Measuring Customer Sentiment on Twitter

Olga Mierzwa 356344

Econometric Institute
Erasmus School of Economics
Erasmus University Rotterdam

Supervisor Dr. Michel van de Velden
Co-reader Dr. Flavius Frasincar

2012

Acknowledgements

The quality of this Master's thesis was greatly enhanced by the gracious assistance and guidance of Dr. Michel van de Velden. I am thankful to him for his interest and help.

I am also grateful to my colleagues at Veneficus who have provided me with programming support.

Abstract

Twitter, a microblogging platform, allows its users to post short messages about any topic and follow others to receive their posts. By means of Twitter people communicate with each other. The goal of this research is to study customer sentiment expressed on Twitter and to develop a framework that allows monitoring it in the real-time. For this purpose 9368 tweets are collected from the KLM Twitter account. Tweets preparation including among others spelling correction, synonym substitution, hyperlink deletion and stop words is performed. Sentiment is manually categorized the sentiment into three classes: objective, positive and negative, in order to create the training set for the classifier. The tweets are classified using linear Support Vector Machines. The classifier obtains 82% precision for the objective class, 59% for the positive and 54% for the negative class. The classified tweets are used to create the positive and negative emotion indexes over time. Looking at the different subset of features created by the ranking algorithm shows how different words influence the prediction and helps to gain insight into the classification. It is shown that the data preparation improves both the precision and recall of the classification. The spelling correction and synonym substitution improves the precision for the negative class of tweets by up to 8%. Relating the predicted sentiment to various events such as the introduction of a new service or operational issues with flights showed how such events influence customer opinion.

Contents

List of Figures

List of Tables

1 Introduction

The explosive growth of social media, where people communicate with each other across various platforms, creates new opportunities to monitor people's views and their personal opinions in real time. The rapid adoption of social media can be best highlighted when put in the context of the adoption of other technologies. Telephones, for example, came to the market in 1876 and it took 89 years to reach 150 million users. Twitter, on the other hand, reached 150 million users in less than four years.

Social media in contrast to traditional advertising platforms, offers users the possibility to freely express opinions, which for companies means that they no longer have control over the published content in cyberspace (Kaplan and Haenlein, 2010). Customers who are dissatisfied with the services or goods that a company offers often engage in virtual complaints in the form of protest websites or blogs (Ward and Ostrom, 2006). This results in the availability of potentially damaging information. On the other hand, there are also customers who manifest their positive attitude towards the brand and want to compliment the company on the quality of offerings. As indicated by Dutta (2010) the on-line discussion around a firm can take place even when it is not participating in social media. An extreme situation, which results from a lack of supervision over social media, involves brand hijacking. An example of this is the case of Exxon Mobil, which was represented on social media by a person not authorized by the company, but introducing herself as a firm's legitimate representative (Thomases, 2010).

Nowadays companies need to not only have a presence on social media, but also be able to instantly follow, monitor and adapt to on-line discussion and learn from this new source of information(Kaplan and Haenlein, 2010). The advent of social media has created a need for new on-line strategies, altered marketing actions and a reallocation of marketing resources. Customer sentiment analysis from content posted on social media offers an automatic, fast, free and large-scale addition to current toolkits that facilitate customer satisfaction studies. This makes social media relevant not only for large corporations, but also for small and medium size firms, and even non-profit and government organizations.

Due to the relative novelty of social media, a majority of business units either do not listen to their customers' opinions on social media, or use inadequate measures, such as number of fans and likes on Facebook, or followers on Twitter, to quantify popularity.

With a focus on social media in the area of micro blogging, we intend to investigate how measures of on-line sentiment obtained from information posted on Twitter relates to actual customer satisfaction. More precisely, in this thesis we propose a framework to measure customer sentiment towards the KLM brand. Monitoring customer sentiment and understanding its relationship with

the airline industry is expected to be an important issue. According to the findings presented by ASCI [1], airlines exhibit one of the lowest levels of customer satisfaction in the transportation sector.

To provide an answer to the research question we aspire to measure customer sentiment by applying multi-class pairwise Support Vector Machines (SVMs) to the collection of tweets posted to KLM (including @KLM or #KLM). SVMs achieve high performance on text classification problems and are considered a current state of art system in data mining (Rich and Niculescu-Mizil, 2006; Tong and Oles, 2001).

Analysis of content from Twitter requires solid data preparation, due to the noisiness of the posted information (no grammar structure, spelling mistakes, abbreviations and use of Internet slang). We address this challenge by conducting extensive tweet "cleaning", which, among other processes, involves deleting duplicates, spelling correction and synonym substitution. This process of tweet preparation results in an improvement of the data quality and an increase of the frequency of specific features. Furthermore, we verify that the data preparation results in a higher performance of the classifier.

The feature selection process also simplifies interpretation of the classification. Gaining insight into which words contribute to the prediction is very important when dealing with "black box" classifiers. Better understanding of the terms involved in the classification could be used in future sentiment retrieval. Words found to be highly influential could be used to create or update the key-words list, prepare new marketing campaigns and spot emerging trends.

The thesis is organized as follows. In Section 2, we conduct a literature review and present the work that has been done in the field of sentiment analysis and micro blogging. Section 3 describes the data and the process of its preparation for the analysis. In Section 4, we present the methods applied for the spelling corrections and classification. Section 5 reports the results. Finally, we formulate conclusions and future research avenues in Section 6.

2 Literature and Previous Research

Available textual information could be put into one of two major categories: facts or opinions. Facts are defined as events or their properties that have actually occurred; they are objective expressions that can be verified and proven. Opinions are subjective beliefs, and are the results of emotion or interpretations of facts. While an opinion may be supported by an argument, counter opinions can often be drawn from the same set of facts. The concept of an opinion is very broad. In this section, we focus on previous research in the field of subjectivity and sentiment analysis.

[1] http://www.theacsi.org/

First, we make an introduction to content analysis. Later, we discuss work that has been done specifically pertaining to social media, with a focus on micro blogging.

2.1 Domain of Sentiment Analysis

As an area of research, sentiment analysis can be considered a part of computational linguistics, natural language processing (NLP), and text mining (Mejova, 2009). Generally speaking, sentiment analysis aims to uncover the author's attitude towards some subject or the overall contextual polarity of a text. This may be a judgment or an evaluation, an affective state or an intentional emotional communication.

Part of the work done on sentiment analysis focuses on the expression of a private state (Wiebe, 1994). It was defined by Quirk et al. (1985) as a state that is not open to objective observation or verification. Opinions, evaluations, emotions, and speculations all fall into this category. A classic example of subjectivity analysis is the detection of opinionated text in order to distinguish it from objective (Finn, 2003; Ng et al., 2006; Ni et al., 2007).

The problem is formulated as such: given an opinion-oriented document or text, whose overall opinion applies only to one item or idea, categorize its attitude by assigning it to one of the two contrasting sentiment polarities, or locate it on a scale between these two polarities (Pang and Lee, 2008). This classification task of tagging a text as containing either a positive or a negative view is called sentiment polarity classification or polarity classification.

Mihalcea et al. (2007) summarize the evidence of several projects on content analysis (Sebastiani and Esuli, 2006; Takamura et al., 2006) as follows "the problem of distinguishing subjective versus objective instances has often proved to be more difficult than subsequent polarity classification, so improvements in subjectivity classification promise to positively impact sentiment classification".

The most widely studied texts are product and movie reviews (Beineke et al., 2004). Such documents have well-defined topics and in addition to the sentiment expressed in the language they also contain the author's rating. This label gives a quantitative indication of the author's opinion. Such documents are usually used as a best case example when measuring sentiment (Chevalier and Mayzlin, 2006; Houser and Wooders, 2001; Hu et al., 2006).

Sentiment classification on these documents can be formulated as a supervised learning problem with two classes: positive and negative. Usually, a review with 4-5 stars is viewed as a positive review ("thumbs up"), and a review with 1-2 stars is labeled as a negative review ("thumbs down"). Therefore, in such studies opinionated words that indicate positive or negative emotions are often crucial, e.g., good, awesome, amazing, terrible, bad, poor, etc. Existing supervised learning methods can be used for sentiment classification, e.g., naive Bayes, support vector machines (SVMs)

and voting algorithms (AdaBoost).

Sebastiani (2002) surveys the application of standard text-categorization algorithms that belong to the machine learning model. He reviews in detail issues related to the document representation problem, classifier construction and its performance evaluation.

Pang et al. (2002) employ three machine learning methods to classify movie reviews into two classes: positive and negative. Neutral reviews are omitted in the study, which made the problem less complicated. Dave et al. (2003) perform various supervised learning classification (Rainbow algorithm, SVMs, naive Bayes) on product reviews from websites C|net and Amazon.com. They are able to obtain satisfactory results for the review classification task through the choice of appropriate features and metrics.

Beside the classification problem of positive or negative opinions, research has also been done on predicting the rating score. In this case, the problem is formulated as a multi-class or a regression problem. In their research Pang and Lee (2005) first obtain human performance at the task and then applied a meta-algorithm, based on a metric labeling formulation of the problem that alters a given multi-class classifier's output in order to guarantee that similar items receive similar tags. They show that the meta-algorithm can offer significant enhancements over multi-class and regression types of SVMs if a novel similarity measure is used appropriate to the problem.

