NTNU – Trondheim
Norwegian University of
Science and Technology

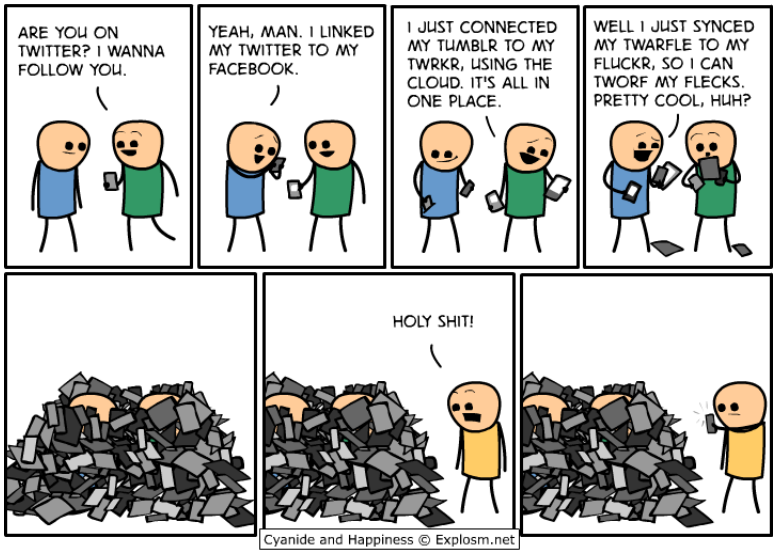
Sentiment Analysis of Norwegian Twitter Messages

John Arne Øye

Master of Science in Informatics
Submission date: Januar 2015
Supervisor: Jon Atle Gulla, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Sentiment Analysis of Norwegian Twitter Messages



John Arne Øye

January 2015

MASTER'S THESIS
Department of Computer and Information Science
Norwegian University of Science and Technology

NTNU
Norwegian University of Science and Technology

Supervisor: Professor Jon Atle Gulla

Preface

This thesis has been submitted in order to meet the requirements for completing the degree of Master of Science in software engineering. The Master programme has been completed at the Department of Computer Science (IDI) at the Norwegian University of Science and Technology(NTNU).

The work in this thesis has been made possible by the guidance of Professor Jon Atle Gulla at IDI.

Sammendrag

Web-baserte sosiale medier (WBSM) har sett økende popularitet popularitet de siste årene, og har som konsekvens fått interesse fra flere grupper som vil dra nytte av de enorme mengder data som finnes på disse nettstedene. Mikro-bloggstedet Twitter er en av disse sidene som har omfavnet utviklere og interessegrupper ved å tillate fri tilgang til sine rammeverk, slik at interessenter kan få tilgang til og søke gjennom disse store mengdene informasjon.

I denne masteroppgaven er det beskrevet et system for å utføre sentiment analyse (SA) på norske Tweets. Systemet bruker en to-trinns binær prosess for subjektivitets- og polaritets klassifisering, som utnytter ulike parametere og tre forskjellige algoritmer - Naive Bayes (NB), Support Vector Machines (SVM), og Maximum Entropy (MaxEnt) - for de to forskjellige klassifiseringsoppgavene. Løsningen gjør også et forsøk på å utnytte grammatisk metadata gitt av NTNU SmartTagger samt tverrspråklig oppslag i SentiWordNet sentimentleksikon for å oppnå bedrede resultater fra de tre klassifikatorene.

Et system for å trekke ut sentimentmål etter klassifiseringen er også beskrevet. Metoden utnytter klassifikatoren for å identifisere sentimentgivende ord for å forsøke å identifisere målet for et gitt sentiment i en tweet.

Abstract

Web-Based Social Media (WBSM) have been on the rise for the recent several years, and have subsequently garnered interest from several groups out to proficiently utilize the vast amounts of data found on these sites. Micro-blogging site Twitter is one of the media sites that have embraced developers and interest groups, Twitter has developed accessible frameworks and allows the use of these frameworks enabling developers access to large amounts of information.

In this master thesis, a system for performing Sentiment Analysis (SA) on Norwegian Tweets is described. The system described uses a two-step binary classification process for subjectivity and polarity classification, utilizing different parameters and three different classifiers - Naive Bayes (NB), Support Vector Machines (SVM), and Maximum Entropy (MaxEnt) - for the two different classification tasks. The solution also makes an attempt at exploiting grammatical metadata given by the NTNU SmartTagger as well as cross-lingual sentiment lookup in the SentiWordNet sentiment lexicon in order to achieve improved results from the classifiers.

A system for extracting sentiment targets after classification is also described. This method is a way of utilizing the classifier in order to identify critical sentiment words in order to augment the detection of the target of a given sentiment in a tweet.

Acknowledgements

First and foremost I would like to thank my supervisor Jon Atle Gulla for his very useful thoughts and responses to the task during our biweekly meetings, as well as for arranging the biweekly SmartMedia workshop discussions allowing me to get feedback from several points of view on my work.

I would also like to thank Arne Dag Fidjestøl for access to and infomartion on the NTNU SmartTagger as well as his part in the workshop discussions.

For valuable thoughts and insights on different aspects and subjects, as well as access to indexed Twitter datasets, I would like to thank Jon Espen Ingvaldsen.

For taking time off her hectic schedule and sitting down and discussing sentiment analysis and cross-lingual dictionary look-up I would like to thank Jeanine Lilleng. Her out-of-the-box thinking and cleverness gave fresh perspectives when it was much needed.

I would like to thank the members of the biweekly SmartMedia workshop meetings for valuable insights, feedback, and discussion points: Özlem Özgöbek, Pål Njølstad, Mahboobeh Harandi, Patrick Romstad, Lars Høysæter, Amir, Henning Wold, and Linn Vikre.

I would also like to thank Sondre Søråsdekkan for writing quiz-questions for me for the quiz at Edgar Cafe, when I realized it was my turn to write and it was one week until the deadline of my thesis delivery. Thus he enabled me to work fully on my thesis the final days before delivery.

For reading through my thesis I would like to thank Nils Ove Tendenes, Jarl Aanes, and Håkon Hauso, and for giving me feedback on spelling errors, sentence structures, and other improvable aspects.

Thank you to Ørjan Helstrøm for reading my thesis and giving me subject-related feedback, and feedback regarding general thesis structure.

Many thanks to the people at explosm[1] for allowing me to use their Cyanide and Happiness comic on my title page.

Last but not least, I want to thank my girlfriend Jill, for her patience, understanding, and care when I was working long hours with this thesis. Her constant receptiveness to my thoughts and ideas and devotion to giving me help and feedback have been invaluable during this last year.

[1] www.explosm.net

List of Figures

List of Tables

Contents

PART I

Introduction

CHAPTER 1

Introduction

> "I tweet, therefore my entire life has shrunk to 140 character chunks of instant event and predigested gnomic wisdom, and swearing."
>
> Neil Gaiman

This chapter will start by giving insight into the motivation behind this thesis, before outlining and describing the task at hand along with the research goal and questions. Lastly the challenges and the structure of this thesis will be described.

SECTION 1.1
Motivation

In the recent years of the evolution of Web-based social networks(WBSN), Twitter[1] has emerged as one of the leading social media sites worldwide, along with Facebook[2] and LinkedIn[3]. Reaching over 500 million users in 2012[1], Twitter has become an enormous platform for information sharing worldwide, and garnered increasing interest both as a social site and as a news medium[2].

The high amount of sentiment data produced by users on Twitter has made this a valuable resource for sentiment analysis. Within the fields of entertainment, news, sports, marketing, politics, socio-economics, and finance, among others, academics have found value in sifting through the vast amounts of data generated on Twitter. Marketeers and entertainers can use sentiment information to predict trends in the market, and thereby change dynamically as

[1] http://www.twitter.com
[2] http://www.facebook.com
[3] http://www.linkedin.com

well as greatly augment their decision-making processes in accordance to the general opinion of their target-groups. Within socio-economics one can use the data to analyse the temporal changes of sentiment and how it corresponds to economic trends. Bollen et al. showed how measured Twitter moods can be used to predict trends on the stock market[3], similar to the work of Njølstad and Høysæter who attempted to predict similar stock market trends using sentiment analysis on business news articles[4]. Psychologists and anthropologist can use sentiment data to view interesting user behaviours of intercommunication in large groups of people, and information can be put in a temporal perspective in order to view interesting social trends and change in opinions.

We can easily see that sentiment analysis systems have practical and commercial uses within several domains across different fields of study. Of course, analysing sentiment data has great practical use within computer science as well. Information Retrieval(IR) methods can for instance utilise sentimental metadata to greatly improve their capabilities, utilising sentiment metadata in order to augment search engines. Within Artificial Intelligence(AI) the ability to extract sentiment from unstructured text and speech can be a vital component in order for intelligent agents to be able to properly interpret unstructured text.

The issue of analysing the large amount of data becomes a very real problem when done manually, it is simply not feasible if one wants to get valuable information without spending millions of man-hours sifting through data. A single tweet message is limited to only 140 characters, but Twitter receives an average number of 500 million tweets every day[1]. With an average message length of 67.9 characters this adds up to over *50 terabytes of new data each day*, not including metadata. There is no doubt that methods for automatic analysis is necessary for such amounts of data. To achieve this automation, machine learning methods are often proposed as a solution. Methods include Naive Bayes(NB), Support Vector Machines(SVM), and Maximum Entropy(MaxEnt), augmented with metadata from Part-of-Speech tagging and sentiment values from pre-created sentiment lexica. Early work on this such as Pang and Lee[5] showed that these methods can perform classification with a percentile accuracy of around 80%. Over the past decade, methods for sentiment analysis have improved, recent performances showing up to 85% accuracy in sentiment analysis systems.

In this thesis I will go about performing SA from mainly an academic perspective, using tools from the fields of AI and IR, but assuming less interest in practical uses and more interest in the results themselves. I will elaborate on how these results were achieved, and discuss ways in which they could be improved. There will also be a longer discussion into the potential practical value and social benefit of the results, later in chapter 10.

SECTION 1.2
Task Description

The main focus for this research is to use machine learning methods and semantic analysis to deduce the sentiment in tweet messages. I will also give an attempt at using linguistic features to augment detection of the target entity of sentiment.

The following describes the task as it was given by the supervisor:

> *Twitter Sentiment Analysis and Sentiment Topic Identification*
> The informal nature of Twitter makes it a viable corpus for sentiment analysis.
>
> In this project the student will develop techniques for analysing the sentiment of Twitter messages and identifying the sentiment target of given sentiment. This involves techniques from information retrieval/search, text mining and semantics. The techniques should be language-independent, though we may use semantic structures/-taxonomies as part of the analysis.

In essence, the main part of this task can be described as mainly a text document classification problem. With regard to this each tweet is defined as a single document, and the task will be to put each tweet in one of three sentiment classes: *Positive*, *Negative*, or *Neutral*. Given the shortness of tweets - limited to 140 characters - the assumption that classification at the document level is adequate has led to keeping this focus throughout the thesis. Seen in a larger scope, the task is to perform systematic analysis of aggregated sentiment, with a topical focus. This means that developing a way of identifying the topic targets of given sentiments is necessary. The former part of the task - twitter sentiment classification - is a well researched area within computer science. The latter - document topic detection in general - is also very well researched. However, the focus on identifying the topic targets of expressed sentiments is an area with room for new and exciting developments.

SECTION 1.3
Research Goal

The following *research questions* arise:

RQ1 How can one proficiently extract sentiment from Norwegian Twitter messages?

- What features should be used for such a task?

RQ2 How can one accurately find the topics towards which the sentiment is directed?

  - Are hashtags proficient in describing the topics of expressed sentiment?
  - Can linguistic metadata be used in order to augment identification of sentiment topics?

RQ3 Can Norwegian-English translation be used in order to apply an English sentiment lexicon for augmenting sentiment analysis on Norwegian tweets?

  - Can one use such a translation to augment sentiment topic identification?

This thesis will aim towards developing a system for sentiment analysis on Norwegian twitter messages, and also attempting to augment this sentiment system with sentiment topic identification. Three **goals** are apparent in this task. *Firstly*, a overview of theory on the subject and a review into the state of the art will be needed. *Secondly*, a sentiment engine with sentiment topic identification for Norwegian tweets will need to be constructed. *Lastly*, evaluation of this engine and its different components will need to be done, in addition to evaluation of aggregated sentiment values.

