Predicting sentiment of comments to news on Reddit

UNIVERSITY OF AMSTERDAM

MSc Thesis

Bruno Jakić - bruno@ai-applied.nl

Intelligent Systems Lab Amsterdam

Science Park 904
1098 XM Amsterdam
P.O. Box 94323
1090 GH Amsterdam
The Netherlands

Supervised by:

Wouter Weerkamp
w.weerkamp@uva.nl

Abstract

Today, methods for automatic opinion mining on online data are becoming increasingly relevant. Over the past few years, methods have been developed that can successfully and with a great degree of accuracy analyze the sentiment in opinions from digital text. These developments enable research into prediction of sentiment. Sentiment prediction has traditionally been used as a tool for stock prediction. In such scenarios, incoming news is analyzed in real-time and the impact of that news on stock prices is estimated, making automatic stock trading possible. Recent developments in sentiment prediction have seen attempts to predict explicit sentiment of the reactions to blogs, before the blogs are even posted. In this thesis, we research the prediction of the general sentiment polarity in reactions to news articles, before the news articles are posted. We use Reddit as our data source for news and comments, and approach the sentiment prediction problem using machine learning methods. To automatically label Reddit comments for sentiment prediction training, we perform automatic domain-knowledge transfer from a classifier trained on Twitter to Reddit. In this thesis, we propose a new machine learning method, a new feature selection method for text and a new machine learning evaluation metric. We provide a thorough analysis of Reddit data, and manually annotate a gold standard from it. Finally, we demonstrate the feasibility of sentiment prediction of the general sentiment polarity in reactions to news articles, before the news articles are posted in limited cases. Ultimately, we provide an analysis of the limitations of our and similar approaches to sentiment prediction, and make recommendations for future research.

Acknowledgments

I would like to thank my thesis supervisor, Wouter Weerkamp, for his support, comments and invaluable advice. I would also like to thank Maarten de Rijke and Valentin Zhizhkun for pointing me in the right direction in the early days of writing this thesis. Also, I would like to thank my colleague and friend Mark Mooij for providing me with invaluable assistance in the matters of domain-knowledge transfer and sentiment analysis in general. Finally, I would like to thank my parents, grandparents and extended family and many good friends for putting up with me all these years that I have been a student.

Table of Contents

1. Introduction

On a variety of online platforms, such as review sites, blogs, as well as social services such as Twitter and Reddit, internet users produce vast amounts of opinionated text about a large range of domains, such as movie reviews, travel experiences, product reviews, opinions about news and others. Automatic opinion mining - the ability to process large amounts of opinionated textual information from online sources without human interference - is necessary. The data sources include opinions about products, brands and developments which increasingly drive the decision making in business and government. Automatic opinion mining is divided into two categories; *qualitative* opinion mining, which attempts to extract pieces of literal information from the data, such as sentences describing an experience relevant to the target of the opinion and *quantitative* opinion mining, which attempts to determine quantifiable dimensions of opinion, such as sentiment. Sentiment analysis is utilized in order to determine the polarity of opinions (positive/neutral/negative), or the emotional charge of opinions across a range of possible emotions (love, fear, anger, understanding etc).

The field of sentiment analysis has recently witnessed a large amount of interest from the scientific community [1] [2] [3]. Sentiment analysis has traditionally been applied to a single domain at a time, such as movie reviews or product reviews [4]. More recently, much effort has been invested into development of sentiment analysis methods that can be used across multiple domains [3]. While the creation of general-purpose (cross-domain) sentiment analysis systems remains an unsolved problem, previous advances in sentiment analysis already yield some domain-specific systems which have near-human performance [1] [2].

In addition to sentiment analysis, research into the prediction of sentiment was conducted by a number of researchers [5] [6] [7] [8]. To expand upon the difference between sentiment prediction and sentiment analysis, we consider in abstract detail the methods used in sentiment analysis.

Sentiment analysis has been approached from a number of different directions, such as the application of lexicons with manually or semi-automatically annotated word polarities [9], Natural Language Processing methods [10] [11] and machine learning-based approaches [4] [1]. All such approaches determine words or phrases which denote subjective opinion in order to determine the sentiment polarity of the message. For example, [9] uses subjective words annotated with a weight and a polarity, such as "excellent" with a positive polarity and the weight of 5, or "poor", a word with a negative polarity and the weight of 5. Additionally, combined with Naive Bayes machine learning methods as in [4], every word is allocated a particular polarity and weight based on some training examples. Training of such systems is supervised by explicit polarity labeling or implicit interpretation of features such as "smileys" (":)" or "> - ("). In sentiment analysis, the subjective opinion of the creator of a message is explicitly present within the same message. In [11] [10] [12], researchers have established that subjective words, when analyzed using Natural Language Processing methods, are often adjectives, denoting the opinion of the message creator about the noun they belong to. For example "great (ADJ) suspense (NN)" or "worthless (ADJ) plot (NN)", indicate that the object of an opinion is, in many domains, located near the opinion in the message text.

On the other hand, when we intuitively consider a domain like news and its character, we observe that news is usually intended to be objective. This means that the opinion of the audience to the themes in a news article is not contained in the article itself. Instead, separate reactions to news, when available, contain the opinions of the audience towards the content of the news. Unlike in sentiment analysis, the text of a news article is not useful to determine the opinions to that article, except perhaps the opinions of the article's author. Since the news article contains objective information, and the opinions to an article are found in the comments to it, the characteristics of the commentators become important. The distinction between sentiment analysis and sentiment prediction is graphically demonstrated in figure 1.

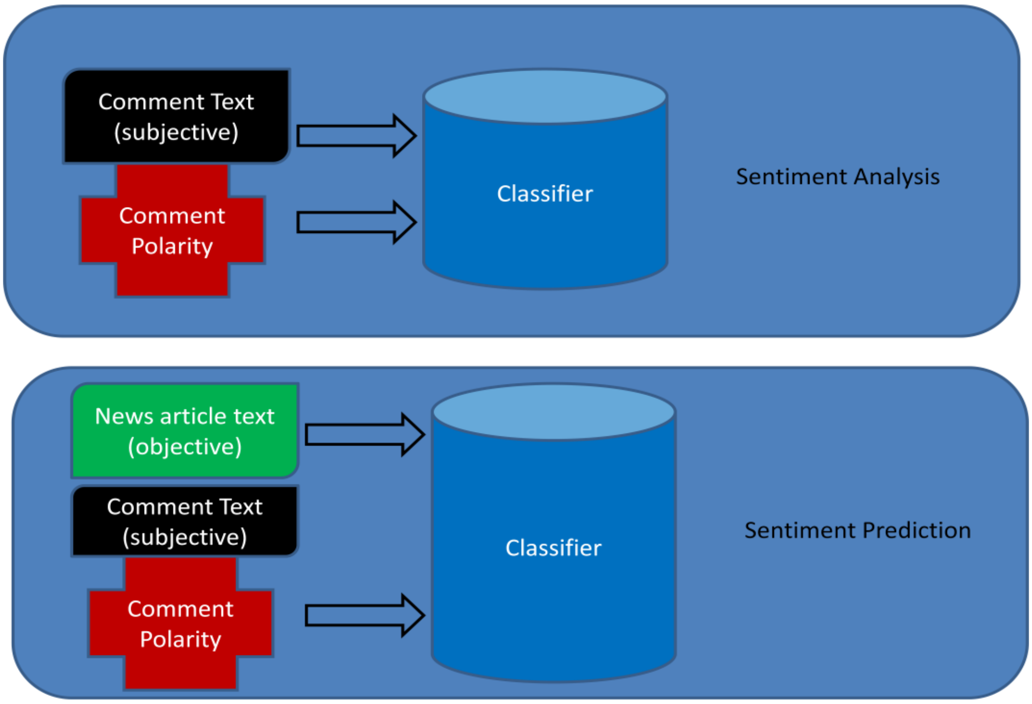

Figure 1: The difference between sentiment analysis and sentiment prediction in news expressed graphically.

Prejudices and attitudes of individual commentators are most determining for their reactions to objective news. It is, for instance, entirely possible that two commentators, can have different and opposite attitudes polarities towards the same article. Consider an article titled *"Tunisia Riots: US warns Middle East to reform or be overthrown".* Without further elaboration, the title of the article can be expected to evoke different reactions amongst people who agree with US foreign policy and those who disagree with it. The first group will probably have a positive attitude towards the message of that article, while the second group will probably have a negative attitude towards it.

Also, consider an article titled *"Congresswoman Giffords wounded, five killed in shooting".* Regardless of political affiliation and philosophical or other orientation, shooting of innocent people by a deranged lunatic can in almost all cases be expected to cause a strongly negative response from anyone who reads such an article. The idea that murdering innocent people is unacceptable is culturally conditioned in most contemporary societies. Conversely, an article titled *"Lady GaGa sweeps MTV awards in Belfast"* can be expected to have highly variable comment polarities, if the same most general group of people gives their opinion about it. The preference for Lady GaGa's music is after all very personal. If the general population, however, was segmented on whether they are Lady GaGa aficionados or not, the predictability of the reaction of the two groups to such an article would likely increase.

Automatic prediction of sentiment in reaction to news has already shown merit in practical applications in specific domains. The research focus on sentiment prediction in news, forums and blogs has traditionally been on the prediction of stock movements [5] [6] [13]. Blogs content and reactions are generally automatically processed to produce a reliable stock trading signal: buy, sell or hold. Other domains of sentiment prediction and analysis to news are the generation of news alerts, capturing the general tone in news articles towards relevant issues and assuming this to have predictive properties. Such systems are used to warn its users of "bad news" [14] [15]. In addition, systems have been created that provide the context to news articles in terms of their political and emotional polarity [16]. Political sentiment in news have been further explored in the estimation of the emotional impact of political news and blogs to specific audiences [12] [7] [8] [17]. They conclude that, despite the highly personal nature of the motivation behind every individual's reaction to news, in some cases groups of people react to news in similar and predictable ways. This effect is dependent on along which lines people are segmented in groups. For instance, the research in [7], focuses on predicting the reactions of people to political blogs. The commentators to these political blogs are divided into two political categories, Liberals and Conservatives. The article [7] demonstrates high sentiment prediction accuracy for this specific task.

In stock prediction [5] [6] [13], while news articles about a listed company themselves might talk of bad performance by the company's CEO, the reaction of the investors is not by default in agreement with the tone of the news article. Instead, the discussion occurring in the comments to the article reflects the investor sentiment with regard to the news message, by containing speculations and interpretations of news.

One flaw of the said approach is that it functions only once comments to news become available. Depending on the topic and news message location, this may never happen, or take a period of time spanning from seconds to hours after the actual news article has been published. For stock exchange prediction and the generation of news alerts where, for the users, timely reaction to news is of paramount importance, the time delay between the article publication and system's predictions can render such systems impractical. A method not relying on fresh comments to news for sentiment prediction, but instead reacting to the news article content itself, would be more prudent. Such a method would open new potential applications, such as automatic estimation of the impact of a news article before it is actually published.

The research in this thesis focuses on predicting the general sentiment polarity of the reactions to news on Reddit before a news article is published. News represents a wide domain in which the general sentiment prediction problem is only beginning to be considered [8]. News itself covers a large range of different categories and sub domains. For instance, there is news about world events, news about economy, news about celebrities etc. In "world news", multiple sub domains are covered, such as the success of a movie release or the outbreak of a war, which are thematically vastly different. The difficulty of the task in analyzing or predicting sentiment in news lies in the creation of robust methods and models which can capture the nuanced differences of word or phrase meanings across different domains present in news and different audiences.

We base our research on data collected from Reddit [18]. Reddit is a convenient source for retrieval of news articles and related comments since it centralizes the links to news and commenting capacities in a single place. It offers different categories of news and has an API that allows easy access to its data. The press has lauded Reddit [19] [20] for its culture of open and often polarized discussion. Much of this discussion is led along political lines [19]. This allows us to research the predictability of sentiment polarity in reactions to news for both general Reddit users, and groups of Reddit users separated along political lines into Liberals and Conservatives.

The goal of our research is to provide the answers to our main research question and it's sub questions:

> Can we predict the general sentiment of the reactions to news on Reddit before a news article is published?
>
> - Can we use bootstrapping from Twitter data to classify comments on Reddit as an step in automation of the training of our sentiment prediction system?
> - Does grouping users into homogenous groups, such as Liberals and Conservatives, and conditioning the prediction on those categories increase the prediction accuracy?

- Is the accuracy of sentiment prediction dependent on the category of news? If so, how can the dependency be explained?

- Does the performance of our approach to sentiment prediction compare favorably to similar methods presented in related work?

- Which of the following feature selection methods can we use to obtain the greatest prediction accuracy when modeling a news article: bag-of-words, Tf-Idf or Augmented Tf-Idf?

- Does context selection during classifier training improve sentiment prediction accuracy?

We begin this thesis with an overview of related work in the fields of machine learning, feature selection and sentiment analysis and prediction in chapter 2.

Subsequently, we provide an extensive data analysis of collected Reddit data in chapter 3. That collection consists of 26,299 postings from Reddit over a period of three months in 2011, which contain links to news articles and discussions directly related to these news articles. In this thesis, we provide a detailed analysis of the type and the quality of data we collect from Reddit, as well as a general overview of the Reddit platform itself.
We provide insights into the procedures and standards of data collection, distributions of data over news categories, the usability of data for machine learning tasks, the relevance of comments in the data and the politically separated user groups in the data. We also implement and demonstrate a method for classification of users into political categories based on their comment histories and the sentiment polarities thereof.

In chapter 4, we describe the procedure of the creation of a gold standard and a training set for benchmarking the experiments conducted in this thesis.

To obtain the sentiment polarity of the comments, we use automatic sentiment analysis. In chapter 5 we explain the utilized machine learning and domain-knowledge transfer methods. Large amounts of comments in our corpus make manual comment sentiment polarity tagging impractical. In this chapter, we describe, implement and verify various methods of domain-knowledge transfer and machine learning. To perform automatic sentiment annotation for our Reddit comments, we successfully use a domain-knowledge transfer method, Frequently Co-occurring Entropy with Estimation Maximization [3] to transfer the sentiment knowledge from a large, publicly available Twitter corpus to our Reddit corpus. The applied domain transfer method works with a Naive Bayes (3,4) machine learning approach to automatically annotate all comments in the training set for sentiment. We evaluate the results of domain transfer and conclude they are satisfactory and similar to those found in related work. We also propose a new performance measure, Classifier Performance, to evaluate the performance of classifiers in general and we explain its benefits when compared to measures such as accuracy.

In chapter 6, we conduct research into the prediction of general sentiment of the reactions to news on Reddit before a news article is published. We begin by implementing the approach from related work [7] that combines domain-knowledge transfer with sentiment prediction using Support Vector Machines (5), which is the most similar to our chosen approach. We evaluate this approach on our training set and gold standard, and use the results as a baseline to compare our systems against. We then describe feature selection approaches. Here we use the bag-of-words and Tf-Idf feature selections commonly found in related work. In addition, we propose an augmentation scheme to Tf-Idf to deal with some of the shortcomings of Tf-Idf feature selection. We also propose a new, context-sensitive method for machine learning, that combines Naive Bayes machine learning and a Nearest Neighbor-based approach. Subsequently, we conduct five experiments; one to determine the optimal number of Tf-Idf features, then to test the proposed augmentation scheme, to compare our classification approaches with the baseline approach and each other, and to measure the effects of predicting sentiment using specific news categories or user groups on sentiment prediction performance. We offer a overview, discussion and explanation of the experimental results.

In chapter 6, we present our conclusions based on the research in this thesis, and we provide explicit answers to our research questions. In our research, we have proven that the prediction of general sentiment of the reactions to news on Reddit before a news article is published is possible under certain conditions. We provide insights into those conditions, and in chapter 7, recommendations for future research on the basis of these insights. The main insight of our research is that categorization of users who react to news plays the most important role in building of robust sentiment prediction systems.

In this thesis, we contribute to the general body of knowledge with regard to the sentiment prediction. In addition, we contribute a novel method of augmentation of features selected using Tf-Idf methods, which yield a significant improvement in classification accuracy. Finally, we contribute a novel hybrid machine learning method which shows promise of better classification accuracies under specific conditions.

2. Related work

The research in this thesis builds on previous work in the fields of machine learning, feature selection, and sentiment analysis and prediction. In this paragraph we explore a body of work related to these methods.

2.1 Sentiment analysis

Sentiment analysis can be defined as the automatic extraction of quantitative opinions from subjective text. Over the past number of years, a large body of work describing methods, applications and insights into sentiment analysis was published [1] [2]. Most work has been done on specific domains, such as movie reviews, and on large texts [2], in some cases providing near-human accuracies of sentiment classification. In this paragraph we explore relevant methods for sentiment analysis.

In the introduction of this thesis we have briefly mentioned work by Go et al [1]. In their paper, Go et al introduce an interesting approach towards the classification of sentiment in Twitter messages. They were the first to perform sentiment analysis on Twitter data. They propose the idea of "distant supervision", defined as training with noisy data. Every Twitter message which contains a smiley character (table 1) is considered a signal denoting the sentiment of that entire tweet. Consequently, they automatically collect information from Twitter by filtering the general Twitter feed for smiley characters. In this way, they create a annotated training corpus, with which they train machine learning methods and create a sentiment polarity classifier.

Positive smileys	Negative smileys
:)	:(
:-)	:-(
:D	:<
:P	:'(

Table 1: Examples of smiley's as used on Twitter and mapped to sentiment class by Go et al [1].

They reduce the feature space in collected tweets by removing the common and uninformative occurrences, such as usernames, links, tweets containing ":P" smiley's, the smiley's themselves from the tweets, retweets and, on word level, repeated letters. This reduces the amount of noise in the training data. They also filter the collected dataset for a list of keywords, in order to constrain the domain on which they experiment. From the remaining tweets they then extract unigrams, bigrams and Part-Of-Speech tags to use as features for the classifiers. They employ Naive Bayes, Maximum Entropy models and Support Vector Machines as machine learning methods.

Go et al train the classifiers using the dataset annotated by distant supervision, and validate the classification performance against a manually constructed gold standard. They conclude that their results are close to state-of-the-art, with an accuracy of around 81%. Unigrams work roughly as well as bigrams as features in their approach, while POS-Tags aren't useful as features. The best results in their paper with unigrams as features are obtained by the SVM classifier.

Go et al refer to the influential paper by Pang et al [4] where those researchers have set a standard for machine-learning based sentiment analysis. Pang et al compare sentiment analysis to topical categorization of documents. Their approach is credited as one of the first attempts at applying machine learning techniques to the problem of sentiment analysis. The authors define the main challenge of sentiment analysis as the fact that the sentiment of a message is conveyed with much more subtle linguistic cues than in topic categorization. Pang et al conduct their research on the Movie Reviews domain. They scrape the text of movie reviews from Internet Movie Database [21] to collect their corpus of data, but only in those cases where the reviews have been additionally annotated with a number of stars or a numerical rating. Based on the rating, the extracted movie reviews are automatically converted into a Positive, Negative or Neutral category. They use only the Positive and Negative categories in their work. To avoid the domination of the corpus by prolific reviewers, they limit the maximal number of reviews per author. They establish a baseline for the evaluation of their systems, by using two human annotators whom they ask to manually construct a short list of words which denote positive or negative sentiment. Using these lists, they perform simple frequency-of-occurrence based classification on the automatically collected corpus of data. They conclude that, while the brevity of the wordlist partially accounts for the low performance of this approach, a list of the same length consisting of statistically common words from the corpus yields a great accuracy increase. Pang et al implement a Naive Bayes classifier, a Support Vector Machine and Maximum Entropy learning in a fashion which was directly followed by Go et al in their work: using unigrams, bigrams and POS-tags as features from text. Their machine learning approach greatly outperforms their human annotators, achieving accuracies of around 80% vs. human annotator accuracy of around 64%. They further explore the use of feature counts in a message vs. the presence of a feature in a message, and conclude that, in contradiction to previous work on topical categorization tasks, the use of only the feature presence improves the classifier performance.

While the early work by Pang et al achieves a good classification performance for sentiment analysis on the domain of Movie Reviews, Go et al with their subsequent work on Twitter achieve similar results applying the same methods to tweets. This is relevant, as tweets carry information about multiple domains. This indicates applicability of machine learning approaches to cross-domain classification of Reddit comments using Go et al's Twitter classifier, with additional optimization methods.