Further, sentiment analysis has been done on determining if a politician is in support of or in opposition to the issue under discussion. Bansal and Lee (2008), Thomas and Pang (2006) investigate whether it is possible to determine opposition to proposed legislation from the transcripts of U.S. Congressional floor debates. They incorporate the fact that the speeches occur in a sequence of a debate and source the information regarding the relationship between the speakers. They find that the incorporation of such information results in a vast improvements over classifying speeches in isolation. There is corresponding work on classifying election discussion forums into "likely to win" and "unlikely to win" (Kim and Hovy, 2007).

2.2 Sentiment Analysis on social media

A field of research where sentiment analysis has been extensively applied is the field of social media. Yang et al. (2007) in their work, investigate the emotion grouping of web blog corpora using support vector machines (SVMs) and conditional random field (CRF) (Lafferty et al., 2001) based machine learning techniques. They perform the emotion classification by taking the sentence context into account. They conclude that the emotion in the last sentence in the document is important in deciding the overall polarity of the examined document. They provide the measures of classifier performance for different experimental setups.

Read (2005) build the training set for the text classification by looking at emoticons such as ":-)" and ":-(". As their source author uses texts containing the emotionicons from Usenet newsgroups. The author elaborates on the classifier dependency problem in sentiment analysis. It is demonstrated that the training set depends both on the time when the data are collected and the topic of the domain. Thus, applying training set from domain A to domain B could be done if domains share domain specific vocabulary. The conducted experiments using training data labeled with the emoticons shows that collecting data in such way could result in datasets which are independent of domain, topic and time. Emoticons-trained classifiers obtains up satisfactory results on the test set.

Further work with blog data involves also tracking the spread of an idea, behavior or style (meme) in the news cycles. As explained by Leskovec et al. (2009) who attempt to model the diffusion of information in social media like blogs and to track the flow of information from professional news media to social networks. The research offers one of the first quantitative analyses of the global news. The authors observe a 2.5 hour time difference between the peaks of attention to a slogan in the news media and in blogs.

2.3 Quantitative research using micro blogging

Micro blogs (e.g., Twitter, Jaiku, Plurk, Tumblr) are becoming an established class within the broad group of various social media. They are a broadcast medium in the form of blogging. Blogs consist of personal posts made by the author/blog owner and displayed in chronological order. Micro blogs differ from traditional blogs through their smaller size of published elements. This medium allows people to exchange small elements of content such as short sentences, or links to videos, images and other media (Kaplan and Haenlein, 2011). Bloggers use a number of services to post updates, including instant messaging, e-mail, or micro blogging portals. Due to the novelty of the micro blogging, only a small body of research considers the medium.

Pak and Paroubek (2010) focus on using Twitter, the most popular micro blogging platform, and conducted sentiment analysis on an extensive collection of tweets. They obtain a corpus for sentiment analysis by collecting positive, negative and objective tweets. The authors label the tweet as positive if the message includes the happy emoticon, as negative if sad emoticon is included. To collect the objective tweets they retrieve posts from Twitter accounts of popular newspapers and magazines. The best performance is achieved using a contiguous sequence of 2 words (bi-grams) as features, including negation. The authors show that increasing the sample improves the performance of the system.

The aggregate of millions of tweets submitted to Twitter at any given time may provide an accurate representation of public mood and sentiment. This led to the development of real time

sentiment-tracking indicator such as Northeastern University's and Harvard University's "Pulse of Nation"[2].

In Go et al. (2009), authors obtain data through Twitter and then implement sentiment classification. Their approach is similar to the one taken by Read (2005). The authors construct the corpora - the large and structured set of texts - by looking at the emoticons. The tweets are labeled according to the emoticons. The best result is obtained using a Naive Bayes classifier with a mutual information measure for feature selection. However, high performance, is only obtained for the two class classification problem. The method shows unsatisfactory results with three classes ("negative", "positive" and "neutral").

Kim and Gilbert (2009) examine almost 2 million tweets about Michael Jackson's death in order to test a variety of methods of sentiment analysis and expand understanding of how people express emotions on Twitter. The authors do that by looking at the usage of Affective Norms for English Words (ANEW) within those tweets. The classification outputs by means of ANEW are compared with manually coded scores. Their results shows that ANEW is a promising tool for sentiment analysis on Twitter.

Bollen et al. (2011) show that Twitter mood can be used to predict the stock market. Their results indicate that changes in the public mood state can be tracked from the content of large-scale Twitter feeds. Further, the public mood can be correlated to the value of the stock market index Dow Jones Industrial Average (DJIA). The content of daily Twitter messages is tracked by two tools, OpinionFinder, which measures positive versus negative mood, and Google-Profile of Mood States (GPOMS), which translates mood into 6 levels (Calm, Alert, Sure, Vital, Kind, and Happy). The results shows that the accuracy of DJIA predictions could be notably enhanced by the inclusion of specific public mood dimensions such as Calm. Authors find the accuracy of 87.6% in predicting the daily fluctuation in the selected index. Calmness of the public (measured by GPOMS) is found to be a better predictor of the changes in DJIA than broad levels of positive sentiment as measured by OpinionFinder.

Achrekar et al. (2011) find that keeping track of influenza outbreaks by monitoring social networking sites such as Twitter has the potential to be quicker and more cost effective than traditional methods of disease surveillance. Based on data collected during 2009 and 2010, they find that the amount of flu-associated tweets is highly correlated with the number of influenza-like illness (ILI) reported by Centers for Disease Control and Prevention (CDC). The regressive models built on the historic CDC data verify that Twitter data substantially improves the model's accuracy in predicting ILI cases.

Sakaki et al. (2010) use a probabilistic spatiotemporal model to build an autonomous earth-

[2] `http://www.ccs.neu.edu/home/amislove/twittermood/`

quake reporting system in Japan using Twitter users as sensors. As an application, they construct an earthquake reporting system in Japan. Frequent seismic and the significant number of people using Twitter in the country enable them to spot an earthquake by monitoring feeds on Twitter with high probability. System reports earth movements rapidly and notified users. The notification message is delivered much faster than the information broadcasted by the JMA.

No papers written in the field of sentiment analysis and text mining use information extracted from Twitter to measure customer satisfaction. The specificity of the language used on Twitter imposes many challenges on the researcher in terms of data preparation for the analysis, which was not addressed by previous work. Such as, this Master's thesis aims to fill that gap.

3 Data

In this Section we describe the data collected from micro blogging platform Twitter. We first make a short introduction into characteristics of Twitter messages and present the steps needed to make the Twitter messages useful for our analysis. We then discuss the process of transforming the text messages into numeric data which form an input to a classifier.

3.1 Introduction to Twitter

We use tweets that people sent to KLM through Twitter using either "@KLM" or "#KLM" in our analysis. A tweet is a message posted via Twitter containing 140 characters or fewer. A retweet is another user's message, forwarded by other person. Retweets are often use to spread and share the news. To mark a retweet "RT" is put in front of the tweet. A duplicate is a tweet posted by the same user more than once. The @ sign is used to call out usernames in tweets. When a username is preceded by the @ sign, it becomes a link to a Twitter profile. On the other hand, the # symbol, called a hashtag, is used to mark keywords or topics in a tweet. It was created originally by Twitter users as a way to categorize messages. In general, @ signs are used to address other users and # to mention them. However, those rules do not always apply and people tend to use these characters interchangeably. Thus, to obtain the tweets that contain an opinion about KLM, Twitter is scraped for both "@KLM" and "#KLM" entries.

3.2 Preparation of the tweets corpus

Tweets were collected from the 8^{th} of February until the 30^{th} of April. Only English-language tweets are collected. This resulted in a dataset of 9368 unique tweets, a "unique tweet" is being defined as a tweet that is neither a retweet nor a duplicate.

Unique tweets do not contain a label, which is essential to perform the supervised learning.

The labels expressing the sentiment of the tweets need to be manually assigned for all tweets. A message obtains label 1 when the tweet content is neutral also referred as objective, 2 when the tweet is subjective and positive and 3 when the message is subjective and negative. The objective tweets account for 62.9% of all tweets, positive for 20% and negative for 17.1%.

The length of the Twitter message limits the user in sharing information. Therefore, many people include hyperlinks in their messages to refer to further information. However, extracting information from the link is only possible by clicking on it, so we decided to delete links from the dataset. This procedure allows for a reduction in the noisiness of the dataset.

We continue with the tweet cleaning process through the following operations: the entries with @ signs and # tags are removed from the tweets, to preserve the information contained in them we create three dummy variables; the positive entry dummy takes the value 1 if the tweet contains either a positive @ sign or # tag; and 0 otherwise. The dummies for negative and objective tweets are created in the same way. The sets of the positive, negative and objective signs are defined by looking at the most common signs used by Twitter users to mark emotions. For example "#fail" or "#epicfail" are popular terms to express dissatisfaction. Following the approach of Pak and Paroubek (2010) we classify mark ups (# or @ signs) from the popular news publications such as "Time" or "The Economist" as objective. Furthermore, all letters in the tweets are converted to the lower case.