SECTION 1.4
Challenges

When it comes to machine learning methods on text classification, one has to consider context. While the informal nature of Twitter makes it a platform where users can express their sentiments, it also creates a context in which tasks such as Parts-Of-Speech(POS)-tagging and machine learning classification can be challenging. Not only is Twitter prone to informality, but its context is also continually changing. This change can prove challenging for sentiment analysis systems. Machine learning algorithms would need to be regularly retrained on contextually updated datasets, which - in the case of supervised learning methods - would require more annotation work to be done, which is potentially expensive and time-consuming. Brew et al. show how unexpected noise and trending memes create difficulties for classifiers on social media sites, without considering context it is hard to get good results with such corpora[6].

Most sentiment analysis systems over the years have been targeted towards use on the English corpora, and several language-specific tools have been developed in order to serve as augmentations to the analysis task. Tools such as context-specific POS taggers, large sentiment lexica, and knowledge on sentimental aspects of the language. When attempting sentiment classification on Norwegian tweets, such useful analysis tools are not readily available. This means that language-specific tools either needs to be developed, or already existing tools need to be translated from one language domain to another.

The development of such applications versus the translation and reapplication procedure was discussed by Bautin et al. [7], where they show that such a procedure is prone to high levels of effort for little yield.

SECTION 1.5
Contributions

This thesis has three main contributions corresponding to the goals of the task. The *first and foremost contribution* is in accordance with *RQ1*, which is a sentiment classification system, using machine learning methods. The system utilises several different feature sets, of which differences in performance is evaluated and discussed.

The *second contribution* is the sentiment topic identification part of the system, which uses linguistic methods and POS-tags in order to identify the topics of the given sentiments. Part of this system is also topical sentiment aggregation, both for sentiment visualisation purposes and proving the practical applications of such a system.

The *third contribution* in this thesis is the translation interface using an English sentiment lexicon to tag Norwegian words with sentiment values. These values are used in an attempt to augment the classification process, and experiments will be executed where the performance of this particular part will be scrutinised.

All the parts of the system are elaborated upon in chapter 6.

SECTION 1.6
Structure

This thesis will start by giving an introduction to the theoretical components needed for the task. Thereafter a look into the state of the art will be given, where literature describing this specific scientific field will be reviewed. The theory part of the thesis will be concluded by describing any pre-existing tools used for solving the task at hand, as well as the introduction of some tools which were contenders for the ones used.

The subsequent part will deal with describing the actual implementation used to solve the tasks at hand. Starting with descriptions of the collected datasets. Then the implementation of the system will be described and the reasoning behind it explained. Lastly the experimental set-ups will be described, and the performance of the system will be elaborated upon.

The third and last part of the thesis will present discussions on various subjects regarding the task and the implemented system, finishing with the conclusions from the work on this thesis.

PART II

Prestudy

CHAPTER 2

Theory

"Those people who think they know everything are a great annoyance to those of us who do."

-Isaac Asimov

This chapter will give an overview of the theoretical components needed for this thesis. This will include going through the basics of sentiment, before going into how to perform sentiment analysis using machine learning methods. The chapter will be introduced with a section about Twitter, describing tweets as units of information, and the parts of metadata which will be utilised in the task.

All the theoretical components written about in this chapter where necessary to solve the task at hand. Together these components contribute towards the possibility of implementing the system described in this thesis.

SECTION 2.1

Twitter

Being the eight most visited website in the world[8] results in Twitter being an ever-growing corpus of information. This sizeable corpus in turn attracts interested parties who wants to apply the information that can be extracted from this data. The giant amount of accessible information sets a never before seen precedence for connectivity between computer science and varying fields of academia, such as sociology, psychology, and anthropology.

Politics is also a field where Twitter sentiment analysis can be of great interest. So let us bring up the tweet example in figure 2.1, a tweet posted by the Norwegian Prime Minister Erna Solberg (or possibly the former Prime Minister if this is being read after the year 2017). As can be seen in this figure,

Figure 2.1: A tweet example

this tweet was posted by the user *erna_solberg*, and inside the textbody of the tweet the user *jensstoltenberg* has been tagged. The @ tag is always used to denote and tag the users of Twitter. We can see the hashtag *smd2014* denoted by the # character, which is intended as a form of topic label for the tweet. At the bottom of the tweet we can see information on how many times this tweet has been retweeted and favorited. All these metadata features are potentially valuable when one wants to use machine learning sentiment classification on tweets.

SECTION 2.2
Sentiment and Opinion

Opinion Mining(OM) and Sentiment Analysis(SA) are expressions that are often used interchangeably within the field of computer science, often bearing the same meaning. *Opinion mining* being an expression that originated in the field of Information Retrieval(IR), while *sentiment analysis* is most often used in Artificial Intelligence(AI). In this thesis, *Sentiment Analysis(SA)* will consistently be used as the term when referring to the task at hand.

While these two expressions are often used with the same meaning in mind, we find distinctions in Liu's definitions of opinion and sentiment. Liu[9] defines opinion as a quintuple:

> (g, s, h, t)
> where g is the opinion target, s is the sentiment about the target, h is the opinion holder and t is the time when the opinion was expressed.

It is clear that Liu defines the sentiment as being a part of an opinion, while the opinion as a whole consists of more contextual information. All 4

components of opinion, as Liu defines it, will be focused upon in this thesis. The opinion target will usually be referred to as the *sentiment topic*.

Figure 2.2: A tweet example with sentiment

Bringing Liu's definition into an example, a tweet by Norwegian politician Trine Skei Grandecan can be seen in figure 2.2, regarding Norway's Prime Minister Erna Solberg. Skei Grande expresses her thanks to Solberg regarding a climate package proposed by Solberg, which Skei Grande seems to be positive towards. If formulated using Liu's quintuple this opinion takes the form of ("erna_solberg", "positive", "Trinesg", "21:96 09.04.2014").

Two words are worth noting in this tweet: "Takk" (eng: "thank you), and "god" (eng: "good"). These are both words bearing positive sentiment orientation in and of themselves, and both would most likely be classified as such by a sentiment lexicon. However, sentiment is rarely that simple. There are negations, idioms, and other ambiguities in language. Worst of them all is sarcasm. Without taking user history and additional user information into account in order to build a profile, detecting whether a user is being sarcastic is a very hard task. Therefore, for the rest of this task we shall pretend that sarcasm doesn't exist and we shall never mention it again.

SECTION 2.3
Document Sentiment Classification

The way sentiment analysis is done is greatly dependent on the level of **granularity** one wants to analyse. Different levels of granularity require different tools and methods, and some granularities are more feasible in some contexts than others.

The highest granularity level is the *document level*. At this level, one is concerned with determining the sentiment of each document as a whole[5]. In order for this level of granularity to have applicable value, one usually wants to assume that each document expresses sentiment on a single topic. Corpora with documents such as customer reviews are very suitable for analysis on this granularity level. A more detailed level of granularity is the *sentence level*, where methods performing sentiment analysis at this level attempts to determine the sentiment of single sentences. Finally, the finest level of granularity is at the *entity level*. In order to analyse the sentiments at the entity level one has to create a more holistic model of sentiment where including the target of expressed sentiment is necessary. This, of course, requires more advanced linguistic computation and information modelling. Systems performing analysis at this level are very useful tools for performing structured sentiment summaries on entities, turning unstructured text into structured data. This aspect of this level is very valuable for different qualitative and quantitative analyses[9].

A common way to perform sentiment analysis is to perform it in **two binary steps**. First, you identify a text to be either objective or subjective. Subsequently, you take the subjective tweets and determine their polarity, i.e. whether they are negative or positive[10]. These two steps are often done using supervised learning methods. Supervised learners are often the methods of choice when there is access to annotated datasets on which to train the classifiers. Especially, in the case of Twitter, there are means of obtaining datasets where the tweet classes can be determined automatically[11]. This enables the acquisition of large training datasets without the tediousness of manual annotation.

When using machine learning techniques for text classification, **feature engineering** is an important part of it. The **features** of a machine learning classifier is a selected subset of the measurable properties that define the documents in the corpus. Selection of the feature set is often performed as a combination of empirical selection by a domain expert and automated methods. The set of feature values for a given document is usually called the **feature vector** of the document. In text classification tasks, term frequencies are commonly used as features. When using term frequencies one regards frequencies for different sizes of **N-grams**, i.e. different sizes of combinations of terms. The most common usage of N-grams are unigrams, bigrams, or a combination

of both.

Several supervised machine learning classifiers exist suitable for sentiment classification. The below subsections detail the theoretical components of some of these classifiers.

SUBSECTION 2.3.1

Multinomial Naive Bayes

Naive Bayes(NB) is a fast and versatile classification algorithm, and it is probably the algorithm for supervised text classification which is easiest to implement. The performance of the NB classifier often reflects its simple nature, in that it is often outperformed by other more sophisticated classifiers such as Support Vector Machines(SVM)[12]. Performance among different classifiers will be shown in more detail when reviewing the state of the art in chapter 3.

The NB classifier is based on Bayes theorem, which gives us the relationship between P(A) and P(B), i.e. the probability of event A and the probability of event B. This relationship takes the form of the theorem below:

$$P(A|B) = \frac{P(A) \times P(B|A)}{P(A)} \tag{2.1}$$

In short, this theorem enables a classifier to calculate the posterior probability of B given A, using prior probabilities. Since this theorem is to be used in a NB classifier for tweets, we will formulate it like so:

$$P(c_p|\vec{d_j}) = \frac{P(c_p) \times P(\vec{d_j}|c_p)}{P(\vec{d_j})} \tag{2.2}$$

where $P(\vec{d_j})$ is the probability that a randomly selected tweet will be represented by $\vec{d_j}$, and $P(c_p)$ is the probability that a randomly selected tweet belongs to class c_p. [13]

The classification functionality then becomes finding the class with the largest probability function given by the product of all the feature probabilities, given their class labels. This functionality is described by the equation below.

$$classify(f_1, ..., f_n) = argmaxp(C = c) \prod_{i=1}^{n} p(F_i = f_i|C = c) \tag{2.3}$$

This equation shows the simplicity of the Naive Bayes classifier. In essence, all we need to do to train our classifier is to count all the features and which classes they appear in, and use these frequencies to compute their probabilities. When classifying, we select the class which gets the highest product of the features given by the target feature vector.

SUBSECTION 2.3.2

Support Vector Machines

Support Vector Machines(SVM) is a relatively new technique for text classification, the method was first used for this purpose by Joachims in 1999[14]. Compared to the NB classifier, the SVM method is a lot more complex, and as a result more difficult to implement.

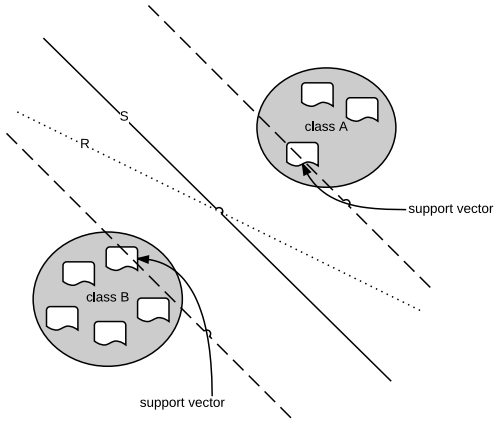


Figure 2.3: Two-dimensional SVM example

In figure 2.3 we can see the basic intuition behind the SVM method. In this example the documents are represented as points in a two-dimensional space. In a practical instance of text classification, the dimension size is decided by the feature size, which means that we usually operate with several dimensions.

The idea behind training the classifier is to find the support vectors which maximise the space - the decision surface - between the two classes, i.e. finding the optimal separation between the features representing the two classes. The two support vectors in the figure are defined by the documents that lie closest to the decision surface.

The task of training an SVM classifier can such be formulated as the optimisation problem of finding the optimal hyperplane. Yates states this optimisation problem as follows[13]:

> Let H_w be a hyperplane that separates all documents in class c_a from all documents in c_b. Let m_a be the distance of H_w to the closest document in class c_a and let m_b be the distance of H_w to the closes document in class c_b, such that $m_a + m_b = m$. The

distance m is the *margin* of the SVM. The decision hyperplane H_w maximises the margin m.

When the optimised decision surface has been calculated, any future instance presented to the classifier is evaluated using their position in the space as represented by the features of this instance. The instance's position relative to the separation between the classes determines which class is decided for the new instance.

SUBSECTION 2.3.3
Maximum Entropy

A Maximum Entropy(MaxEnt) classifier is a conditional probabilistic classifier. Implementations of it use logistic regression in order to find the probability distribution with the largest entropy, which - given by the Theory of Maximum Entropy[15] - should be the one best to represent the current state of knowledge, given precisely stated prior data[16].

Unlike the NB classifier, MaxEnt assumes no conditional independence for the features. This means that MaxEnt handles feature overlap better than the NB classifier[17]. It also means that for text-only features, the MaxEnt classifier will perform better given that most of the time we work with words that are conditionally dependent of each other.

