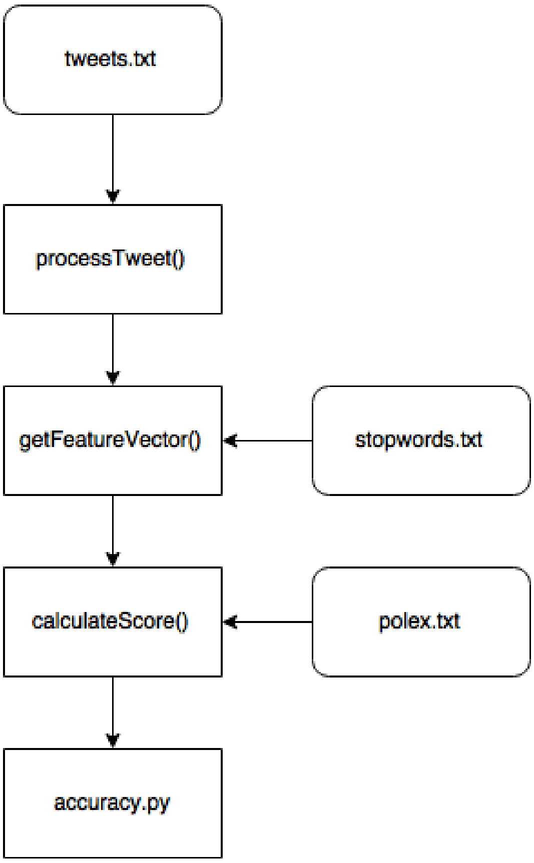


Figure 7: lex.py structure

After loading the tweets from tweets.txt, they are preprocessed. First of all, all text is converted to lower case. Then, all URLs are converted to the string 'URL'. @username is converted to the string AT_USER. Additional white space are then removed and hashtags are replaced with the string following the # symbol. In the next step, the feature vector for each tweet is generated, that is the individual tokens are stored in an array. At this point, stopwords are removed. The stopwords used are provided by Marco Götze and Steffen Geyer[1]. Additionally, words containing string sequences which contain letters that are repeated more than two times are reduced to two letters. All punctuation is removed as well. Next, the score for the tweet is calculated by the calculateScore() method. Basically, it checks for each word if it appears in polex.txt (and polex_emote respectively for lex2.py) and if it does, it takes the polarity value noted there, thus aggregating the sum of these values. If the final value of the whole tweet is greater than 0, the tweet is labeled as positive, if the sum is smaller than 0 as negative and if the sum is exactly 0, it is labeled neutral. Finally, the results of the classification is compared to the pre-labeled data and the accuracy of the classifier is calculated..

3.2 Machine Learning Approaches

The following sections describe the methods used for constructing the machine learning based approached for the sentiment analysis task. First of all, the data acquisition process is described, followed by detailed descriptions of the individual machine learning techniques that were used.

3.2.1 Data Acquistion

In order to train the classifiers, a relatively large amount of data had to be acquired, i.e. a corpus of tweets, which are labeled according to their polarity is necessary to train the naïve bayes, maximum entropy and support vector machine based classifiers. Since there is no (large enough) freely available corpus of tweets available for German language tweets, the decision was made to create a corpus using a similar approach as Pak and Paroubek [2010], i.e using tweets that contain either positive or negative emoticons. By accessing the Twitter api, crawling tweets is fairly simple and a large number of them can be downloaded quickly. Using a simple search string containing either positive (e.g. :), :D, =), ;), :]) or negative (e.g. :(, :(() emoticons (the full list can be found in crawl.py and crawlneg.py). This yielded a total amount of 62'685 tweets containing positive emoticons and 56'186 tweets containing negative emoticons. Of course, this method of acquiring positive and negative tweets by looking for positive/negative emoticons will yield also a large number of tweets that are either not positive, even though they contain positive emoticons or are not negative when containing negative ones, as well as tweets that potentially contain both positive and negative emoticons. Still, the assumption here is that most tweets

[1] http://solariz.de/649/deutsche-stopwords.htm

containing emoticons can be classified according to them, since most tweets contain no more than one sentence, and if this sentence contains either a positive or negative emoticon, one can assume the polarity of the whole sentence to be in accordance with that particular emoticon.