We notice that a lot of messages contain spelling mistakes and Internet abbreviations. Hence, we also perform a spelling correction process. This is done using the list of common misspellings for machines available from Wikipedia[3]. The idea is to look for the misspelled word from the list in the tweet and replace it with its correct equivalent. We supplement the list with abbreviations commonly used by Internet users selected from InternetSlang[4] and with corrections for typos characteristic to the dataset, such as "amestrdam" to "amsterdam".

After spelling corrections, we perform a substitution of synonyms in order to increase the frequency of certain words. The words replacement takes the form:

- names of competitive airlines are replaced by "otherairline",
- abbreviations for KLM destinations[5] are replaced by "destination",
- expressions of satisfaction which contain the word "service" are replaced by "positiveservice",
- expressions of dissatisfaction which contain the word "service" are replaced by "negativeservice",
- synonyms to good (positive adjectives) are replaced by "positivesynonyms",

[3] http://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings
[4] http://www.internetslang.com/
[5] http://en.wikipedia.org/wiki/KLM_destinations

- synonyms to bad (negative adjectives) are replaced by "negative synonyms",
- synonyms to luggage are replaced by "luggage",
- synonyms to congratulations are replaced by "kudos".

In subsequent steps, punctuation, numbers and all non-alphanumeric signs are deleted from the tweets. To control for the information that is contained in the emotion icons, which are deleted during punctuation removal, we create dummies for positive and negative emotion icons.
Emotion icons are grouped as follows:

- positive group:
  ":-)", ":)", ":-d", ":d", ":->", ":>", "<=)", "<=-)", "=)", "=-)", "x)", "x-)", ":-bd", ":bd",
- negative group:
  ":-(", ":(", ":-/", ":/", ":-s", ":s", """:-&", ":&", ":-w", ":w".

Further, in order to control for negations in the sentences we add a dummy variable to the dataset that takes value 1 if there is a negation in the tweet and 0 otherwise. A number of words frequently appear in tweets, but do not contain any sentiment information. These are referred to as "stop words". Most commonly, these short function words are,*as such as, the, is, at, which, and, on.* We delete the stop words from the tweets. The last operation which we apply to the corpus of tweets is stemming, which is the process of reducing inflected words to their stem.

3.3 Constructing the term matrix

To this point all of the required operations for cleaning the data have been applied to the corpus of Twitter messages. To make the data available for classification it must be converted into numbers. Therefore, we break the tweets into single words. This process is called tokenization and the output words are known as tokens. We use the tokens to create a term matrix. The term matrix is the matrix of dimensions $m \times n$, where m is the number of the tweets in a corpus and n is the number of unique tokens. Thus, having only 2 tweets in a corpus "*I love flying with @KLM*" and "*Give me my luggage back*" the term matrix would look like the example in Table 1. After selecting the unique tokens we create a term matrix which has 9368 rows and 10224 columns.

Table 1: An example of a term frequency matrix

	love	fly	give	luggage	back
1	1	1	0	0	0
2	0	0	1	1	1

Note: assuming that all the described operations were applied.

3.3.1 Merging the tokens

Despite the fact that the unique and spell-corrected tokens are the input into the matrix, it still contains the features that are misspelled. For example, instead of *cool* and *coool* being one feature they are the separate ones. This problem results from the specificity of the language used on Twitter. Unfortunately, the simple spelling correction is insufficient to create a clean dataset for tokenization.

We attempt to find the misspelled tokens and merge them under the one label. It requires finding the sets of similar variables that consist of the correct word and the misspelled candidate. In order to be able to group the words we need criteria to identify them. The distance between two strings provides our measure of string similarity. We define the distance $d(a, b)$ between two words a and b as the minimal cost of a set of operations that convert a into b. The operations are a series of rules, such as $\tau(z, w) = t$, where z and w denote different strings and t is a nonnegative real number. Once substring z is transformed into w, no further transformations can be done on w.

If for each operation of the form $\tau(z, w)$ a particular operation $\tau(w, z)$ exists at the same cost, then the distance is symmetric $d(a, b) = d(b, a)$. Furthermore, $d(a, b) \geq 0$ for all strings a and b, that $d(a, a) = 0$, and that $d(a, z) \leq d(a, b) + d(b, z)$ always holds. Hence, if the distance is symmetric, the space of strings is a metric space. We restrict the possible operations to:

- insertion, i.e., inserting a letter into string a to obtain b,
- deletion, i.e., deleting a letter from string a to obtain b,
- substitution or replacement, i.e., substituting one letter from a string a by another to obtain string b.

To match misspelled words with the correct ones and merge them under one correct label we utilize a Nearest Neighbor (NN) search (Ford et al., 1970). Nearest Neighbors search, also known as proximity search is a technique used for identifying similar items in a list. We assign two words into one group if the distance between them is equal to the fixed value called the radius.

Given n tokens, there are $n(n-1)/2$ pairs of strings (and relative distances) that need to be compared. This requires a lot of computation time. Hence, we limit the search to process called blocking. First, blocks are obtained in which all variables share a substring of a given size (number of characters in a string). The number of features that are checked is reduced, as the comparison is only done within the block. Thus, instead of $n(n-1)/2$ there are $km(m-1)/2$ words to match, where k is the number of blocks and m is the average size of the block. The number k is specified before the analysis. As the method shows good results on longer strings we perform the search for k equals 6 and 5. In order to perform the NN search we have to define the metric for calculating the distance between the tokens.

The Levenshtein distance is perhaps the most straightforward distance function between strings (Levenshtein, 1965). It is considered very effective due to its general applicability. We consider operations: insertions, deletions and substitutions, where each is of cost 1. This can be rephrased as "the minimal number of insertions, deletions and substitutions to make two strings equal" (Navarro, 2001). The distance is symmetric, and it holds $0 \leq d(a,b) \leq max(|a|,|b|)$. For example, "amsterdam" and "Amsterdam" have an edit distance of 1 as changing A into a is the only operation required. "New York" and "newyork" has edit distance 3 - 2 substitutions and 1 removal.

To maximize the precision of the NN search we fix the radius to 1 which equals to the distance $d(a,b)$ of the cost of 1. The tool to perform NN search with Levenshtein distance was implemented in Google-Refine and is a part of the VINCIO project[6]. In the Google Refine tool we can confirm if the labels to be replaced are correct before merging the selected terms under one label. We can accept the choice, decline it or modify the label proposed. Therefore, the process is not fully automatic but allows the researcher to have control over the operation. In our case, interference by the researcher is minor.

3.3.2 Reducing the dimensionality of the matrix

The features' labels in the term matrix are replaced with the labels from Google Refine. We then merge the columns with the same labels. This procedure allow us to reduce the number of features by 7%. The size of the matrix however, remains substantial. In response to this, sparse terms, i.e., terms occurring only in very few tweets, are deleted from the matrix. We remove the terms that appear in the dataset less than 20 times (0.2%). This reduces the size of the matrix dramatically without losing significant relations inherent to the data. The matrix that we further analyze with SVMs has 9368 rows and 557 columns. This is a 94.6% reduction of the number of columns.

4 Methods and Techniques

In this Section we present the methods used to answer our research question. As we aim to perform tweets classification we need a learning algorithm that performs well in automatic text categorization. Support Vector Machines are chosen, because they display high performance within sentiment analysis (Dumais et al., 1998). First we introduce the general concept of binary SVMs. Later, we elaborate to the multi-class classification problem, because in our research the aim is to classify tweets into three different classes. Next, we present the way of dealing with the

[6] `http://code.google.com/p/simile-vicino/`

unbalanced nature of the dataset. In the last part of this section we discuss techniques for model selection and evaluation.

4.1 Support Vector Machines

Before we discuss the more complex multi-class classification problem we begin with the more basic concept of a binary classification task. The multi-class algorithm relies heavily on that concept, and hence an understanding as such is essential. A classification task usually involves separating data into training and testing sets. The training data are used together with the supervised learning method to train the classifier. The test set is used to asses the performance of the classifier. Each observation in the training set contains one target value (i.e. the class labels) and several predictors (i.e. the features which in our case are words or dummies that are created). We denote a training set of a number of features by $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_l$ and associated class labels as $y_1, y_2, ..., y_l$. We assume that features belong to some set $X \subset \Re^n$ and $y \in \{-1, 1\}$. Furthermore, an important concept in defining a classifier is the dot product between two vectors, also referred to as an inner product or scalar product, defined as $\mathbf{w} \cdot \mathbf{x} = \sum_i \mathbf{w}_i \mathbf{x}_i$. The dot product is a value expressing the angular relationship between two vectors.

The goal of SVMs is to build a model (trained on the training data) which predicts the target values of the test data given only the test data attributes.

4.1.1 Binary Support Vector Machines

We begin with a description of the linear classifier and later extend the framework to the nonlinear case. The linear classifier is based on a decision function. The decision function, also known as the discriminant function, is a simple weighted sum of the training features plus a constant referred in the SVM literature as a bias. In our notation

$$D(x) = (\mathbf{x} \cdot \mathbf{w}) + b, \tag{1}$$

where the vector $\mathbf{w}$ is the weight vector, and b is called the constant (bias).
The hyperplane

$$(\mathbf{x} \cdot \mathbf{w}) + b = 0, \tag{2}$$

divides the space into two subsets I for which $y = 1$ and another subset II for which $y = -1$. The sign of the discriminant function $D(x)$ denotes on which side of the hyperplane a point is located.