Go et al formulates the model the following way[17]:

$$P(c|d,\lambda) = \frac{exp[\sum_i \lambda_i f_i(c,d)]}{\sum_{c'} exp[\sum_i \lambda_i f_i(c,d)]} \tag{2.4}$$

where, in this case, c is the class and d is the tweet. Here λ represents a weight factor, which corresponds to the relevance of a feature for a given class. The numerical operations of the task of optimising these lambdas is what makes the MaxEnt implementation non-trivial and time-consuming.

For text classification tasks, MaxEnt classifiers have been shown to have an accuracy performance which is on par with SVM[16].

SECTION 2.4
Preprocessing

Preprocessing is the act of making text documents more consistent in order to better facilitate text representation, and is necessary for most text analysis tasks. Traditional preprocessing methods typically involve *stemming/lemmatisation*, *tokenisation*, and *stop word removal*[18].

Stemming/Lemmatisation is the attempt at reducing words to more basic lexemes. Stemming being the crude method of simply cutting of a suffix of an inflected or derived word in order to reduce it to a base form, often the

morphological root of the word. Lemmatisation usually utilises a dictionary in order to map inflected words to a single entry, in order to identify them as one word.

Tokenisation is the task of breaking a sentence or document up into words or phrases. This process also usually handles digits, letter casing, hyphens, and other special characters.

Stop word removal is the act of removing words which are considered more or less meaningless with regards to the task at hand. They are often computed beforehand and stored in a stop word list used at the time of pre-processing. There are several different methods of stop word removal. **Max DF(Document Frequency)** is a stop word removal scheme often used along with using a TF-IDF weighing scheme. It removes stop words on the basis of a document frequency cutoff value. Given an instance where max df is set to 0.5, only n-grams with a df below 0.5 will be used for classification, and all words above this frequency will therefore be disregarded like stop words.

SECTION 2.5
Computational Linguistics

Computational linguistics encompass several methods of deriving grammatical metadata from text. The theoretical background of the computational linguistics tools used in this thesis are elaborated upon in the subsections below.

SUBSECTION 2.5.1
Sentiment Lexicon

Sentiment lexica are useful given the fact that the sources of sentiment can often be identified from specific words. In Norwegian, adjectives like *"bra"*, *"fantastisk"*, and *"vakkert"* are words that are used to express positive sentiment, while words like *"elendig"* and *"forferdelig"* are used to express negative sentiment. Sentiment lexica are made by creating dictionaries of sentiment words and their corresponding sentiment values. These dictionaries are often compiled by using manual annotation augmented by automated methods, and expanded by using dictionary based methods or corpus based methods[9].

The main issue with using a sentiment lexicon is that in itself it does not take any context into consideration. Some sentiment words have very different meanings in different contexts. This aspect is particularly visible in verbs that also appear as idioms bearing sentimental values. E.g. the verb *"suger"*(eng: *sucks*) is often associated with a negative sentiment when appearing as an idiom as seen in the left-hand tweet in figure 2.4. However it can also appear as a verb giving implicit sentimental value, a statement expressing that a vacuum cleaner *"suger"* as in the right-hand tweet in figure 2.4, can implicitly express

Figure 2.4: Tweet examples showing use of idioms

positive sentiment towards the vacuum cleaner since this is actually what a vacuum cleaner is supposed to to. Disambiguating between different entries in a lexica becomes impossible unless one has contextual metadata which can help provide an answer as to which is correct. Metadata on word classes obtained from a Part-of-speech tagger can be of good help with this disambiguation task.

SUBSECTION 2.5.2

Part-of-Speech Tagging

Part-of-speech tagging is a useful text data mining procedure essential to obtaining structural grammatical metadata from text corpora. It is the act of assigning descriptive grammatical tags to the words in a text, e.g. deciding whether a word is an adjective, adverb, noun etc[19]. These taggers are often created using statistical machine learning algorithms such as Maximum Entropy[20].

For tasks in sentiment classification, part-of-speech tagging can be a useful tool. The presence of certain word classes can for instance help indicate whether a text is subjective or not. E.g, a presence of personal pronouns in a text can be an indicator that a text contains a subjective sentiment[11].

Table 2.1 show ten word classes in Norwegian. The tag sets used to assign words to classes are often of higher morphological complexity than these classes mentioned above. For instance the TypeCraft tag set found in appendix E which is used in the NTNU SmartTagger, contains 20 different verb tags - transitive verbs, modal verbs, reflexive verbs, etc. - and 12 different noun tags - common noun, masculine noun, proper noun, etc.

SUBSECTION 2.5.3

Valence Shifters

Valence shifters are words that can interact with parts of a sentence and shift the sentiment. There are three main classes of valence shifters: *negators*, *intensifiers*, and *diminishers*[21].

Negators are words that reverse the sentiment of a word from positive to negative or vice versa. *Intensifiers* are words that strengthen the sentiment

Table 2.1: Word-classes used in part-of-speech taggers

Class	Description	Examples
Nouns	Things, people, and places	*Erna Solberg*, *hus*
Verbs	An action	*rope*, *synge*
Adjectives	Describes properties of pronouns or nouns	*smart*, *fire*
Adverbs	Describes properties of verbs, adjectives or other adverbs	*mye* in "jeg trente *mye*"
Pronouns	A replacer for nouns	*hun*, *de*
Prepositions	Describes where a noun is relative to another noun	*under*, *omkring*
Conjunctions	Binds similar words, phrases, or sentences	*og*, *men*
Interjections	Utterenaces which can stand alone and still bear meaning	*uff!*, *nei*
Two Norwegian wordclasses with no English equivalent		
Bestemmerord	Decides more detail in the noun	*ingen*, *hver*
Subjunksjoner	Initiates joint sentences	*Når* in "*Når* jeg sykler"

of a word, and *diminishers* are words that weaken the sentiment of a word. Several examples of negators, intensifiers, and diminishers can be seen in table 2.2.

SECTION 2.6
Sentiment Topic Detection

When it comes to performing standard topic detection in documents, unsupervised methods like clustering are often proposed as solutions. Topical classifiers can also be trained on specific topics in order to answer the binary query on whether a given document belongs to a given topic or not with a relatively high accuracy. However, in order to jointly identify both sentiment and the sentiment topic of a document, more advanced methods are needed, often utilising linguistic methods[22].

Pointwise Mutual Information is a method often used to evaluate the importance of a topic term by calculating its mutual dependence with either a document class or another term. PMI between two terms t_1 and t_2 is calculated as

$$PMI(t_1, t_2) = log \frac{P(t_1 \bigwedge t_2)}{P(t_1) \times P(t_2)} \tag{2.5}$$

Table 2.2: Examples of different valence shifters in Norwegian

Shifter	**Example**
	Negators
ikke	"Han er *ikke* en snill fyr"
ingen	"*Ingen* er flinke til å skrive masteroppgave"
hverken	"Du er *hverken* kul eller smart"
	Intensifiers
ganske	"Du er *ganske* alarmerende"
veldig	"Det var en *veldig* lekker ostekake"
mer	"Strikkhopp er *mer* spennende nå"
	Diminishers
lite	"Dette var en *lite* smart masteroppgave"
noe	"Han har *noe* intelligens i seg"
mindre	"Strikkhopp er *mindre* spennende nå"

where $P(t_1 \bigwedge t_2)$ is the co-occurrence probability of t_1 and t_2 and $P(t_1) \times P(t_2)$ express the probability that these two terms occur together if they are statistically independent[22].

SECTION 2.7

Testing and Evaluation

In order to properly test and evaluate a trained classifier, a dataset of instances not previously seen by the classifier is needed. In order to obtain such a set, a common practice is to partition the dataset before training, into a training set and a testing set. The most basic method is the **Holdout method** which simply partitions the set into a given fraction, and saves the smaller part of the set to be used as the testing set. The problem with the holdout method is that it does not take advantage of the full potential of the dataset for training the classifier, which can result in poorer performances.

K-fold cross validation is a method taking better advantage of the full dataset. This method partitions the original dataset into k folds, and performs training and testing k rounds, holding off each of the folds for testing and training with the rest of the dataset in turn. Performance of the classifier is then calculated as the mean of all k rounds of training and testing.

There are several different evaluation metrics used to measure the performance of a classifier, each one giving different insights into the classifier performance[13]. A tabulation of known labels and predicted labels can be seen in table2.3, values which are used in the description of the computations of the various evaluation metrics below.

The *Accuracy* of a classifier is one of the simplest metrics. It is the propor-

Table 2.3: Tabulation of predictions

		Predicted label	
		Positive	**Negative**
Known label	**Positive**	True positive	False negative
	Negative	False positive	True negative

tion of all the instances which has been correctly classified. Thus it is given by

$$Accuracy = \frac{T_p + T_n}{T_p + F_p + T_n + F_n} \tag{2.6}$$

where T_p is the number of true positives, T_n is the number of true negatives, F_p is the number of false positives, and F_n is the number of false negatives.

Precision and *Recall* are two evaluation metrics originally used in IR systems, but have been adopted and widely used in classification tasks. Precision is a measure of the relevance of the results returned by a classifier, and recall is a measure of how many relevant results are returned. In a binary classification task this means that the precision score is given by

$$Precision = \frac{T_p}{T_p + F_p} \tag{2.7}$$

and the recall score is given by

$$Recall = \frac{T_p}{T_p + F_n} \tag{2.8}$$

The F1 score - or the balanced F-score - is the harmonic mean between precision and recall,

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{2.9}$$

Two variations upon this metric are the F_2 score which evaluates with greater emphasis on recall, and the $F_{0.5}$ score which puts greater weight on precision[23].

SECTION 2.8
Feature Engineering

Feature engineering for machine learning text classification is the task of selecting a set of properties for the documents in a corpus. These properties are then the ones used in classification of the documents, their values determining the outcomes of the classification task. Several specific automated methods exists for feature engineering.

TF-IDF is one of the most popular feature weighing schemes. The method is composed of using the Term Frequency(TF) together with the Inverse Document Frequency(IDF) in order to weigh the importance of a term as classification feature. The TF is often calculated as the raw frequency of the term divided by the total number of terms in the document:

$$tf(t,d) = \frac{f(t,d)}{|d|} \tag{2.10}$$

where $f(t,d)$ is the raw frequency of term t in document d, and $|d|$ is the length of the document.

The IDF is then calculated as

$$idf(t,d,D) = log\frac{|D|}{|\{d \in D : t \in d\}|} \tag{2.11}$$

where D is the corpus. In plain English, the IDF is the *log* of the total number of documents in the corpus divided by the number of documents in which the term occurs. The total TF-IDF is finally calculated such

$$tdidf(t,d,D) = tf(t,d) \times idf(t,D) \tag{2.12}$$

The idea behind TF-IDF is that a term is usually considered less informational to the given document if it also appears in several other documents in the corpus. A higher frequency in other documents results in a low IDF, in turn de-evaluating the given term[13].

SECTION 2.9
Annotation

When performing supervised learning the issue of performing and evaluating manual annotation is usually a necessity. Since manual sentiment annotation is inherently subjective the annotations can differ from person to person, therefore it is important to analyse the annotated dataset in order to assess the reliability of the performed annotations. The most basic way of doing so is calculating the **Joint-probability of agreement**, which simply calculates the percentage of instances where the annotators have agreed. The **Cohen's Kappa coefficient** is a more robust way of since it takes into account the agreement occurring by chance. It is calculated by

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \tag{2.13}$$

where $Pr(a)$ is the observed percentage of agreement, and $Pr(e)$ is the expected percentage of agreement[24].

CHAPTER 3

The State of the Art

"If you steal from one author, it's plagiarism; if you steal from many, it's research."

-Wilson Mizner

This section goes into the current state of the art within SA-systems and sentiment topic detection. It will start by presenting the use of a semi-structured literature review and its findings, then the state of the art will be presented in a topically structured manner.

SECTION 3.1

A Systematic Literature Review

There is a great need for systematic literature reviews in software engineering, due to the fact that during the recent years the field has seen a rise in the amount of empirical studies[25]. For this thesis, the need for a review into the state of the art is evident. The task in this thesis bases itself on theoretical components which have been largely researched and published about. Therefore, this thesis adopts a systematic literature review. The following section details the process of this review.

With the intention of meeting the need for information on the subjects comprising the task at hand, several search queries were formed. The queries were then posted in the Google Scholar search engine[26]. Google Scholar accumulates its search results from several different sources, and it was found to be a good provider of relevant articles.

The search queries used in the review were as described below.