A recent overview compiled by Montoyo et al [2] provides a very useful insight into the state-of-the-art within the field of sentiment analysis. They define the field of research into sentiment analysis as the task of detecting, extracting and classifying opinions to different topics using text. They see this as a subset of the field of subjectivity analysis, which according the them deals with the detection of "private states", such as opinions, beliefs and speculations. They provide a detailed exploration of the key concepts and methods which have been used for the sentiment analysis task, presenting their respective achievements and directions for future research. They indicate that sentiment analysis systems and methods which are based on machine learning approaches the best performing in their fields. We present a short summary of other papers in this overview. The paper “A Lexicon Model for Deep Sentiment Analysis and Opinion Mining Applications” by Isa Maks and Piek Vossen deals with the use of SentiWordNet manually annotated sentiment corpora for sentiment analysis in a way that is dependent on the person expressing the sentiment. The main idea in this paper is to condition the sentiment models on users expressing the sentiment. We already have a variant of this approach planned in our experiments. Another paper, “Creating Sentiment Dictionaries via Triangulation” by Steinberger et al. deals with the automatic creation of multi-lingual sentiment lexicons. While interesting, this is not relevant for our research. The third paper in the overview is “Experiments with a Differential Semantics Annotation for WordNet 3.0” by Tufis and Stefanescu. Their contribution deals with ambiguity of words and their sentiment labels using manually annotated resources. The fourth paper in the overview is “Feature Selection for Sentiment Analysis Based on Context and Syntax Models” by Duric and Song. This paper proposes context-based methods for intelligent feature selection in sentiment analysis. The improvement in results compared with the work by Pang et al [4] are marginal at the expense of much greater complexity in system design. The following article by Lane et al, “On Developing Robust Models for Favourability Analysis: Model Choice, Feature Sets and Imbalanced Data” deals with the difficulty of machine learning on real media data in which issues change of context over time. The paper provides insight into the role of the balance between the positive and negative sentiment classes in documents, and provides ways to cope with the context change over time. Next is “That's your evidence? Classifying Stance in Online Political Debate” by Walker et al. This paper analyzes the sentiment polarities of stances in dialogs between people. The authors demonstrate that even humans have trouble identifying stances in dialogs when they are taken out of context. They provide adequate solutions to this issue which take context into account. Saif Mohammad presents in “From Once Upon a Time to Happily Ever After: Tracking Emotions in Mail and Books” a method to build a large word-emotion association lexicon trough crowd sourcing, and uses this lexicon to track emotions in books and mail. He explores how different visualizations can be used to interpret the results of emotion analysis, and describes experiments in which he measures gender preference for the use of certain emotion-related words. Finally, he analyzes books from Project Gutenberg to measure the association between emotions in novels and in fairy-tales. Further work on the challenges posited by Mohammed is undertaken by Balahur et al in “Detecting Implicit Expressions of Emotion in Text: a Comparative Analysis”. They specifically discuss the issues arising when analyzing text for sentiment which contains very few affective cues. They propose a number of methods based on the use of commonsense knowledge to account for this problem. In “Making Objective Decisions from Subjective Data: Detecting Irony in Customer Reviews”, Reyes and Rosso tackle one of the toughest problems in sentiment analysis: detection of irony and sarcasm. While the authors present a number of methods and experiments to tackle this problem, and contribute to the body of knowledge surrounding it, their work is limited to very specific cases and no generalizable results are presented. Finally, the last paper reviewed is “The Socialist Network”, by Matje van de Camp and Antal van den Bosch. They use sentiment analysis methods to extract person-centered social networks from historical texts. In this process, they use a high number of different machine learning techniques and providing a good comparative analysis of their performance. One of the simplest methods, Naive Bayes, is one of the best performing.

Montoyo et al conclude that while a large body work has been published on sentiment prediction, "much remains still to be done to create systems that perform reliably in real world situations". They discuss the lack of language corpora in languages other than English and set out challenges for the future. We observe that the results of the work by Pang et al and Go et al. compare favorably with the results of state-of-the approaches presented in this overview.

After reviewing previous work on sentiment analysis, we have not found any work performed on Reddit comments. This means that, to construct a sentiment analysis system for this thesis, we need to apply some form of domain-knowledge transfer. Recent work by Tan et al [3] outlines a straightforward method for performing cross-domain knowledge transfer in sentiment analysis. They do this by re-estimating the feature weights within a Naive Bayes classifier. Tan et al elaborate on the reasons for poor performance of classifiers trained on one domain and applied to another domain. They attribute this to the shift in the meaning of features depending on the context in which they are used. For instance, a word such as "unpredictable" will have opposite meanings depending on whether is used in the context of a movie review (a "unpredictable plot" is a good thing) or a car review ("unpredictable steering" tends to be bad). They assert that, despite the contextual sensitivity of some words, other words do have a similar meaning across domains, such as the words "good" or "excellent". Based their analysis Tan et al demonstrate a two-step approach to domain-knowledge transfer. First, they identify the "useful" words from the domains using the intuition that the words with similar amounts of relative occurrences in the base and the target domain have a similar meaning across these domains. In order to find such words, Tan et al use Frequently Co-Occurring Entropy, and extract the useful words to serve as a bridge between the domains.

Tang et al perform domain adaptation using the Estimation Maximization algorithm. This algorithm iteratively adapts the weights of words from the old domain to the new domain to a point of convergence. They first train a Naive Bayes classifier for the source domain, and then they re-estimate the word values towards the new domain, constructing a Adapted Naive Bayes classifier.

Tan et al evaluate their cross-domain knowledge transfer method by comparing the performance of their Adapted Naive Bayes classifier to a number of baseline approaches. These are a standard Naive Bayes classifier, a Estimation Maximization Naive Bayes classifier [22], similar in approach to their work but without the restriction the use only the generalizable features for domain transfer, and a earlier Naive Bayes Transfer Classifier by [23]. They evaluate the cross-domain performance over three domains: Education Reviews, Stock Reviews and Computer Reviews. For every domain, they train a classifier, then perform the sentiment classification on the other two domains (after transferring the knowledge, in the case of adaptive classifiers). They compare the performance of their approach using a Micro and Macro F1-Measure [24], and come to the conclusion that their system dramatically outperforms all baselines in cross-domain sentiment analysis. The improvements on the F1-Measures are between 0.10 and 0.20 against the second best approach.

For this thesis, we require a sentiment analysis system for automatic annotation of Reddit comments. After reviewing the state-of-the art in sentiment analysis by Montoyo et al, we decide to use the work by Go et due to the simplicity and state-of-the-art accuracy of their methods and the public availability of their large data corpus to construct our automatic sentiment analysis system. While Twitter data itself contains tweets representing many different domains, the distribution of these domains is likely to be different then in the Reddit data. For this reason we perform additional cross-domain adaptation from Go et al's classifier trained on Twitter data to Reddit comments for the sentiment analysis system using Tan et al's method.

2.2 Sentiment prediction

Sentiment prediction can be defined as the automatic prediction of what the quantitative opinions of some audience will be to some message, based on the contents of that message and the earlier observation of a similar audience's response to similar messages. Sentiment prediction is a more difficult task than sentiment analysis, due to the lack of explicit opinion in source data that is classified and the dependency on audience similarity for accurate prediction. Most research into sentiment prediction for news has focused on predicting the movements of the stock market [5] [6] [13]. In some recent work [12] [7] [8] authors are beginning to attempt to predict the sentiment polarity of reactions to news, which we too research in this thesis. In this paragraph, we explore some of the relevant methods of sentiment prediction.

In their paper [5], Cheong Fung et al present a system for stock prediction based on the information contained within the news articles. They investigate the impact of news articles on time series flow of stock signals, based on the Efficient Markets Hypothesis. Their work is relevant to our research, as the movement of the stock price represents a signal about the desirability of a stock, which encodes the sentiment reaction of the investors. They use data mining and text mining techniques in a novel way. Their system is event-driven, meaning that it can generate predictions in real-time. This constitutes an innovation compared to previous work on stock prediction using news. Cheong Feung et al utilize trend segmentation based on linear regression and clustering of interesting trends. Based on the generated and assigned trend signal label {Rise, Drop}, they cluster and align interesting news articles relative to the trend labels using the semi-supervised Incremental K-Means algorithm. This ensures that only the news relevant to a trend is linked to that trend. Every document in every cluster is represented as a normalized space-vector with individual words as the features of a document vector. Cheong Fun et al use the Tf-Idf algorithm for feature selection.

Using the selected features, Cheong Feung et al proceed to split their original clusters into two clusters each using the incremental K-Means algorithm. This algorithm uses the Cosine Similarity Measure to determine whether any document belongs to the cluster it originated in. To achieve this, the clustering algorithm first calculates a centroid for every cluster. After this calculation, the Cosine Similarity measure is applied to every document in a cluster comparing it to that cluster. Both documents and clusters in this case are represented as vectors of word-weights obtained by Tf-Idf (23).

The four newly created clusters are compared using Cosine Similarity (25), and discard, for both Rise and Drop signals, the one cluster which is more similar to the two clusters of the opposing signal. For example, one of the two Rise clusters most similar to both Drop clusters is removed. Hereby they manage to filter out the news documents which are less relevant to their signal class.

Additionally, Cheong Feung et al give higher weights to features present in those articles which support only one trend type. They define and measure a inter-cluster discrimination coefficient CDC, and a intra-cluster similarity coefficient CSC. They combine the CDC and CSC measurements together with a term frequency score for every word in every document, in order to determine the weight of that word for that document. The intuition behind this is that, the rarer a word in an article and a article in a cluster, the more important it is as a description feature for that article and cluster. During evaluation, they demonstrate that this scheme improves the recall of their model. We do not elaborate in detail on those procedures, since they are not of direct relevance for our research.

Cheong Feung et al utilize the features in the clustered documents as input features for machine learning using a Support Vector Machine (5). They use the signal labels {Rise, Drop} as the possible outcomes. They train two SVM classifiers, one for Rise detection, and one for Drop detection. For each classifiers, the positive examples consist of the cluster of documents that has been kept during pre-processing, and the negative examples consist of the documents in the cluster that has been rejected during pre-processing. To provide a stock trade recommendation, they rely on a positive classification by either one of the classifiers. If the classifiers both give a positive or negative classification, the signal is considered neutral and the system gives no recommendation.

The authors evaluate their system against an archive of about 350,000 financial news articles obtained from Reuters Market 3000 Extra, with their corresponding stock prices. They demonstrate, trough regression analysis, that the quality of their predictions is "high". By plotting a ROC curve of the clustering approach against a non-clustering approach, they demonstrate that their systems shows better ROC characteristics. Their main evaluation metric, however, is cumulative profit derived from their system's recommendations in a trading simulation. Here they demonstrate that they outperform the baseline from previous work up to a certain frequency of documents collected for learning, after which the baseline outperforms their system. The authors speculate that this effect is due to higher robustness of the baseline with regard to noise in the data.

While the article by Cheong Feung is not directly related to our research, it demonstrates a number of useful and common feature-selection and data-comparison methods. The evaluation metrics utilized in their research are inadequate, however, since they are qualitative and therefore hard to compare with other work in the field. The authors do demonstrate that features from news articles have predictive properties for their field of research.

Same authors have proposed a more extended version of their approach [6]. In this work, they focus more on the modeling of the stock time-series themselves, than news article content. Only changes to news article analysis compared with their earlier work are the use of stemming, stopwords and the removal of tokens from texts, such as numbers, URLs, e-mail addresses etc. This extended work demonstrates improved prediction performance, this time expressed as a quantity. The authors suggest that their research reveals the possibility that interrelations between different news articles might play a important role in prediction accuracy, when applied to the financial news.

Another stock sentiment prediction system, proposed by Seghal and Song [13], doesn't use any text features from the news articles. Rather it takes subjective opinion data from financial message boards in order to predict stock movements. This approach is on the border between sentiment analysis and sentiment prediction, since a sentiment polarity is being predicted, yet the prediction is conditioned on subjective data. Nevertheless, this article is relevant for our research, as it elaborates on feature selection for text and the modeling of user-groups for prediction purposes. Main point of this article is its use of a Trust Value given to every individual user on the message board. The users are segmented by the predictive values of their opinions, which are used as weights in the final recommendation signal calculation.

The authors obtain 260,000 posts for 52 popular stocks from Yahoo Finance message boards, and use this data for training and evaluation of their system. For feature selection, they use the Tf-Idf (23) formula, selecting the 10,000 highest-weighing features over all posts as the global feature space of their system. In addition, they label a unspecified number of the messages explicitly: StrongBuy, Buy, Hold, Sell or StrongSell. The authors use the labeled instances together with the features to train a number of machine learning based classifiers, such as Naive Bayes (3,4), Decision Trees, and Bagging. Using the trained classifiers, the authors classify all posts from the message board. Then, the authors compare the opinion of every message's author for every message for every stock, with the actual outcome of the movement of that stock. This way, they calculate an author's Trust Value per stock. Trust Value is a ratio between the amount of correct predictions and near predictions, and the total number of predictions by the author. A modifier called Activity Constant is used, which penalizes authors with a low activity on the boards. Finally, the authors re-train the machine learning classifiers, using as new features the sentiment predicted for a message board post by their previous classifier and the Trust Value of the author of that post.

Seghal and Song evaluate their system using precision and recall. Their systems are trained conditionally on text features and the stock ticker symbol, using the StrongBuy and StrongSell labeled instances. They conclude that their system attains a high accuracy and recall. We disagree with this conclusion, as the presented results in fact suggest overfitting in some cases. They inspect the features selected by the Decision Trees classifier manually, and conclude that the system is choosing appropriate terms to represent the classes. Finally, the authors evaluate the performance of their improved stock prediction system, which uses the predicted sentiment based on text in conjunction with Trust Value as features. They conclude that Trust Value yields significant improvements over the model with only sentiment analysis for some cases, while it performs worse on others.

We believe the precision/recall metric to be insufficient to evaluate this type of classification task. This paper is relevant to us for its successful use of Tf-Idf feature selection in order to perform sentiment analysis and stock prediction. Additionally, the research indicates that user modeling tends to improve prediction performance. They succesfully use the same feature selection method has been used by Cheong Feung et al, Tf-Idf. For this reason, we utilized the Tf-Idf feature selection method in our work. In this thesis we further explore the inadequacy of use of precision/recall metric for the measurement of general sentiment classification tasks.

Research into explicit prediction of sentiment has been conducted by authors interested in the impact of news and blogs on the political landscape. Research by Balasubramanyan et al [7] focuses on providing an explicit prediction of the sentiment polarity of comments based on the content of political blogs. This research has a very similar goal to our research in this thesis. In addition, they too use a "bootstrapping" approach, initially using automatic sentiment analysis to classify the comments to the blogs and then using this data to train their prediction system. They limit their research to political blogs. This makes the prediction of sentiment of the comments to posts easier, as the political orientation of the authors and the readers of these blogs is known in advance. Blogs, unlike news, also contain subjective text (opinions of the authors) next to objective data. Nevertheless, the feature selection and sentiment prediction training methods presented in [7] provide a useful guideline for our research.

Balasubramanyan et al claim that researchers must recognize that emotional reactions to blogs are different for different people. They limit their research to the domain of politics, and indicate that in that specific domain there is exploitation of "code words" by politicians in order to invoke a desired reactions in specific sub-sections of their electorate. Data used by the authors in this research is collected from five political blogs, each with a strong political affiliation shared amongst the readership and the writers of each blog. The blogs are about politics in the USA, and the political affiliations are Liberals and Conservatives. The authors use machine learning for the training of their sentiment prediction classifiers. Before prediction training, the authors use sentiment analysis methods to automatically classify the sentiment polarities of the comments to blog posts. For this task, the authors attempt two classification approaches: the use of a SentiWordNet lexicon, based on manual assignment of sentiment to words in WordNet, and a corpus-specific technique utilizing Pointwise Mutual Information (PMI).

SentiWordNet lexicon-based sentiment analysis works by computing the net score for all the words in all comments to a single blog post. Since every word in SentiWordNet has a certain sentiment weight (positive or negative) associated with it, the sum score for all the words in all comments is used to determine the sum sentiment polarity of the comments to a blog post. If the sum score is positive, the comments receive a positive label, and if the score is negative, the comments receive a negative label.

The PMI-based technique, has similarities of approach to domain-knowledge transfer with [3]. The authors statistically analyze the frequency of word occurrences in comments to blogs, and compare of these counts with the SentiWordNet list of positive and negative words. For each blog, the authors compile a separate seed-list consisting of 100 positive and 100 negative words using the measured frequency of occurrence and SentiWordNet scores of these words. For every other word in the comments to blogs, the authors calculate the average PMI of that word and all words in the positive and negative seed-lists respectively. The authors then calculate the difference between the positive and the negative PMI average, after which they select the top 1000 words as the positive, and the bottom 1000 words as the negative lexicon. They then use the created lexicon in the same way in which they used SentiWordNet for automatic sentiment polarity classification of comments.

The authors manually annotate the comment polarities for a gold standard for 30 blog posts. They evaluate the accuracy of the automatic sentiment analysis this gold standard, and claim an "excellent" performance of around 77% accuracy with the PMI-based method. The authors only report the accuracy of the system without any additional metrics or a confusion matrix.

The authors train their sentiment prediction system by using a bag-of-words representation of every blog post and the corresponding comment sentiment polarity label. Machine learning methods utilized by the authors are SVM (5) and Supervised Latent Dirichlet Allocation (sLDA). LDA [25] is a method for generative modeling of text which detects relations between terms and constructs topics based on their (latent) semantic proximity by assigning all terms weights per topic. Every document can then be described as a mixture of these topics. The Supervised variant of LDA, sLDA [26], is a supervised machine learning method using LDA topic distributions as features for every document and the supervision label provided for that document for training. Balasubramanyan et al use a limited number of documents in their experiments. They therefore set the sLDA classifier to assume the existence of 15 topics in the blogs, which is considered a low number for LDA. The authors perform regression analysis between the topics uncovered by LDA and the respective labels for topic-carrying documents. They indicate, upon manual observation, that LDA is very capable in linking the topics to their audience sentiments within the context of a political blog.

The authors evaluate the sentiment prediction accuracy of their system. They claim "excellent" performance of the prediction of the polarity of comments to blog posts, with accuracies between 74% and 90%. They also perform a cross-blog experiment in which they predict the sentiment in reaction to blog posts for blogs of the opposite political affiliation. They report a degradation in performance, indicating to the authors "that emotion is tied to the blog and the community that one is involved in."

Balasubramanyan et al also present an extension to their work [8]. While the greatest part of their research is identical to [7], they present a new LDA-based blog-response modeling method, Multi-community Response LDA (MCR-LDA) in addition to the ones mentioned in [7]. For MCR-LDA, the authors combine the training data of all blogs into a single corpus. Then, they train MCR-LDA classifier on both the sentiment polarity of the comments to blog posts, as well as to the number of comments, in a way that, when training on Conservative blog data ignores the comment polarities and volumes of Liberals and vice versa. The joint modeling still allows the topic commonality to be identified. They evaluate the performance of MCR-LDA and conclude that it performs as well as the original classifiers in the prediction of comment polarities within single blogs, but that it performs just as well in the cross-blog prediction. The authors provide no analysis of the role the additional feature, the volume of comments, plays in the improved performance.

Lerman et al explore another application of sentiment prediction to politics [12]. In their work, the authors use the content of news articles in order to predict the public perception of political candidates as expressed in prediction markets. Prediction markets provide a setting very similar to stock markets. Instead of representing a company or commodity, the "share prices" represent the outcome of sporting, financial or political events. An example given in their work is Iowa Electronic Markets [27] prediction market. The authors explicitly note that with their research they attempt to predict opinions, based on objective news, rather than subjective opinions. They want to "learn the impact of events in news on public opinion". In their approach towards the forecasting of the public perception of political candidates, the authors use linguistic information from news and internal market indicators as features. They note that, unlike in a sentiment analysis task, where they could process data across the whole dataset in a batch, in this type of approach they must use a chronological ordering in data processing. Due to this restriction, their system operates "online": learning follows prediction which follows learning. They use the data of all previous days until the day of classification for training with a logistic regression classifier, and then they classify the current day's instance. At the day's end, after having received the actual movement of the public opinion, their system stores the daily news and the movement label for future training.

The authors use multiple feature extraction techniques for news. They begin by representing the documents as bags-of-words. They also devise News Focus Features, which track the shifting focus of news across time, which are defined as the difference in frequency counts of unigram features from documents between two consecutive days. This is based on the intuition that old news will not create a change in opinion in either direction. Next, the authors use Named Entity Recognition in order to extract the names of people and places as features. They use only those sentences from documents in which exactly one named entity is detected. In this approach, all unigrams from the same sentence in which the named entity is discovered are considered features. Finally, the authors use Dependency Features, constructed by first POS-tagging the sentences from the text which contain a Named Entity, and then by using a dependency parser on the tagged sentences in order to map object-subject dependencies.

In addition to news data, Lerman et al use market history data in order to detect and learn general market behavior. They do this by using ridge regression on the price and volume data for the two days prior to the day used for labeling. Finally, they combine the market history features and the news features into a single system.

Lerman et al evaluate their approach in a similar manner to [5] [13]. They use a trading simulation on the prediction market [27]. As a baseline, they set use a trading method that assumes a continuation of the situation at the previous day. While all feature selection methods separately outperform this baseline, on average the Dependency Features outperform all other feature extraction methods. Interestingly, the performance of the bag-of-words approach performs roughly as well as the News Focus Features, and only slightly worse than Named Entity extraction. All feature selection methods lead to a greatly increased performance when used in conjunction with market history data.

Of the reviewed papers, the work by Balasubramanyan et al [7] [8] is the only one which presents a system for explicit prediction of sentiment polarity of comments to news before the news is published. We therefore use this approach as the baseline for our research, and build our system along the general lines proposed in their research.

2.3 Other works

While not directly concerned with the prediction of sentiment or sentiment analysis, a number of articles present interesting methods and insights for related tasks. These tasks include feature selection, prediction of the number of comments to news or the number of retweets, and user segmentation along political lines. We discuss them shortly in this paragraph and indicate their relevance for our research.

The work by Hong et al [28] explores the task of popularity prediction of tweets in terms of the number of retweets that the tweets would attract in the long run. The authors treat the task as a classification problem. They approach the problem binary classification to determine whether a tweet will be retweeted. Afterwards, they use multiple binary classifiers in order to determine in which expected retweet-volume class the tweet belongs. For feature selection, they use Tf-Idf (23), and LDA [25], as well as a number of temporal and binary features. These are: whether a tweet had been retweeted before and how much time had passed since the tweet posting. The authors demonstrate that a combination of features, including Tf-Idf and LDA, provides best prediction performance, which the evaluate using recall (16), precision (15) and the combined F1-score (17). Their best systems attains a F1-Score of 0.603, roughly an improvement of 120% over the baselines. This work is relevant for our research due to the used feature selection and machine learning methods.