3.2.2 Naive Bayes Classifier

The Natural Language Toolkit (NLTK), a suite of libraries for python, that can be used for a wide array of NLP tasks was used in order to implement the classifiers. The NLTK is intended to "support research and teaching in NLP or closely related areas, including empirical linguistics, cognitive science, artificial intelligence, information retrieval and machine learning" [Bird et al., 2008, 2]. The NLTK also encompasses a book, describing the underlying theories and methods that are implemented in the python modules. The python script bayes.py contains the methods used for training the classifier as well as for classifying the test set. The basic structure is as shown in Figure 8

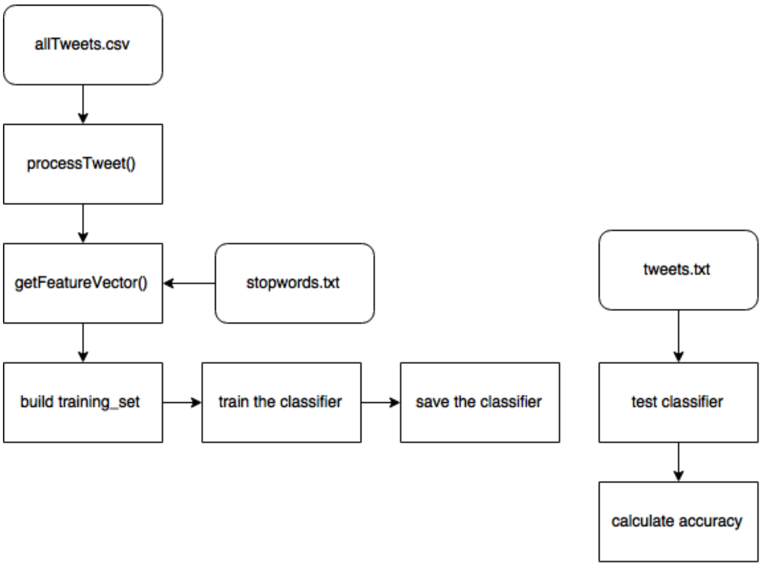


Figure 8: bayes.py structure

Due to the processing limitations of the system that was used to classifiy the tweets, the amount of tweets had to be reduced. In order to measure the effect of the amount of data that is used as input for the system, two classifier instances were trained: one with an input of 10'000 positive and 10'000 negative tweets and one classifier with an input of 15'000 positive and 15'000 negative tweets. Similar to the lexicon based system, the tweets are first preprocessed, additionally, all emoticons are removed. After the feature vector is generated, consisting of unigrams, the training set is built, which is then passed on to the classifier in order to train the model. The classifier is then saved so that it can be reused later to classify the test set. The human annotated set of tweets (test_tweets.txt) is then passed to the classifier and finally, the accuracy is calculated. In addition, a hybrid system, using both the naive Bayes classifier as well as the lexical approach was created. Here, the results of the naive Bayes classifier (i.e. the output, either 'positive' or 'negative' is added to the calculateScor() method from the lexicon based approach. In the case of a positive output of the Bayes classification algorithm, a positive value of 0.5 is added to the aggregated score, in the case of a negative output a value of -0.5 is added.