Q1: "opinion mining" OR "sentiment analysis" (16 100 results)

Table 3.1: Overview of information labels used in review tables

Label	Description
ML	Machine Learning techniques, such as NB, SVM, MaxEnt etc.
Lexicon	Lexical methods, pattern matching, etc.
PMI	Point-wise Mutual information method for topic-sentiment detection
Twitter	Uses Twitter as corpus
SM	Uses other social media as corpus
News	Uses news articles, or the news domain
Reviews	Uses customer reviews - e.g. movie reviews - as corpus
Forums	Uses web discussion forums as corpus
Blogs	Uses weblogs as corpus
MPQA	Multi-perspective Question Answering Opinion Corpus[27]
Survey	Theoretical survey of different methods

This *first query* was performed in order order to establish a base of informational articles regarding the main subject. Since the two phrases *opinion mining* and *sentiment analysis* are often used pertaining to the same, the query was structured in order to get articles using either of the two. The 20 first results from this query can be seen in table 3.2 and table 3.3.

Q2: twitter "opinion mining" OR "sentiment analysis" (5 410 results)

The *second query* was used to get articles concerning themselves with the twitter corpus. This was important because of the fairly large part context plays in SA-systems. The top 10 results from this query can be seen in table 3.4

Q3: sentiment "topic" OR "target"

The *third query* was designed to get information on SA-systems focusing on identifying the target of sentiment. The top 10 results from this query can be seen in table 3.5

Several labels are used in the tables - 3.2 3.3 3.4 3.5 - in the Comment, Corpus, and Method columns. These labels are used to denote various aspects of the systems described in the literature review. These labels and their descriptions can be found in table 3.1.

In addition to literature from the semi-systematic review, more literature comprise this description of the state of the art. This collection of literature

has been found by varying means. Some of the articles and theses have been recommended by the supervisor, but most of them have simply been fished up while frolicking in the vast sea of information that is the World Wide Web. A total of 20 relevant articles were found by mostly using the highly unscientific method of informational frolicking.

Table 3.2: Table over query results from Q1

Title	**Author(s)**	**Comment**	**Year**	**Corpus**	**Method**
A sentimental education: Sentiment analysis using subjectivity[10]	Pang, Lee	Two-step MC	2004	Reviews	ML
Sentiment analysis using support vector machines with diverse information sources[28]	Mullen, Collier	Several metadata sources, and syntactic relations	2004	Reviews	ML and Lexicon
Recognizing contextual polarity in phrase-level sentiment analysis[29]	Wilson, Wiebe, Hoffmann	Phrase-level SA	2005	MPQA	Lexicon
Using appraisal groups for sentiment analysis[30]	Whitelaw, Garg, Argamon	Builds feature lexicon of appraisal groups	2005	Reviews	ML
Determining term subjectivity and term orientation for opinion mining[31]	Esuli, Sebastiani	Semi-supervised SA	2006	-	ML
Sentiwordnet: A publicly available lexical resource for opinion mining[32]	Esuli, Sebastiani	Presents SentiWordNet	2006	-	Lexicon
Fully automatic lexicon expansion for domain-oriented sentiment analysis[33]	Kanayama, Nasukawa	Builds sentiment lexicon	2006	Reviews	Lexicon
Large-scale sentiment analysis for news and blogs[34]	Godbole, Srinivasaiah	SA and entity identification	2007	News	Lexicon
Structured models for fine-to-coarse sentiment analysis[35]	McDonald, Hannan, Neylon, Wells, Reynar	Several granularities	2007	Reviews	ML
A holistic lexicon-based approach to opinion mining[36]	Ding, Liu, Yu	Holistic	2008	Reviews	Lexicon

Table 3.3: Continued table over query results from Q1

Title	Author(s)	Comment	Year	Corpus	Method
Sentiment analysis: Capturing favorability using natural language processing[37]	Nasukawa, Yi	Identifies sentiment subjects	2003	News	Lexicon
Opinion Mining and Sentiment Analysis[38]	Pang, Lee	Survey of methods	2008	-	-
Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums[39]	Abbasi, Chen, Salem	Genetics algorithm for feature extraction	2008	Forums, Reviews	ML
Joint sentiment/topic model for sentiment analysis[40]	Lin, He	Unsupervised classification	2009	-	ML
Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis[41]	Wilson, Wiebe, Hoffmann	Contextual feature engineering	2009	MPQA	ML
Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining[42]	Baccianella, Esuli, Sebastiani	Improved SentiwordNet[32]	2010	-	Lexicon
Sentiment analysis and subjectivity[43]	Liu	Chapter from Handbook of NLP	2010	-	-
Lexicon-based methods for sentiment analysis[44]	Taboada, Brooke, Tofiloski	Survey	2011	-	Lexicon
Twitter as a corpus for sentiment analysis and opinion mining[11]	Pak, Paroubek	N-gram experiments	2010	Twitter	ML
Sentiment analysis and opinion mining[9]	Liu, Zhang	Chapter from Mining Text Data	2012	-	-

Table 3.4: Table over query results from Q2

Title	**Author(s)**	**Comment**	**Year**	**Corpus**	**Method**
Twitter sentiment analysis[45]	Go, Huang, Bhayani	Automatic annotated corpus acquisition	2009	Twitter	ML
Twitter sentiment classification using distant supervision[17]	Go, Bhayani, Huang	Emoticons as noisy data	2009	Twitter	ML
Twitter sentiment analysis [46]	Sharm, Vyas	Six mood dimensions (POMS)	2010	Twitter	Lexicon
Twitter sentiment analysis[47]	Jose, Bhatia, Krishna	Subjectivity filtering using AFINN	2010	Twitter	ML
Twitter as a corpus for sentiment analysis and opinion mining[11]	Pak, Paroubek	-	2010	Twitter	ML
Sentiment knowledge discovery on twitter streaming data[48]	Bifet, Frank	Survey	2010	Twitter	-
Sentiment analysis of twitter data[49]	Agarwal, Xie, Vovsha, Rambow, Passonneau	POS specific prior polarity features	2011	Twitter	ML
Target-dependent twitter sentiment classification[50]	Jiang, Yu, Zhou, Liu, Zhao	Target-dependence	2011	Twitter	ML
Twitter sentiment analysis: The good the bad and the omg![51]	Kouloumpis, Wilson, Moore	Utility of linguistic features using Adaboost	2011	Twitter	ML
Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach[52]	Wang, Wei, Liu, Zhou, Zhang	Hashtag-level sentiment	2011	Twitter	ML

Table 3.5: Table over query results from Q3

Title	Author(s)	Comment	Year	Corpus	Method
Thumbs up? Sentiment Classification using Machines Learning Techniques [5]	Pang, Lee	Unigram bag-of-words	2002	News	ML
Sentiment Analyzer: Eztracting Sentiments about a Given Topic using Natural Language Processing Techniques[53]	Yi, Nasikawa, Bunescu, Niblack	Sentence-level subjectivity	2003	Reviews, News	ML, Lexicon
Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs[54]	Mei, Ling, Wondra, Su, Zhai	Specific HMM structure	2007	Blogs	ML
Leveraging Sentiment Analysis for Topic Detection[55]	Cai, Spangler, Chen, Zhang	Uses PMI to identify sentiment topics	2008	SM	ML
Domain-specific Sentiment Analysis using Contextual Feature Generation[56]	Choi, Kim, Myaeng	Notes here	2009	Several	ML
Joint Sentiment/Topic Model for Sentiment Analysis[40]	Lin, He	LDA-based sentiment-topic model	2009	Twitter	ML
Target-Dependent Twitter Sentiment Classification[50]	Jiang, Yu, Zhou, Liu, Zhao		2011	Twitter	Lexical
Alleviating Data Sparsity for Twitter Sentiment Analysis[57]	Saif, He, Alani	Compensates for shortness of tweets	2011	Twitter	ML
Weakly-supervised Joint Sentiment-Topic Detection from Text[58]	Lin, He, Everson, Ruger	Portable topic-sentiment model based on sLDA	2012	Twitter	ML
Semantic Sentiment Analysis of Twitter[59]	Saif, He, Alani	Semantic concept modelling	2012	Twitter	ML

SECTION 3.2
Sentiment Analysis

Automated sentiment analysis originated within the field of computational linguistics, traditionally utilising known linguistic patterns and textual behaviour, and using rule-based systems and frameworks in order to identify deeper linguistic meanings. Some of the first instances of sentiment analysis came back in 1992, when Hearst suggested refining information access by mining text *directionality* - i.e. is the agent in favor of or opposed to the event - arguing that traditional topical analysis methods were lacking[60]. In 1994 Wiebe made an attempt at tracking a characters' *psychological point of view* in third-person narrative texts[61]. When identifying a specific character and tracking the point of view of this character, Wiebe showed results of 60% accuracy on identifying subjective sentences.

SECTION 3.3
Machine Learning Methods

The prevalence of machine learning methods within the field of sentiment analysis and opinion mining has risen greatly within the last decade, still showing increases in performances by utilising more sophisticated algorithms, richer metadata acquisition, and ingenuitive feature engineering methods.

Already in 2002, Pang and Lee showed that using machine learning methods with simple unigram bag of word features can achieve accuracies of around 80%, results which were generally higher than their human baseline annotations[5]. Two years later Pang and Lee[10] proposed a solution to sentiment classification using two-step classification for subjectivity and polarity classification, using Minimal Cuts(MC) to augment SVM and NB classification on sentence level subjectivity and document level polarity on movie reviews. They showed a performance of up to 86.4% accuracy.

Machine learning methods can be found used at various granularities of sentiment analysis. Determining sentiment at the term level, sentence level, document level are all viable tasks. McDonald et al. presents a hierarchical sequence learning model similar to CRF, used to classify sentiment at two different levels of granularity simultaneously - at the sentence level and the document level[35]. Their system yields an accuracy of 82.8% at the document level, and they show that the model performs better when classification is done at the two levels jointly as when compared to separately.

Feature engineering, or **feature selection**, is an important aspect when using machine learning methods for sentiment analysis. Proper engineering of contextual features can provide higher informational value and reduce chances of noise. In order to achieve this, several sources for features are often used.

Using SVM to combine a set of features from topic model knowledge, syntactic relations, pre-annotated favorability-measures for phrases and adjectives, along with standard unigram text features was shown by Mullen and Collier to achieve a performance of 86.0%[28]. Whitelaw et al. constructed appraisal groups for features, more complex features consisting of adjectives along with their modifiers[30]. Using Weka's Sequential Minimal Optimisation(SMO) algorithm on a movie review corpus from the Internet Movie Database they showed an accuracy of 90.2% with the appraisal group feature sets.

Abbasi et al. showed an accuracy of over 95% on a benchmark data set of web movie reviews[39]. They develop their Entropy Weighted Genetic Algorithm(EWGA) and use SVM with an extensive set of stylistic features - letter frequencies, character n-grams, special characters, word lengths, etc. - features selected by using their genetic algorithm.

Wilson explored the importance of feature engineering for contextual polarity[41], developing an automatic system of identifying features that can distinguish between prior and contextual polarity for sentiment lexicon terms. Experimenting with several different ML methods, they show that most classifiers perform better with the contextual polarity features.

SECTION 3.4
Sentiment Lexica

Lexica with sentimental metadata have been very popular for performing sentiment analysis. Creation of these often involves a combination of human annotation and automated methods such as synonym and antonym classes or machine learning techniques. After a successful creation, a sentiment lexicon combined with some simple syntactic rules can be a powerful method of sentiment classification in and of itself.

Wilson et al. built a sentiment lexicon using thesauri and dictionaries to expand upon a list of sentiment clues from Riloff and Wiebe[62], then use an AdaBoost classifier in order to perform subjectivity classification and polarity classification with metadata from their lexicon as features in the classifier[29]. They show that a lexicon with sentiment clues is a very viable method of performing subjectivity identification.

The corpus method is another automatic way of expanding sentiment annotated lexica with words and phrases. Early ideas presented by Hatzivassiloglou and McKeown describe a method starting with a corpus and a set of seed adjectives with sentiment values, and utilise a set of linguistic patterns and conventions on connectives in order to expand and create a more substantive sentiment lexicon [63]. The corpus method was expanded upon by Kanayama and Nasukawa and applied on Japanese customer reviews, they show that the automatically acquired lexicon achieves a precision score of 94% on average[33].

Ding et al. focused on the issues of lack of context when using sentiment

lexica[36]. They propose a holistic lexicon-based approach which uses linguistic rule-based conventions of natural language expressions.

Esuli et al. in 2006 used semi-supervised learning to determine term orientation and subjectivity, in order to create sentiment lexica [31]. They create the English sentiment lexicon SentiWordNet[32], containing tens of thousands of entries and their objectivity-negativity-positivity triples. The lexicon was then later improved upon in 2010 by Bacianella et. al [42], where they show improvements of up to 20% when evaluated against manually annotated words.