Tsagkias et al [29] deal with the prediction of the number of comments to news. The authors treat the task as a parameter finding problem for models of user commenting behavior. The research compares the commenting behavior of users to blog posts and to news, and concludes that people are more likely to react to news in their dataset. The researchers also uncover that users are more likely to react to articles presented by certain news providers, and explain this by relative ease of commenting at those news providers. They further model the temporal cycles of comment frequencies, on a monthly, weekly and daily base. They consider two types of distributions to model comment volume: log-normal and negative binomial. They use the Chi-Squared test to evaluate the goodness-of-fit between their system's predictions and the observed data. After parameter estimation for the model distributions, they observe overall good performance for both types of distributions. The authors then explore the relation between the early and late comment volume to news. They observe that not all stories have the same probability of being commented on, which also depends on their publication time. To counter this, they introduce the concept for comments of "source time", or time since publication, instead of real-time. The source time is scaled to equalize the commenting probabilities for each article based on publication time. The authors observe random behavior in articles with few comments at early times. They remove such articles using k-means clustering. They then train a model which predicts the number of comments to a news article after a long time based on the observation of the number of comments at a time just after the article publication. The authors evaluate their approach using the relative squared error metric, and conclude that prediction of long term comment volume is possible with a small error after 10 source-hours of observation. This work provides to us valuable insights into the dynamics of user commenting behavior to news.

Work by Kim et al [30] presents a different approach to the prediction of politically motivated reactions of the audience based on news. They develop a computational model of political attitudes and beliefs on the basis of contemporary theories from social and cognitive psychology about "motivated reasoning". This model, John Q. Public, is a variant of a neural network. The authors simulate the behavior of political candidate evaluators based on empirical data, and find that their model outperforms a Bayesian learning model. They conclude that any learning model that does incorporate motivated reasoning will have difficulty accounting for the persistence and polarization of political attitudes. The authors provide a valuable insight into the capacity of different machine learning methods in tracking changes of political attitudes. This is relevant for our research as it provides proof that political attitudes do change, which can influence the segmentation of users into political categories.

A large amount of work has been done [31] [32] [33] [34] on improving the ubiquitous feature-selection method Tf-Idf (23). Without going into much detail, most performance gains obtained by different authors are either marginal [31] [32] or very domain specific [33] [34]. Previous work overall has demonstrated that despite its vintage, Tf-Idf algorithm is still a very feasible choice for feature selection from text.

Momtazi et al [35], compare four known methods for extraction of co-occurring terms from textual corpora in order to improve sentence retrieval. They describe the problem of sentence retrieval itself, and explain the approach to corpus driven clustering of terms. They consider four types of word co-occurrence: document-level, sentence level, window of text level, when a word co-occurs with another word in very near proximity, and co-occurrence in syntactic relationship, for instance when two words are the objects of the same verb. The authors proceed to describe experiments in which they benchmark the performance of different co-occurrence measurement methods in the sentence retrieval task. Finally, they conclude that he window of text level retrieval yields the best performance for this task. In our thesis we propose a system of feature augmentation for Tf-Idf retrieved features. This research provide useful guidelines for the set-up of a augmentation system based on word co-occurrence retrieval

The work by Park et al [17] presents an automatic approach towards the annotation of the political determination of news articles by sentiment analysis of comments to the news articles. Their approach focuses on the behavior of individual commentators. They sample a number of commentators to news from a popular news site and segment them into a Popular Set and a General Set. The Popular Set contains the users who react to 20 most read political articles in a day over a 6 month period. The General Set is samples from a historical directory of political issues, taking the users who reacted to 11 chosen issues over a 11 month period. The authors select the 50 most active users from both sets. They analyze the continuity of reaction behavior of these users, and conclude that the users in the Popular Set react to news much more consistently. They manually annotate the political affiliation of the commentators based on their comments and conclude that the Popular Set is somewhat dominated by Conservatives, while the General Set is somewhat dominated by Liberals. They also manually annotate 100 news articles and 100 comments as a gold standard. The authors implement a simple sentiment analysis system to automatically analyze the comments. They then propose a single-commenter and two multi-commenter methods for the prediction of the political polarity of a news article. The single-commenter approach is a multiclass Naive Bayes classifier. The first multi-commenter approach combines single-commenter classifications for multiple users and uses a voting scheme to determine the final classification. The other multi-commenter approach calculates the maximum posterior probability for a classification by summing the predictions of individual commentators. The two approaches are validated on the two datasets. The authors report the accuracy of their approaches as well as coverage, expressed at the proportion of the article that the selected users have reacted to. We are mainly interested in the accuracy of the approach. The authors report a classification accuracy of about 65% for the single-commenter and accuracies from 70% to over 80% for the multi-commenter approach. This article is relevant to our research since we perform automatic segmentation of users into political categories in this thesis. The methods presented in this article allow us to do this accurately.

Work by Brodersen et al [36] considers evaluation metrics for classification tasks. They observe that the most used performance measure for such tasks, accuracy, often provides a overestimation of the generalization potential of the such methods. This is attributed to intrinsically flawed treatment of the distribution of the results per fold, but also to not taking into account classifier bias on imbalanced datasets. The authors propose a alternate classifier evaluation method posterior balanced accuracy. The ratios of true positives and against all positives and true negatives against all negatives are averaged to obtain this metric. In this way, the reported result is balanced as to punish biased classifiers. They use the balanced accuracy to statistically analyze the results reported per classification fold, to calculate a posterior accuracy. Finally, they provide examples which demonstrate the superiority of their metric in representing generalization potential of classification methods. This paper is relevant for our research because we evaluate performance of classifiers, which can potentially be biased. Therefore we seek a metric that can reflect bias of a classifier in a single score.

In their word, Frank et all [37] propose a modification to the standard Naive Bayes classifier in order to adapt the learned classification model at run-time to the data that is being classified. This relaxes the independence assumption present in Naive Bayes. They re-estimate the probability distributions from the learned model for a *k* number of features related to the features observed in the test instance they want to classify, weighing the observed and related features based on a distance metric. For every new test instance, they perform model values re-estimation based on the originally trained model. They evaluate their approach and conclude that it seldom underperforms standard Naive Bayes, and most of the time outperforms it. As we propose a improvement of the standard Naive Bayes algorithm in this thesis in order to relax the independence assumption present in Naive Bayes, this work provides one example on how this can be approached.

In addition, Jiang et al [38] in their work present a system that combined Nearest Neighbor classification methods with Naive Bayes for accurate document ranking. They first compare k-Nearest Neighbor (KNN) approaches in several variations with Decision Trees and Naive Bayes on a ranking task. Then, they retrieve related data with KNN and train a Naive Bayes classifier using that data to create a new ranking approach. They address the issue of poor performance of Naive Bayes classifiers when a low amount of training examples are available by cloning the nearest neighbors to expand the training corpus. They finally evaluate their approach and conclude that it outperforms all individual classifiers. In our research, we follow a similar approach towards the construction of a classification (instead of ranking) system, to relax the independence assumption present in standard Naive Bayes. This research provides a analysis of the characteristics and weaknesses of such systems.

3. Data and Reddit

The research in this thesis focuses on predicting the general sentiment polarity of the reactions to news on Reddit before a news article is published. To answer our research questions regarding the influence of category of news and user group homogeneity on the accuracy of sentiment prediction, we require a news source that serves multiple categories of news and has commentators to news who are divisible into different user groups. It is imperative that reactions of user group at a time are about the same news article, to prevent source bias. For this reason, we need to obtain news and responses for different groups from the same platform. In addition, the volume of different topics recurring in news daily requires our training set to be sizable, as to capture sufficient training data on as many topics as possible. Experimental data therefore needs to be obtained from a source that displays a high number and variety of news topics, in multiple categories, which has an active and somewhat heterogeneous user community. For these resons, we collect data from Reddit.

Reddit [18] is a social news website, where the users, called "Redditors", post links to online content or create text postings themselves. This content often consists of news, but also of links to videos, images, blog posts and other material. For every submitted link, Reddit users can vote on how important or relevant they find the associated content. They can give positive and negative votes. Based on the votes and using a metric [39], the postings are ranked. This ranking is used to determine the display position of the postings on Reddit front-page(s) in the default setting. The articles can also be manually ranked by the number of reactions they have received, or chronologically. Every user can react to every posting, or to the reactions of other users. Users can also vote on the quality of the reactions. Same voting and ranking principles and options which are available for postings are used for the reactions as well.

All postings on Reddit are categorized by the poster into whatever category they feel the material belongs (a so called "subreddit"). Individual users can subscribe to subreddits to personalize what they see on Reddit homepage when they visit. The ten most popular subreddits as of 28-6-2012 are:

1. Pics
2. Funny
3. Politics
4. Gaming
5. Askreddit
6. Worldnews
7. Videos
8. Iama
9. Todayilearned
10. Wtf

All subreddits are open to all users to post and comment. Reddit's open culture [20], simple premise and basic user interface attract a large audience. The site serves a very wide demographic, social and political range of users, with most of its users coming from the United States [40]. As of 7-5-2012 there are around 67,000 subreddits, most popular of which have around 1,700,000 subscribers. As not all users explicitly subscribe to subreddits, the actual number of users is likely to be higher. Alexa.com [40] site visit ratings suggest this as well, ranking Reddit as 57th most visited site in the United States on 28-6-2012.

3.1 Data collection

While the wider Reddit community and social mechanisms are a interesting subject in itself, in this thesis we focus on the subreddits which contain the relatively largest amount of postings linking to established online news sources. Manual inspection of the content in most popular subreddits yields the following shortlist of news-content rich subreddits, which represent topics:

1. WorldNews
2. Politics
3. Technology
4. Science
5. Environment
6. Entertainment
7. Business
8. Economics

Reddit offers a simple Application Programming Interface (API). To facilitate data collection from Reddit, we implement the Reddit API and a scraper application in Python 2.7. The scraper polls the chronologically ordered posting listings for the shortlisted topics every 5 minutes. New postings are added to an internal watch list. Every posting present in the watch list is collected from Reddit in full exactly 24 hours later, to ensure that the comments for every posting are always collected over the same time period. We ignore the fact that the likelihood of a posting attracting comments could be influenced by the posting time. We assume that, since redditors from same geographical areas both post articles and comment to them, the posting and commenting volumes are correlated. For every posting, we collect both the posting link and all available comments.

In order to collect the full text of the linked news articles, we devise and implement a heuristic system in Python 2.7. The system collects the web-page from the link in the posting, and extracts candidates for article text from the web-page. For every candidate in the web-page, we calculate the relative weight of text versus HTML tags. Additionally, we modify the scores of the candidate sections with their Okapi BM25 score [41], using the webpage title as the search query. Finally, we select the highest scoring article text candidate and remove all HTML tags from it. We add the extracted article text to the Reddit posting and save the posting to disk.

3.2 Data overview

We collect 26,299 postings from the shortlisted topics (subreddits) between 15-1-2011 and 13-3-2011 which conform to the following rules:

1. Posted link does not refer back to Reddit (not a self-post)
2. Posted link refers to text content, consisting of at least 50 words in the article text. (not to a picture, video or other type of post)

A distribution of posting frequencies across the topics is displayed in Table 2.

The 26,299 postings contain a grand total of 221,799 comments, of which 12,746 postings contain zero comments and 13,553 contain one comment or more. There is an average of 8.43 comment per collected posting, or an average 16.37 comment per posting with nonzero comments.

Category (subreddit)	Postings
Business	668
Entertainment	1,423
Science	2,049
Economics	418
Environment	1,323
World News	3,554
Politics	14,236
Technology	2,628
Total	*26,299*

Table 2: Distribution of postings over Reddit categories in collected data

Manual inspection of comments reveals that Reddit users sometimes use sarcasm in their comments. We expect this to have a negative impact on the accuracy of our sentiment prediction system, as our base classifier is unable to deal with sarcasm.

Category	Comments
Business	5,389
Entertainment	6,775
Science	30,669
Economics	8,910
Environment	5,518
World News	50,121
Politics	90,447
Technology	23,940
Total	*221,799*

Table 3: Distribution of comments over Reddit categories in collected data

Fortunately, the number of sarcastic comments seems to be relatively low. Another possible source of noise is the occasional inclusion of a comment text within a comment on that comment. This leads to dilution in meaning of that comment.

The distribution of comments amongst the categories is displayed in Table 3. We observe that almost half of all comments are present in the Politics category. This supports the general claims and findings in [7] [12], that political themes in particular tend to drive a lot of discussion. In Table 4, we observe that Economics has a particularly high comments-to-postings ratio of 21.32. Manual inspection of the collected postings for that category reveals that the issues of global financial crisis and Obama administration's economic policy are major discussion drivers for this category. The proponents and opponents of these issues tend to engage in lengthy discussion, often including external sources supporting their claims. Simultaneously, a large number of other economy related postings, such as stock market info, often remain without any comments.

Category	C/P Ratio
Business	8.07
Entertainment	4.76
Science	14.97
Economics	21.32
Environment	4.17
World News	14.1
Politics	6.36
Technology	9.11

Table 4: Comments-to-Postings ratio over Reddit categories in collected data

Observing the comments-to-postings ratios in Table 4, we conclude that Reddit users in our collection like to discuss economics, science and world news the most, while they do most posting in politics, world news and technology.

3.3 Comment relevance

On Reddit, it is possible to comment on someone else's comment. This creates a nested hierarchy of comments during discussions. In this thesis, we are interested in predicting the sentiment polarity in comments to news. To train our systems, we need comments which only contain an opinion about the news article. For this reason, we manually inspect the collected corpus, to see whether the nested discussions contain opinions towards the article itself. We find that in many cases, nested discussions deviate thematically from what is being discussed in the news article. Instead they often consist of sarcastic comments, opinions about opinions or personal attacks.

To determine the level of nesting that still returns useful opinions, we randomly select 100 postings from the three categories with the greatest C/P ratio, under the condition that they contain at least one nested discussion. From these postings we extract 141 nested discussions. We divide the comments in these discussions into direct comments (not nested) and indirect comments (comments to comments, 1st level nesting). We ignore the comments with deeper nesting.

Using a simple web interface, which simultaneously displays the article text and the comments, we annotate every comment with a binary class {Relevant/Not Relevant} denominating the relevance towards the article. For this, we use two human annotators with a scientific background. Every human annotator annotates every comment. We compare the two annotations for every comment, and keep only those annotations on which the annotators agree, ignoring others. We evaluate the agreement between the annotators by using the Cohen's Kappa measure:

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)} \tag{1}$$

where κ is the Cohen's Kappa score, $\Pr(a)$ the relative observed agreement between two raters, and $\Pr(e)$ is the hypothetical probability of chance agreement.

Our annotators score a Kappa of 0.69, scores around 0.7 being considered as a high measure of agreement. We find that a indirect comment is almost 2 times less likely (P=0.49) to contain a clear opinion about the news article compared to a direct comment (P=0.92).

To attain a high relevance of comments towards the news articles, we modify our corpus to only contain direct comments to news. This yields a total of 25,346 comments. An additional benefit of this filtering is that we eliminate the noise from the inclusion of the text of a direct comment into the nested comment, which we have occasionally observed. Table 5 shows the distribution of comments per category, and table 6 the comments-to-postings ratio.

Category	**Comments**
Business	352
Entertainment	927
Science	1,452
Economics	411
Environment	911
World News	4,710
Politics	14,720
Technology	1,863
Total	*25,346*

Table 5: Distribution of direct comments over Reddit categories in collected data

The average comment-to-posting ratio is now 0.96 comments per collected posting, or 1.87 comments per posting that has nonzero comments.

The new comments-to-posting ratio distribution indicates that the categories most opinionated towards the news are politics, world news and economics. Manual inspection of the remaining comments reveals the presence of a strong polarity division along the political lines in those three categories. Even when discussing world news or economics, users tend to react from their own political framing. This is consistent with findings in [12] [7].

We calculate sentiment polarity in comments to a news article by averaging the polarities of individual comments to that news article. This means that the number of reactions to a news article potentially plays a important role in the calculation of sentiment polarity in comments to news.

Category	**C/P Ratio**
Business	0.53
Entertainment	0.65
Science	0.71
Economics	0.98
Environment	0.69
World News	1.33
Politics	1.03
Technology	0.71

Table 6: Comments-to-Postings ratio for direct comments over Reddit categories in collected data

In figure 2 we present a normalized distribution of comment frequencies per posting category. We cut away the frequencies higher than 10, as they occur only incidentally.

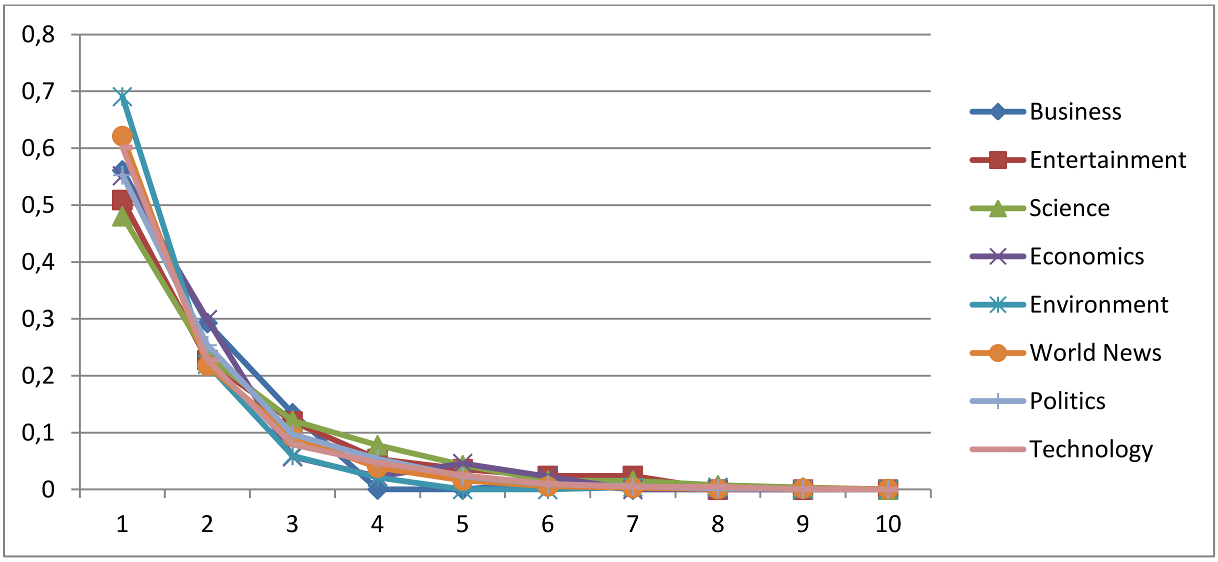

Figure 2: Normalized distribution of comments-per-posting frequency per category

In figure 2, we observe that Environment is the category with the highest relative amount of single-comment postings, and Science the lowest. This is consistent with the total distribution of comments over categories. Interestingly, Business and Politics, categories with the lowest and the highest number of comments respectively, have a very similar relative amount of single-comment postings.

We expect our sentiment prediction system to obtain a lower performance on the categories with a higher relative amount of single-comment postings, due to the high influence on the sentiment polarity of the low comment frequency. Having considered setting a threshold of 2 or higher to the minimal number of comments in a posting for our corpus, we reject this due to the need of a large number of news examples to cover a wide area of topics. Interestingly, while faced with the same problem, [7] did not report a worse performance of the classifier. Therefore, we expect this to present less of a problem when the sentiment is predicted for specific user groups as was the case in [7].

3.4 User groups

In this thesis, we research whether the division of users into homogenous groups can increase sentiment prediction accuracy. Previous work [12] [30] [7] [8] suggests that dividing the users in groups along political lines increases the prediction accuracy. During manual inspection of the data, we have observed that many opinions in discussions, even when the discussions are not explicitly about politics, are influenced by the political frame of the user. For example, in the Science category comments to news about advances in renewable energy technology regularly turn into discussions along political lines, with a clear pro- and against-side of the argument. Similarly, users in discussions about Economy news, for instance news about stimulus packages, often start using polarizing political terms for the other side such as "communists" or "teabaggers" [1]. For comparability with related work [7], and considering that, upon manual inspection, one of the most visible dividing lines on the opinions in our corpus is politics, we create three user groups from our corpus: Liberals, consisting of users with progressive political views, Conservatives, consisting of users with conservative political views, and All, consisting of the former two groups.

Based on manual observation, we define Liberals as users commenting in favor of renewable energy solutions, minority rights protection (having pro-immigration or pro-gay marriage views), being in favor of state-sponsored economy stimulus, having a positive attitude towards Obama administration’s policies, and being against the wars in Iraq and Afghanistan, the Tea Party movement, tax exemptions for large corporations, right to bear arms and (Christian) religion-based politics. We define conservatives as being in favor of lower taxation, cutting government spending, (Christian) religion-based values in politics, less state interference in economy, less minority rights protection, strong defense, the wars in Iraq and Afghanistan, right to bear arms, development of fossil and nuclear fuels, and against renewable energy solutions, state-sponsored economy stimulus, government

[1] A derogatory term for a (conservative) Tea Party movement member

entitlement programs, immigration, secular values in politics and Obama administration's policies. While the list of issues favored or disfavored by either side is longer, these are recurring themes.

Reddit does not support an explicit annotation scheme for user's preferences or identity. Based on the posting, commenting and voting activity of a user, "karma" points for the user are calculate. Separate "karma points" are awarded for postings and for comments. In addition, every user's posting history is visible. As "karma points" do not explicitly denote user preferences in any way, we use the posting history to infer the user groups.

Our corpus contains comments by a total of 14,110 users. The distribution of the number of users per comment frequency is displayed in Figure 3.