The boundary between the examples classified as positive and negative is the decision boundary determined by the hyperplane. It is linear because it is linear in the input instances, and hence

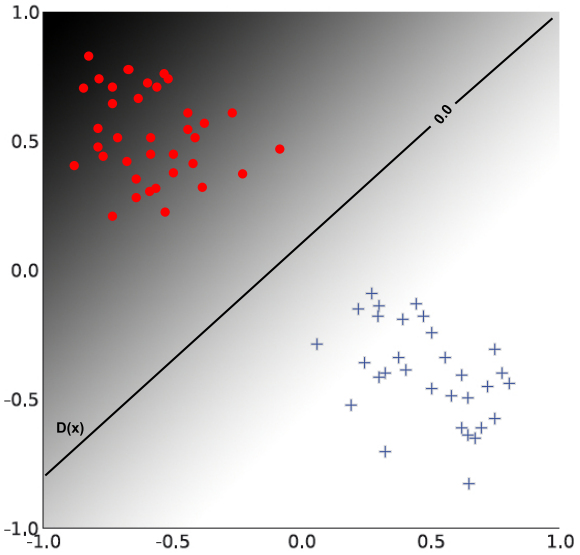

Figure 1: An example of decision hyperplane separating two classes of the data. Source: own.

the classifier is linear. If the decision boundary depends on the data in a non-linear manner the classifier is said to be non-linear. An example of a decision hyperplane separating two classes (red dots and blue crosses) in presented in Figure 1.

The optimal hyperplane is the hyperplane that has a maximal distance, i.e., margin to the closest vector $\mathbf{x}$ from the training data which is equivalent to minimizing $||\mathbf{w}||^2$. Here the $||\mathbf{w}||$ is the norm of a vector $\mathbf{w}$. This results in the following minimization problem

$$\min Q(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2 \tag{3}$$

subject to

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, i = 1, ..., \ell. \tag{4}$$

The constrained minimization can be solved with Lagrange multipliers (Schölkopf et al., 1999). Finding a solution involves constructing a dual problem where a Lagrange multiplier α_i is associated with each constraint $y_i(\mathbf{w} \cdot \mathbf{x_i} + b) \geq 1$. The problem is transformed into

$$\max W(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j), \tag{5}$$

subject to constraints

$$\alpha_i \geq 0, i = 1, ..., \ell, \tag{6}$$

$$\sum_{i=1}^{\ell} \alpha_i y_i = 0. \tag{7}$$

The solution is

$$\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i, \tag{8}$$

$$b = y_i - \mathbf{w} \cdot \mathbf{x}_i \text{ for any } \mathbf{x}_i \text{ such that } \alpha_i \neq 0. \tag{9}$$

The examples $\mathbf{x}_i$ for which $\alpha_i > 0$ are called support vectors (Schölkopf et al., 1999). The constraints in this formulation ensure that each example is classified correctly, which is possible if the data are linearly separable.

Computing the hyperplane that provides a separation to the training data without errors is very unlikely. To allow errors we introduce the non-negative variables $\xi_1, ..., \xi_\ell$, along with relaxed constraints

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 - \xi, \ i = 1, ..., \ell, \xi \geq 0. \tag{10}$$

We supplement $\frac{1}{2}||\mathbf{w}||^2$ in (3) with $C\sum_i^\ell \xi_i$ to penalize training errors. The updated objective function takes form

$$Q(\mathbf{w}, \xi) = C\sum_{i=1}^{\ell} \xi_i + \frac{1}{2}||\mathbf{w}||^2, \tag{11}$$

subject to the constraints in (10), for some value of the constant $C > 0$. This formulation is called the soft-margin SVM introduced by Cortes and Vapnik (1995). The constant C is referred in the literature as soft margin constant or cost parameter. The solution to new optimization problem is found by maximizing (5) under different restrictions

$$0 \leq \alpha_i \leq C, i = 1, ..., \ell, \tag{12}$$

$$\sum_{i=1}^{\ell} \alpha_i y_i = 0. \tag{13}$$

The only difference from the separable case is the upper bound C. We present the example of linear SVMs in the Figure 2. The circled examples are the support vectors. They determine the margin that separates the classes. The thick black line is the decision hyperplane.

4.1.2 Kernels - from Linear to Non-Linear Classifiers

So far we have only considered a linear classifier. However, in some text classification problems with data which contain features for which the non-linear classifier performs better. One way of transforming the linear classifier into a non-linear classifier is to map the data from the input space X to a feature space $F = \Phi(x) : x \in X$ using a non-linear function Φ: $X \to F$. The discriminant

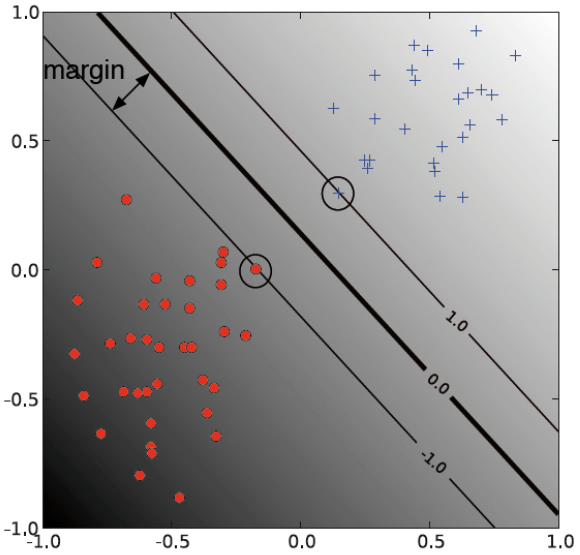

Figure 2: An example of linear SMVs. Source: Ben-Hur and Weston (2010).

function then takes the form:

$$D(x) = (\Phi(\mathbf{x}) \cdot \mathbf{w}) + b. \tag{14}$$

We can avoid computing the explicit coordinates in the feature space, which result in increasing both the usage of storing memory and time needed to compute the discriminant function, by applying a kernel function (Cristianini and Shawe-Taylor, 2000). A kernel function is a continuous, symmetric and has a positive define gram matrix. It can directly calculate the value of the dot product of the mapped data points in the feature space. All the variables in X can be replaced by their kernel basis. The kernel function measures the similarity between $\Phi(x_i)$ and $\Phi(x_j)$.

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\Phi(x_i) \cdot \Phi(x_j)). \tag{15}$$

Popular choices for k in the SVM literature (Hastie et al., 2001) are dth-Degree polynomial functions $k(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^T \mathbf{x}_j + r)^d$, and Radial basis function (RBF) $k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma ||\mathbf{x}_i - \mathbf{x}_j||^2}$. The linear kernel is the special case of polynomial kernel, with parameters γ and d and r equal to 1. Here γ, r and d are the kernel parameters. If observations exhibits non-linear relationships the linear kernel will not separate the data properly. For the higher degree of polynomial kernel the discriminant function is more flexible. The radial basis function corresponds to mapping the observations into a infinite dimensional vector space.

Substituting the kernel into the hyperplane decision function we obtain

$$D(\mathbf{x}) = sgn\left(\sum_{i=1}^{\ell} y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}_j) + b \right). \tag{16}$$

This substitution is also referred to as the *kernel trick* and could be use to extend the framework to nonlinear SVM (Boser et al., 1992).

4.1.3 Pairwise Support Vector Machine

A concept of pairwise classification is to use $K(K-1)/2$ classifiers, where K is the number of classes in the dataset, covering all pairs of classes instead of applying K binary classifiers "one-class versus all others" (Kressel, 1999). In out dataset K equals 3, hence we build 3 binary classifiers.

For training data from the ith and the jth classes, the classification problem is

$$\min_{\mathbf{w}_{ij},b_{ij},\xi_{ij}} 1/2(\mathbf{w}_{ij}) \cdot \mathbf{w}_{ij} + C \sum^{\ell} (\xi_{ij})_{\ell}, \tag{17}$$

subject to

$$(\mathbf{w}_{ij}) \cdot \Phi(\mathbf{x}_{\ell}) + b_{ij} \geq 1 - \xi_{ij}, \tag{18}$$

if $\mathbf{x}_t$ in the jth class,

$$(\mathbf{w}_{ij}) \cdot \Phi(\mathbf{x}_{\ell}) + b_{ij} \leq -1 + \xi_{ij}, \tag{19}$$

if $\mathbf{x}_t$ in the jth class,

$$\xi_{ij} \geq 0. \tag{20}$$

Classification is done by a "max-wins" voting strategy, in which one binary classifier for every pair of the distinct classes assigns an example to one of two classes. Then the vote for the assigned class is added by one, and finally the class with the largest number of votes determines the example classification. Figure 3 presents the classification scheme.

For each case we build 3 decision functions, objective against positive (1 vs. 2), objective against negative (1 vs. 3) and positive against negative (2 vs. 3). Let's assume that we classify the example with the first binary classifier (1 vs. 2) and the classifier decides the point to be in objective class, therefore the objective class gets one vote. Further, the consecutive classifiers are applied and voted. We assume that for the second classifier the example is also voted as objective and for the third classifier as positive. Summing up the votes we see that the objective class got 2 votes, positive 1 and negative 0, hence finally the point is classified as objective.