Sentiment lexica can be a lot of work to create, and when created one would want to exploit their usefulness to the fullest. Sadly, this method is of course language-restricted, which in turn has spawned several systems utilising machine translation in order to be able to use sentiment lexica designed for other languages than the language to be classified in. Translations of documents and performing SA using English SAs [64], and Wan performed sentiment analysis on Chinese product reviews by translating the entire reviews into English, and using English-specific sentiment tools including a sentiment lexicon to achieve an accuracy of up to 86.1%[65]. Bautin et al. used similar methods for Spanish, and discussed the use of translators for use of English sentiment lexica on corpora of several different languages[7].

SECTION 3.5
Sentiment Topic Detection

A lot of work have been done on identifying the topics of documents, Blei[66], Griffiths[67], and Titov[68] are a few among many others. However these works focus on only the text topics and disregards any sentiment towards them. Discounting the sentiment limits the usefulness of such topic mining results. They can say something about the popularity of the topics and the frequencies of their discussions, but the lack of any sentiment evaluation on these topics results in an information deficit regarding the statements in which the topics are brought up. This in turn arguably reduces their practical value, especially within areas such as politics, marketing, and finance. Take marketing as an example: Although the saying goes "all PR is good PR", there is more data to be considered. Everything expressed by a person or organisation is subject to bias, and thus a biased sentiment. Even supposedly objective news organisations can express sentiment towards the news items on their agenda. For a marketeer, using a model which encompasses this sentimental nuance is important if one wants to analyse the real-world situation regarding any entity viewed in the public light. Also, for certain corpora topic analysis method may not be effective at all. Nigam and Hurst found that within USENET corpora only 3% of sentences contained any topical information[69], while Subasic and Huettner showed that informal web discourse corpora are rich with sentimental information[70].

In order to pair sentiments and topics in documents, Natural Language Processing tools are very often used. Nasukawa and Yi used in 2003 a syntactic parser and a sentiment lexicon to extract sentiments on specific subjects in documents, showing a precision ranging from 75 to 95%[37]. Yi et al. also developed a system able to extract sentiment regarding a specific topic on the web. This system analysing sentiment of web entities by identifying links to the objects, and augmenting with a sentiment lexicon for textual sentiment analysis[53].

The news article domain has also been used for sentiment topic detection. Godbole et al. created a system which associates human-provided relevant entities with expressed opinions along with sentiment aggregation with regards to these entities[34]. The system is built on top of the *Lydia* system which utilises the WordNet synonyms and antonyms both for sentiment lexeme expansions and entity identification[71]. Njølstad and Høysæter exploited inherent metadata in their dataset of news articles - news articles posted with pre-existing firm-specific tags - along with carefully selected contextual feature sets in order to attempt to predict stock prices at the Oslo Stock Exchange [72]. They proved that for articles classified as positive by their system a rise in traded volume for the selected stocks could be seen.

Work has also been done with sentiment topics within the context of weblogs. Opinion tracking, user behaviour prediction, and search result summarisation using weblog data was shown possible using a Topic-Sentiment Mixture Model by Mei et al[54]. The modelling of topics often takes a hierarchical form, like Saif et al. identifying semantic concept parents for topics to augment the classification [59]. More advanced topic modelling methods are also introduced. Wei and Gulla dealt with the hierarchical structure of the fine-grained aspects of products in customer reviews by introducing a Sentiment Ontology Tree(SOT) with Hierarchical Learning(HL)[73].

Lin and He showed in 2009 a *Joint Sentiment-Topic(JST)* model based on *Latent Dirichlet Allocation(LDA*), a topic modelling and document level sentiment classification system with accuracy of up to 85.6%[40]. They improve upon this system in 2012[58], where they argue the applicability of their model in a larger scale, as well as experiment with reversing the sequence of sentiment and topic detection.

SECTION 3.6
On Contextuality and Using Twitter as a Corpus

When performing sentiment analysis in an informal context the traditional computational semantic methods are no longer as viable as before. The once strict syntactic rules and semantic conventions are suddenly thrown out the window for *lol*s, *omg*s, elongated vowels, and hearts and smiley-faces. The differences in context have been shown to have effects on the performances of

machine learning classifiers. In order to view the effects of differing contexts Aue and Gamon trained SVM classifiers on four different domains[74]. They show a drop of almost 20% when the classifier was trained on informal text and tested on formal text, and vice versa.

The Microblog context is also an ever-changing context. With so many new tweets every day, a classifier must take this change into consideration if it is to have any practical value over a longer time period. Bifet and Frank elaborate on several considerations when designing a classifier for a constant stream of tweets[48]. Brew et al. elaborated upon a system identifying the underlying causes behind sentiment shifts, highlighting cases susceptible to noise and trending memes causing sentiment shifts[6].

Go et al. utilised the Twitter framework and it's informal context to acquire an annotated Twitter corpus automatically, by fetching tweets containing happy emoticons - ":)" - and labelling them as positive and tweets containing sad emoticons - ":(" - and labelling them as negative[45]. Using only bigrams and unigrams they achieved an accuracy of 83% with MaxEnt as their classifier. Pak and Paroubek expanded upon this automatic method of collecting a Twitter corpus by including neutral tweets from various newspapers and magazines. They show - using NB, SVM, and CRF - the differences in results for various n-grams, concluding with bigrams yielding best performances for the Twitter corpus[11].

Arguably, proper contextual features are as important - if not more important - in an informal context as in a formal one. However, Agarwal et al. performed a comparison between the feature based model and the Tree Kernel method[49]. They showed the two methods performing at equal levels, arguing for the Tree Kernels possibility to leviate tedious manual feature engineering. At the same time Kouloumpis et al. argued for the importance of microblog features such as emoticons and intensifiers. By comparing features from the MPQA subjectivity lexicon[75], part-of-speech features, and microblogging features they show substantially increased accuracy when using microblog features[51]. Jiang et al. used a dependency parser to generate topic dependent features for a human provided query[50], and show increased accuracy scores when utilising the target dependent features. Similarly, Saif et al. attempts to augment sentiment classification using extracted topic from tweets[57], using the JST model from Lin and He[40].

Sentiment lexica are still viable tools for sentiment classification in an informal context. Jose et al. showed promising results using the AFINN lexicon for subjectivity filtering on tweets[47]. Sharma and Vyas used an extended version of the *Profile of Mood States(POMS)* lexicon[76], containing sentiment values on six mood dimensions. They show a strong correlation between their findings of aggregated sentiment and real-world events[46].

Grammatical metadata can be very helpful in text classification tasks, and as such, POS-taggers have become very popular for use in text classification tasks. POS-taggers for the English language are in abundance. However, tag-

gers trained on formal contexts are known to perform with less accuracy on corpora of informal texts, shown with several augmentations to reach 88.7% accuracy on a Twitter corpus[77]. In order to achieve better precision there have been developed contextual POS-taggers which have been trained for the informal context of social media sites. The informal vocabulary used on WBSMs makes it generally a hard task to perform tagging without taking context into account. Contextually trained taggers have been shown to perform with an accuracy nearing 90%[78].

CHAPTER 4

Tools

This chapter will give a description of the tools and data used in this thesis, as well as an overview of the tools that were contenders for being used instead of the selected tools.

SECTION 4.1

Twitter and Tweepy

Twitter grants free developer access to its REST API [79]. This API allows developers access to searching and downloading tweet text and metadata, as well as access to the Twitter stream; access to a stream of new public tweets. The Twitter Firehose access can give a user access to acquiring live access to ALL public tweets on Twitter, this part of the API however requires special access usually given through third-party handlers and a subscription fee. When using the API Twitter asks that developers follow their "developer rules of the road"[80].

The Tweepy Python API was used for access to the Twitter API, it is a framework simplifying this process[81]. It allows easier access to the search and stream functions of the Twitter API. Tweepy made handling the Twitter API connection a breeze, a task which would be far more complex if one was to use the Twitter API directly. Connection was set up with the routine

```
1     def __init__(self):
2         auth = tweepy.OAuthHandler(OAUTH_API_KEY, OAUTH_API_SECRET)
3         auth.set_access_token(ACCESS_TOKEN, ACCESS_SECRET)
4         self.api = tweepy.API(auth)
5         print "Connection to Twitter API is up."
```

where the *ACCESS TOKEN* and *ACCESS SECRET* are tokens received upon performing registration for Twitter developer access.

After establishing connection, searching was performed by establishing a *Cursor* object, and iterating over the result:

```
    def retrieve_for_dataset(self):
        c = tweepy.Cursor(self.api.search, q=self.query, lang="no")
        results = []
        for tweet in c.items(500):
            results.append(tweet)
        results_list = utils.get_resultsets_text(results)
```

SECTION 4.2

Part-of-Speech Taggers

There exists a few POS-taggers for the Norwegian language, the number of which is dwarfed substantially in comparison to the number of taggers for the English language.

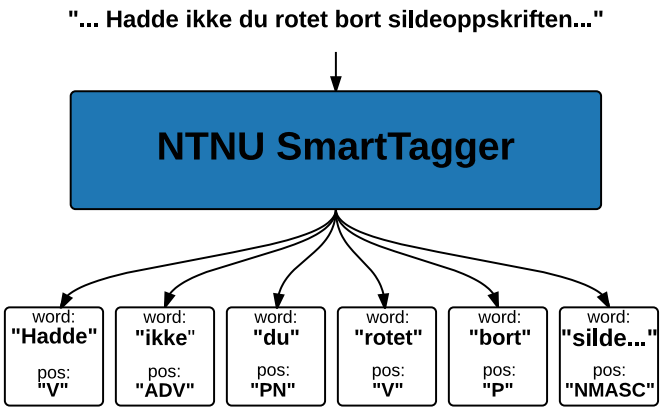


Figure 4.1: An example of a tagged sentence from the NTNU SmartTagger

The tagger used in the classification task in this thesis is the **NTNU SmartTagger**[82]. An example of a tagged sentence from this tagger can be seen in figure 4.1. The tagger is developed at NTNU and the tag set utilised is the TypeCraft tagset, which can be seen in its full in appendix E. In order to perform HTTP requests to use the tagger the python Requests framework was used[83]. Other taggers for Norwegian which were contenders for the use in this thesis are described below.

The **Oslo-Bergen Tagger** is a rule-based tagger originating from a joint project between the University of Oslo(UiO) and the University of Bergen(UiB).

The tagger can show an F1 score of 97.16 [84]. However, any unsolved ambiguities from the output of the tagger is not dealt with. An improvement to deal with these ambiguities was however included in the latest version.

The **Noursource Tagger** was also a contender for use in the classification, which is another tagger developed at NTNU[85].

Recently, a tagger was developed using existing resources originally used for an English tagger, reapplied to develop a tagger for the Norwegian language[86]. Using statistical methods Marco describes a tagger for Norwegian Tweets reaching an accuracy of over 97% for morphosyntactic tagging, and 95.2% for lemmas.

SECTION 4.3
Machine Learning Tools

Several tools exist for performing different machine learning tasks, such as classification, regression, and clustering. The tool used in this thesis was the **Scikit Learn** framework[87]. This is a Python framework built on NumPy[88], SciPy[89], and matplotlib[90]. Scikit Learn supports several classification methods, SVM, NB, MaxEnt, nearest neighbours, and random forest, to name a few. The main reason for choosing to use this framework for the task was the wide range of different classification methods, as well as a well-written and complementary framework documentation. Last but not least this framework is open-source and completely free to use. Several other ML tools were contenders for use, below are some of them described in short.

Google Prediction is Google's collection of cloud-based machine tools[91]. The advantage of this API would be its availability using different platforms, as it is represented as a REST API, which also can allow for powerful asynchronous training. However, the API allows for only 100 predictions per day on the free quota, and requires a monthly subscription fee in order to get more. The number of support classification methods were also unimpressive.

The **Apache Mahout** framework is a ML framework developed through the Apache Foundation, and supports four different classifiers; NB, Hidden Markov Models, MaxEnt, and Random Forest [92]. The main strength of the Mahout framework is the implementation's distributed nature allowing scalability and use for use with large datasets.

Datumbox is an open-source framework written in Java [93]. This framework supports a number of classification methods on par with Scikit Learn, and it's also free to use. Scikit Learn was chosen over this tool due to the author's preference of developing in Python.

Like Scikit Learn, **TextBlob** is also an open-source Python framework[94], with good support for language-specific tasks such as translation, language detection, and POS-tagging. However, the only supported classifiers are NB and Decision Trees.

PyBrain is another open-source Python ML framework[95] which mainly supports learning methods through neural networks.