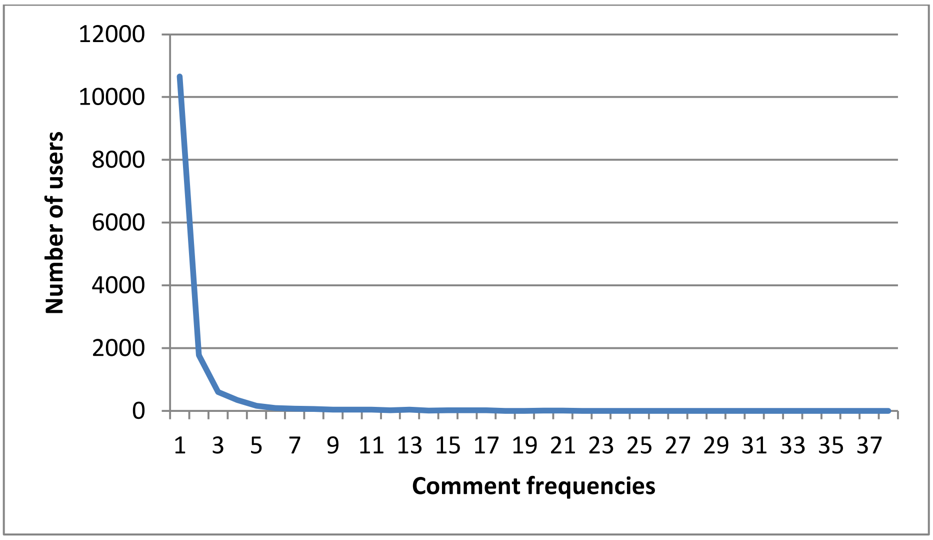

Figure 3: The distribution of user counts per comment frequency

We observe that 75% of the users in our corpus have contributed one comment only, while the direct comment average per user is 1.80. 25% of the users are responsible for 58% of all comments, and 5% of the users are responsible for 31% of the comments. The maximal number of direct comments contributed by a user is 72. For all comment frequency counts from 38 to 72, there are less than 2 users associated with them.

To divide users from our corpus into Liberals and Conservatives, we define the problem as a document categorization task. We solve it using a document classification approach, combined with the approach described in the work by Park et al [17]. We define documents in the task by grouping all comments made by one user into a single document. We observe most users in our corpus have contributed a low number of comments, making the task of grouping them into Liberals or Conservatives based on their comments difficult. For this reason and only within this task, we obtain all comments from the top-100 Reddit postings for every user from our corpus directly from Reddit. From our corpus, we randomly select 100 users. Using a simple web-interface which displays all additionally collected comments per user, we annotate every user with a class {Conservative/Liberal/Neutral} denominating their political allegiance based on the Liberal and Conservative definitions. The users who remain absolutely politically impartial in their comments we annotate as Neutral, and remove them from the selection. We use two human annotators with a scientific background. Every human annotator annotates every user. We compare the two annotations for every user, and keep only those annotations on which the annotators agree, removing others. The remaining selection we use as a gold standard for our bootstrapping scheme. Cohen's Kappa score for the agreement between the annotators is 0.67, denoting a high degree of agreement. The gold standard contains 96 users, of which 77 are Liberals and 19 are Conservatives.

Next, we obtain the 500 most popular postings with associated comments from the Conservative subreddits “Conservative”, “Republican”, “Prolife”, “Social Conservative” and “Paleoconservative”, together with 500 most popular postings with their comments from the Liberal subreddits “Liberal”, “Progressive”, ”Democrats”, “Green” and ”AllTheLeft”. From all postings we select the users which exclusively comment in the Conservative or Liberal subreddits. Next, we extract all direct comments by the Liberal or Conservative users from the postings in their respective categories. In addition, for every selected user from those categories, we collect all comments from their personal top-100 postings on Reddit. We use all extracted comments to train a Naïve Bayes (3,4) classifier for the {Liberal, Conservative} classes.

With this classifier, we classify all users in our corpus. From these users, we select 1,000 users classified as Liberals and 1,000 users classified as Conservatives with the highest individual comment counts. We proceed to implement the approach demonstrated by Park et al in [17]. We use the NB+FCE classifier described in paragraph 5.1 of this thesis to perform sentiment analysis of the comments left by the selected users.

We then apply a bootstrapping approach. Consider a posting with a news article *N*, and comments by both the users with known political affiliations *K*, and those with unknown political affiliation, *U*. We analyze the sentiment of the *K* users towards *N*, and from that we infer the polarity of *N*. Then, we analyze the sentiment of the *U* users towards *N*. With this knowledge, we can infer the likely political affiliation of a user *U*, as displayed in the following decision matrix:

		U=?	
		S=Positive	S=Negative
K = Liberal	S=Positive	Liberal	Conservative
	S=Negative	Conservative	Liberal
K=Conservative	S=Positive	Conservative	Liberal
	S=Negative	Liberal	Conservative

Table 7: Decision matrix for the inference of political orientation in unknown users. S = Sentiment.

For every comment by *U* over all postings in the corpus we take the inferred political affiliation and calculate the fraction of Liberal v.s. Conservative inferences for *U*. Finally, we assign the user *U* to the more prevalent inference over all postings, making them a *K*. Since we can only classify a limited number of users in one pass, we repeat the bootstrapping process until no more users can be assigned a political affiliation in this way.

The affiliation for some users *U* can't be inferred in this way, since they are never simultaneously present with a *K* in a posting. Usually, these users only have 1 posting across the whole corpus. To group these users, we return to document classification. We retrain our Naive Bayes classifier with all Liberal and Conservative comments of the known users *K*, and classify the comments of all unknown users with this classifier, assigning them to Liberals or Conservatives.

We evaluate the of this approach against the manually annotated gold standard. The evaluation results are displayed in Table 8.

Actual	**Predicted**		*Total*
	Liberal	Conservative	
Liberal	61	16	*77*
Conservative	5	14	*19*
Total	*66*	*30*	*96*

Table 8: Document categorization results for automatic user segmentation into political classes

We observe a document categorization accuracy of 0.78. We conclude that our bootstrapping approach is sufficient for automatic document categorization, as at this accuracy level we should see the effect of user grouping on the sentiment prediction accuracy. With the trained classifier, we classify every user in our corpus as Liberal of Conservative. This yields 10,771 Liberals and 3,339 Conservatives. We observe that most Reddit users in our corpus have a liberal political orientation.

4. Gold Standard

For the purpose of evaluation of our sentiment analysis and sentiment prediction systems, we manually create a gold standard. We begin by filtering the 26,299 postings collected from Reddit and the news sites, removing all postings with zero comments. The remaining corpus contains 13,553 postings, each consisting of the full text of a news article and the related direct comments from Reddit. The postings are distributed over Reddit categories as displayed in Table 9.

We believe that 500 postings is a good sample size for our gold standard, as it is bigger than most gold standards in related work [1] [3] [2] [7]. For our gold standard to be representative with regard to the full corpus, we select the posting frequencies for it in such a way that the category distribution for the gold standard is similar to the category distribution of the corpus (table 9).

Category	**Frequency**
Business	322
Entertainment	715
Science	1,064
Economics	201
Environment	679
World News	1,811
Politics	7,416
Technology	1,345
Total	*13,553*

Table 9: The distribution of posting frequencies over categories in the Reddit corpus

This is necessary to ensure that we minimize classification error due to divergence between the number of topics present in the training corpus and those in the gold standard. For every category from the corpus, we select a linearly scaled number of samples as displayed in Table 10.

We use the following selection procedure: first, we order the postings within every category chronologically, in order to obtain a reasonable temporal distribution for the reasons mentioned in [12]. Next, we calculate a selection interval, by dividing the total number of postings in that category by the number of postings we want to select. For example, the selection interval for the category "Business" is $\frac{322}{12} \approx 27$, meaning that we select every 27th posting from the chronologically ordered category "Business" in the corpus, until we have selected the desired 12 postings for that category.

Category	**Frequency**
Business	12
Entertainment	26
Science	39
Economics	7
Environment	25
World News	67
Politics	274
Technology	50
Total	*500*

Table 10: The distribution of selection frequencies for the gold standard from the Reddit corpus

To perform manual annotation of the comments in the postings, we use a simple web based interface. For every posting, the interface displays the full text of the news article in order to establish the context to the comments for the human annotators, for the reasons mentioned in [17]. In addition, below the full news text, all comments to the news are displayed with 3 category choices per comment for the human annotator: Positive, Neutral or Negative. The interface requires the human annotator to provide a manual annotation for every comment. After providing the annotations, the human annotator submits the posting to a central server, which stores the annotations. The human annotator is subsequently presented the next posting, until all comments in all postings have been annotated.

We provide our human annotators with the following annotation guidelines:

1. Always annotate the polarity of the comment intent related to the context of the news article. For instance, in case of perceiving sarcasm in the comment, annotate the *intent* of the comment, not its literal content.
2. When the comment is positive towards the article, regardless of the perceived intensity of that emotion, annotate as Positive.
3. When the comment is negative towards the article, regardless of the perceived intensity of that emotion, annotate as Negative.
4. When the comment is perceived as inconclusive, or if the content contains only objective statements, annotate as Neutral.
5. When the comment appears to be a reaction to another comment, annotate it's intent polarity towards the news article, if at all possible. Otherwise, annotate as Neutral.

The annotation process is conducted by two human annotators, both with a scientific background and strong familiarity with sentiment analysis methods. Every human annotator annotates the entire gold standard. We compare the two annotations for every comment in every posting, and keep only those annotations on which the annotators agree, removing the annotations on which the annotators disagree. In addition, we remove all annotations marked as "Neutral" from the gold standard, since we are only interested in the explicit sentiment polarity. Cohen's Kappa score is 0.70 indicating a high level of agreement amongst the annotators.

As a consequence of the removal of comments from postings, some postings no longer have any comments associated with them. We remove 111 such postings from the gold standard, remaining with a gold standard with 389 postings and a category distribution as displayed in Table 11.

To compare the fit between the original category distribution and the gold standard, we use Pearson's Chi-Square test to calculate the goodness-of-fit between the distributions:

$$\chi^2 = \sum_{i=1}^{N} \frac{(O_i - E_i)^2}{E_i} \tag{2}$$

where χ^2 is the Pearson's cumulative test statistic, $N$the number of available expectations and observations, O_i observation at *i* and E_i the expectation at *i*. We use the original corpus distribution as expectations, and the gold standard distribution as observations. We observe that the difference between the distributions is statistically significant[2]. We do not expect this to have a major detrimental effect on the sentiment prediction performance measurement due to the sensitivity of χ^2 to the number of categories, but that possibility exists.

While every posting in the gold standard contains both the full news article text as well as the comments, for the evaluation of the base classifier only the comment polarity is relevant. The gold standard contains a total of 744 comments, of which 264 are positive and 480 are negative. In order to build an evaluation set for the base (sentiment analysis) classifier, we extract all comments from the postings in gold standard and store them separately.

Category	**Frequency**
Business	10
Entertainment	20
Science	35
Economics	3
Environment	20
World News	40
Politics	218
Technology	43
Total	*389*

Table 11: The distribution of posting frequencies over categories in the gold standard

For every posting, we also calculate the prevalent sentiment polarity of the reactions to news. The table 12 displays the distribution of prevalent sentiment in comments polarity over news categories.

Category	**Positive**	**Negative**	*Total*
Business	7	3	10
Entertainment	17	3	20
Science	23	12	35
Economics	1	2	3
Environment	7	13	20
World News	16	24	40
Politics	66	152	218
Technology	26	17	43
Total	*163*	226	389

Table 12: The distribution of sentiment polarities over categories in the gold standard

We observe that both the extracted comments and the full postings have sentiment polarity distribution of roughly $\frac{2}{3}$ negative v.s $\frac{1}{3}$ positive polarities. Sentiment polarities in business, entertainment, science and technology categories are generally positive, while economics,

[2] p = 0.013474, lowest significance level (p< 0.05)

environment, world news and politics generally yield negative polarities. We explain this difference by the sensitivity of the subjects discussed in their respective categories; for instance, political themes tend to be both polarizing and concentrated on the negative issues in politics [12] [7].

Finally, we analyze the distribution of sentiment per political class. In Figure 4, we display the average positive sentiment distribution per user group for every category in the gold standard.

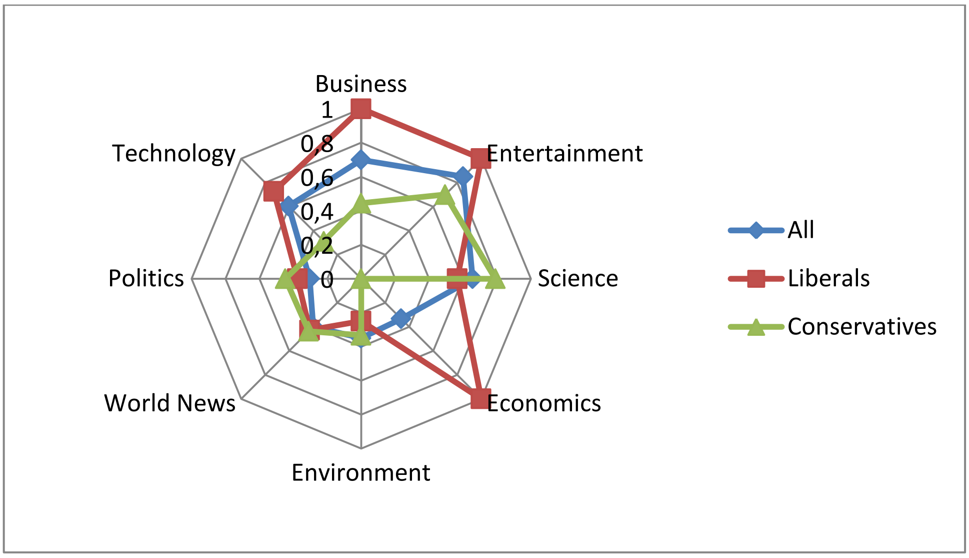

Figure 4: The positive sentiment distribution per user group per category in the gold standard.

To improve the clarity of this graphic, we display the positive sentiment only. Considering the binary nature of our sentiment representation, negative sentiment normalizes with positive sentiment to 1 and can therefore be left out. We observe that Liberals are generally more positive towards different categories than Conservatives or the general sentiment, with the exception of the category “science”. We also observe that in “economics” the sentiment difference between the user classes tends to be more extreme. We explain this by the low number of samples in that category.

5. Base classifier

In this thesis, we explore methods for sentiment prediction of comments to news. We utilize a machine learning approach for the training of our sentiment prediction systems. We train our prediction systems on (objective) features from news and the prevalent subjective opinion label of the comments to that news. As news itself contains a large amount of varying subjects, we require a large number of news articles in order to get a reasonable coverage of the news domain. To train our prediction classifiers, we have collected 13,053

Reddit postings that contain news articles. In order to determine the prevalent sentiment in comments for every news article required for prediction training, we need sentiment polarity annotations for every comment to that article. In their raw form, the comments from Reddit are not annotated with sentiment. Manual annotation of the sentiment polarity for every comment is a time and labor-intensive task, making a manual approach to sentiment annotation unfeasible. We pose the research question "Can we use bootstrapping from Twitter data to classify comments on Reddit as an step in automation of the training of our sentiment prediction system?".

In this section, we describe a machine learning approach for sentiment analysis, which we utilize to automatically annotate the sentiment of the comments in our training corpus. For this, we train our sentiment analysis system on data collected from Twitter using the methods proposed by Go et al [1]. We evaluate the performance of our sentiment analysis classifiers against the manually annotated gold standard.

In order to improve the performance of our base classifier, we apply and evaluate a number of cross-domain knowledge transfer methods [3] [7]. Additionally, we try a naive domain-adaptation approach by manually influencing class priors for Naive Bayes. Finally, we demonstrate improved performance of our sentiment analysis system attained by applying the cross-domain knowledge transfer approach proposed by Tan et al [3].

5.1 Classification methods

We use the basic approach by Go et al [1] for the construction of our base classifier. This approach has demonstrated good performance and relative simplicity in construction of the classifier. In addition, Go et al provide the training corpus used in their research, which is obtained from Twitter. This corpus contains 1,600,000 training examples, split equally between the Positive and Negative sentiment annotations. We expect this corpus size to be sufficient to contain most discussion-related terms in the English language which are also used in Reddit comments. Twitter itself is a platform where many different topics are discussed, so we expect a lower amount of bias towards any particular domain than might be the case with other annotated training corpora [3] [2] [7].

We take into consideration the option of creating both training and test corpora manually from the original Reddit data, but the process of manual annotation is too time consuming to consider. We also consider using more elaborate sentiment analysis methods [2] in conjunction with the Twitter corpus, but reject this approach due to unavailability of the results of such methods using Twitter data and near state-of-the art performance of Go et al's approach. Experiments in this direction would also be beyond the scope of this thesis. While Twitter data itself contains tweets representing many different domains, the distribution of these domains is likely to be different than in the Reddit data. Therefore, in addition to training machine learning methods for automatic sentiment prediction, we also attempt automatic domain-knowledge transfer between the two datasets.

Go et al use three machine learning approaches to construct their classifiers: Naive Bayes. Maximum Entropy and Support Vector Machines (SVM). Of these, Naive Bayes and Support Vector Machines demonstrate the best (and similar) performance. We implement and train baseline Naive Bayes and SVM classifiers, and try to improve their performance using cross-domain knowledge transfer methods [3] [7].

Naive Bayes classification is a simple model that is well suited and often used for text categorization. It applies the Bayes theorem under strong independence assumptions: it assumes that the presence or the absence of a feature in a class is unrelated to the presence or the absence of any other feature in that class. When applied to language, this assumption is incorrect. Despite this oversimplification, Naive Bayes classifiers have demonstrated good performance on language classification tasks [1] [4] [24]. Simply stated, the classifier is trained by counting the frequencies of word occurrences within the context of a predefined class (e.g. positive or negative sentiment). During classification, every word of the document being classified is fed into the trained model, and the final classification is made by selecting the class with the largest probability given the document. Written as a formula:

$$c* = argmax_c P(c|d) \tag{3}$$

$$P(c|d) = P(c)\prod_{w\in d} P(w|c) \tag{4}$$

with *c** denoting the class the document *c* belongs to according to the classifier, *P(c|d)* denoting the probability that the current document *d* belongs to the class *c*, with *P(c)* denoting the prior probability of the class *c*, and *P(w|c)* denoting the probability that the word *w* from document *d* is a member of the class *c*.

Support Vector Machine is a popular [1] [2] non-probabilistic binary classifier, which uses the input data provided during training in order to find a hyperplane in the space described by the data. This hyperplane best separates the data with an as large as possible distance to the inlaying data points. The mathematics and implementation details for SVM are more complex than those of Naive Bayes. Mathematically, separation hyperplane discovery can be considered a constrained optimization problem, expressed in the formula:

$$\vec{w} = \sum_j a_j c_j \vec{d}_j,\ a_j \geq 0 \tag{5}$$

The separation hyperplane in this formula is represented by $\vec{w}$, c_j represents the binary class (1 or -1), and those $\vec{d}_j$ in which a_j is larger or equal to zero represent the "support vectors" since they are the only document vectors contributing to $\vec{w}$. A much more elaborate treatise of SVM is available in the book by Cristianini and Shawe-Taylor [42], which provides a accessible explanation of the principles behind the theory and implementation of SVM.

Considering the mathematical nature of SVM, it is not possible to perform manual optimization on an already trained model. This makes direct domain-knowledge transfer impossible. Instead, Balasubramanyan et al [7] attempt cross-domain knowledge transfer using Pointwise Mutual Information (PMI) with manually compiled vectors of 100 positive and 100 negative words before they train a SVM. For every word in the comments to blogs, they calculate the average PMI of that word and the positive and negative seed-vectors respectively. They order all words by the difference between the positive and the negative PMI average and select the top 1000 words as the positive, and the bottom 1000 words as the negative features. Pointwise Mutual Information gives a relative score of co-occurrence of words within a corpus, by utilizing the formula:

$$PMI(a,b|C) = \log\frac{P(a,b|C)}{P(a|C)*P(b|C)} \tag{6}$$

where a and b are words of which the co-occurencein corpus C is calculated. We follow their approach in an attempt to improve the performance of our baseline SVM classifier.

Naive Bayes classifier allows the modification of an already trained model by influencing the probabilities of word occurrences in their respective classes within the model. We attempt two methods to exploit this possibility and improve the performance relative to our baselines.

The first method is very simple, and consists of a manual adjustment of prior probabilities in the trained Naive Bayes classifier. This is motivated by the fact that in the training data from Twitter, the distribution between the negative and positive examples is exactly equal, at a 0.50 probability for each. During the annotation of our gold standard, however, we have observed that the negative/positive comment ratio is roughly 0.67/0.33. We manually adjust the prior probabilities accordingly for the respective classes under the assumption of roughly similar probability distributions for the words between the two domains. We call this Manual Priors Adjustment (MPA).

The second method uses the Frequently Co-Occurring Entropy measure as described by Tan et al [3] to select the words from the source domain most suited to re-estimate the word values for the target domain. After that, the word values for the target domain are re-estimated using the Expectation Maximization algorithm. Mathematically, they describe the Frequently Co-Occurring Entropy (FCE) as:

$$f_w = \log(\frac{P_o(w)*P_n(w)^{\pi}}{(|P_o(w)-P_n(w)|+\beta)^{1-\pi}}) \tag{7}$$

where f_w is the FCE measure, $P_o(w)$ and $P_n(w)$ represent the word probabilities in respecitvely the old and the new domain, the modifier β is set to a very low value to prevent the extreme case of division by zero if $P_o(w)$= $P_n(w)$, and the weight π is introduced to allow the scaling of the importance of the two domains relative to one another.

The Estimation Maximization algorithm consists of separate Estimation and Maximization steps. When applied to the context of the work by Tan et al, the estimation step consist of using the current Naive Bayes estimates and a classifier to classify the new domain data. The subsequent maximization step takes the form of:

$$P(c_k) = \frac{(1-\lambda)*\sum_{i\in D^o} P(c_k|d_i)+ \lambda*\sum_{i\in D^n} P(c_k|d_i)}{(1-\lambda)*|D^o|+\lambda*|D^n|} \tag{8}$$

$$P(w_t|c_k) = \frac{(1-\lambda)*(\eta_t^o*N_{t,k}^o)+\lambda*(N_{t,k}^n)+1}{(1-\lambda)*\sum_{i\in|V|}(\eta_t^o*N_{t,k}^o)+ \lambda*\sum_{i\in|V|}(N_{t,k}^n)+|V|} \tag{9}$$

where c_k denotes a class (positive,negative) and $P(c_k)$ denotes it's prior probability, D^o and D^n represent the old and the new domain respectively, expressed as collection of instances d_i belonging to that domain, $P(w_t|c_k)$ expresses the re-estimated conditional probability of a word relative to the class and λ represents a parameter that controls the impact between the old and new domain data. The value of λ is controlled by:

$$\lambda = \ min\{\delta * \tau, 1\} \tag{10}$$

where τ indicates the iteration step $\tau\epsilon\{1,2,3,4 \ldots M\}$, and δ is a constant which controls the strength of the update parameter in the the range between 0 and 1. $N_{t,k}^o$ and $N_{t,k}^n$are the numbers of word appearances in the old domain and new domain respectively, within the class c_k. They are described as:

$$N_{t,k}^o = \ \sum_{i\in D^o}(N_{t,i}^o * P(c_k|d_i)) \tag{11}$$

$$N_{t,k}^n = \ \sum_{i\in D^n}(N_{t,i}^n * P(c_k|d_i)) \tag{12}$$

where $P(c_k|d_i)$ is the outcome of the estimation step for the sample instance d_i given the class c_k.