3.2.3 Maximum Entropy Classifier

The method for training the Maximum Entropy Classifier is similar to the one used for the Naive Bayes Classifier. The same feature vector is used, including all the steps leading up to that point. The Maximum Entropy Classifier is then trained, using the training set and utilizing the improved iterative scaling (IIS) algorithm, a "hillclimbing algorithm for calculating the parameters of the classifier given a set of constraints."[Nigam et al., 1999, 2]. A complete description and derivation of the IIS algorithm is presented by Pietra et al. [1997]. Due to the potentially complex interactions between related features, Maximum Entropy classifiers "choose the model parameters using iterative optimization techniques, which initialize the model's parameters ro random values, and then repeatedly refine those parameters to bring them closer to the optimal solution" [Bird et al., 2009]. Since there is no way of determining when this optimal solution is reached (i.e. after which amount of iterations), one has to manually set the number of iterations. For the purposes of this paper, an external classifiaction algorithm, MEGAM was chosen. MEGAM is an implementation based on the Ocaml system, the main implementation of the Caml language, which is a general-purpose programming language. The algorithms used by MEGAM are much more efficient than the standard iterative scaling techniques used by the NLTK . As Daumé [2004] writes: "It has been recognized that the typical iterative scaling methods used to train logistic regression classification models (maximum entropy models) are quite slow." MEGAM is freely available [2] online and may be used for any research purposes. Still, due to processing restrictions (optimization techniques may take a long time to learn, especially when the training set and the number of features are high), the number of iterations was set to 5. Therefore, the training data had to be reduced even more than for the Naive Bayes

[2] http://www.isi.edu/ hdaume/megam/

classifier, since the amount of input tweets was to large for the system to handle. The set was thus reduced to 10'000 positive and 10'000 negative tweets. After training the classifier, it is saved and then given the test set. Finally the accuracy score is calculated. Similar to the Bayes/Lexicon hybrid method, the maximum entropy method was combined with the lexicon based approach as well. The output of the maximum entropy classifier was included to the calculation of the score from the lexicon based system, adding either 0-5 or -0.5 for a positive or negative output respecitvely.

4 Results and Analysis

The following chapter contains the results acquired through the various methods that were tested for the sentiment classification task. First, all results will be presented, followed by in-depth explanations of the individual systems.

4.1 Lexicon based Approach Results

The lexicon based approaches (lex.py and lex2.py) were both tested on the human annotated test set (1688 individual tweets). In order to analyze the system in detail, it was tested on all tweets, all tweets minus neutral tweets (this was done in order to compare the results to the other approaches, which discriminate only between positive and negative tweets), only positive tweets, only negative tweets and only neutral tweets. Table 1 shows the results of the accuracy measurements, both of lex.py and lex2.py.

test set	lex.py	lex2.py
All	0.6	0.63
All without neutral	0.22	0.4
Positive	0.23	0.5
Negative	0.2	0.24
Neutral	0.8	0.75

Table 1: Accuracy scores of lexicon based systems

As can be seen, the results of the baseline system (lex.py) were overall quite low. Even though the value over all tweets (positive, negative and neutral) is relatively high, with an accuracy score of 0.6, the value for the set of tweets without neutral instances, as well as for only the positive and only the negative instances are very low (0.22, 0.23 and 0.2). The main reason why the overall accuracy score is at 0.6, is the relatively high score for neutral instances. This means that the system performs relatively badly for discriminating between positive and negative tweets and therefore classifies most tweets into the neutral category, mainly due to the limited range of the sentiment lexicon that was used, as well as the shortness of twitter messages, which makes it difficult for the classifier, since there are only few words that can contribute to the polarity score. The results for the system incorporating emoticons into the lexicon are slightly improved over the baseline. (+0.03 overall tweets). The largest improvement can be seen in the positive category (+0.27), but the recognition of negative instances improved as well (+0.04). On the other hand, the system lost 0.05 points of accuracy in the neutral category, implying the wrong categorization of tweets due to the presence of positive or negative emoticons. Due to the limited size of the test set, it is difficult to tell whether the larger value in contrast to the negative set for the positive category has reasons other than the higher frequency of positive emoticons in the test data. Nevertheless, the simple method of extending the basic sentiment lexicon with emoticons and incorporating it into the baseline system, has shown to produce improved results.