SECTION 4.4
Sentiment Lexicon

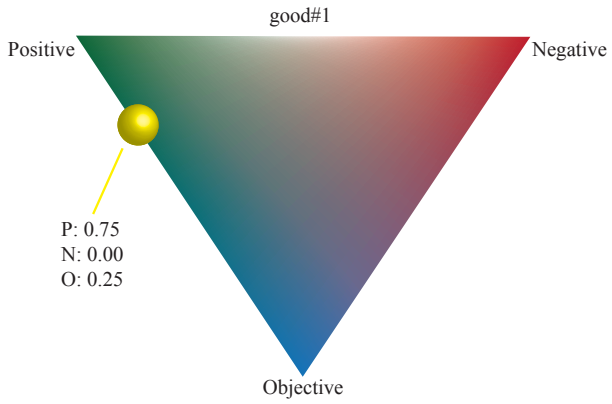


Figure 4.2: An example of an entry in SentiWordNet

SentiWordNet is an open sentiment lexicon[96]. It contains triples of values describing sentiment for over 100 000 entries. Its architecture is based on WordNet[97]. An example of an entry in SentiWordNet can be seen in figure 4.2, where the first entry of the word *"good"* is shown, showing its triple containing values for positivity, negativity, and objectivity.

The Sentiwordnet reader is in this task used to perform look-up in SentiWordNet[98], it builds on The Natural Language Toolkit (NLTK)[99] and the NLTK WordNet implementation, and helps the look-up process in SentiWordNet.

SECTION 4.5
Translation Tools

Bing Translator is one of the translator tool used in this thesis. Microsoft's translation API[100] is cloud-based and supports a wide array of languages. The main reasoning behind choosing this translation tool was that it supports Norwegian to English translations, and that Google Translate API now needs a subscription fee for use.

Google's Translate API is the machine translator used in Google's popular web interface translator[101]. It is represented as a REST API and supports translation between thousands of language pairs. As of December 2011 this API

was no longer free to use and requires a monthly subscription fee to use.[102] However, the web interface can still be used to get translations for free[101], by sending request to the web interface and scraping the web site afterwards. The Google translate web interface is also used in this thesis addition to the Bing Translator.

PART III

Contributions

CHAPTER 5

Datasets

In order to work on different aspects of the classification system, it was found to be a good choice to collect several sets of tweets. A collection of three different sets was compiled in order to have sets with slightly different characteristics. The three different datasets can be seen in figure 5.1. All these sets were collected using the Twitter REST API, with the help from the Tweepy framework[81].

Firstly, a set with a random distribution of tweets was collected. A random set like this was needed in order to be able to train and test the general sentiment classification quality of the system. This set contained 606 tweets, which was split into a training set and testing set with 546 and 60 tweets, respectively.

A set was also collected in order to use for attempting sentiment entity identification. For this reason it was chosen to collect a dataset containing only tweets where the phrase *Erna Solberg* had been used, a phrase which of course refers to the prime minister of Norway Erna Solberg. A total of 662 tweets were collected and preprocessed in this dataset, with a split between training set and test set of 596 and 66 tweets. The idea behind this set was that it would be possible to attempt entity identification, and perform testing of this aspect, and then of course attempting to improve upon this aspect of the system even further.

In addition, it was also necessary to have a dataset which could be used for testing aggregated sentiment values. This dataset needed to have events where the increase or decrease of aggregated sentiment values could be fairly easily predicted, so that the aggregated sentiment values could be test for correctness in some fashion. This dataset was collected only from tweets containing the word *Rosenborg*, referring to the home soccer team in Trondheim. The thought behind this choice was that soccer matches can function as real-life indicators for sentiment increase or decrease, and therefore provide an indicator for evaluating aggregated sentiment values. 578 tweets were collected and preprocessed for this dataset, and split into 521 and 57 tweets for training and testing.

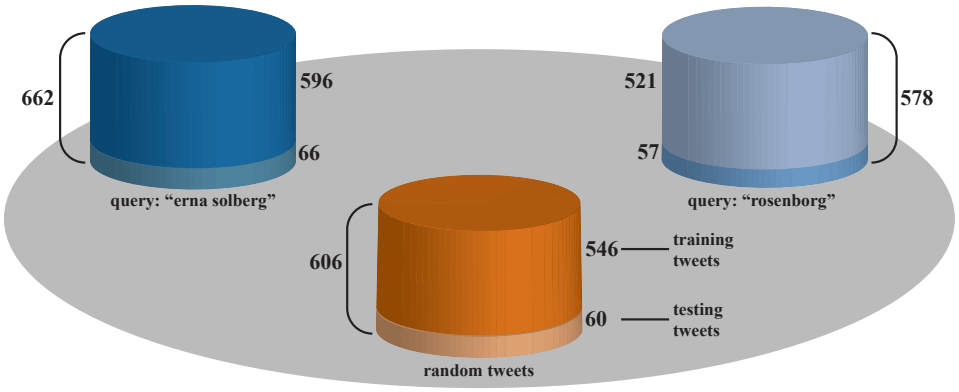


Figure 5.1: Three different datasets were collected

SECTION 5.1

Retrieval

All the datasets used were retrieved with the help of the Tweepy[81] framework. By using the *search()* function provided it was possible to fetch several hundred tweets at a time. However, this search function restrains the result set to a fairly short time period prior to the request, and several requests in a short time period would therefore yield duplicate tweets in the dataset. Therefore, in order to get a larger dataset of non-duplicates, the search request was performed at several times with day-long breaks inbetween each search. Since the *search()* function allows for no random search queries, the dataset containing *random* tweets were collected using several different search queries. Each search query was comprised solely of a random member of the five most frequent Norwegian words: *jeg*, *det*, *er*, *du*, and *ikke*[1]. This was done in order to get a fairly random distribution of tweets in the Norwegian language.

The collection of larger datasets was done by tapping into the Tweepy *stream*. This was done in order to get large datasets comprised of tweets with a higher temporal density for a given time period, to be used for aggregated temporal sentiment visualisations. The stream was accessed continuously for a period of 7 days, storing all the tweets from the stream in this period of time.

[1] n.wiktionary.org/wiki/Wiktionary:Frequency_lists/Norwegian_Bokmål_wordlist

SECTION 5.2

Preprocessing

Before being able to properly analyse textual data, the raw twitter data retrieved needed to be properly preprocessed. Figure 5.2 displays all the stages of preprocessing the datasets went through before classification.

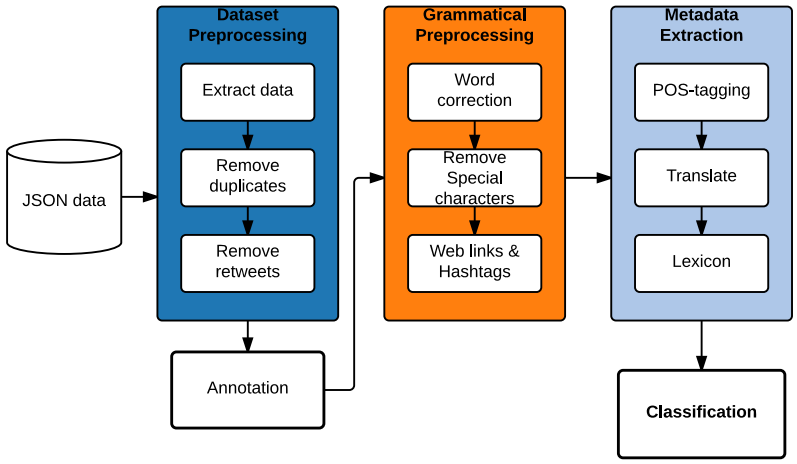


Figure 5.2: Preprocessing pipeline of data before sentiment classification

An initial sequence of preprocessing was performed before annotation, where **retweets** and **duplicates** were removed. After this, all the datasets were manually processed as well. While we do want objective tweets in our datasets - part of the classification task is distinguishing between objective and subjective tweets - in some cases sentiment in tweets is scarce. Therefore, it was found beneficial to manually adjust the datasets so as to contain a higher percentage of subjective tweets, i.e. deleting some objective tweets. One might argue that this off-balance simply reflects the nature of the tweet corpus in its entirety. However, by giving the classifier a more balanced amount of training instances for the classes at hand, we can make sure we have enough instances for each class without needing unreasonably large datasets. Very large datasets can be cumbersome when it comes to the task of manual annotation of the tweets, as is needed for any unsupervised classification task. In order for the annotation task to be feasible for humans to perform, it was necessary to have datasets of a reasonable size.

Word correction was also performed as part of the preprocessing sequence before the annotation of the datasets. It was however a very simple form of correction, consisting of shortening any use of double or more vowels into only one vowel, and shortening uses of triple or more consonants into only two consonants. The former correction would result in any vowel elongation - often used for emphasis, e.g. "Jeg er syyyyyykt lei av ostekake" - being properly changed into its correct form. The latter correction will result in the correction of typographical errors where several consonants were used but only two consonants were supposed to be used. According to Brandwatch[103], Twitter had the most illiterate users of the social networks they looked at, with a 0.56 percentage of words posted on Twitter being incorrectly spelled or deviations from traditional English.

Web links were removed from the tweets, replaced with a link label. This was done since most of the links pasted in tweets are pasted in the shortened *t.co* Twitter link format. While this makes the tweets short and manageable as well as protecting Twitter and it's user from harmful content, it makes the links unintelligible when using text processing and therefore useless as a sentiment target. This means that whenever a *"link"* tag is encountered later in analysis, if it is ever found to be the target of a sentiment it will be considered as an unidentified sentiment target.

Hashtags are often found in tweets, and they can sometimes accurately denote the topic of a tweet. However, hashtags are also widely used informally and indiscriminately, resulting the labels to be generally unreliable when it comes to identifying the sentiment target of the tweet. Hashtags were therefore removed from the tweet texts during preprocessing, however while doing this the hashtags in each tweet were stored and counted in the tweet object, in order for this information to be possibly used as features for classification.

In tweets, users can mention other **users** in their tweets by applying the *"@"* tag. Like the hashtags, these labels also needed removal from the tweet texts, otherwise they would complicate the texts for the POS-tagger. Any word prefaced with the "@" tag were therefore deleted from the texts, but stored in the tweet objects for possible use as features for classification and potential sentiment targets.

Part-of-speech tagging and **look-up in SentiWordNet** was done as a part of preprocessing. With the help ot the POS-tagger, potentially sentiment-bearing words - i.e. adjectives, nouns, verbs, and adverbs - were identified and sent to the translation application. A small C# console application was developed for the purpose of translation, in order to get access to the Microsoft Translator API. The Google Translate web interface was also used by posting requests with text and scraping the page for the results. The English translations of the words were then used to perform look-up in the lexicon, where triples of sentiment values were obtained and stored back with their corresponding Norwegian words.

Table 5.1: The joint probability of agreement for the datasets

	Neutral	**Positive**	**Negative**	**Both**	**Total**
Neutral	521	52	93	0	666
Positive	49	359	15	8	431
Negative	56	8	659	4	727
Both	0	12	0	12	24
Total	636	416	769	24	**1847**

SECTION 5.3
Sentiment Annotation

Since determining sentiment is an inherently subjective task, it was necessary to take measures in order to minimise bias. A total of three people were brought in to help annotation of the datasets, pairing two and two of them together on each dataset. A simple console application was developed in order to collect annotations from the subjects. The subjects were each given a introduction with a textual definition of sentiment, the definition used can be found in appendix D. This way, for each of the three datasets, two different annotation sets where procured.

In order to then evaluate the reliability of the annotated datasets, the *joint probability of agreement* and *Cohen's Kappa* were calculated. Acquiring annotated datasets from independent annotators was done in order to obtain two differing annotation sets. The agreement contingency table can be seen in figure 5.1, showing an overall agreement of 83.9% with a Kappa value of 0.73. This value falls within what Landis and Koch deem "moderate agreement"[104].

The annotation of sentiment targets in the dataset is a less subjective task than the task of annotating the sentiment itself. Therefore this task was not performed using annotation subjects, it was performed solely by the author. This annotation task was done by going through the *rosenborg* dataset, and performing a binary response to whether or not the sentiment in each of the tweets were targeted towards the football team *Rosenborg*. This annotated dataset could then be used to measure the accuracy of the sentiment entity extraction scheme developed in this thesis.

SECTION 5.4
Dataset Analysis

In order to get a better understanding of the nature of the datasets used, a few analyses were performed. Understanding the differences, or lack thereof, could be important when classification is due. In order to gauge the differences between the datasets, several of their statistical traits were analysed. The

average distribution of words per tweet within the four major word classes - Adjectives, Adverbs, Nouns, and Verbs - can be seen in figure 5.3.