4.2 SVM with Unbalanced Data

Many datasets are unbalanced, i.e., one class contains much more data points than the other. This poses a challenge when training a classifier (Provost, 2000), because the classifier has unequal chances of learning for each class. To correct for an imbalance in the data different costs can be assigned for a misclassification for each class. The total misclassification cost, $C \sum_{i=1}^{\ell} \xi_i$ is

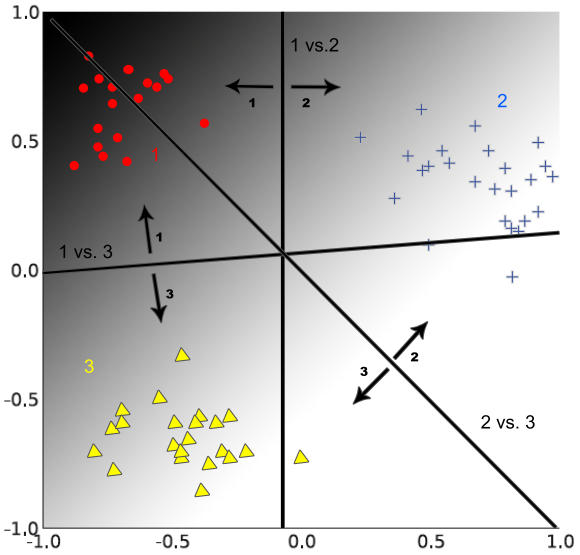

Figure 3: An example of pairwise multi-class SMVs. Source: own.

replaced with two terms, one for each class

$$C \sum_{i=1}^{\ell} \xi_i \rightarrow C_I \sum_{i \in I} \xi_i + C_{II} \sum_{i \in II} \xi_i, \tag{21}$$

where C_I (C_{II}) is the cost constant for the examples in the subset I (II). To give equal weight to each class the total penalty for each class has to be the same. Assuming that the number of misclassified observations from every class is proportional to the number of all observations in every class, it is advised to choose:

$$\frac{C_I}{C_{II}} = \frac{n_{II}}{n_I}, \tag{22}$$

where n_I (n_{II}) is the number of examples of a given class in subset I (II). This provides a method for setting the ratio between the costs. Such solution was implemented, among others, in the SVM package LIBSVM (Chang, 2008). We assign different cost parameters to three classes of data. We do that by multiplying the cost constant with the weights that correspond to the ratio calculated accordingly to Formula (22).

4.3 Model Selection

To construct the SVM classifier we have to make a number of choices. First we need to select a kernel and cost parameter for SVMs and decide on a subset of attributes to use. The choice of both the parameters and features is crucial for SVMs performance. Several authors have shown that tuning the parameters and feature selection improves the classifier's results (Huang and Wang, 2006). We first introduce the concept of cross-validation which we use along the process of building

and tuning the model. Further, the SVMs parameter selection is presented. We then describe our approach to feature selection and show the metrics that we use to validate the classifier's performance.

4.3.1 Cross-validation

We use cross-validation for development and fine-tuning a model. Cross-validation is a statistical method that obtains reliable indicators of quality for a given model. In other words it allows an estimation of prediction errors. K-fold cross-validation uses part of the available data (training set), to fit the model, and a different part (test set) to test it. The data are split into K roughly equal-sized parts, when $K = 5$, the scenario looks like this: For the k-th part, the model is fitted to the other $K-1$ parts of the data, and the prediction error of the fitted model is calculated when predicting the k-th part of the data. It is done for $k = 1, 2, ...,5$ and five estimates of prediction error are combined and averaged. We use 5-fold cross-validation throughout the analysis.

4.3.2 Parameter Selection

SVMs have a set of parameters called hyperparameters: a cost constant C and parameters that are dependant on a kernel function. The dependence of the SVM decision boundary on the hyperparameters results into a dependence of the accuracy of the classifier on those parameters. Therefore, when building the SVMs classifier we want to choose the parameters that are optimal for a given classification problem.

Figure 4 shows the effect of different values of the cost parameter on the orientation of the hyperplane and the margin size. The black thick line is the decision hyperplane and the thiner lines are the boundaries of the margin. For a large value of C errors, the points that violate the margin constraint, receive larger penalty. Consequently the points that are closest to the hyperplane affect its orientation. Decreasing the value C those points are ignored and become margin errors. The hyperplane gives greater margin for the rest of the data. The smaller the C value the wider the margin around the decision boundary.

Kernel form and its parameters also have an effect on the decision boundary. Figure 5 depicts how the γ parameter of the Gaussian kernel and the degree of polynomial kernel determine the flexibility of the classifier in fitting the data. The decision which kernel and parameters to use dependents on the data. We can use polynomial kernel to capture words conjunctions. Thus, if we aim to model pairs of words, like for example *customer* and *service* we will require 2 degree polynomial kernel. Radial basis kernel enables to model observations that pick out circles. Increasing the value of *gamma* results into greater curvature of the decision hyperplane.

We follow the guidelines provided by Ben-Hur and Weston (2010) and first try the linear kernel

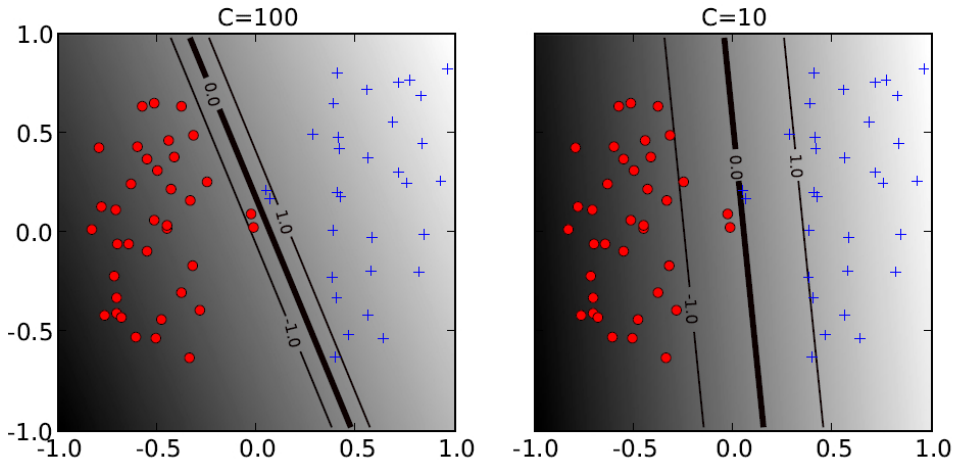

Figure 4: The effect of the parameter C on the decision hyperplane. Source: Ben-Hur and Weston (2010).

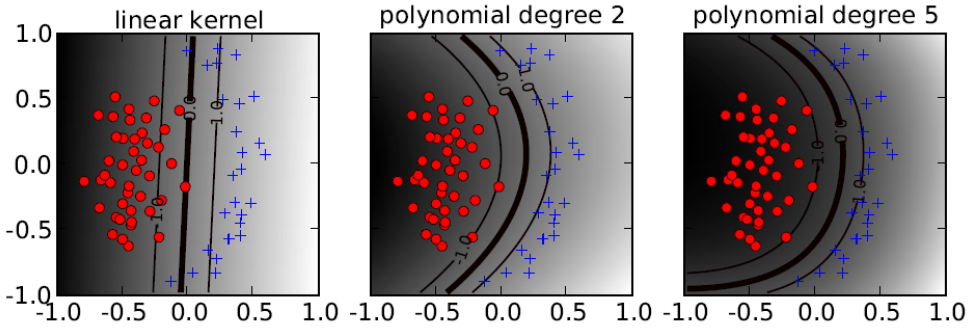

Figure 5: The effect of the degree of the polynomial kernel on the flexibility of the decision boundary. Source: Ben-Hur and Weston (2010).

and then check if the performance can be improved using a the non-linear kernel. The linear SVM has an advantage that only one parameter to tune, the C parameter. The hyperparameter tuning is done using a grid search over a specified parameter range on the full training test with 5-fold cross-validation. We use the classification error as a performance measure.

4.3.3 Feature Selection

Feature selection or variable selection is a technique of choosing a subset of relevant attributes available from the data in order to build robust learning models. There are many advantages of feature selection: simplify data visualization and data interpretation, decreasing the memory and storage requirements, reducing computation time and improving prediction results.

There are two approaches of feature selection: Forward and Backward. Forward selection starts with no attributes and adds them one by one, at each step evaluating whether the new subset is an improvement over the previous. The backward feature selection begins with the full set of features and sequentially removes the attribute that allows for the improvement over the previous bigger subset. Miche et al. (2007) state that forward selection is computationally more efficient than backward elimination, because it selects the important features effectively. On the other hand, the backward selection method could outperform the forward selection by finding two variables that together result in the best performance (Guyon and Elisseeff, 2003).

Many algorithms that perform feature selection include variable ranking as a major technique, because of its simplicity and empirical success. It aims to select the features that discriminate between the different classes. This could be insightful when performing classification by so called "black box" methods like SVMs. The important question is which variable to add or remove from the dataset when performing the selection. Kohavi and John (1997) suggest to use the difference in a objective function value caused by removing one feature at the time as a ranking criteria to evaluate which variable to include or remove.