η_t^o is another constant, which is defined by:

$$\eta_t^o = \begin{cases} 0 \ if \ w_t \notin \ V_{FCE} \\ 1 \ if \ w_t \in \ V_{FCE} \end{cases} \tag{13}$$

where $V_{FCE}$are the generalizable features selected using the method described in (7). (8,9) indicate that, during the maximization step, only the generalizable features from the old domain and that all features from the new domain are used in the calculation of the new probability estimates. The EM algorithm iterates, until a local maximum likelihood for the estimated model is reached. The log likelihood of the parameters to be maximized is expressed by:

$$l(\theta|D,z) = \ \sum_{i=1}^{|D|}\sum_{k=1}^{|C|} z_{ik} \log(P(c_k|\theta)P(d_i|c_k;\theta_k)) \tag{14}$$

with θ indicating the model with value estimates, D indicating all data (form old and new domain combined), and z_i indicating a binary vector where the authors indicate a z_{ik} value of 1 for the class they want to track, and otherwise a z_{ik} of 0.

5.2 Experimental setup

To answer our research question "Can we use bootstrapping from Twitter data to classify comments on Reddit as an step in automation of the training of our sentiment prediction system?", we implement a Naive Bayes classifier, the FCE/EM domain-transfer algorithm [3], all preprocessing scripts and the PMI-based domain-transfer algorithm [7] in Python 2.7.

To prepare the training data, we apply the pre-processing techniques as described in [1] to the Twitter corpus consisting of 1,600,000 tweets, of which 800,000 are positive and 800,000 are negative. We begin the preprocessing by splitting the tweet strings into words on the whitespace character. We remove "@"-tags, "#"-tags, numerical values, "smileys" (table 1) and URLs from tweets. We normalize the character table in tweets, so that characters such as "á" or "ë" become "a" or "e" respectively. We subsequently remove all non-alphabet characters with the exception of " ' " which is often used in phrases such as "don't". Further, we remove all in-word character repetitions if the character occurs more than twice consecutively. For example, "i juuuustttt looooveeee it" become "I just love it", while "moore's law" remains the same. In addition, we lowercase all words and remove excess spaces between the words. From the resulting word vectors, we remove 214 commonly occurring stop words which are known to carry little emotional weight [43]. We submit the gold standard, consisting of 744 Reddit comments, to the same preprocessing procedure.

From the 13,053 Reddit postings which are not in the gold standard, we extract all 24,572 comments and pre-process them in the same way as the training data. We use them as the not-annotated goal domain corpus for the FCE/EM algorithm [3].

To execute domain adaptation by PMI as described in [7], we use the first 100 positive and 100 negative words provided in the seed lists in the said work. Instead of using the 24,572 comment list as for FCE/EM, we create 13,053 bags-of-words pooling all comments per Reddit posting, as prescribed in [7]. Following the procedure, after the PMI calculations we select the 1,000 top scoring words as positive features, and 1,000 bottom scoring words as negative features.

For SVM classification, we use the freely available libSVM classifier described in [44]. libSVM requires the data to be formatted as a sparse matrix of class tags and data points. To convert our language-based training corpus into the appropriate format [44], we create a script that calculates a set of all words observed in the tweets. This script subsequently maps every word in every tweet to a position in the sparse matrix for SVM. We likewise process the gold standard, using the all-word set computed on the training data. If a word exists in the gold standard but not in the all-word set, we ignore that word. We normalize the feature values and the class labels in the SVM training set, following instructions in [44]. For the classifier itself, we use the linear kernel and other default settings.

We perform the same type of processing to the PMI-domain-adaptation feature set, with the difference that the all-word set consists of the 2,000 features selected by that algorithm. We evaluate the adapted classifier against each comment from the gold standard separately, deviating from [7], to be able to compare to baseline measures. The lower number of features possibly leads to instances in which none of the features known to classifier are present in a comment during evaluation. In such cases, we set the default outcome to be "negative", due to the observed distribution of positive vs negative comments in Reddit data.

We run all experiments under Ubuntu Linux 11.01 64-bit operating system, on a machine with a Intel Core Quad Q6600 2,6Ghz processor and 4GB DDR3 RAM. All experiments are run in-memory.

5.3 Results and discussion

We evaluate the performance of the trained classifiers on the manually annotated gold standard with Reddit comments. For every experiment, we calculate a full confusion matrix, containing the intersections of the true values from the gold standard and the classifier estimates (see table 13). In related work [1] [13] [7], authors have used the metrics of Accuracy, precision and recall, and F1-Measures to evaluate the performance of their methods.

They [13] define precision as:

$$precision = \frac{True\ positive\ instances\ predicted}{Total\ instances\ predicted} \tag{15}$$

and recall as:

$$recall = \frac{Positive\ instances\ predicted}{Total\ positive\ instances} \tag{16}$$

Those metrics are combined into a harmonic mean score of recall and precision, the F1-Measure defined as:

$$F1 = \frac{2*recall*precision}{recall+precision} \tag{17}$$

Finally, the accuracy is defined as:

$$accuracy = \frac{correctly\ identified\ instances}{total\ instances} \tag{18}$$

We believe that these metrics alone is sufficient for a thorough understanding of the classifier performance, as they are highly dependent on the character of the test data against which they are evaluated. Let's take a hypothetical "sad corpus" of 100 instances, 90 negative and 10 positive.

Actual	**Predicted**		*Total*
	Positive	Negative	
Positive	203	61	*264*
Negative	235	245	*480*
Total	*438*	*306*	*744*

Table 13: Example of a confusion matrix for the Naive Bayes classifier performance.

Using the accuracy measure on this corpus, with a classifier which always chooses the "negative" class, this classifier would attain a very high accuracy of 0.90. If we had a "happy corpus" with the exact opposed distribution, the same classifier would perform terribly. Were precision and recall to be used as metrics [13], and applied only to the "negative" class for the "sad corpus" or the "positive" class for the "happy corpus", they would also give an unfairly optimistic performance assessment. This problem was considered by Brodersen et al [36] and a alternative measurement, balancing the score against classifier bias, has been proposed. The measurement by Brodersen et al, however, relies on cross-validation in classification experiments. In our setting, no cross-validation can be performed as we evaluate the data against a gold standard. In high similarity with work by Brodersen et al, and in order to include the bias and accuracy of a binary classifier in a single measure using a proven [13] balancing mechanism under our evaluation conditions, we propose a new metric: Classifier Performance (CP). Just as precision and recall are combined in a harmonic mean, the F1-Measure, for CP we propose taking the harmonic mean of the F1-Measures for both classes:

$$CP = \frac{(2*F1_{positive}*F1_{negative})}{F1_{positive}+F1_{negative}} \tag{19}$$

where, to balance out zero values:

$$F1_c = \begin{cases} \frac{2*R_c*P_c}{R_c+P_c} & iff\ R_c + P_c > 0 \\ 0 & iff\ R_c + P_c = 0 \end{cases} \tag{20}$$

with $F1_c$ being the F1-Measure for a class {positive, negative}, R_c the recall of that class and P_c the precision of that class.

This gives us a harmonic mean of the harmonic means of the precision and recall per binary class. Using this measure, we let the data distribution in the test corpus play a less important role in the evaluation of the classifier performance. Biased classifiers obtain a CP that is much smaller than their accuracy ratings, while fully biased classifiers, such as the one from the example, attain a CP of 0.

We also benchmark the statistical significance of the obtained results using Student's T-Test for paired samples:

$$t = \frac{\bar{X}_D}{\sqrt{s_D^2/n}} \tag{21}$$

where $\bar{X}_D$ is average of the differences between the baseline scores vector and the new scores vector, $s_D$is their standard deviation and n is the degree of freedom for the comparison (in our case the number of samples in the gold standard). We indicate the level of significance (if any) using the star rating: * if p<0.05, ** if p<0.01 and *** if p<0.001. We compare all Naive Bayes experiments with the first Naive Bayes baseline, and the cross-domain SVM experiment with the SVM baseline.

In table 14, we show the performance of different classifiers and domain transfer methods tested against the gold standard.

We observe that the accuracy of the trained baseline classifiers applied to Reddit comments is much worse than when applied to their native domain of Twitter [1]. This is a expected effect, which is widely reported in previous work [3] [7] [23]. In addition, the observed baseline accuracies are very similar to those reported in [7] before domain knowledge transfer. SVM demonstrates a slightly better baseline accuracy and performance than Naive Bayes, which is consistent with findings reported elsewhere [1] [24]. The training of the SVM classifier takes 4.5 days - a very long time compared to Naive Bayes, which trains within 8 minutes. The FCE/EM adaptation takes 18 hours to converge.

	NB[3]	*SVM*[4]	NB+MPA	SVM+PMI	NB+FCE
Accuracy:	0.6	0.62	0.74***	0.59	**0.77***
Positive Precision	0.46	0.48	0.67	0.41	**0.68**
Positive Recall	0.77	**0.79**	0.55	0.33	0.7
Negative Precision	0.8	0.82	0.77	0.67	**0.83**
Negative Recall	0.51	0.53	**0.85**	0.73	0.82
Positive F1-Score:	0.58	0.6	0.6	0.37	**0.69**
Negative F1-Score:	0.62	0.64	0.81	0.7	**0.82**
Classifier Performance:	0.6	0.62	0.69*	0.48	**0.75***

Table 14: Evaluation results of classification and domain transfer methods. NB = Naive Bayes, SVM = Support Vector Machine, MPA = Manual Priors Adjustment, PMI = Pointwise Mutual Information, FCE = Frequently Co-occurring Entropy and Estimation Maximization

We note that both the MPA and the FCE/ME domain-knowledge transfer approach significantly outperform the Naive Bayes baseline. This is particularly interesting in the case

[3] Naive Bayes baseline

[4] SVM baseline

of MPA, as the method employed is very simple. This at the same time suggests a somewhat similar distribution of words in their classes between Twitter and Reddit domains, with different document distributions (priors). Also, comparing the negative and positive F1-Scores for NB+MPA approach, we conclude that the manual adaptation of prior probabilities does introduce bias towards the negative class into the classifier. That the classifier is biased is also evident from its Classifier Performance, which is much lower than the classifier accuracy.

Interestingly, the application of the PMI domain transfer method demonstrates the worst performance by far. This is in contrast with the findings reported in [7]. We explain this by a number of differences between the approach from [7] and our approach. In [7], the authors analyze political blogs. This makes it likely that the comments to the blog posts are of a very consistent sentiment, e.g. everyone is happy with their political party or outraged at the opponent, while the entire discussion is centered on the politics domain. On Reddit, discussion between people of different affiliations is much more common in the comments. This means that the PMI scoring scheme produces much less clear discrimination between the positive and negative word associations in the pools of comments. Also, [7] evaluates the classifier on pools of comments, and not per individual comment as in our gold standard. This makes it more likely that a sufficient amount from 2,000 feature terms selected as the result of domain adaptation is present in the pool, which is much less likely in the much shorter individual comments from Reddit. Also, our classifier treats the comments which contain no known features as negative. Finally, the authors in [7] manually tune the initial positive and negative lists, to fit the data they are adapting the domain to. We omit this due to the fact that this process is not explained in the source article. The use of only 2,000 features per instance to train the SVM classifier greatly shortens the training time.

The domain knowledge transfer method utilizing Frequently Co-occurring Entropy and Estimation Maximization [3] shows a large and highly significant improvement over de Naive Bayes baseline. This is in agreement with the results reported in [3]. Compared to MPA, this method adapts both the prior probabilities distribution and the word distributions between the two domains. This prevents the bias which MPA displays.

Based on the experimental evidence, we use the NB+FCE as our base classifier. Using this classifier, we annotate the comments in 13,053 postings remaining in the Reddit corpus to be used for sentiment prediction training.

6. Sentiment prediction

In this thesis, we explore methods for sentiment prediction of the comments to news before the news is published. We utilize a machine learning approach for the training of our sentiment prediction systems. We train our prediction systems with the (objective) features from news and the prevalent sentiment in the comments to the news. In this chapter, we explore the characteristics of training data, describe and evaluate a baseline sentiment prediction system, define methods for feature selection from news articles, formulate two methods for sentiment prediction and evaluate the performance of the sentiment prediction systems against a gold standard. We seek to answer our research questions:

Can we predict the general sentiment of the reactions to news on Reddit before a news article is published?

- Can we use bootstrapping from Twitter data to classify comments on Reddit as an step in automation of the training of our sentiment prediction system?
- Does grouping users into homogenous groups, such as Liberals and Conservatives, and conditioning the prediction on those categories increase the prediction accuracy?
- Is the accuracy of sentiment prediction dependent on the category of news? If so, how can the dependency be explained?
- Does the performance of our approach to sentiment prediction compare favorably to similar methods presented in related work?
- Which of the following feature selection methods can we use to obtain the greatest prediction accuracy when modeling a news article: bag-of-words, Tf-Idf or Augmented Tf-Idf?
- Does context selection during classifier training improve sentiment prediction accuracy?

We demonstrate two well-performing sentiment prediction methods, mixed results in sentiment prediction for distinct categories of news and an improvement of the classifier performance for specifically defined groups.

6.1 Training data

We train our sentiment prediction systems on 13,053 Reddit postings in 8 categories, with comments automatically classified for sentiment. In addition, we train our sentiment prediction systems on user groups {Liberals, Conservatives, All} and news categories. In order to test the similarity between the training data and the gold standard for group and category divisions, we calculate the distribution of the positive polarity strengths over categories and user groups. Figure 5 displays this distribution.

To give a quantitative analysis of the differences in positive polarity distributions over news categories between the training data and the gold standard for different groups, we use the following formula:

$$d(g) = \frac{\sum_{i=1}^{n_c} |G(g)_i - T(g)_i|}{n_c} \tag{22}$$

where $d(g)$is the average difference between the gold standard positive polarity percentage and the training data positive polarity percentage for all categories for the group g, $G(g)_i$ is the average gold standard positive polarity percentage for the group g and the category i, and $T(g)_i$ is the average training corpus positive polarity percentage for the group g and the category i, while n_c is the number of categories.

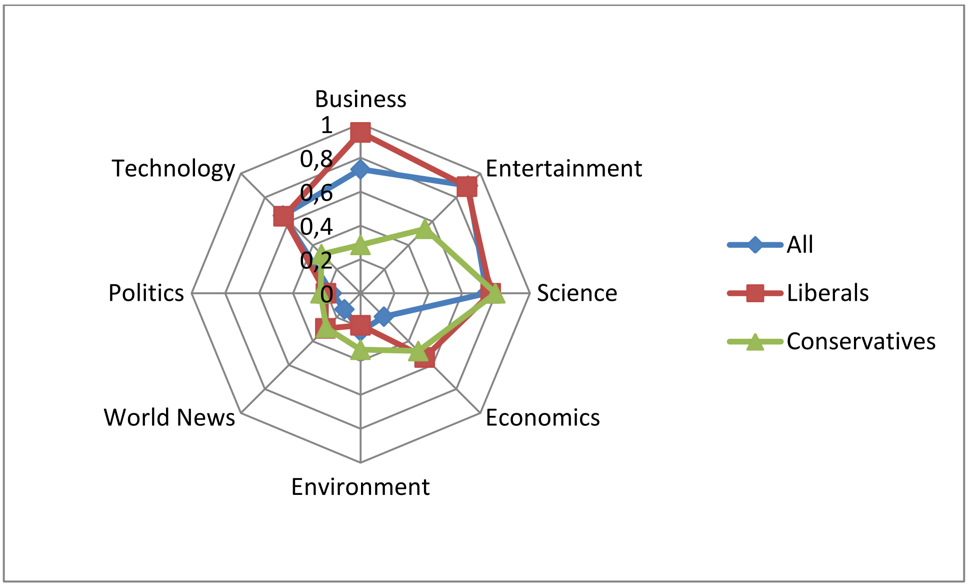

Figure 5: The positive sentiment distribution per user group per category in the training data.

The quantitative differences are displayed per user group in table 15.

	All	Liberals	Conservatives
Average Difference	0.11	0.29	0.37

Table 15: The polarity difference between the training data and the gold standard per user group.

While visual comparison of the positive sentiment plot of the training data in Figure 5 is generally similar to the same plot of the gold standard represented in Figure 4, there are a number of notable, yet not significant[5], differences. In the gold standard Liberals tend to have much more positive attitude towards Economics, while in the training data Conservatives tend to have a much more positive attitude towards Science. We explain this

[5] As verified using Pearson's Chi-Square

by the fact that the gold standard contains a very low number of samples in Economics, and that reactions in Science have, upon manual inspection, proven to be extremely conditioned on whether the related news article is about global warming or not. The low number of samples in the gold standard makes it unlikely that the samples in the gold standard are representative for the training data for those categories. In addition, the higher difference between the polarities of the gold standard and the training data for the specific user groups, Liberals and Conservatives, goes against our intuition that the division of users into more homogenous groups should yield more homogenous sentiment polarities. Low sample counts and polarized themes in the mentioned categories also account for this effect; if we ignore the mentioned categories, we obtain the results which are in line with our intuition that users in more homogenous groups have more a more consistent sentiment, as demonstrated in table 16.

	All	Liberals	Conservatives
Average Difference	0.11	0.08	0.10

Table 16: The polarity difference between the training data and the gold standard per user group, with Economics and Science removed.

Overall, the differences between the polarity distributions over categories are not significant. Table 17 shows the general distribution of the polarities per category within the training corpus, as well as the general distribution of Reddit postings over the categories in the training corpus. Both distributions do not differ significantly[5] from those in the gold standard. We use the entire training corpus as the training base for our experiments including the mentioned suboptimal categories.

In addition to category, polarity and user group distributions, topical coverage of the training data when compared to the gold standard is an important factor for the performance of the sentiment prediction system. The main idea behind sentiment prediction is to model the objective topics in news articles and link them to subjective reactions found in the comments within the same Reddit posting.

Category	Positive	Negative	*Total*
Business	149	55	*204*
Entertainment	374	42	*416*
Science	494	169	*663*
Economics	42	175	*217*
Environment	132	456	*588*
World News	356	2,295	*2,651*
Politics	1,286	6,325	*7,611*
Technology	456	247	*703*
Total	*3,289*	*9,764*	*13,053*

Table 17: The distribution of Reddit posting comment polarities in the training corpus.

Therefore, it is important to ensure that there are no large differences in the coverage of topics in news article texts between the gold standard and the training set. The measurement of topic distribution similarity between articles is highly dependent on the chosen model for article representation. Generally, in this thesis we model the news articles using the words from news article text as features. To compare the training set and the gold standard, we count the word occurrence frequencies in the training set and in the gold standard, ignoring the stop words[6]. The ten most popular words in the training set and the gold standard are displayed in table 18. Counting all words from the gold standard, we observe that the coverage of words from the training corpus compared to the gold standard is 0.92, or around 92%.

Training set	Gold Standard
People	One
One	Can
Government	People
State	Boehner
Us	Percent
Just	Obama
Time	President
President	Year
Even	Public
Year	Government

Table 18: The ten most popular words in the training set news articles and the gold standard news articles.

While the "coverage gap" of around 8% seems large, the difference between the normalized word distributions in the gold standard and the training set is not significant[7]. We expect no fundamental issues due to insufficient coverage of the gold standard by the training set with regard to the sentiment prediction performance. In specific cases, for instance when a very low number of features is used, the coverage might play a role. If applicable, we mention this in experimental results.

Manual observation of the most common words in both corpora reveals that the most talked-about themes involve (U.S.) politics, as can be expected from the distribution of Reddit postings in both corpora.

6.2 Baseline

In order to establish a baseline for our work, we have implemented a sentiment prediction system as proposed by Balasubramanyan et al in [7]. While in [7] both Support Vector

[6] Full list of used stop-words is provided in Appendix A

[7] As verified using Pearson's Chi-Square

Machines and Supervised Latent Dirichlet Allocation are used for sentiment prediction, we only implement the Support Vector Machine (SVM) variant, since the difference in performance between the classifiers is marginal in most experiments [7]. In addition to the SVM classifier, we implement the PMI-based cross-domain knowledge transfer method from their paper as described in chapter 5.1 of this thesis.

Considering the different approach towards labeling of the comments used in [7], for baseline evaluation we create a separate variant of the training set and the gold standard. Instead of trying to modeling the comment polarity label for a news article by using the ratio of positive v.s. negative comments, we pool all comments into a single bag of words and perform the PMI-based cross-domain knowledge transfer as per [7]. This provides a single label per news article for the training corpus. This creates a Reddit posting distribution different from our original training set. There is a clear tendency towards more negative labeling, as displayed in table 19. This tendency can be explained by the higher prevalence of "negative" seed-words for the PMI domain-transfer method from [7] in the comments of Reddit.

Category	Positive	Negative	*Total*
Business	111	93	*204*
Entertainment	292	124	*416*
Science	371	292	*663*
Economics	37	180	*217*
Environment	97	491	*588*
World News	249	2,402	*2,651*
Politics	846	6,765	*7,611*
Technology	312	391	*703*
Total	*2,315*	*10,738*	*13,053*

Table 19: The distribution of Reddit posting comment polarities in the specific baseline-training corpus.