4.2 Naive Bayes Results

Table 2 shows the results of the accuracy measurements of the four classifiers using the Naive Bayes algorithm as well as the two lexicon based baseline systems.

test set	lex	lex2	bayes1	bayes2	bayeshybrid1	bayeshybrid2
All without neutral	0.22	0.4	0.5	0.49	0.61	0.62
Positive	0.23	0.5	0.46	0.48	0.67	0.69
Negative	0.2	0.24	0.55	0.51	0.52	0.51

Table 2: Accuracy scores of naive Bayes based systems

As can be seen, the results for the standard Bayes models are rather low, with values around the 0.5 accuracy mark. Interestingly, in both systems, detection of negative tweets yielded the highest accuracy. Furthermore, enlarging the training data did not seem to improve the results. Accuracy even declined, although not significantly. Concerning the hybrid systems, scores were on average around 2 points higher. In comparison with the lexicon based systems, the hybrid systems improved by a fair margin. This has mainly to do with the fact that the machine learning part of the system prevents the system to classify tweets as neutral when there is no word in the tweet that is also present in the polarity lexicon, i.e. it circumvents the data scarcity of the lexicon. Interestingly, the hybrid system using the classifier trained with more data performed a little bit better here, although only marginally. Thus, the hybrid systems seem to produce acceptable results, whereas the standard models only reach values that are close to randomness. Nevertheless, the accuracy values attained by the naive Bayes classifier are still nearly double as high as the baseline lexicon based systems (which suffer from data scarcity) and the results show that using a machine learning classifier in complement with a fairly simple lexicon based approach, using automatic retrieval methods, produces acceptable results.

4.3 Maximum Entropy Results

Table 3 shows the results of the accuracy measurements of the two classifiers using the maximum Entropy algorithm as well as the two lexicon based baseline systems.

test set	lex	lex2	maxent	maxenthybrid
All without neutral	0.22	0.4	0.44	0.65
Positive	0.23	0.5	0.1	0.59
Negative	0.2	0.24	0.95	0.72

Table 3: Accuracy scores of maximum enrtopy based systems

The basic maximum entropy system, trained on unigram features performed in the same range as the naïve Bayes classifier, altough the accuracy scores were a little bit lower. Especially concerning the positive category, accuracy was extremely low. The reasons for this are unclear. Most of the test tweets were classified as negative, reaching an accuracy score of 0.95. Overall, the performance is underwhelming, considering the large amount of data that was used as input, implying either the inadequacy of the training data or a classification error due to the limited amount of input features. On the other hand, the hybrid system, a combination of the lex2 and maxent systems, using the classification output of the machine learning algorithm as an additional input source for the lexicon based system, performed significantly better, reaching an overall accuracy of 0.65, similar to the naivy Bayes/lexicon hybrid system. Here, the low accuracy in the positive category of the standard system is revoked, reaching an accuracy score of 0.59. Overall, the maximum entropy hybrid system performed better than all the other systems, even though the standard system did not perform as well as (both) naïve Bayes standard systems. Additionally, another model was trained using the maximum entropy classifier and a larger amount of data (15'000 positive and negative tweets), but no improvement in the accuracy resulted thereof.