For a classification problem the optimal objective function is the expected value of the error rate. The objective function can be replaced by the cost function J, computed on the training set only. The cost function is the approximation of the optimal objective function. We want to calculate the change in the cost function resulting from deleting an attribute. For the linear SVM classifier the cost function is the quadratic function of w_i. The cost function J equals $\frac{1}{2}||\mathbf{w}||^2$ and $(w_i)^2$ is used as a feature ranking criterion. The ranking criterion can be computed by iterative procedure Recursive Feature Elimination (RFE) (Guyon et al., 2002). It is the backward feature elimination (Kohavi and John, 1997).
The RFE consists of three steps:

1. Train the algorithm (optimize the weights w_i with respect to the cost function).

2. Compute the ranking $(w_i)^2$ criterion for all features.

3. Delete the feature with the smallest criterion.

SVM RFE can be seen as an application of RFE with the weight magnitude as a ranking criterion. We apply the SVM RFE to calculate the feature ranking list. We then use the output list to construct the different subset of features.

4.3.4 Assessment of a classifier quality

Before we interpret the final results we check the performance of the classifier in several ways. To appropriately quantify the performance of the classifier with unbalanced dataset we calculate 3 metrics of performance - a precision, recall and F-measure for every class.

In a classification task, the precision for a class is the number of correct results (i.e., true positives) divided by the number of all returned results (i.e., the sum of true positives and false positives). The recall in this context is the opposite measure. It is defined as the number of correct results divided by the number of results that should have been returned (i.e., the sum of true positives and true negatives). The trade-off between the precision and the recall depends on the system application.

Having 3 classes in our model we obtain 6 measures of model performance, thus to facilitate model comparison we combine the precision and recall into the F measure. It is the weighted harmonic mean and is expressed by

$$F = \frac{1}{\alpha \frac{1}{Precision} + (1-\alpha)\frac{1}{Recall}}, \tag{23}$$

where α is the weight which expresses the utility trade-off between the precision and recall. If α equals to 0.5 the same emphasis is put on the precision and recall. For the model selection we use a a rule that recall for all the classes has to be above 50%, because we want to be sure that the predicted tweets accounts for at least 50% of the true sample. Further, for the recall values above the 50% threshold we choose the highest precision values for the subjective classes of the tweets.

For the selected model we display the results using a confusion matrix, which helps in the results interpretation. In the field of machine learning, the confusion matrix (Kohavi and Provost, 1998) contains information about the actual and predicted classifications and is used to evaluate the forecast. Each column of the matrix (Figure**??**) represents the instances in a predicted class, while each row represents the instances in an actual class. The cells in the table contain the number of incorrectly classified cases - false positives (FP), false negatives (FN) - and correctly ones - true positives (TP) and true negatives (TN).

Table 2: An example of a confusion matrix.

	Predicted class	
Actual class	+	-
+	TP	FN
-	FP	TN

5 Results

In this Section we present our results obtained with the learning algorithm described in the Section 4. The process of building the final classifier is described by us. We compare the performance of the nonlinear and linear classifier for which both parameter and feature selection are implemented. The best performing classifier is selected according to the rule described in the Section 4. We interpret the results for the selected model by looking at the precision and recall for each class of the tweets and by analyzing the features. The tweets' preparation is shown to influence the classifier performance. Finally we demonstrate how our approach could be used to track customer sentiment over time.

5.1 Classifier selection

On overview of the steps that are taken to build the final classifier is presented on the Figure 6. First we randomly split the dataset into training and test sets (75:25) and then build the linear and nonlinear classifier. The SVMs hyperparameters are tuned by performing the grid search with 5-fold cross validation over the supplied values of cost parameter $C = 2^{-3}, 2^{-2}, ..., 2^{4}$ for SVMs with linear kernel and RBF kernel for which the kernel parameter $\gamma = 2^{-4}, 2^{-2}, ..., 2^{1}$. For the linear classifier this yields the value of $C = 0.5$ and for the nonlinear $C = 0.25$ and $\gamma = 2$.

We obtain the feature ranking list through the Recursive Feature Elimination and use it to build the 557 subsets of attributes. The first subset consists of the 1^{st} feature from the ranking and subsequently to every following subset the next feature from the ranking list is added. Thus, the last subset consists of all the features. The SMVs are trained on the features' subsets using 5-fold cross-validation and afterwards predict the tweets labels for the training set. We compare the results for linear SVMs and SVMs with radial kernel by looking at the F-measures that are calculated for models built with the 50, 100, 150 best features (according to the RFE) and all the features. Table 3 presents the results which show that the linear classifier outperforms the nonlinear. SVMs with RBF kernel give better results only for the objective class using all the features.

We decide to proceed with the linear SVMs with 0.5 value of the cost parameter. This value

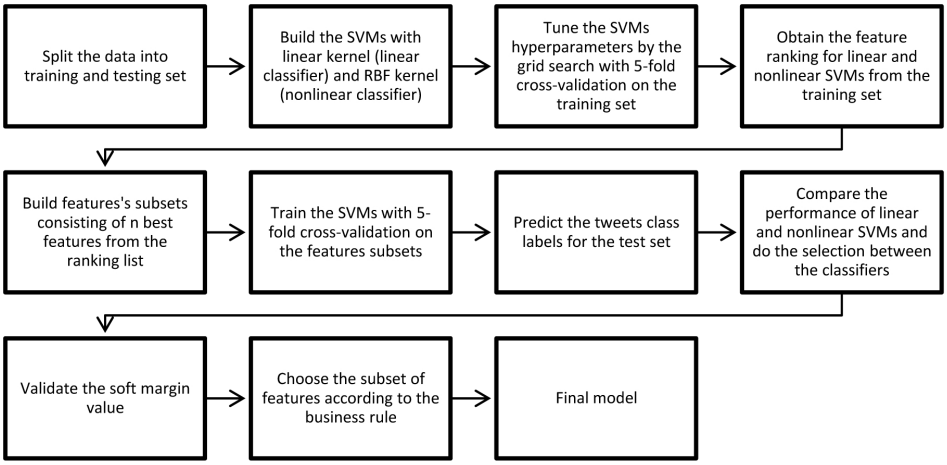

Figure 6: Process of building the SVMs classifier.

Table 3: F-measures for the linear and non-linear SVMs, 5-fold cross-validated.

		# of features in the model			
Tweets class	Classifier	50	100	150	all
Objective	Linear	0.78	0.78	0.76	0.76
	Non-linear	0.74	0.75	0.75	0.77
Positive	Linear	0.65	0.67	0.66	0.66
	Non-linear	0.57	0.45	0.37	0.17
Negative	Linear	0.5	0.52	0.58	0.57
	Non-linear	0.44	0.33	0.2	0.07

Note: For linear SVMs $C = 0.5$ and for non-linear with RBF kernel $C = 0.25$ and $\gamma = 2$.

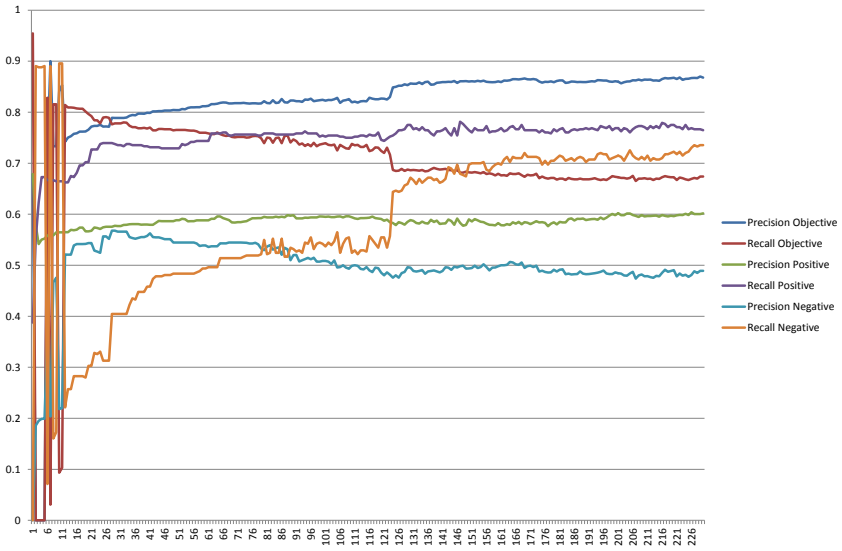

Figure 7: Performance of linear SVMs models (C=0.5) for different subset of features.

of cost parameter was optimal for the full set of attributes. However, to provide the optimal value of the cost for each model would require performing the grid search for each subset of features which computationally would be very expensive. We compare the performance for $C = 0.5$ with the results for the default value $C = 10$ and conclude that the solution is insensitive to the value of the cost parameter. These results are found sufficient to continue with the value of the cost parameter 0.5, which is in line with findings presented by Guyon et al. (2002). For the sake of completeness the results for the SVMs with the higher cost (C=10) are presented in Appendix I.

For additional feature selection we consider the performance of the first 230 models, because adding more features does not alter the performance for any of the classes. Figure 7, depicts the precision and recall for the three classes of the tweets as function of number of features.

The subset of features to select depends on a application of the classifier, a trade-off between precision and recall. We follow the rule that we describe in previous Section (4.3.2 Assessment of a classifier quality). Using this criteria it leads us to the selection of the model using subset of 66 best features from the ranking.