We have already established in the section 5.1 of this thesis that the PMI-based domain-transfer method proposed in [7] performs poorly for Reddit data. This explains the large and highly significant ($p<0.005$)[8] difference between the polarity distributions from our original training set, and those from the training set adapted for the baseline measurement.

In order to create a gold standard conforming to the methods used in [7], we split the manually annotated comments in the gold standard into words. To every word in these comments manually annotated as positive in the gold standard, we assign the value "1", and to every word from comments annotated as negative we assign the value "-1", converting all comments into a single bag of words per Reddit posting.

If the resulting bag of words carries more positive words, the entire comment section is labeled as positive, and otherwise as negative. This provides a single sentiment polarity label

[8] As verified using Pearson's Chi-Square

per news article in the gold standard. The distributions in the adapted gold standard are displayed in table 20.

The changes to the gold standard labeling are minor and insignificant[8]. They are generally caused is the (arbitrary) length of individual comments manually labeled as positive or negative, and now grouped together in a single bag of words.

Category	Positive	Negative	*Total*
Business	7	3	*10*
Entertainment	16	4	*20*
Science	21	14	*35*
Economics	1	2	*3*
Environment	8	12	*20*
World News	16	24	*40*
Politics	70	148	*218*
Technology	25	18	*43*
Total	*164*	*225*	*389*

Table 20: The distribution of news article comment polarities in the specific baseline-gold standard.

After the adaptation of the data for the baseline measurement, we deploy the LIBSVM [44] SVM classifier. We implement the necessary middleware and the PMI-based domain transfer method in Python 2.7 using methods described in paragraph 5.2 of this thesis. We train the SVM classifier for the prediction task using the words from news articles in the adapted training corpus as features together with the polarity labels. We run all experiments under Ubuntu Linux 11.01 64-bit operating system, on a machine with a Intel Core Quad Q6600 2,6Ghz processor and 4GB DDR3 RAM. All experiments are run in-memory.

We evaluate the baseline sentiment prediction performance against the adapted gold standard. The results of the evaluation are displayed in table 21. The baseline performance is poor, with an accuracy of 0.56 and a Classifier Performance of 0.49. The bias we observed in the adapted training set seems to play a large role in the poor performance of the baseline sentiment prediction system. The evaluation results show a clear negative bias.

Compared to the results presented in the original article by Balasubramanyan et al [7], when applied to the task of predicting the comment polarity for Reddit postings, the original system from [7] performs poorly. There are a number of reasons for this poor performance.

In chapter 5 of this thesis we have already established that the PMI-domain knowledge transfer method as described in [7] yields poor results when applied to our corpus. This issue itself has a number of reasons. The authors in [7] point out that they have obtained their results by a manual adaptation of the seed lists for PMI-domain knowledge transfer to best suit the data, without providing explicit guidelines how to repeat this process. Consequently, we use the general seed lists provided in Appendix A of [7]. Additionally, the authors evaluate the performance of their system by using cross-validation on a single, completely automatically annotated, corpus of data. This would be analogous to us performing both training and classification using cross-validation on the adapted training set. If the automatic annotation system in such a case is biased, same bias applies both to training data and the test data in every fold. Greater accuracy is attained by this method than when validating against a objective, manually annotated gold standard devoid of that bias.

	Baseline
Accuracy:	0.56
Positive Precision	0.47
Positive Recall	0.33
Negative Precision	0.6
Negative Recall	0.73
Positive F1-Score:	0.39
Negative F1-Score:	0.66
Classifier Performance:	0.49

Table 21: Baseline classification results using the adapted training set and gold standard with the methods from [7].

Balasubramanyan et al apply their method to political blogs. Blogs, by their nature, contain opinion themselves, and as such are likely easier to model compared to objective data present in news. In addition, said blogs are always linked to a single general theme, politics, and usually to clearly defined groups of users - the adamant supporters of the political orientation of the blog, or die hard detractors, the latter being a small minority. Manual inspection of the blogs mentioned by [7] reveals that the language used in the comments is very clear and polarizing. The political blog environment from [7] differs from our Reddit corpus in that our corpus both covers multiple categories of news and that almost every discussion contains comments by people with views varying both in sentiment polarity and intensity. The task of predicting sentiment for news on Reddit therefore seems harder, which we believe explains a part of the poor performance of the baseline system.

6.2 Feature selection

A number of methods for feature selection in sentiment prediction based on news article text have been proposed in previous work [5] [6] [13] [12] [7], including the use of full article text as a bag-of-words, Tf-Idf-based selections, Latent Dirichlet Allocation variants and the use of POS tagging and Named Entity Recognition. Most used methods are Tf-Idf (and variants) and bag-of-words. While in most research [5] [6] [13] [12] Tf-Idf provides an improvement over a bag-of-words approach, a comparative analysis [12] of the prediction performance demonstrates that the use of more sophisticated feature extraction methods for this task only yields marginally improvements in classifier performance, if any at all. For this reason, in our research we use bag-of-words and different configurations of Tf-Idf for the selection of features from news article text. While a number of improvements to the Tf-Idf algorithm have been proposed for different tasks, comparative research [31] [32] [33] [34] has demonstrated that performance increases are either marginal [31] [32] or strongly domain specific [33] [34]. For this reason, we use the original Tf-Idf algorithm for feature selection. To select important feature terms from the documents, for every word w_i in the document d_k we calculate:

$$TF * IDF(w_i, d_k, N) = \frac{f_{w_i}}{\sum_{w_j \in d_k} f_{w_j}} * \log(\frac{\sum_{w_c \in N} F_{w_c}}{F_{w_i}}) \quad (23)$$

where N represents all words with their document-counts present in the language corpus, f_{w_i}represents the frequency of the word w_i in the documentd_k, f_{w_j} represents the frequency of any word w_j in the document d_k, F_{w_c} represents the document-frequency of any word w_c present in the language corpus N, and F_{w_i} represents the document-frequency of the word w_i in the language corpus N. In this way, the words more common in the current news article text, but less common in the general language – the more "descriptive" words - obtain a high Tf-Idf value, and vice versa. The Tf-Idf values assigned to words can be used to sort the words and select a number of most descriptive words as features to represent the document.

Regardless of the chosen feature-selection method, we process every news article text consistently in order to remove noise. Firstly, we remove any HTML tags that might be present in the text. Secondly, we lowercase all text and convert all accentuated characters to the standard characters; for instance, characters such as "é" or "ï" are converted to "e" and "i". Subsequently, we remove all non-alphabetical characters from text, including numbers and interpunction characters, with the exception of the apostrophe ' character often used in words such as "don't" and "can't". We do not use stemming or more advanced processing on words, as such has been demonstrated to create no improvement of the results in related work [12].

When using the bag-of-words feature selection method, we split the preprocessed news article text into words on the blank-space (" ") character. We remove duplicate words, as related work [4], has demonstrated that using just the presence of a feature consistently and significantly yields improved results compared to the use of the frequency of the feature in Naïve Bayes or SVM classifiers.

We define two of variants of Tf-Idf based feature selection. Initially, we build a language corpus using the words from all 13,053 news article texts from Reddit postings in order to obtain document frequency counts for every word present in the entire corpus. Subsequently, for every document, we split the preprocessed news article text into single word features on the blank-space (" ") character, and count their frequencies within that document. Using the document word counts and the known document frequency counts, for every word of the document we calculate a Tf-Idf value.

An important issue often not explicitly addressed in previous work is the number of Tf-Idf-selected words to use as features. This issue is important, as the selection of an appropriate number of features can ensure that noise, such as stopwords, are naturally left out of feature selection. In addition, in Naïve Bayes classification, terms which are very rare in one class and are often present in another best discriminate between the classes. Hence, it is beneficial to have fewer, high-value terms representing the documents. Therefore, we expect the selection of the number of features to have a high influence on the classifier performance of the sentiment prediction system.

The features are sorted by their Tf-Idf values, in a descending fashion. Previous work does not provide consistent guidelines on the selection of the number of Tf-idf features. We therefore select configurations of 5, 10, 25, 50, 100, 250 and 500 features at a time in order to experimentally determine the best scale of features for sentiment prediction.

In previous feature-selection methods, we have used features consisting of a single word each. One of the main disadvantages of the use of single words as features is that such features are devoid of semantic meaning. A classifier using such a feature-word is conditioned on that word only. Any morphological deviation from that word, such a misspelling or a different case, constitutes a totally different feature for the classifier, and as such does not trigger the corresponding classification. For example, a classifier conditioning a Positive classification on the word “cookie” will be clueless to the related word “cookies”, or the presence of highly related terms such as “crumble”, “chocolate”.

We propose a method of augmentation of the features selected by Tf-Idf with related words. This is how we can better relate the concepts behind the words selected by Tf-Idf to the classifier, both during training and classification. This approach is very similar to query expansion, as is known in information retrieval [45]. In contrast to methods that serve similar goals, such as Latent Dirichlet Allocation (LDA) which models texts as mixtures of topics, feature augmentation is both simple and can be used in a fully unsupervised fashion. To obtain word relations, we apply basic processing to every news article text in the training set, and extract a set number of Tf-Idf selected features from the news article.

Based on the work by Momtazi et al [35], we construct a co-occurrence mining system to compible an corpus of augmentation words. For every selected feature in the every text we count its co-occurrences with the other features for that text. This yields a co-occurrence matrix of words. Table 22 shows an example of the 10 most co-occuring words for 10 randomly selected features from news articles in the training set at 100 Tf-Idf features. These numbers are chosen arbitrarily as an example; the number of Tf-Idf features extracted from the news article is paramount to the content of the co-occurrence matrix. Manual inspection of the word co-occurrences for 500 most frequent features reveals that, on average, the 5 most frequently co-occurring words with any feature are, by the judgment of the author, either a different morphological representation of the same feature, or semantically highly connected to that feature within the context of the training set. We test the validity of this assumption by experimenting with different numbers of augmentation words: 1,2,3,5,10,25,50.

10 feature-words	10 most co-ocurring
Stunned	Magazine,Works,School,Reached,Courtesy,Pages,Election,Presentations,Hit,Guess
Flotilla	Gaza,Israel,Israeli,Turkish,Turkey,Bound,Mavi,Raid,Blockade,Activists
Egypt	Mubarak,Arab,Egyptian,Israel,Democracy,Regime,Tunisian,Cairo,Peace,Stability
Marketplace	Sales,Online,Stores,Taxes,Inc,Industry,Legislation,Retailers,Customers,Mart
Pelosi	Nancy,Democrats,Gop,Committee,Rep,Reid,Speaker,Election,Cuts,Senate
Embarrassments	Rational,Goof,Potentates,Bedfellow,Barf,Resolutely,Drooling,Lavishly,Tiniest,Supreme
Prohibits	Legislation,Legal,Prohibit,Deny,Constitutional,Rep,Protection,Arizona,November, Amendment
Label	Product,Products,Comments,Consumers,Labels,Truth,Consumer,Word,Charge, Educate
Facial	Crowd,Individual,Recognition,Identify,Special,Discovery,Camera,Performance,Star, Adds
Outlaws	Offenders,Leg,Passing,Ar,Commitment,Peaceful,Squad,Potentially,Senators, Transform

Table 22: Ten most frequent co-occurrences for ten randomly selected features.

As is evident from table 22, in some cases more than 5 co-occurring words are relevant for the selected feature, and in other cases less than 5. We propose a automatic selection method allowing our system to select variable numbers of augmentation words for features, depending on the feature. Our intuition is that, in a list of co-occurrences of any given feature-word, which is sorted in a descending order based on the co-occurrence frequency, the cutoff point for the selection should be the location of the largest difference in frequency between two consecutive co-occurrence words, relative to maximal co-occurrence frequency of that list. This can be expressed as:

$$i_c = argmax_i(\frac{v_{i-1}-v_i}{v_{\max}}) \tag{24}$$

where $i_c$is the cutoff index for the sorted list of co-occurrence frequencies in descending order, v_{i-1} the frequency of the co-occurrence at the index position $i-1$ in that list, v_i the frequency of the co-occurrence at the position i and $v_{\max}$ the maximal frequency count in that list. Figure 6 demonstrates this principle graphically for the feature “pelosi”.

In Figure 6, we observe that the largest relative drop in frequency occurs between the “democrats” and “gop”, or co-occurrences at index numbers 1 and 2. The cutoff is therefore set between these index positions. The selected augmentation words for the feature “pelosi” are “nancy” and “democrats”.

We compare the performance of Tf-Idf based feature selection to the feature augmentation approach, with different parameters.

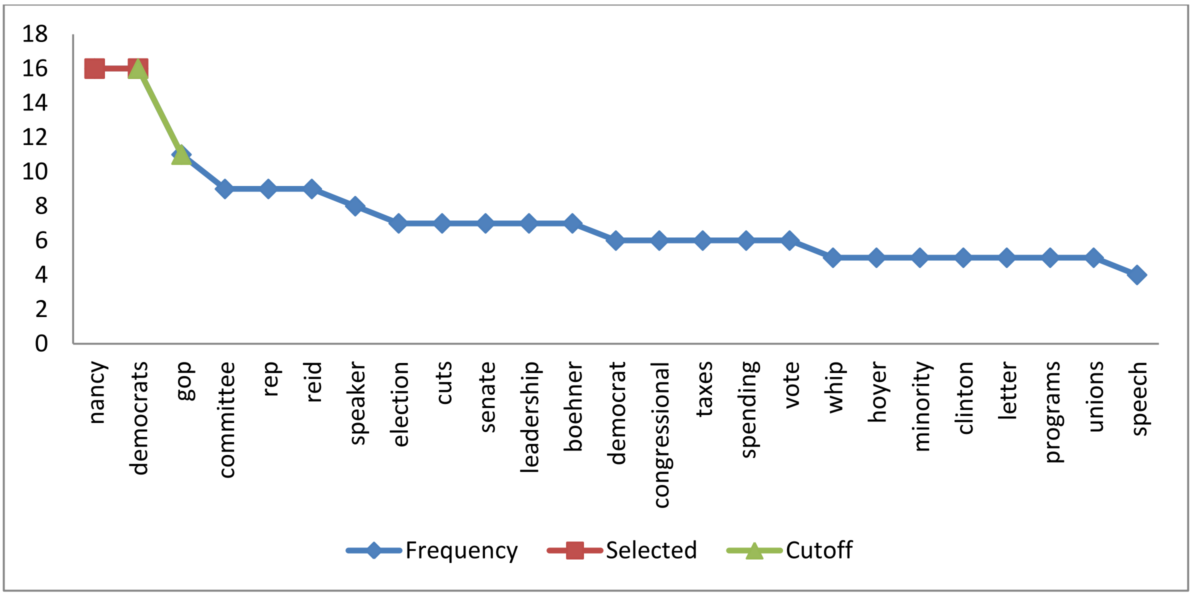

Figure 6: Graphical representation of automatic co-occurrence cutoff determination for the feature "pelosi".

6.3 Prediction classifiers

In this thesis we employ two classification methods for the prediction of general sentiment polarity in the reactions to a news article on Reddit before it's publishing. In previous work [5] [6] [13] [12] [7], classification methods such as Naïve Bayes (NB), Support Vector Machines (SVM), Nearest Neighbor (NN) and Supervised Latent Dirichlet Allocation (sLDA) have all been used for sentiment prediction. Reported results indicate that, depending on the task, more recent classification methods such as SVM and sLDA tend to improve classification accuracy, albeit marginally. Our goal is to research the feasibility of the basic idea behind sentiment prediction for news on Reddit. For this, it is not paramount to attain the highest possible classification performance during prediction. In addition, in those cases where multiple feature selection methods have been used prior to classification [5] [13] [7], feature selection played a much greater role in the accuracy of the classification than the used classification method. sLDA and SVM are more complex in the implementation and require long training times, when compared to NN and NB. For these reasons, we utilize two simple machine learning approaches to classification: Naive Bayes, and a combination of Naive Bayes and a Nearest Neighbor classifier.

Our first method consists of training the Naive Bayes classifier, described in the paragraph 5.1 of this thesis, for sentiment prediction. We provide it with the extracted features from news article texts and the automatically classified polarity labels for the comments to the article, using the postings from the training set.

Our second approach consists of a combination of a Nearest Neighbor classifier and a Naive Bayes classifier. Naive Bayes classifiers have proven to provide robust classification accuracies for many tasks. However, Naive Bayes classification operates under an independence assumption: it assumes that the provided features are independent of one another. This means that any context that might have been represented by the combined presence of the features is lost. Work by Frank et al [37] indicates that relaxing the independence assumption yields improvements in classifier performance and proposes a locally weighed variant of Naive Bayes as an improvement. Work by Jiang et al [38] applies Frank et al's work [37] to a document ranking task, expands upon it by constructing a system that combines Nearest Neighbor and Naive Bayes methods, and demonstrates superior performance of their approach. We propose a variant of Jiang et al's system which does not require the setting of a manual parameter *k* for its functioning.

A Nearest Neighbor (NN) classifier operates under the assumption that the data which is to be classified is similar to some earlier observed training example. To compare the classification data and the training data, NN uses some metric to calculate the distance between all known training examples and the classification data. Using the classification labels of a number of most similar training examples, it provides a classification majority vote. The metric generally used in related work, e.g. in [5], to determine the similarity between documents is cosine angle distance:

$$\cos(d,t) = \frac{\sum_{i=1}^{n} d_i * t_i}{\sqrt{\sum_{i=1}^{n}(d_i)^2} * \sqrt{\sum_{i=1}^{n}(t_i)^2}} \tag{25}$$

where d represents the document up for classification, t represents a training document, n represents the total number of features for both documents, d_i the feature in document d at position i and t_i the feature in document t at position i. The value used in the similarity calculations is the Tf-Idf value of the features.

We follow Jiang et al's [38] blueprint: we use basic Nearest Neighbor methods to retrieve relevant news articles and to measure the distance between the article that is being classified and those training articles. Then we train a Naive Bayes classifier using the retrieved news articles and classify the document with it. By limiting the training dataset for Naive Bayes to only those news articles which have some similarity with the news article being classified, we relax the independence assumption, by ensuring that all training and classification finds place in a similar context. After all, the training data is guaranteed to be somewhat similar to the document that is being classified. In the training of the NB classifier, we retrieve all documents that contain at least a single same feature. We use the cosine angle distance similarity as document weight for the training documents, to prevent the need for manually setting a parameter *k* to determine how many documents should be used in training. Document weighing in the training of a NB classifier can be performed by counting the supplied weight instead of the frequency of the words in that document. This has no adverse consequences for the calculation of the probabilities in the NB classifier, as the Bayesian function normalizes the data. We train our second classification method by remembering all training set examples with their corresponding labels. We call our hybrid approach Contextualized Naive Bayes (CNB).

6.4 Experimental setup

To answer our research questions, we define a number of experiments using the previously described training and test methods and metrics. We perform all training on the automatically labeled training set consisting of 13,053 news articles or subsets thereof, and all validation on the manually annotated corpus containing 389 news articles or subsets thereof.

Every experiment in which we vary over the categories of news, such as "business" or "politics", or specific user groups, such as "Liberal" or "Conservative" is run in such a way that we take subsets of both the training set and the test set which contain only those news articles belonging to the respective category or user group.

We implement all experiments in Python 2.7. We use the classes that perform basic text processing and normalization as described in chapter 6.2 of this thesis. All experiments are ran under Ubuntu Linux 11.01 64-bit operating system, on a machine with a Intel Core Quad Q6600 2,6Ghz processor and 4GB DDR3 RAM. The experiments are fully run in-memory, except where otherwise indicated.

We implement the Naive Bayes implementation as specified in the paragraphs 5.1 and 5.2 of this thesis. For our basic classification approach, we train the Naive Bayes classifier with the features extracted from the news articles from the training set together with their comment sentiment polarity labels. The classification is performed using identical feature selection methods. In classification, we send the features to the classifier, which returns the label for the most probable class the features belong to.

To answer our reseach question, "Does context selection during classifier training improve sentiment prediction accuracy?", we implement the Contextualized Naive Bayes approach proposed in this thesis. For this, we develop a simple information retrieval system and implement it in Python 2.7. The system consists of an in-memory reverse index of news article pointers and a Python "shelf". Features representing the news articles are index keys, while the news articles themselves are "shelved" using Python. During training, all features from news articles are indexed and "shelved". During classification, the information retrieval system is queried with all features of the article that is being classified. Only those documents are retrieved from the "shelf" which, according to the reverse index, contain one or more features from the query. For every document retrieved by the system, the cosine angle distance to the article being classified is calculated. Then, a new Naive Bayes classifier is trained using the features and labels of the retrieved documents with the cosine angle distances as weights. This classifier is then used to predict the sentiment polarity of the comments to the article being classified. We use this classifier in every experiment we perform.

To help answer our research question "Which of the following feature selection methods can we use to obtain the greatest prediction accuracy when modeling a news article: bag-of words, Tf-Idf or Augmented Tf-Idf?", we implement our first two experiments.

In our first experiment, we attempt to discover the appropriate number of Tf-Idf features to select from news article texts. We implement a Tf-Idf selection algorithm using the guideline set forth in (23). We train the Idf component of the implemented algorithm by processing the text of all news articles in the training set and counting the frequencies of all words over documents. A separate document frequency count is stored for every news category and user group. This prevents the influencing by the difference in training corpora for the respective categories of the Tf-Idf selection. We set up a automated experiment that first trains the NB and CNB classifiers, then verifies their performance for varying numbers of features extracted from the news articles using the Tf-Idf method. The numbers of features to be selected are 5, 10, 25, 50, 100, 250 and 500. These numbers have been chosen to represent different size scales. While we evaluate the outcome of the experiment against the gold standard, we somewhat fit our classifier parameters to our gold standard somewhat. However, we do not seek to fully optimize the selection parameters for the performance of the classifier possible. Rather, we try to experimentally establish the general scale for the appropriate number of features.