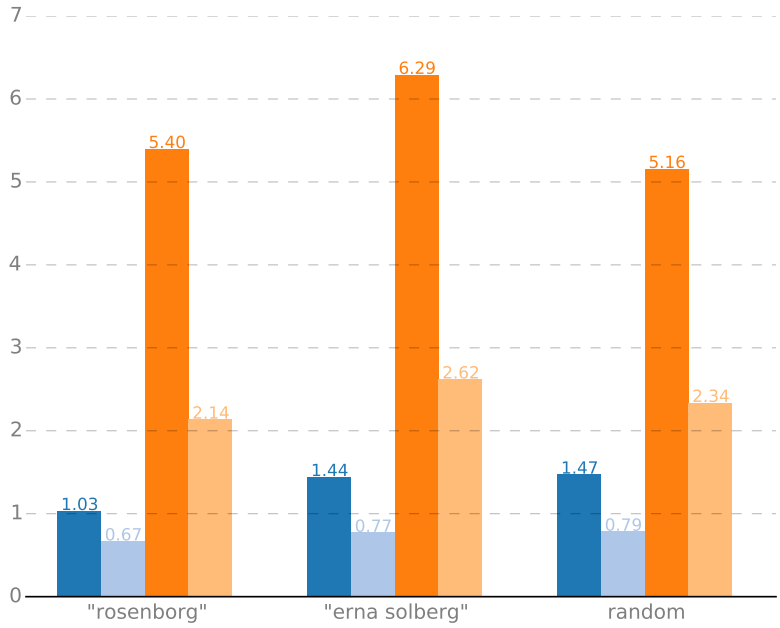


Figure 5.3: Average Adjectives, Adverbs, Nouns, and Verbs per tweet for the datasets

These averages show the similarities between the datasets. Even though they have been retrieved by different means of queries, they still appear to have quite similar distributions within the major word classes. The main difference we can see here between these three datasets is that there is a slightly higher average of nouns and verbs in the *erna solberg* dataset. The reasons for this may be that people who write about or to politicians are more likely to formulate more structurally correct and diverse sentences.

If we look more closely into the finer granularities of the parts-of-speech in figure 5.4, we can see a more detailed picture. Adjectives, adverbs, nouns, and verbs are displayed in the same colour here as the previous diagram, blue, light blue, orange, and light orange, respectively. We can see greater differences in the datasets here when we look at this distribution. The most striking difference is the rightmost bar of nouns for the *rosenborg* dataset. This bar displays the average for proper nouns. The disparity of proper nouns in this dataset compared to the others may be a result of the mention of football teams and football players. Tweets advertising scores in a match can also appear in this dataset - which only display for instance *"Rosenborg - Aalesund 3 - 0"* - could lead to a skewed perspective of the amount of proper nouns per tweet. Other than this the figure shows that the datasets are of quite similar nature.

Several other statistics for the datasets were obtained. A view of general

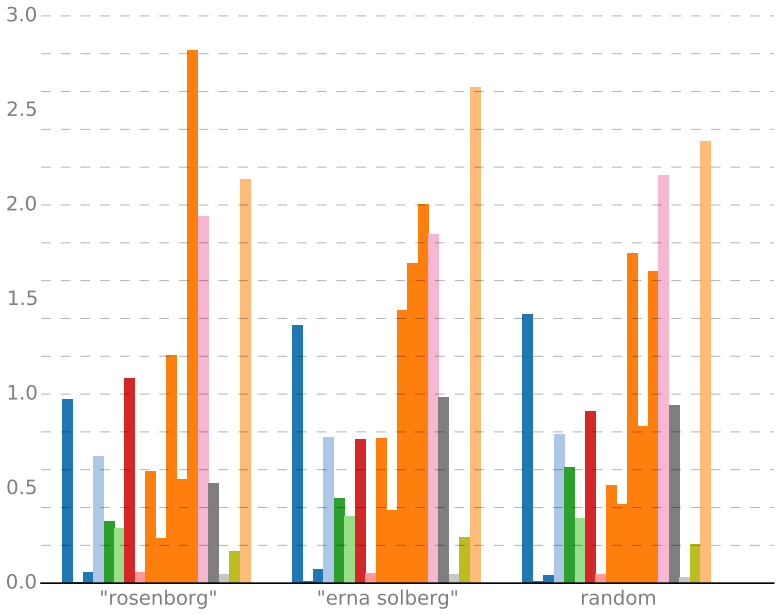


Figure 5.4: Detailed word class averages per tweet for the datasets

statistics of the three datasets an be found in table 5.4, 5.2, and 5.3.

In order to view the difference in characteristics between annotated classes of the tweets, a pairwise comparison of POS-tag distributions in these classes was carried out. The following equation was used to illustrate differences in subjective and objective tweets, as well as negative and positive tweets. We calculate P^T for each POS tag using

$$P^T = \frac{N_1^T - N_2^T}{N_1^T + N_2^T} \tag{5.1}$$

where N_1^T and N_2^T are the number of occurrences of the tag T in the first and second class respectively[11]. Figure 5.5 shows P^T using the subjective class as class 1 and the objective class as class 2.

Most interestingly, we see that proper nouns, determiners, prepositions, and nouns tend to weigh heavier in objective tweets, while subjective tweets seem to have a higher amount of adjectives, adverbs, and interjections. These finding suggest that adjectives, noun, adverb and interjection frequencies can be informative features when it comes to subjectivity classification.

Figure 5.6 shows P^T using the positive tweets class as class 1 and the negative class as class 2.

Here we see in general a heavy weight of words on positive tweets. Comparative adjectives and subordinating conjunctions being the major word tags in positive tweets, while negative tweets apparently consist mainly of interjections and proper nouns. This may suggest that interjections and adjective

Table 5.2: Table of statistics for *random* dataset

Number of tweets	606
Words	9937
Users	532
Words per tweet	16.39
Tweets per user	1.13
Users mentioned	406
Emoticons	22
Negative tweets	80(13.20%)
Neutral tweets	403(66.50%)
Positive tweets	123(20.29%)
Time period	30/9-14 - 28/10-14

Table 5.3: Table of statistics for *rosenborg* dataset

Number of tweets	579
Words	8842
Users	320
Words per tweet	15.27
Tweets per user	1.80
Users mentioned	154
Emoticons	8
Negative tweets	85(14.68%)
Neutral tweets	349(60.27%)
Positive tweets	145(25.04%)
Time period	28/9-14 - 20/10-14

Table 5.4: Table of statistics for *erna solberg* dataset

Number of tweets	662
Words	10974
Users	460
Words per tweet	16.57
Tweets per user	1.43
Users mentioned	1284
Emoticons	9
Negative tweets	245(37.01%)
Neutral tweets	307(46.37%)
Positive tweets	110(16.61%)
Time period	26/9-14 - 26/10-14

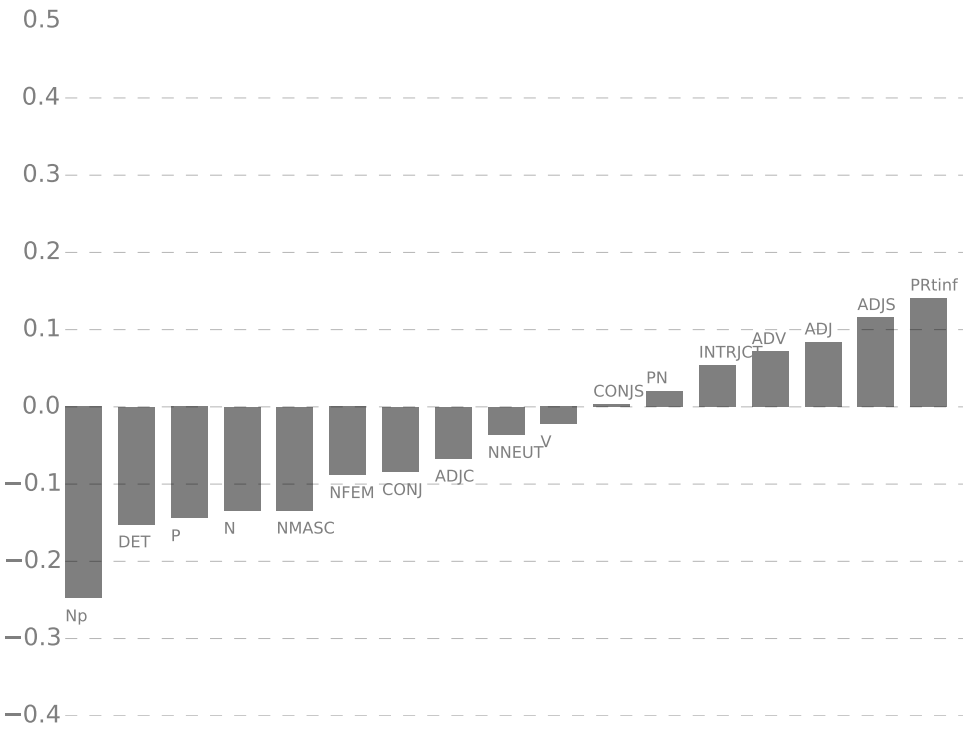


Figure 5.5: POS-tag analysis of objective and subjective tweets

frequencies can be informative features for polarity classification. It seems also to suggest that message length can be informative when it comes to classifying negative and positive tweets.

Some of the phenomena in these frequencies may be explained by the fact that a large part of the dataset consist of tweets regarding a football team. For instance the large frequency of proper nouns in objective tweets in figure 5.5 may be a result of the substantial amount of tweets from sport news outlets in the *rosenborg* dataset, tweeting only to enlighten their followers regarding a new score development in a football match.

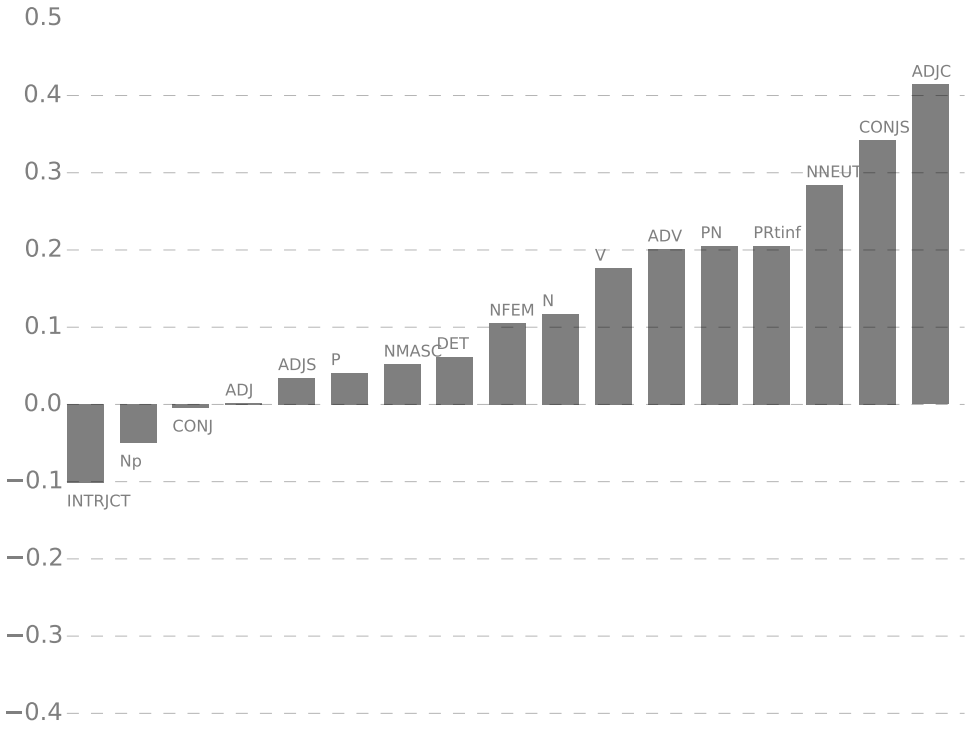


Figure 5.6: POS-tag analysis of negative and positive tweets

CHAPTER 6

Architecture

This chapter will give insight into the architectural overview of the system developed for this thesis. Starting by going into the feature extraction process using the NTNU SmartTagger and SentiWordNet. Then an overview of the general classification process will be given. Finally the part of the system dealing with the sentiment topic detection will be described.

SECTION 6.1

Sentiment Lexicon Feature Extraction

Given that SentiWordNet is an English sentiment lexicon and we are dealing with Norwegian tweets, a translation process was needed in order to get sentiment values from the lexicon. Two different methods of extracting sentiment values was used. The first method using the Bing Translator[100], can be seen in figure 6.1 and the other method using the Google Translate Web Interface[101], can be seen in figure 6.2. The two methods differed in both the tools used for translation as well as the way translation was performed. This section will elaborate into detail how the two different translations were done along with the subsequent lookup in SentiWordNet.