5.2 Performance the selected classifier

The performance of our final classifier using 66 features is presented in Table 4. Values on the diagonal are the correctly predicted examples. In the columns we can see how examples are labeled by SVMs and the numbers given in the rows represent the actual (true) classes that the

tweets belong to.

Table 4: Confusion matrix for linear SVMs (C=0.5) using 66 features, 5-fold cross-validated.

	Predicted class		
Actual class	Objective	Positive	Negative
Objective	1109	89	156
Positive	216	265	35
Negative	144	26	202

Metrics summarizing the classier performance that are calculated from the confusion matrix are presented in Table 5.

Table 5: Performance of the linear SVMs (C=0.5) using 66 features, 5-fold cross-validated.

Tweets class	Precision	Recall	F-measure
Objective	0.82	0.75	0.78
Positive	0.59	0.76	0.66
Negative	0.54	0.51	0.52

It follows that the classifier achieves the best precision for the objective class 82% and the recall 75%. The F-measure for the objective tweets is the highest. Obtaining higher precision for objective tweets would be possible by increasing the features' set. For positive tweets the classification achieves 59% precision and 76% recall, while only 54% and 51% for negative tweets. Despite the fact that corrected for the unbalancedness in the data the least satisfying performance for the negative class follows from the fact that negative tweets belong to the minority class.

The most commonly made mistakes are classifying subjective tweets as objective and vice versa. The positive tweets are misclassified as objective 89 times (objective as positive 216) and negative tweets as objective 156 times (objective as negative 144). The most "unwanted" error is mislabeling positive tweet as negative and negative tweet as positive. For the selected case only 26 positive tweets are labeled as negative and 35 negative as positive. Thus, the SVMs distinguish well among the subjective classes.

5.3 Features analysis

To interpret out results we investigate the first 20 features from the ranking list. We present the attributes with their frequencies from the full dataset are in Table 6.

From Table 6, we can see that artificial dummies such as "synonymspositive" or "positive.emo" are the features that obtain high position in the ranking. Suggesting that they contain valuable

Table 6: First 20 best features from the ranking list.

Ranking	Feature	Frequency	Ranking	Feature	Frequency
1	blog	37	11	launch	60
2	synonyms.positive	872	12	hashnegative	120
3	positive.emo	684	13	delay	160
4	love	252	14	positive.service	66
5	error	69	15	cancel	80
6	please	292	16	kudos	47
7	seatmate	220	17	example	32
8	synonyms.negative	283	18	pick	171
9	media	285	19	cool	76
10	meetandseat	260	20	book	424

information for the classifier. For example using only the two best features "blog" and "synonymspositive" the classifier achieves 58% precision and 55% recall for the positive class which already could be a satisfactory result if we aim only at selecting the positive tweets. The features "meetandseat" and "seatmate" are also found important. These two tokens are associated with the new program introduced by KLM "Meet and Seat". The program was launched in February and enables passengers to pick their seat mates on the basis of their Facebook or LinkkedIn profiles. The new service was advertised by the airline and caused much controversy. Around 70% of the all "meetandseat" attributes were used before the mid march. I conclude that the training set exhibits seasonal patterns, i.e., introduction of new product/service causes increased traffic of key words for the campaigns. These findings are in line with the results reported by Read (2005).

5.4 The influence of tweets preparation on the model performance

Finally, we to asses the impact of the data preparation on the performance of the classifier. We compare the performance of our classifier found on the dataset of prepared tweets with the classifier obtained using only partially prepared data, i.e., where no spelling correction, synonym substitution and controlling for the mark-up signs. The rest of the operations described in the Section 3 were applied to create the dataset. Thus, we call this dataset partially prepared. The framework presented in Figure 6 is used to obtain the final model, with the only difference being that we start from the beginning with the linear classifier. The selected model is the linear SVMs model with cost value 2 using 189 features.

Table 7 shows the performance of the selected model. The findings suggest that to meet the 50% recall rule while obtaining the highest precision for the subjective class more features need

Table 7: Performance of the linear SVMs (C=2) using 189 features from the limited prepared data, 5-fold cross-validated.

Tweets class	Precision	Recall	F-measure
Objective	0.81	0.72	0.76
Positive	0.59	0.73	0.65
Negative	0.46	0.50	0.48

to be included in the model. We compare the values in Table 7 with the vales for model using prepared tweets in Table 5. The partially prepared tweets model exhibits comparable precision on the objective and positive class, however the precision on the negative class is 8% lower than on prepared data. The impact of data preparation is also reflected in the recall measures. For the objective and positive class, there is 3% improvement and 1% for the negative class. For every class of the tweets the F-measure for the SVMs using prepared data is higher than for the SVMs using tweets with less preparation. This suggests the improvement in the overall performance of the classifier.

5.5 Tracking Sentiment

In order to be able to track the predicted sentiment the predicted objective tweets are excluded from the sample. We aggregate daily the predicted positive and negative tweets to create the sentiment indexes over time. The obtained sentiment is presented in Figure 8.

For the time period from the beginning of February until the mid March the positive and negative sentiments are negative correlated. The spikes of the green line (positive sentiment) are contrasted with the dips of the red line (negative sentiment). It could be explained by the fact that during this period KLM launched the "Meet and Seat" campaign, which by part of the users was seen as a great and interesting idea but by the others as creepy and new way to stock people.

On the 7^{th} of March the information about the current financial situation of KLM-Air France was made public and caused the negative buzz. The actions to mitigate the negative tweets were undertaken by KLM therefore the negative peak is followed by the raise in the positive sentiment. Furthermore, the negative sentiment increase on the 15^{th} and the 20^{th} of March resulted from the operational issues with the flights. The compensation actions were taken by KLM which are reflected by the gains in positive sentiment.

Moreover, from the analysis of the moving average we conclude that the negative sentiment maintains on the constant level, while the positive index slowly raises. This might be explained by the phenomena that people's mood is affected by the season of the year. Thus, people tend to get happier in spring when the number of sunny hours increases. Winter period also results in

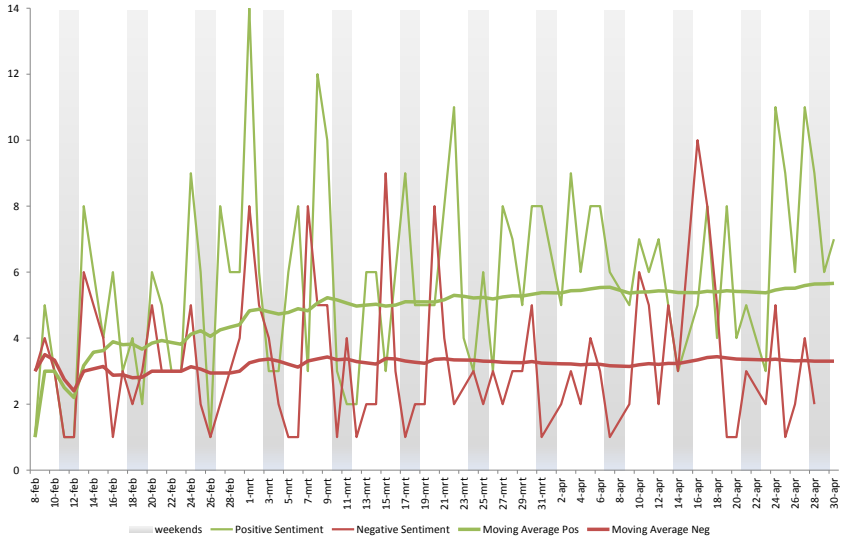

Figure 8: Aggregated daily sentiment over time. Source: own.

longer delays due to disturbances caused by the snow.

6 Conclusions and Recommendations

In this Master's thesis we aim to develop the framework that allows to measure the sentiment expressed on Twitter. This goal is achieved by building the linear SVMs classifier with 66 features selected by the ranking algorithm. We obtain 82% precision for the objective class, 59% for the positive class and 54% for the negative while the recall for all the classes is above 50%. Our classifier distinguishes well between the subjective classes and makes the most errors while making distinction between objective and positive tweets. This might be due to the fact that some positive tweets are very similar to the objective tweets. We illustrate this problem with the following example. The tweet "*@KLM Can you help me with booking thanks?*" is labeled by us as a objective tweet, because it does not contain any sentiment. On the other hand, the tweet "*@KLM Thanks for the help with booking*" is marked as a positive tweet as the person was satisfied, because received help from the KLM Twitter team. However, the features extracted from such tweets are equivalent.

The example above points also to the other problem which is the representativeness of the sentiment expressed on-line. The positive sentiment captures the opinions of the customers who are satisfied with the KLM Twitter service. The answer to the question whether those tweets should be disregarded in the analysis depends on the researcher goal. We decide to count those Tweets as positive as KLM manages to provide the customer with satisfactory answer and therefore create a positive brand experience.

Unfortunately, we are not able to benchmark the results with the data on customer satisfaction such as NPS scores, satisfaction survey results or number of complaints received by KLM. This data were not made available by KLM. Data that were provided allow us to conclude that in the course of negative events, such as publishing the statements about the company losses or having operational issues with flights, negative sentiment raises. The role of KLM's Twitter team is to mitigate the negative tweets therefore usually the negative sentiment peaks are followed by the positive ones. It shows how successful the brand is in managing the on-line conversation. It is important to prevent the negative opinions from spreading.