For the second experiment, we implement a system which constructs an augmentation corpus. For every news article in the provided training set, we apply the basic text processing and extract a set number of features. We count the number of co-occurrences of every word from the feature vectors of news articles with every other word in those vectors. This process is mainly conducted in-memory. A different augmentation corpus is calculated for every different number of Tf-Idf features, news category or user group. Using the experimentally established best performing number of features in the first experiment, we construct our second experiment to establish the best number of augmentation words to attach to the selected features. We set up a automated experiment that trains the NB and CNB classifiers, then verifies their performance for varying numbers of augmentation words. From the constructed augmentation corpora, we select different numbers of augmentation words: 1,2,3,5,10,25,50. In addition to the manually selected sizes, we implement the automatic cutoff-algorithm described in (24) for a feature-specific variable selection of the number of augmentation words. We do not seek to fully optimize the selection parameters for the performance of the classifier possible. Rather, we try to experimentally establish the scale for the appropriate number of augmentation words. While we evaluate the outcome against the gold standard in this experiment, the main point of this experiment is to evaluate the performance of the automatic cutoff-algorithm against the experimentally established optimal number of augmentation words.

With our third experiment, we answer the questions "Which of the following feature selection methods can we use to obtain the greatest prediction accuracy when modeling a news article: bag-of words, Tf-Idf or Augmented Tf-Idf?" and Does the performance of our approach to sentiment prediction compare favorably to similar methods presented in related work?".

In the third experiment we compare the performance of different feature selection methods and classifiers on sentiment prediction. We compare the performance of our approaches to the baseline scores. For this experiment we additionally train the classifiers using the representation of the news articles from the training sets as bags-of-words retrieved from the articles after basic text processing.

To answer our research question "Is the accuracy of sentiment prediction dependent on the category of news? If so, how can the dependency be explained?" we conduct our fourth experiment. We compare the sentiment prediction performance over different news categories using the best performing classifier from the third experiment. We create separate augmented corpora and train the best performing classifier from the third experiment for every news category separately. Table 17 specifies the category-specific training subsets. The performance of the separate category-based classifiers is evaluated against the corresponding subsets in the gold standard, specified in table 12.

Finally, to answer our research question " Does grouping users into homogenous groups, such as Liberals and Conservatives, and conditioning the prediction on those categories increase the prediction accuracy?", we compare the prediction performance between the general Reddit population and the "Liberal" and "Conservative" user groups. We create separate augmented corpora and train the best performing classifier from the third experiment for the user groups {Liberal, Conservative. The performance of the separate user group based classifiers is evaluated against corresponding subsets in the gold standard.

6.5 Results and discussion

For evaluation of our experiments, we use the Accuracy and Classifier Performance measures as explained in the paragraph 5.3 of this thesis. We believe that Classifier

Performance is the superior measure, since it harmonically balances the score of the classifier performance and punishes classifier bias. The relationship between the accuracy and the classifier performance scores gives an indication of that bias: the higher the difference between the two, the higher the bias.

In our first experiment, we calibrate the Tf-Idf feature extraction method. We seek the number of features that yields the best sentiment prediction performance on NB and CNB classifiers. Figure 7 displays the results of this experiment.

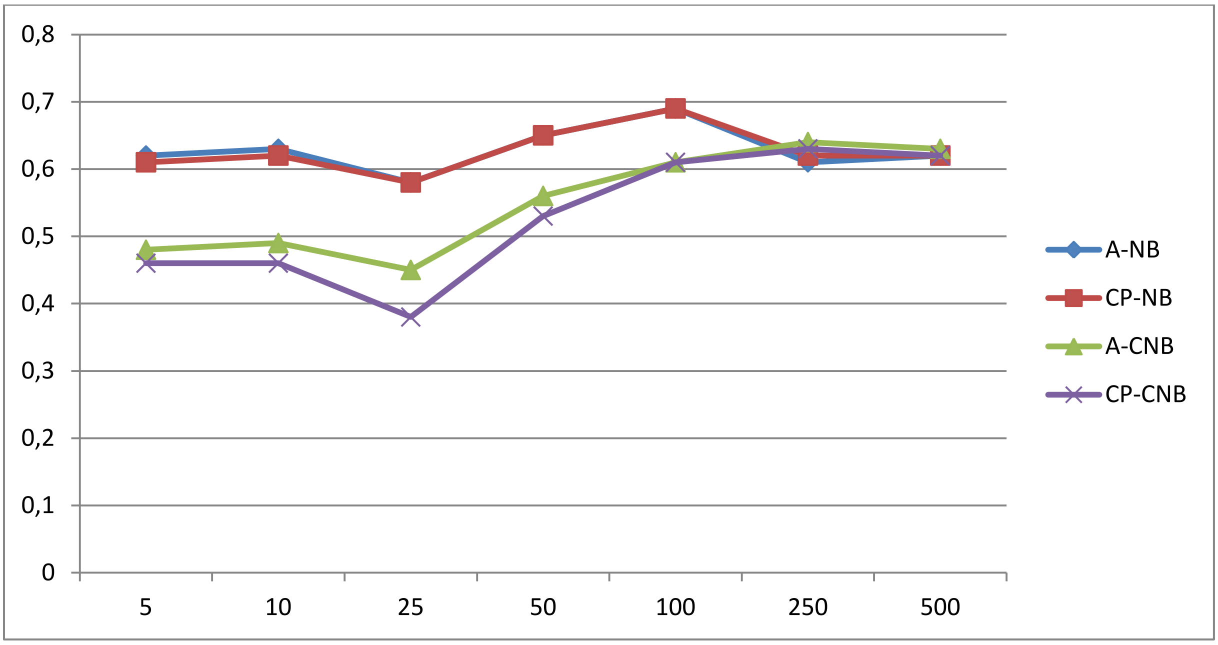

Figure 7: The results of Tf-Idf calibration experiment for determining a optimal number of features. A-NB: Accuracy of the Naive Bayes classifier. CP-NB = Classifier Performance of the Naive Bayes classifier, A-CNB = Accuracy of the Contextualized Naive Bayes Classifier, CP-CNB = Classifier Performance of the Contextualized Naive Bayes Classifier.

We evaluate the consequence of different numbers of features on classification performance against the gold standard, meaning that we are partially training our classifiers against the evaluation set. In this, we partially train our systems on the validation set, which is generally to be avoided. However, in this experiment we do not seek to obtain an optimal parameter selection, but merely an appropriate scale of the number of features for selection. Considering the careful construction of the gold standard described in chapter 4 of this thesis, intended to accurately reflect this Reddit corpus in general, we believe these results to be representative for the training set as well.

We observe a dip in performance going from 10 features to 25 features for both classifiers, after which the performance bounces back at 50 features. This can be explained by the presence of noise (e.g. the usual "stopwords") in the selected features; at 5 or 10 features, the classifier is trained with features that contain very little noise. At 25 features, the relative amount of noise in the feature selection drives down the performance, yet after that the ratio of noise v.s. useful features is again lowered. Same argument counts as explanation of the performance dip of NB after 100 features, or CNB after 250 features.

The best performance is attained by the Naive Bayes classifier at 100 Tf-Idf features, attaining an accuracy of 0.69 and classifier performance of 0.69. Relative to lower and higher feature counts, as well as to the CNB score, this result is significantly[9] higher. This indicates that the behavior of the classifier changes depending on the number of selected features

We observe that the evolution of the performances of the two classifiers is different. CNB classifier begins with a much lower accuracy than NB, but eventually outperforms NB at high feature counts. This observed effect is likely to be the result of overfitting in CNB. In the case of the NB classifier, all training data is applied to a single classifier. CNB, however, is always trained with some subset of all data, the number of features being also the query length for retrieval of the training documents. With low query lengths, low numbers of documents are likely to be retrieved, leading to overfitting of the on-the-go constructed Naive Bayes classifier.

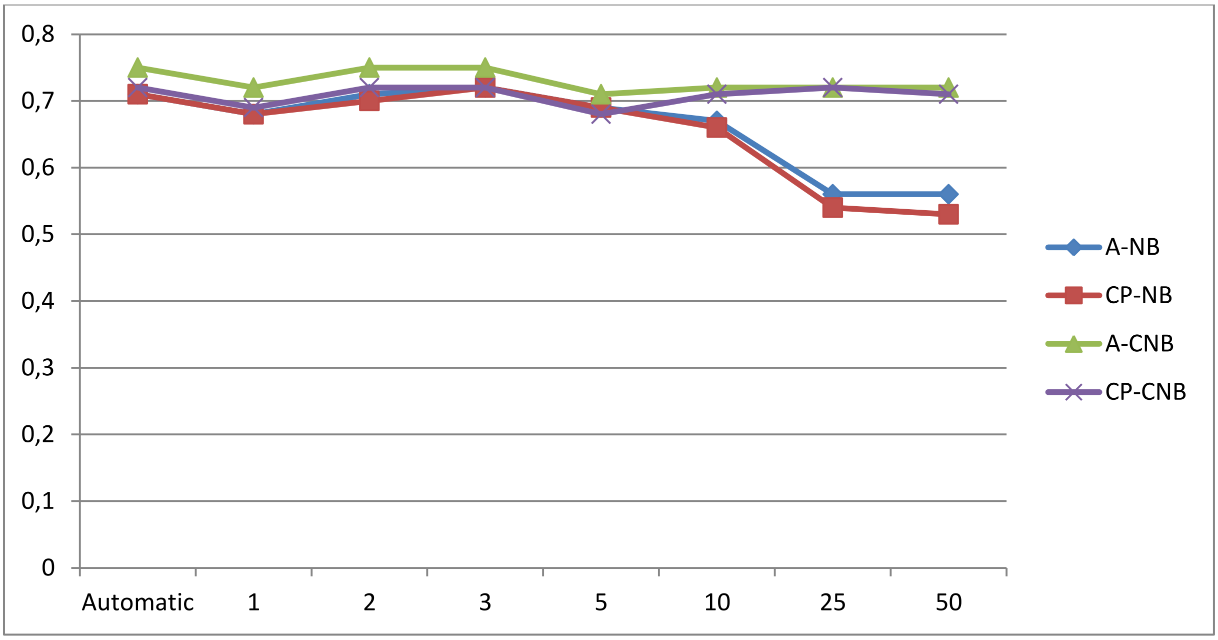

Figure 8: The results of feature augmentation experiment for determining a optimal number of augmentation words. A-NB: Accuracy of the Naive Bayes classifier. CP-NB = Classifier Performance of the Naive Bayes classifier, A-CNB = Accuracy of the Contextualized Naive Bayes Classifier, CP-CNB = Classifier Performance of the Contextualized Naive Bayes Classifier. The "Automatic" label displays the result of automatic variable augmentation number word selection.

This explains why with increased query lengths the relative performance of CNB improves. Another sign of overfitting is the large difference observed between the accuracy and the classifier performance, indicating classifier bias that is often the result of overfitting. Based on the observed results we expect CNB to outperform NB when the training set contains

[9] As verified using Student's T-Test, p<0.05

more examples of every topic, and at higher numbers of feature words. Overall, the CNB performance in this feature configuration is lower than the standard NB.

Despite the slightly better CNB performance at 250 features than at 100 features, we select 100 as the overall number of Tf-Idf features to use for both classifiers. In the following experiment, we augment the features with additional words, expanding the queries used in CNB to at least the double size. We expect this expect to negate the CNB overfitting problems we observe at 100 features.

In the second experiment, we calibrate the number of augmentation words to be appended to every feature. We determine the approximate scale that yields the best sentiment prediction performance using NB and CNB classifiers based on 100 Tf-Idf features. Figure 8 displays the results of the second experiment.

The best performing number of augmentation words with the NB classifier is 3, reaching an accuracy and classifier performance of 0.72. This result is significantly[10] higher compared to the non-augmented Tf-Idf classifier at the same number of features. Classification by CNB yields the best accuracy of 0.75 with classifier performance 0.73 on CNB with 3 augmentation words. This result is significantly[10] higher compared to the non-augmented Tf-Idf classifier at the same number of features. The performance improvement relative to the NB classifier from this experiment at 3 augmentation words is insignificant.

Our earlier expectation, based on manual inspection of the augmentation data, that 5 augmentation words seem to be a good number, has been proven wrong by this experiment. We learn that functional properties of feature extraction parameters are best determined experimentally, if they can't be optimized theoretically.

While we evaluate the outcome against the gold standard in this experiment, which in itself would mean the fitting of selection of parameters on evaluation data, the main point of this experiment is to evaluate the performance of the automatic cutoff-algorithm against the experimentally established optimal number of augmentation words. We observe that the automatic selection of a variable number of augmentation words yields an average of 2.74 words. This number and the classifier performance of the automatic word selection system are both very similar to best performing manual configuration. We conclude that the automatic selection works as intended and that the intuition behind the approach is valid. In a practical situation, it would be better to use the automatic selection method than to set a manual threshold, considering that how exact character of new data is unknown. For this reason, in our further experiments, we use the automatic selection of a variable number of augmentation words.

In this experiment, the CNB performance consistently exceeds the NB performance. Due to the now increased number of feature words (features + augmentation words), the query

[10] As verified using Student's T-Test, $p<0.05$

reach within CNB is increased. Consequently, overfitting seems to be less of a problem. Nevertheless, it is not entirely eliminated as CNB displays a slight negative bias which is also reflected in the difference between its accuracy and classifier performance. We speculate that bias would decrease as the amount of training data increases. This, however, is a speculation that need to be researched in future work.

While the performance of CNB remains relatively consistent at all numbers of augmented words, the NB performance deteriorates from 3 augmented words on. The high numbers of augmentation words tends to dilute the context that every feature represents. For instance, it is very common that an augmentation word such as "guess" is somewhere in the first 50 co-occurrences of any feature. When the NB classifier is trained with such augmented words, they lose their expressiveness and become noise, just like stopwords. Due to the fact that CNB is always trained on (small) subsets of the whole training set, there isn't often enough training data present for this to become a problem. This explains why CNB takes a much lower performance hit on higher numbers of augmented features.

Overall, the use of augmentation words significantly improves the performance of sentiment prediction relative to the use of Tf-Idf feature selection alone.

In our third experiment, we compare the performance of different classifiers and feature-selection configurations to the baseline system from [7] as was implemented and evaluated in the paragraph 6.2 of the thesis. Table 23 displays the results of these comparisons.

	Baseline	NB-BOW	CNB-BOW	NB-TD	CNB-TD	NB-TD-A	CNB-TD-A
Accuracy:	*0.56*	0.67**	0.67*	0.69***	0.61**	0.71***	**0.75*** **
Positive Precision	*0.47*	0.61	0.65	0.61	0.53	0.61	**0.76**
Positive Recall	*0.33*	0.61	0.48	0.77	0.77	**0.84**	0.59
Negative Precision	*0.6*	0.71	0.68	0.79	0.75	**0.84**	0.75
Negative Recall	*0.73*	0.71	0.81	0.64	0.49	0.62	**0.86**
Positive F1-Score:	*0.39*	0.61	0.55	0.68	0.63	**0.71**	0.66
Negative F1-Score:	*0.66*	0.71	0.74	0.71	0.59	0.71	**0.8**
Classifier Performance:	*0.49*	0.66**	0.63*	0.69***	0.61**	0.71***	**0.72*** **

Table 23: Evaluation results of sentiment prediction using a variety of classifiers. Baseline = combined domain knowledge transfer and sentiment prediction method from [7] applied to Reddit dataset, NB-BOW = Naive Bayes classifiers with news articles represented as a bag-of-words, CNB-BOW = Contextualized Naive Bayes classifiers with news articles represented as a bag-of-words, NB-TD = Naive Bayes classifiers with 100 features selected by Tf-Idf, CNB-TD = Contextualized Naive Bayes classifiers with 100 features selected by Tf-Idf, NB-TD-A = Naive Bayes classifiers with 100 features selected by Tf-Idf augmented with a automatically selected number of co-occurring words, CNB-TD-A = Contextualized Naive Bayes classifiers with 100 features selected by Tf-Idf augmented with a automatically selected number of co-occurring words. Level of significance[11] is indicated by the star rating: * if p<0.05, ** if p<0.01, * if p<0.001 .**

We observe that all classifiers used in this thesis, regardless of the utilized feature selection method, significantly outperform the baseline system in sentiment prediction on Reddit

[11] As verified using Student's T-Test

data. The reasons for the poor performance of the baseline system are explained in the paragraph 6.2 of this thesis. Similar to related work, we observe that feature selection plays the greatest role in the accuracy of the classifier. While the standard NB classifier displays the most balanced performance, the CNB method proposed in this thesis displays the highest accuracy and classifier performance. Having analyzed the behavior of CNB over different feature selection configurations, we are confident that the unbalanced performance of CNB is due to overfitting. If more training data would be available, we believe that CNB would have a balanced performance, while standard NB performance would decrease due to ambiguities introduced by the use of individual words in many more different contexts.

Final results of the best-performing classifier are, in our opinion, reasonable, albeit worse than the results reported in related work on similar tasks [12] [7]. However, in the related news articles, used datasets are collected from domains which contain one or two very (politically) distinct user categories, which makes the prediction task easier. Additionally, the authors state fitting the classifiers to their data, as well as removing "troublesome" data, which explains a part of the higher reported performance. The use of other, marginally better classifiers, also explains a part of the reported higher accuracies. Finally, the authors do not publish confusion tables or an analysis of the bias of their classifiers. This leaves open the possibility that their high accuracies are attained by a combination of unbalanced data and biased classifiers.

Another reason for the observed results is error propagation originating in the automatic classification of comments for training (chapter 5). When we compare the results and the bias in our sentiment prediction system against the bias in the original sentiment analysis system from chapter 5 (table 14), we observe many similarities. We speculate that the error in the original sentiment analysis system sets an upper bound to the attainable accuracy of general sentiment prediction. Between the automatic sentiment analysis of the comments and sentiment prediction, we group the polarities multiple individual comments within a Reddit posting into a single label based on the prevalent polarity. This process is highly dependent on the number of comments within a single Reddit posting. It is therefore not possible to exactly indicate what the upper bound is based on the results of automatic sentiment analysis. However, we observed earlier that many Reddit postings contain only one comment (figure 2). Based on this, we would expect the upper bound for general sentiment prediction on this corpus to be near the levels observed in sentiment analysis of the comments, which we indeed observe in table 23.

In all, we conclude that sentiment prediction on this Reddit corpus using bootstrapping from Twitter is possible for a general case, with reasonable results. The results are mainly limited by the accuracy of automatic sentiment analysis used to annotate the comments in this corpus for training of the prediction system.

	Busines	Entert.	Science	Econ.	Enviro.	WNews	Politics	Techn.
Accuracy:	0.7	0.7	0.77	0.67	0.8	0.73	0.76	**0.79**
Positive Precision	1	1	**0.84**	0	0.83	0.67	0.7	0.83
Positive Recall	0.57	0.63	0.76	0	0.63	0.63	0.61	**0.8**
Negative Precision	0.5	0.4	0.69	0.67	**0.79**	0.76	**0.79**	0.74
Negative Recall	1	1	0.79	1	**0.92**	0.79	0.85	0.78
Positive F1-Score:	0.73	0.77	0.8	0	0.72	0.65	0.65	**0.81**
Negative F1-Score:	0.67	0.57	0.74	0.8	**0.85**	0.77	0.82	0.76
Classifier Performance:	0.7	0.66	0.77	0	**0.78**	0.7	0.73	**0.78**

Table 24: Evaluation results of the sentiment prediction per news category. Busines = Business, Entert = Entertainment, Econ = Economy, Enviro. = Environment, WNews = World News, Techn. = Technology

Our fourth experiment explores whether sentiment prediction performance improves when news categories are clearly and separately defined in training and prediction. This experiment is conducted to evaluate the effect of news categories in prediction performance. For this experiment we use the CNB-TD-A classifier from the previous experiment. Table 24 displays the results of this experiment.

We observe that for some categories, the specific training per category does provide an advantage in classifier performance and accuracy compared to the general case. For others it notably does not. We discuss the observed classification results per news category, in order to determine the effect of the division of data into news categories.

The category "business" has a lower prediction accuracy and classifier performance than the general case. This is probably due to the low number of samples available for this category, so that misclassifications at this point have a large impact on the overall statistic. The classifier shows positive bias, which is expected due to both the low number of training examples and the fact that most training samples are positive.

The category "entertainment" also shows a worse performance compared to the general case classifier. We believe that the main reason for low classifier performance in this category is the fact that opinions about entertainment and entertainers are amongst the most personal ones. For example, fandom of a particular music band is probably much more personal to people than their attitude towards global warming or the politics. We also observe a large positive bias for this category, which we explain by both the low number of training examples and the fact that most training samples by far are positive.

The "science" category shows an improved performance compared to the general case. During data analysis, we observed that many discussions in this category are clearly polarized along political lines, which ostensibly makes sentiment prediction easier. Other materials in this category tend to trigger predictable responses as well, for instance the universally positive reactions to the discovery of new cancer treatments. We observe a positive bias for this category, which we explain by the fact that most training samples are positive, and that the used classifier tends to carry the training bias with relatively low numbers of training examples.

The category "economy" displays a disastrously low performance. This category, however, only has 3 members in the gold standard. Therefore, any classification error causes drastic changes in the performance statistic. In addition, the training examples for this category lean heavily towards the negative, explaining the classifier bias in this direction. We disregard this finding due to too little data for meaningful comparison.

Another category with improved performance compared to the general case is "environment". Not unlike "science", "environment" contains a large amount of politically motivated discussions vis-a-vis "global warming". The polarization in these discussions contributes to the predictability of the sentiment for this category. The category has the highest negative bias of all categories as a consequence of a large number of negative training examples and classifier characteristics.

"World news" is a category that in itself contains news which could be assigned to other categories as well, since world news in general contains news about politics, economics, science etc. The lower prediction performance in this category is therefore expected. World news training data contains a majority of negative examples, which is reflected in classifier bias for that category.