5 Discussion

The lexicon based approaches described in this paper, utilizing a polarity lexicon in order to classify tweets, produced results in the range of 0.6 to 0.63 accuracy for discriminating between positive, negative and neutral tweets. Overall, considering the simplicity of this approach, these results are in the vicinity of what was expected. After all, these systems were implemented mainly as a baseline systems to which to compare the machine learning systems as well as for integration/combination with the machine learning approaches. Still, when looking at the results of the classification when neglecting the neutral category, thus changing the taski into a binary classification problem, it becomes clear that the lexicon based systems mainly reached a relatively high accuracy over the three classes, due to the high number of tweets that were classified as neutral. The reason for this is the limited extent of the polarity lexicon that was used, i.e. in many tweets there were no words present that were part of the lexicon. Additionally, the system would have potentially profited of performing tokenization, since sometimes the token of a word appearing in a tweet may have been present in the lexicon. Furthermore, additional features, such as counting elongated words (i.e. words with one character repeated more than two times, e.g. 'sooooo'), negation resolution (i.e. connecting negations with the following word, thus preventing it to be counted as positive if it appears in the lexicon) or including punctuation as a signal of increased sentimental value (e.g. increasing the score of a positive sentiment word when it is followed by exclamation marks) could have been used to improve the results of the lexicon based approach. Concerning the results of the standard machine learning classifiers (naive Bayes and maximum entropy), one has to say that the performances are not on par with the results of other works, e.g. Bifet and Frank [2010] or Go et al. [2009], with accuracy scores reaching barely the 0.5 threshold. The main reason for this has to be the data that was used in order to train the models. Since the main problem of doing sentiment analysis in German is the (in)availability of suitable data, the decision was made to experiment with automatic data acquisition, using emoticons in order to create a training set of positive and negative tweets, similar to the approach used by Pak and Paroubek [2010], which produced acceptable results for English Twitter sentiment analysis. In this case, the results imply that the training data was not efficient in training the classifiers, since the results are relatively low. Another reason for the low performance may also be the test set that was used, which was fairly limited in size (around 600 tweets excluding the neutral category). Another reason for the poor performance of these systems may be the decision to only use word unigrams. A number of additional features could be incorporated into these systems, potentially improving their performance regardless of the inadequacy of the training data. Potential features that could be used include extension of the unigram approach to bigrams or n-grams, integration of all-caps (i.e. counting the number of words with all characters in upper case), counting the number of elongated words, hashtags, negation contexts, part-of-speech tags, punctuation (e.g. the number of sequences of exclamation marks, question marks and so on) and character ngrams. On the other hand, the results for the hybrid methods showed that this approach may prove to be efficient, even if suitable, human annotated data for the specific language is not available. By combining the lexicon based approach with the two machine learning approaches, the performance of the systems was improved by a fair margin. With accuracy values of around 0.65, these systems, still using only a simple approach, both concerning the lexicon based approach as well as the machine learning methods (only using unigrams as features), could potentially be further improved by integrating additional features (as mentioned above).

6 Conclusion

In the paper, the possibility of using automatically acquired data for training machine learning algorithms to be used in sentiment analysis of tweets was explored. The data was acquired by crawling German language tweets and classifying them into the two categories positive and negative, according to the presence of either positive or negative emoticons. The machine learning algorithms were then tested using a human annotated test set and compared to a baseline lexicon based system. The two approaches were also combined into to hybrid system, incorporating both lexicon lookup as well as machine learning. The results have shown that both the lexicon based approach, as well as the two machine learning approaches did not perform very good on their own. In case of the lexicon based approach, the performance was improved by incorporating emoticons into the polarity lexicon. Especially the machine learning methods did not perform as the results of other works, using similar algorithms, would suggest. This implies that the process of automatically acquiring Twitter data to be used as a training set for machine learning, has to be either improved, or is not worthwhile at all. Furthermore, the combination of the lexicon based and machine learning based systems yielded results approaching accuracy values of around 0.65. These results are promising, since even though the training data for the machine learning algorithms was suboptimal, and the lexicon based systems perform poorly on their own, suffering from data scarcity in the lexicon, the combination of the two methods seems to improve performance. Furthermore, only basic features were used in the machine learning approaches, that is only the frequency of unigrams in the tweets. By extending the feature sets by more complex features, such as bigrams, part-of-speech tags and so on, the performance of both the standard machine learning approaches as well as of the hybrid systems may very well be improved further. Nevertheless, it was shown that using automatically acquired data can be used in order to train classifiers, even though postprocessing of the data may be necessary in order to assure the quality of the training set. Since research on German language Twitter sentiment analysis is very limited at the current time, especially in comparison with, for example, English, this is a first step into expanding the possibilities of research in this area. A freely available twitter corpus for German would go a long way to support the development of better sentiment analysis techniques for German and a method for building such a corpus automatically would be a valuable contribution towards this goal. Even though the results of the systems tested here may not be on par with other state-of-the-art publications, they constitute a first step towards more complex systems, using the automatically acquired data. In order to test this approach of automatic corpus generation, the machine learning classifiers have to be tested with a wider range of input features, so as to see if their performance can be improved further or if the problem lies in the data itself. Still, the hybrid methods show promising results even with potentially skewed data, thus further research into these methods seems to be necessary.