Furthermore, we notice that the on-line sentiment is influenced by the introduction of the new products, which can give very fast feedback on how the new service is perceived by the customers.

We try to verify whether the tweets preparation influences the classifier prediction. The tweets preparation improves the precision for the negative class by 8% and the recall for the objective and positive class by 3%. We expect that the better performance is due to the increase in the frequency of the features resulting from the spelling correction and synonym substitutions. Some words of the same meaning or misspells might originally have a low frequency therefore could be deleted

through the sparse term removal procedure. Further classifier tweaking could be obtained by developing the lists of mark-ups and users associated with the investigated classes and synonyms. We recommend improving the spelling correction process by incorporating more sophisticated techniques such as NLP models (Brill and Moore, 2000; Otero et al., 2007).

In our research we do not address the problem of sarcasm. It is very difficult to detect it from the text which maximal length is 140 characters. Since the tweet "*thanks #KLM for loosing my luggage again*" is an obvious example of sarcasm the "*again a great flight with @KLM...*"[7] is not. Therefore, we only label the clear cases as negative tweets. The rest is classified as positive. The way to handle the sarcasm more appropriately is to control for the polarity in tweets sent by the user in the corresponding time period, i.e., look whether he tweeted something before or just after and examine the sentiment of those tweets.

Future research might consider investigating words that contribute to the prediction. Some authors (Wu and Davison, 2006) recommend using the decision trees after labeling data with SVMs, as they provide user-friendly and interpretable output.

In our research we decide to delete the retweets because in our opinion the retweets do not contain the sentiment of the author but only of the original tweet sender. However, this approach might be altered and retweets can also be included in the analysis.

Worth considering would also be trying a two step approach. First classifying tweets either as objective or subjective, excluding the objective ones after classification and classifying subjective as either positive or negative. Such an approach has the advantage of having more balanced data but requires two trainings and test sets.

[7] This is the example of sarcasm, because it was followed by the tweet "*@KLM you did it again lost my luggage 2nd time this month*".

References

H. Achrekar, A. Gandhe, R. Lazarus, S.-H. Yu, and B. Liu. Predicting flu trends using twitter data. *Computer Communications Workshops (INFOCOM WKSHPS)*, 2011.

Cardi C. Bansal, M. and L. Lee. The power of negative thinking: Exploring label disagreement in the min-cut classification framework. *Proceedings of the International Conference in Computational Linguistics (COLING)*, 2008.

P. Beineke, T. Hastie, C. Manning, and S. Vaithyanathan. *Exploring Sentiment Summarization*, volume 07, pages 1–4. AAAI Press, 2004.

Asa Ben-Hur and Jason Weston. A user's guide to support vector machines. *Methods In Molecular Biology Clifton Nj*, 609:223–239, 2010.

J. Bollen, H. Mao, and X-J. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.

E. Brill and R. C. Moore. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 286–293, 2000.

Chih-Chung Chang. Libsvm. `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`, 2008. URL `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

J. A Chevalier and D. Mayzlin. The effect of word of mouth on sales: Online book reviews. *Journal of Marketing Research*, 43(3):345–354, 2006.

C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20(3):273–297, September 1995.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 1 edition, 2000. ISBN 0521780195.

K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 519–528, New York, NY, USA, 2003. ACM.

S. Dumais, J. Platt, J. Sahami, and D. Heckerman. Inductive learning algorithms and representations for text categorization. pages 148–155. ACM Press, 1998.

S. Dutta. Marketing yourself: What is your personal brand social media strategy? *Harvard Business Review*, 88(11):127–130, 2010.

N. Finn, A.and Kushmerick. Learning to classify documents according to genre. In *In IJCAI-03 Workshop on Computational Approaches to Style Analysis and Synthesis*, 2003.

N. L. Ford, B. G. Batchelor, and B. R. Wilkins. A learning scheme for the nearest neighbour classifier. *Inf. Sci.*, 2(2):139–157, April 1970. ISSN 0020-0255.

A. Go, L. Huang, and R. Bhayani. Sentiment analysis of twitter data. *Entropy*, pages 30–38, 2009.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Mach. Learn.*, 46(1-3):389–422, March 2002. ISSN 0885-6125.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction: with 200 full-color illustrations.* New York: Springer-Verlag, 2001.

D. Houser and J. Wooders. *Reputation in Auctions: Theory and Evidence from eBay.* 2001.

N. Hu, P. A. Pavlou, and J. Zhang. Can online reviews reveal a product's true quality?: empirical findings and analytical modeling of online word-of-mouth communication. pages 324–330, 2006.

C-L. Huang and C-j. Wang. A ga-based feature selection and parameters optimization for support vector machines, 2006.

A. M. Kaplan and M. Haenlein. The early bird catches the news: Nine things you should know about micro-blogging. *Business Horizons*, 54(2):105–113, March 2011.

A.M. Kaplan and M. Haenlein. Users of the world, unite! the challenges and opportunities of social media. *Business Horizons*, 2010.

B. E. Kim and S. Gilbert. Detecting sadness in 140 characters : Sentiment analysis and mourning michael jackson on twitter. *Web Ecology*, 03(August), 2009.

S.-M. Kim and E. Hovy. Crystal: Analyzing prediction opinions on the web. *Proceedings of the Joint Conference on Empirical Methods in Natural Language Pro- cessing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1): 273–324, 1997.

R. Kohavi and F. Provost. Glossary of terms. *Machine Learning*, 30(2):271–274, 1998.

U. Kressel. Pairwise classification and support vector machines. In B. et al. Schölkopf, editor, *Advances in Kernel Methods—Support Vector Learning*, pages 255–268. Cambridge, MA, 1999.

J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. *International Conference on Knowledge Discovery and Data Mining*, 495(978), 2009.

V. Levenshtein. Binary codes capable of correcting spurious insertions and deletions of ones. *Probl. Inf. Transmission*, 1:8–17, 1965.

Y. Mejova. *Sentiment Analysis, An Overview*, 2009.

Y. Miche, P. Bas, A. Lendasse, C. Jutten, and O. Simula. Advantages of using feature selection techniques on steganalysis schemes. In *Proceedings of the 9th international work conference on Artificial neural networks*, IWANN'07, pages 606–613, Berlin, Heidelberg, 2007. Springer-Verlag.

R. Mihalcea, C. Banea, and J. Wiebe. Learning multilingual subjective language via cross-lingual projections. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 2007.

G. Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 2001.

V. Ng, s. Dasgupta, and S. M. Niaz Arifin. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews, 2006.

X. Ni, G-R. Xue, X. Ling, Y. Yu, and Q. Yang. Exploring in the weblog space by detecting informative and affective articles. pages 281–290. ACM, 2007.

J. Otero, J. Graña, and M. Vilares. Contextual spelling correction. In *Proceedings of the 11th international conference on Computer aided systems theory*, EUROCAST'07, pages 290–296. Springer-Verlag, 2007.

A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, 2010.

B. Pang and L. Lee. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 115–124. Association for Computational Linguistics, 2005.

B. Pang and L. Lee. *Opinion Mining and Sentiment Analysis*. Foundations and Trends in Information Retrieval. 2008.

B. Pang, L. Lillian, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *IN PROCEEDINGS OF EMNLP*, pages 79–86, 2002.

F. Provost. *Machine learning from imbalanced data sets 101*, pages 1–3. AAAI Press, 2000.

R. Quirk, S. Greenbaum, G. Leech, and J. Svartvik. *A Comprehensive Grammar of the English Language*. Longman, 1985.

J. Read. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*, ACLstudent '05, pages 43–48. Association for Computational Linguistics, 2005.

C. Rich and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. *Proceedings of the 23rd international conference on Machine learning*, 2006.

T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. *19th international conference on World wide web*, 2010.

B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors. *Advances in Kernel Methods—Support Vector Learning*. MIT Press, Cambridge, MA, 1999.

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

F. Sebastiani and A. Esuli. Determining term subjectivity and term orientation for opinion mining. In *In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'06*, 2006.

H. Takamura, T. Inui, and O. Manabu. Latent variable models for semantic orientations of phrases. In *In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL*, pages 201–208, 2006.

M. Thomas and L. L. Pang. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2006.

H. Thomases. *Twitter Marketing: An Hour a Day*. SYBEX Inc., Alameda, CA, USA, 2010. ISBN 0470562269, 9780470562260.

Z. Tong and F. J. Oles. Text categorization based on regularized linear classification methods. *IR*, 2001.

J. C. Ward and A. L. Ostrom. Complaining to the masses: The role of protest framing in customer-created complaint web sites. *Journal of Consumer Research*, 33(2):220–230, 2006.

J. Wiebe. Tracking point of view in narrative. *Computational Linguistics*, 20:233–287, 1994.

Baoning Wu and Brian D. Davison. Detecting semantic cloaking on the web. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 819–828. ACM, 2006.

C. Yang, K. H-Y. Lin, and H-H. Chen. Emotion classification using web blog corpora. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, WI '07, pages 275–278. IEEE Computer Society, 2007.

A Appendix I

Table 8: Performance of the linear SVMs with cost parameter C=10, 5-fold cross-validated.

	# of features in the model			
Tweets class	50	100	150	all
Objective	0.8	0.78	0.73	0.79
Positive	0.63	0.65	0.61	0.63
Negative	0.39	0.49	0.49	0.48