The category with the greatest number of training and gold standard samples is "politics". This category performs slightly better than the general case, while having very similar negative bias. This similarity is explained by the fact that the category "politics" constitutes more than half of all training samples. While political discussions are often polarized the discussions in this category are about a variety of subjects. The classifier is sometimes overfitting due to inability to retrieve training examples for the particular subject being discussed. In addition, manual inspection of data in different news categories shows that Reddit users are most prone to using sarcasm in the category "politics", which lowers the prediction performance somewhat.

Finally, the category "technology" does show a better performance compared to the general case. In this category, there is news about new (electronic) product releases and similar consumer related information which can usually count on a positive reaction from the users. This is reflected in the positive classifier bias.

In general we observe that, depending on the type of the content of a category and the number of training and test examples, segmentation into news categories of sentiment prediction training and classification can lead to improved results, but does not do so in all cases. The reasons for better or worse performance are category-specific and require a individual analysis of category data for full understanding. Inconsistent prediction performance over news categories, however, is an important observation. Even when we ignore data-related problem such as the low number of samples in "economy", we still observe some categories perform better and some worse compared to general performance. We therefore conclude that having clearly defined news categories is not a determining factor in sentiment prediction, as it does not show a consistent effect on prediction performance. Incidentally, the manual observation of the data made in chapter 3 of this thesis, that many comments in news categories other than "politics" are politically motivated and polarized, coincides with higher accuracies in said categories. This suggests that a clear division of users along political lines plays a key role in attaining a good sentiment prediction performance, rather than a division of news by categories.

Manual observation (chapter 3) of the data in our Reddit corpus has revealed that in many news categories discussions are politically motivated, even if the news itself is not explicitly political in nature. For instance, news about "global warming", in the news category "environment", often attracts comments of political liberals, who urge for the use of alternative fuels, and conservatives who explicitly voice their discontent with what they term "the liberal agenda for killing American jobs". We observe such distinct reacting behavior in "technology" news as well, when green technology is commented upon, but for instance also in articles about the "Large Hadron Collider" - conservatives talk about "typical liberal waste of money on quack science" while liberals in turn make generalizing statement over a alleged lack of intelligence with conservatives. This type of polarizing reaction behavior of different user categories is observed to be consistent over news categories. We therefore expect the division of users into groups along the observed political lines to lead to improved prediction accuracy. In the fifth and final experiment, we determine the role that the division of users along political lines into Liberals and Conservatives has on the classifier performance. For this experiment we use the CNB-TD-A classifier from the third experiment. In the gold standard, we filter the postings that contain only a single user group.

	Liberals	Conservatives
Accuracy:	**0.82**	0.78
Positive Precision	**0.74**	0.66
Positive Recall	**0.85**	0.6
Negative Precision	**0.89**	0.83
Negative Recall	0.79	**0.86**
Positive F1-Score:	**0.79**	0.63
Negative F1-Score:	**0.84**	**0.84**
Classifier Performance:	**0.81**	0.72

Table 25: Evaluation results of sentiment prediction applied to separate users groups.

For Liberals, 298 postings remain in the gold standard and for Conservatives 144. In every posting we remove all comments not belonging to the current user group. The results of this experiment are displayed in table 25.

We observe that the division of users into specific political categories universally improves the accuracy of the classification, and for Liberals it also improves classifier performance.

This validates our intuition that greater sentiment prediction accuracies can be attained in predicting the reactions of more homogenous groups of users, as during training their responses to issues are more consistent.

The difference in prediction performance between Liberals and Conservatives is also expected due to a number of observations we made earlier in this thesis. Firstly, there are more Liberals than Conservatives in our training corpus, meaning that the training set (and the gold standard) for Liberals is larger than for Conservatives, as observed in the paragraph 3.4 of this thesis. When combined with the characteristics of the used classifier, this accounts for less overfitting for Liberals as is also evident from results. Secondly, the consistency between the average opinions of Liberals over news categories between the training set and the gold standard is greater than that of the Conservatives, as observed in the paragraph 6.1 of this thesis. Thirdly, in paragraph 3.4 of this thesis we also observe that the accuracy in automatic classification of the user group for every user is more accurate for Liberals than for Conservatives. This explains both the observation from the paragraph 6.1, and consequently, the observation in table 25.

We conclude that the division of users into specific groups along political lines improves the performance of sentiment prediction as expected. Compared to the effect of the division of news into categories and the general prediction results, we conclude that appropriate modeling of user groups seems to be the single most important factor for successful sentiment prediction.

7. Conclusion

In this thesis, we have researched the feasibility of prediction of general sentiment polarity in the reactions to a news article on Reddit before it's publishing. We have explored the role of the types of users on the sentiment prediction performance, as well as the role of the news categories. In addition, we have provided an extensive analysis of the data collected from Reddit, and manually annotated a gold standard for performance testing. We have also used bootstrapping techniques for sentiment analysis and the segmentation of users into groups. From a technical perspective, we have explored several ways of news article text modeling and several methods of text classification for sentiment prediction. Finally, we have compared our findings to previous work.

We answer our research question "Can we use bootstrapping from Twitter data to classify comments on Reddit as an step in automation of the training of our sentiment prediction system?" positively. To perform the prediction of general sentiment polarity in the reactions to news on Reddit before a news article is published (in short, "sentiment prediction"), we have used a dataset of news articles and comments collected from Reddit. To train our sentiment prediction systems, because of the size of the Reddit corpus, we required automatic annotation of the sentiment polarity of Reddit comments. To achieve this, we used a bootstrapping approach, using a Twitter corpus and domain-knowledge-transfer methods to adapt the sentiment polarity knowledge from tweets to Reddit comments. After testing several approaches of domain-knowledge transfer, we conclude that Frequently Co-Occurring Entropy and Estimation Maximization method [3] offers the best performance and a sentiment analysis classifier usable for this task.

We answer our research question "Does the performance of our approach to sentiment prediction compare favorably to similar methods presented in related work?" positively. We conclude that prediction of general sentiment polarity in the reactions to news before an news article is published is feasible within the Reddit corpus used in this thesis and can be performed with a reasonable degree of accuracy (0.75). In this, our approach to sentiment prediction significantly outperforms a state-of-the-art approach to a similar problem [7].

We answer our research question "Does grouping users into homogenous groups, such as Liberals and Conservatives, and conditioning the prediction on those categories increase the prediction accuracy?" positively. The segmentation of Reddit users along political lines into Liberals and Conservatives, and the separate prediction of sentiment polarity in reactions to news for these groups additionally increases the prediction accuracy (to 0.82 for Liberals). This is in line with our intuition that the groups of Reddit users more homogenous in their political orientation have more predictable sentimental responses to news than the general Reddit population.

We answer our research question "Is the accuracy of sentiment prediction dependent on the category of news? If so, how can the dependency be explained?" inconclusively. In the results we observe a dependency on both the news category and the audience to the news articles. The segmentation of news into separate news categories does not universally improve accuracy or classifier performance; while this segmentation is beneficial for prediction accuracy in some categories, it is detrimental in others. The effect of segmentation of news into categories on the prediction accuracy is dependent on the variety of themes discussed in the categories, the type of audience interested in that category, as well as how much the opinions on subjects in the category are conditioned on personal preferences (e.g. whether a user likes a particular type of food) v.s. general cultural attitudes (e.g. the universal condemnation of murder of innocent people).

We answer our research question "Which of the following feature selection methods can we use to obtain the greatest prediction accuracy when modeling a news article: bag-of-words, Tf-Idf or Augmented Tf-Idf?" with "Augmented Tf-Idf". Observing experimental results, we conclude that feature selection has a major impact on the accuracy of sentiment prediction. In this thesis we use common feature selection methods such as bag-of-words and Tf-Idf. We experimentally calibrate the number of features for Tf-Idf selection, and conclude that 100 Tf-Idf features yield the best sentiment prediction accuracy. In addition, we implement an extension to Tf-Idf, which augments the selected Tf-Idf features by adding words co-occurring with a feature when viewed over the entire training set. We also implement an automatic selection algorithm for determination of the appropriate number of such augmentation words. Then, we experimentally demonstrate this algorithm to perform comparably with the best manual selection method. We conclude that the automatic selection system is usable for this task. Also, we experimentally demonstrate and conclude that feature augmentation significantly improves sentiment prediction performance for our task.

Finally, we answer our research question "Does context selection during classifier training improve sentiment prediction accuracy?" with nuance: under certain conditions, depending on the number of training examples and features, the answer is positive. We implement a hybrid approach to sentiment prediction and classification combining Nearest Neighbor and Naive Bayes machine learning methods, which we call Contextualized Naive Bayes (CNB). This approach is based on the idea of limiting the scope of a classifier's knowledge to the context of the news article up for classification, to prevent the dilution of context-specific knowledge with non-related knowledge. We conclude that this approach yields mixed results, and is highly dependent on the amount of available training data and the utilized feature selection method. Regardless, the best performing classifier evaluated for sentiment prediction in this thesis is a CNB classifier, using automatically augmented Tf-Idf features. Understanding some of the issues with the classifier, we conclude that CNB is a good approach towards classification in general when a large amount of training data and sufficient features are available. To fully determine added value of CNB versus NB, more research is needed.

We give a mixed answer to our main research question, "Can we predict the general sentiment of the reactions to news on Reddit before a news article is published?": yes, but only if the Reddit demographic and their preferences doe not significantly change in the future. Overall, we conclude that sentiment prediction is a *very* difficult task. It is highly dependent on the audience whose reactions to news are being predicted. The high amount of variability in the audiences to news articles, whether a particular user decides to react to an news article or not, the personal preferences of these audiences, all influence their reactions to the news. This makes it unlikely that a reliable sentiment prediction system can be constructed which is not conditioned on audience profiles. While we obtain reasonable performance on the general sentiment prediction task in this thesis, we conclude that this is a lucky coincidence between the large amount of politically motivated users and reactions in our corpus, and the large number of news articles which elicit a politically motivated response from the audience. If, instead, that same number of news articles had topics about "entertainment", the general sentiment prediction performance with this audience would have been much lower. This is evident from experimental results on the influence of news categories on sentiment prediction accuracy. The approach described in this thesis is therefore only reliably usable within the constraints of the Reddit corpus we collected. In future work, we propose a improved system for sentiment prediction based on the lessons learned during the work on this thesis.

8. Future work

Using the lessons learned during the work on this thesis, we propose a basic idea for a general sentiment prediction system for news. Such a system must be based on modeling the consistency of user reactions to different topics from the news articles. Users would then be split into groups along the lines of their most consistent reactions to groups of topics. This system must be able to discover clusters of users who's reactions have the same comment polarities towards the same topics, the topics themselves being related. For every topic, the distance to all other topics should be calculated using metrics such as cosine angle distance, after which the topics are clustered. Then, for every user, their "likes" and "dislikes" towards different topic clusters are determined. The users themselves are then clustered into groups based on the distance between their individual "likes" and "dislikes". In this way, homogenous user groups can be created along dimensions meaningful according to the news and reactions data using unsupervised methods. This approach can automatically create interesting groups depending on the data, e.g. the lovers of "iOS" v.s. "Android" mobile operating systems for mobile technology-themed news etc. These groups can then be manually or automatically labeled, if necessary, and used for accurate sentiment prediction along the lines of this thesis.

We believe that systems for prediction of general sentiment polarity in the reactions to news before an news article is published could benefit from inclusion of features other than just the news article text. Prediction of other characteristics based on the news article text, such as of the number of reactions to an article, is possible. We propose a research effort that would benchmark the impact of such additional predictions used as features in the prediction of sentiment.

In this thesis, we proposed a hybrid Nearest Neighbor - Naive Bayes machine learning method as a context-preserving variation of Naive Bayes, and we called it Contextualized Naive Bayes. Based on observations made in this thesis, we propose a thorough and diligent research effort, testing CNB across multiple domains, feature selection methods and data set sizes against NB, in order to determine it's true potential and added benefit.

Additionally, in this thesis we propose a simple method of augmentation of often used Tf-Idf feature selection. Preliminary results show significantly improved prediction performance when this method is applied. We propose a research effort into this method as a comparison with topic-modeling methods such as LDA, to determine its effect on a larger scale and compared with well-established topic-modeling methods.

The output of a reliable sentiment prediction system can be used as a feature to predict of the number of comments to a news article. We propose a research effort that maps the effects this approach might have as compared to the current state-of-the-art [29].

Citations

[1] A. Go, L. Huang and R. Bhayani, "Twitter Sentiment Classification using Distant Supervision," The Stanford Natural Language Processing Group, 2008/2009.

[2] A. Montoyo, P. Martínez-Barco and A. Balahur, "Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments," Elsevier, 2012.

[3] S. Tan, X. Cheng, Y. Wang and H. Xu, "Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis," in *31th European Conference on IR Research on Advances in Information Retrieval*, Toulouse, France, 2009.

[4] B. Pang, L. Lee and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *ACL-02 conference on Empirical methods in natural language processing - Volume 10*, Philadelphia, PA, USA, 2002.

[5] G. P. C. Fung, J. X. Yu and W. Lam, " News Sensitive Stock Trend prediction," in *Advances in Knowledge Discovery and Data Mining : 6th Pacific-Asia Conference*, Taipel, Taiwan, 2002.

[6] G. P. C. Fung, J. X. Yu and W. Lam, "Stock prediction: Integrating text mining approach using real-time news," in *Computational Intelligence for Financial Engineering*, Hong Kong, 2003.

[7] R. Balasubramanyan, W. W. Cohen, D. Pierce and D. P. Redlawsk, "What pushes their buttons? Predicting comment polarity from the content of political blog posts," in *Workshop on Language in Social Media*, Portland, Oregon, USA, 2011.

[8] R. Balasubramanyan, W. Cohen, D. Pierce and D. Redlawsk, "Modeling Polarizing Topics: When Do Different Political Communities Respond Differently to the Same News?," in *6th International AAAI Conference on Weblogs and Social Media*, Dublin, Ireland, 2012.

[9] M. Taboada, J. Brooke, M. Tofiloski, K. D. Voll and M. Stede, "Lexicon-Based Methods for Sentiment Analysis," *Computational Linguistics Vol. 37, Nr. 2,* pp. 267-307, 2011.

[10] T. Nasukawa and J. Yi, "Sentiment analysis: capturing favorability using natural language processing," in *2nd international conference on Knowledge capture*, NY, USA, 2003.

[11] J. Yi, T. Nasukawa, R. Bunescu and W. Niblack, "Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques," in *Third IEEE International Conference on Data Mining*, Melbourne, Florida, USA, 2003.

[12] K. Lerman, A. Gilder, M. Dredze and F. Pereira, "Reading the Markets: Forecasting Public Opinion of Political Candidates by News Analysis," in *22nd International Conference on Computational Linguistics*, Manchester, UK, 2008.

[13] V. Seghal and C. Song, "SOPS: Stock Prediction using Web Sentiment," in *Seventh IEEE*

International Conference on Data Mining Workshops, Washington, DC, USA, 2007.

[14] A. Balahur, R. Steinberger, M. Kabadjov, V. Zavarella, E. Van der Goot, M. Halkia, B. Pouliquen and J. Belyaeva, "Sentiment Analysis in the News," in *7th International Conference on Language Resources and Evaluation*, Valletta, Malta, 2010.

[15] "MediaSentiment system," [Online]. Available: http://www.mediasentiment.com. [Accessed 21 6 2012].

[16] M. Gamon, S. Basu, D. Belenko, D. Fisher, M. Hurst and A. C. König, "BLEWS: Using Blogs to Provide Context for News Articles," in *2nd AAAI Conference on Weblogs and Social Media*, Seattle, Washington, USA, 2008.

[17] S. Park, M. Ko, J. Kim, Y. Liu and J. Song, "The politics of comments: predicting political orientation of news stories with commenters' sentiment patterns," in *ACM 2011 conference on Computer supported cooperative work*, New York, NY, USA, 2011.

[18] "Reddit," [Online]. Available: http://www.reddit.com. [Accessed 22 6 2012].

[19] "Inside the Reddit AMA: The Interview Revolution That Has Everyone Talking," [Online]. Available: http://www.forbes.com/sites/ryanholiday/2012/05/01/inside-the-reddit-ama-the-interview-revolution-that-has-everyone-talking/. [Accessed 11 5 2012].

[20] "Could Reddit be the world's most influential website?," [Online]. Available: http://www.blueglass.com/blog/could-reddit-be-the-worlds-most-influential-website/. [Accessed 26 4 2012].

[21] "Internet Movie Database," [Online]. Available: http://www.imdb.com. [Accessed 26 4 2012].

[22] K. Nigam, A. McCallum, S. Thrun and T. Mitchell, "Learning to classify text from labeled and unlabeled documents," in *15th Conference of the American Association for Artificial Intelligence*, 1998.

[23] W. Dai, G. Xue, Q. Yang and Y. Yu, "Transferring Naive Bayes Classifiers for Text Classification," in *22nd national conference on Artificial intelligence - Volume 1*, Vancouver, British Columbia, Canada, 2007.

[24] F. Sebastiani, "Machine learning in automated text categorization.," *ACM Computing Surveys, Vol. 34,* p. 1–47, 2002.

[25] M. Blei, A. Y. Ng and M. I. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research 3,* p. 993–1022, 2003 .

[26] D. Blei and J. McAuliffe, "Supervised Topic Models," *The Journal of Machine Learning,* p. 121–128, 2008.

[27] "Iowa Electronic Markets," [Online]. Available: http://tippie.uiowa.edu/iem/. [Accessed 25 6 2012].

[28] L. Hong, O. Dan and B. D. Davison, "Predicting Popular Messages in Twitter," in *20th International World Wide Web Conference*, Hyderabad, India, 2011.

[29] M. Tsagkias, W. Weerkamp and M. d. Rijke, "News Comments: Exploring, Modeling, And Online Predicting," in *32nd European Conference on Information Retrieval*, Milton Keynes, UK, 2010.

[30] S.-y. Kim, C. S. Taber and M. Lodge, "A Computational Model of the Citizen as Motivated Reasoner: Modeling the Dynamics of the 2000 Presidential Election," *Political Behavior Vol 32, Nr. 1,* pp. 1-28, 2010.

[31] K. K. Bun and M. Ishizuka, "Topic Extraction from News Archive Using TF*PDF Algorithm," in *Proceedings of the 3rd International Conference on Web Information Systems Engineering*, Washington, DC, USA, 2002.

[32] M. J. Giarlo, "A Comparative Analysis of Keyword Extraction Techniques," Rutgers, The State University of New Jersey, New Jersey, NY, USA, 2005.

[33] W. Yong-qing, L. Pei-yu and Z. Zhen-fang, "A Feature Selection Method based on Improved TFIDF," in *Third International Conference on Pervasive Computing and Applications*, Alexandria, Egypt, 2008.

[34] T. Xia and Y. Chai, "An Improvement to TF-IDF: Term Distribution based Term Weight Algorithm," *Journal Of Software Vol. 6,* pp. 413-420, 2011.

[35] S. Momtazi, S. Khudanpur and D. Klakow, "A Comparative Study of Word Co-occurrence for Term Clustering," in *The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, CA, USA, 2010.

[36] K. Brodersen, C. Ong, K. Stephan and J. Buhmann, "The balanced accuracy and its posterior distribution," in *20th International Conference on Pattern Recognition*, Istanbul, Turkey, 2010.

[37] E. Frank, M. A. Hall and B. Pfahringer, "Locally weighted naive Bayes," University of Waikato, Department of Computer Science, Waikato, New Zealand, 2003.

[38] L. Jiang, H. Zhang and J. Su, "Learning k-Nearest Neighbor Naive Bayes For Ranking *," in *First International Conference on Advanced Data Mining and Applications*, Beijing, China, 2005.

[39] "Reddit vote weighting metric," [Online]. Available: http://www.seomoz.org/blog/reddit-stumbleupon-delicious-and-hacker-news-algorithms-exposed. [Accessed 28 6 2012].

[40] "Reddit visit statistics," [Online]. Available: http://www.alexa.com/siteinfo/reddit.com. [Accessed 28 6 2012].

[41] "Okapi BM25," in *An Introduction to Information Retrieval*, Cambridge University Press, 2009, p. 233.

[42] N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, 2000.

[43] J. Bollen, H. Mao and A. Pepe, "Modeling Public Mood and Emotion: Twitter Sentiment and Socio-Economic Phenomena," in *Fifth International AAAI Conference on Weblogs and Social Media*, Barcelona, Spain, 2011.

[44] C.-C. Chang and C.-J. Lin., " LIBSVM : a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology,* pp. 2:27:1--27:27, 2011.

[45] Y. Qiu and H. Frei, "Concept Based Query Expansion," in *16th ACM International Conference on Research and Development in Information Retrieval*, Pittsburgh, USA, 1993.

[46] "Ranks.nl stopwords list," [Online]. Available: http://www.ranks.nl/resources/stopwords.html. [Accessed 10 8 2012].

Appendix A

Stopword list

This is the list of stop-words used at different points in this thesis, obtained from [46]:

about, above, after, again, against, all, am, an, and, any, are, aren't, as, at, be, because, been, before, being, below, between, both, but, by, can't, cannot, could, couldn't, did, didn't, do, does, doesn't, doing, don't, down, during, each, few, for, from, further, had, hadn't, has, hasn't, have, haven't, having, he, he'd, he'll, he's, her, here, here's, hers, herself, him, himself, his, how, how's, i, i'd, i'll, i'm, i've, if, in, into, is, isn't, it, it's, its, itself, let's, me, more, most, mustn't, my, myself, no, nor, not, of, off, on, once, only, or, other, ought, our, ours , ourselves, out, over, own, same, shan't, she, she'd, she'll, she's, should, shouldn't, so, some, such, than, that, that's, the, their, theirs, them, themselves, then, there, there's, these, they, they'd, they'll, they're, they've, this, those, through, to, too, under, until, up, very, was, wasn't, we, we'd, we'll, we're, we've, were, weren't, what, what's, when, when's, where, where's, which, while, who, who's, whom, why, why's, with, won't, would, wouldn't, you, you'd, you'll, you're, you've, your, yours, yourself, yourselves

Table 26: The list of stop-words used in text processing at different points in this thesis.