References

Asur, S. and Huberman, B. A. (2010). Predicting the future with social media. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '10, pages 492–499, Washington, DC, USA. IEEE Computer Society.

Bifet, A. and Frank, E. (2010). Sentiment knowledge discovery in twitter streaming data. In *Proceedings of the 13th International Conference on Discovery Science*, DS'10, pages 1–15, Berlin, Heidelberg. Springer-Verlag.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

Bird, S., Loper, E., Klein, E., and Baldridge, J. (2008). Multidisciplinary instruction with the natural language toolkit. In *In Proceedings of Workshop on Issues in Teaching Computational Linguistics*.

Bishop, C. (2010). *Pattern Recognition and Machine Learning*. Springer.

Bollen, J., Mao, H., and Zeng, X.-J. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*.

Clematide, S. and Klenner, M. (2010). Evaluation and extension of a polarity lexicon for german. In Montoyo, A., Martínez-Barco, P., Balahur, A., and Boldrini, E., editors, *Proceedings of the 1st Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA)*, pages 7–13, Lisbon, Portugal.

Das, S. and Chen, M. (2001). Yahoo! for amazon: Extracting market sentiment from stock message boards. *Proceedings of the Asia Pacific finance association annual conference (APFA)*, 35:43.

Daumé, H. (2004). Notes on CG and LM-BFGS optimization of logistic regression. Paper available at http://pub.hal3.name#daume04cg-bfgs, implementation available at http://hal3.name/megam/.

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

Indurkhya, N. and Damerau, F. J. (2010). *Handbook of Natural Language Processing*. CRC Press, 2nd edition.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan amd Claypool.

Liu, Y., Huang, X., An, A., and Yu, X. (2007). Arsa: A sentiment-aware model for predicting sales performance using blogs. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 607–614, New York, NY, USA. ACM.

Martínez-Cámara, E., Martín-Valdivia, M. T., Ureña-López, L. A., and Montejo-Raéz, A. (2012). Sentiment analysis in twitter. *Natural Language Engineering*, 20(01):1–28.

McGlohon, M., Glance, N., and Reiter, Z. (2010). Star quality: Aggregating reviews to rank products and merchants. In *Proceedings of Fourth International Conference on Weblogs and Social Media (ICWSM).*

Narr, S., Hulfenhaus, M., and Albayrak, S. (2012). Language-independent twitter sentiment analysis. *Knowledge Discovery and Machine Learning (KDML), LWA*, pages 12–14.

Nigam, K., Lafferty, J., and McCallum, A. (1999). Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67.

O'Connor, B., Balasubramanyan, R., Routledge, B. R., and Smith, N. A. (2010). From tweets to polls : Linking text sentiment to public opinion time series.

Pak, E. and Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *In Proceedings of the Seventh Conference on International Language Resources and Evaluation.*

Pietra, S. D., Pietra, V. D., and Lafferty, J. (1997). Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393.

Read, J. (2005). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*, pages 43–48. Association for Computational Linguistics.

Rennie, J. D. M., Shih, L., Teevan, J., and Karger, D. R. (2003). Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 616–623.

Tumasjan, A., Sprenger, T., Sandner, P., and Welpe, I. (2010). Predicting elections with twitter: What 140 characters reveal about political sentiment.

Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., and Liu, B. (2011). Combining lexiconbased and learning-based methods for twitter sentiment analysis. *HP Laboratories, Technical Report HPL-2011*, 89.

Zhang, W. and Skiena, S. (2010). Trading strategies to exploit blog and news sentiment. In *In Fourth Int. Conf. on Weblogs and Social Media (ICWSM).*