The method using the Bing Translator can be seen in figure 6.1. As can be seen the process of extracting features from a new tweet document starts with the SmartTagger, before a subset of the tagged words are sent to the Bing Translator. This subset contains words which are members of the word classes represented in SentiWordNet - i.e. adjectives, adverbs, nouns, and verbs - and therefore may contain sentiment value. Each word is translated one at a time, then each word is sent to the lexicon handler for lookup in SentiWordNet, where the word tags gotten from the tagger are used in the disambiguation between lexicon entries.

The difference in process when using the Google Translate method can be seen in figure 6.2. Instead of sending tagged and tokenised words to the

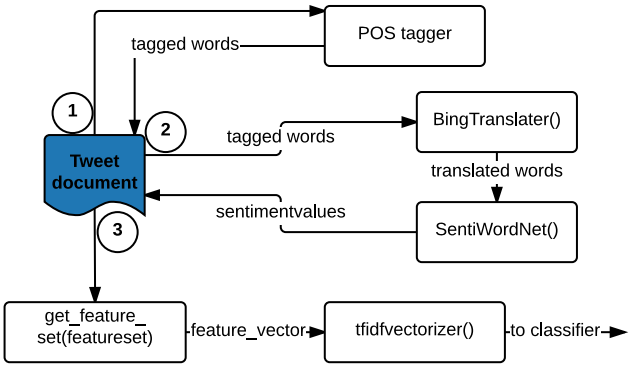


Figure 6.1: The feature extraction process using the Bing Translater

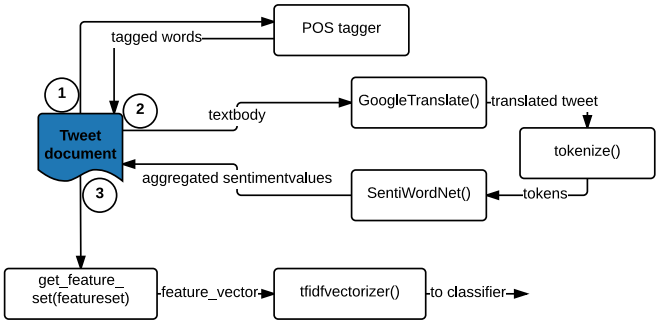


Figure 6.2: The feature extraction process using the Google Translate Web Api

tranlator, the entire tweet textbody is sent. The Google translator translates the entire textbody, and then tokenisation is performed. The tokens are then sent to the lexicon for lookup in SentiWordNet. This means that since no word-specific sentiment values are obtained, only the aggregated total sentiment values - positive, negative, and objective value - for each tweet can be used in classification.

The *get_feature_set(featureset)* handles the extraction and creation of the given feature set, which correspond to one of the strings "SA", "SB", or "SC" for subjectivity classification, and "PA", "PB", or "PC" for polarity classification. These strings correspond to the total of six different feature sets which will be elaborated upon in chapter 7. The feature vector is created and subsequently sent to the *tfidfvectorizer()* which turns the standard feature vector into a tf-idf weighted vector.

The thought behind doing two different methods of translation and lexicon lookup was that it gave the possibility to view how the different methods affected the classification process. The *Bing Translator* method being able to utilise POS tags in order to disambiguate both when translating and when attempting to find the correct entry in the sentiment lexicon. The *Google Translate* method on the other hand, has no such way of disambiguation, however the hope was that it could obtain better translations of the words since the whole context - i.e. the whole tweet text - is translated.

SECTION 6.2
Sentiment Classification

A general overview of the two-step classification process of classifying a new tweet document is shown in 6.3. This process is performed after feature extraction has been performed on the tweet document which means that the classifier receives a tf-idf feature vector representation of a tweet. The tweet is first classified for subjectivity, and subsequently - if it is subjective - classified for polarity.

In chapter 7, the subjectivity classifier and the polarity classifier will be evaluated both separately and combined, using various classifiers and feature sets for the two steps in the classification tasks.

SECTION 6.3
Sentiment Topic Detection

The sentiment topic was detected using metadata from the POS tagger and sentiment lexicon, as well as a crude approach at finding sentiment breaking points using the subjectivity classifier. The method depicted in figure6.4 uses the subjectivity classifier to classify substring combinations of a subjective

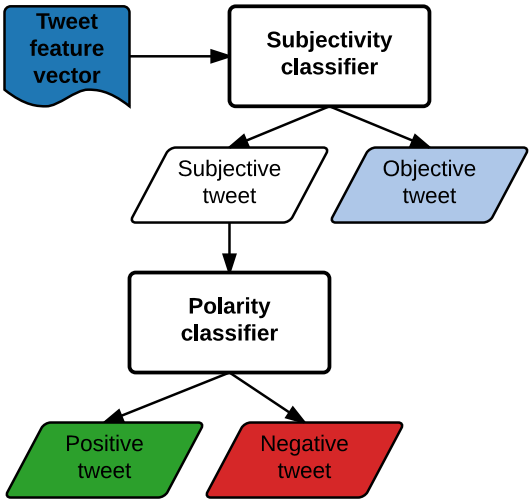


Figure 6.3: The tweet two-step classification process

tweet until it encounters a substring where classification results in the new class having changed from the original. The method then assumes that the lastly removed word from the substring bears some sentimental meaning, as its removal resulted the classifier to go from subjective to objective.

Then, in order to attempt finding the sentiment topic given a correctly classified subjective sentiment, several steps were used. Firstly, all nouns tagged by the POS tagger were regarded as potential topics for the given sentiment in the tweet. Then, any nouns not within a certain vicinity of a sentiment point - either a breaking word or a word with lexical sentiment value - were removed.

Finally, any remaining topics were then disambiguated by ranking them according to their PMI values with the sentiment point, where the topic with the highest PMI value was selected as the target topic for the sentiment.

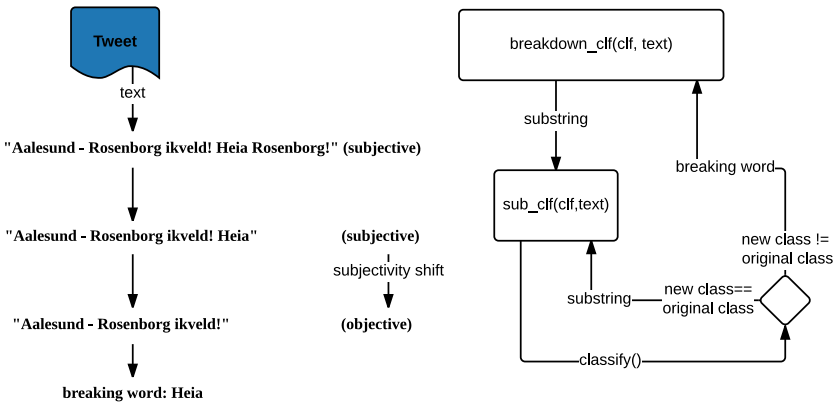


Figure 6.4: Breakdown classification of the texts in order to identify possible sentiment bearing words

CHAPTER 7

Experimental Setup and Results

"Five bananas. Six bananas.
SEVEN BANANAS!"

-Count von Count

This chapter will elaborate on the different experiments performed, and the results from these experiments will be presented. The chapter is divided in four parts; firstly the results from subjectivity classification will be presented, followed by the results from the polarity classification. The combined results of the two classification schemes will then be shown, and lastly an evaluation of the topic detection performance will be done.

SECTION 7.1

Parameter Optimisation

Exhaustive optimisation searches were done for the classification tasks, where several variations upon the input parameters were experimented with. This was done in order to attempt finding the optimal parameter setting for the classification algorithms. The optimal parameters for a classification task depends on the classifier used and the contextual aspects of the datasets on which it is used. Finding the optimal parameters for a task can greatly impact the performance of said task. All the parameter combinations used for optimisation can be seen in table 7.1.

Two of the parameters used were for text vectorisation; the *range of N-grams* used as features, and the *Max document frequency* for using the grams as features. Three parameters were for TF-IDF vectorising; *Use IDF*, *Smooth IDF*, and *Sublinear TF*, all three of them boolean values. Finally, 4 algorithm-specific parameters were used. The *Alpha* parameter of the NB classifier, which is the Laplace/Lidstone smoothing weight. The C parameter in the

Table 7.1: Parameter combinations for optimisation

N-gram range: 1-1 - 1-2 - 1-3 - 2-2 - 3-3
Use IDF: True - False
Smooth IDF: True - False
Sublinear TF: True - False
Max DF: 0.5 - 0.7 - 0.9 - 1.0
Alpha(NB-specific): 0.1 - 0.3 - 0.5 - 0.7 - 0.8 - 1.0
C(SVM-specific): 0.1 - 0.3 - 0.5 - 0.7 - 0.8 - 1.0
C(MaxEnt-specific): 0.1 - 0.3 - 0.5 - 0.7 - 0.8 - 1.0
Penalty(MaxEnt-specific): l1 - l2

Table 7.2: Parameter values with best performance in subjectivity classification

	NB	**SVM**	**MaxEnt**
N-gram range	1-1	1-3	1-2
Use IDF	True	True	True
Smooth IDF	True	True	True
Sublinear TF	False	True	True
Max DF	0.5	0.5	0.5
Alpha	0.3	-	-
C(SVM)	-	0.7	-
C(MaxEnt)	-	-	1.0
Penalty	-	-	l2

Table 7.3: Parameter values with best performance in polarity classification

	NB	**SVM**	**MaxEnt**
N-gram range	1-1	1-1	1-1
Use IDF	True	True	True
Smooth IDF	True	True	True
Sublinear TF	True	True	True
Max DF	0.5	0.5	0.5
Alpha	0.3	-	-
C(SVM)	-	0.7	-
C(MaxEnt)	-	-	0.8
Penalty	-	-	l2

SVM, which influences the margin of the SVM hyperplane. And there is lastly two MaxEnt-specific parameters; the C and the *penalty* parameters.

The parameter sets with the best performances were the ones used in the classification tasks. The resulting best parameters differed for the three classifiers and for the two classification tasks. The parameter values showing the best performances can be seen in tables 7.2 and 7.3, showing the parameter values for subjectivity classification and polarity classification respectively. As can be seen, there are not very big differences in the parameter sets. All three classifiers perform best using IDF and IDF smoothing, and all three classifiers prefer 0.5 as their max Document Frequency value. Most of the classifiers also perform best using only 1-gram text features.

SECTION 7.2

Feature Sets

In total, six different feature sets were experimented with, three for subjectivity classification and three for polarity classification, each of these three utilising increased levels of contextual metadata. Six different denominations are used referring to the different feature sets. For the three subjectivity feature sets: *SA* for *subjectivity set A*, *SB* for *subjectivity set B*, and *SC* for *subjectivity set C*. For the three polarity feature sets: *PA* for *polarity set A*, *PB* for *polarity set B*, and *PC* for *polarity set C*. Feature sets SA and PA, used for subjectivity classification and polarity classification respectively, utilise only word tokens in classification. Feature sets SB and PB have additional features utilising grammatical metadata given using the POS-tagger. Feature sets SC and PC have additional features utilising the sentiment metadata given by lexicon lookup. The feature sets used for subjectivity classification can be seen in table 7.4 and features used in polarity classification can be seen in table 7.5.

SECTION 7.3

Performances

Performance of the classifiers were measured using the 4 previously discussed metrics; the diagrams in this section show the *accuracy*, *precision*, *recall*, and *F1-score*, respectively, for each of three classifiers in the optimised experimental test-runs.

All of different experimental runs in this chapter have been performed with 10-fold cross validation.

Table 7.4: Feature sets for subjectivity classification

Feature set SA
Word Features Word tokens
Feature set SB
Word Features Word tokens, POS-tag occurrences **Sentence Features** Number of exclamation marks, number of emoticons, number of adjectives in sentence, number of adverbs in sentence(except "ikke"), pronoun in sentence(binary), negation in sentence(binary)
Feature set SC
Word Features Word tokens, POS-tag occurrences, subjectivity scores **Sentence Features** Exclamation marks, emoticons, adjectives in sentence, adverbs in sentence(except "ikke"), pronoun in sentence(binary), negation in sentence(binary), total subjectivity score from word polarities, total objectivity score, number of subjective words, number of objective words

Table 7.5: Feature sets for polarity classification

Feature set PA
Word Features Word token
Feature set PB
Word Features Word token, POS-tag occurrences **Sentence Features** Number of happy emoticons, number of sad emoticons, message length
Feature set PC
Word Features Word tokens, POS-tag occurences, polarities **Sentence Features** Number of happy emoticons, number of sad emoticons, message length, total polarity score, number of positive words, number of negative words

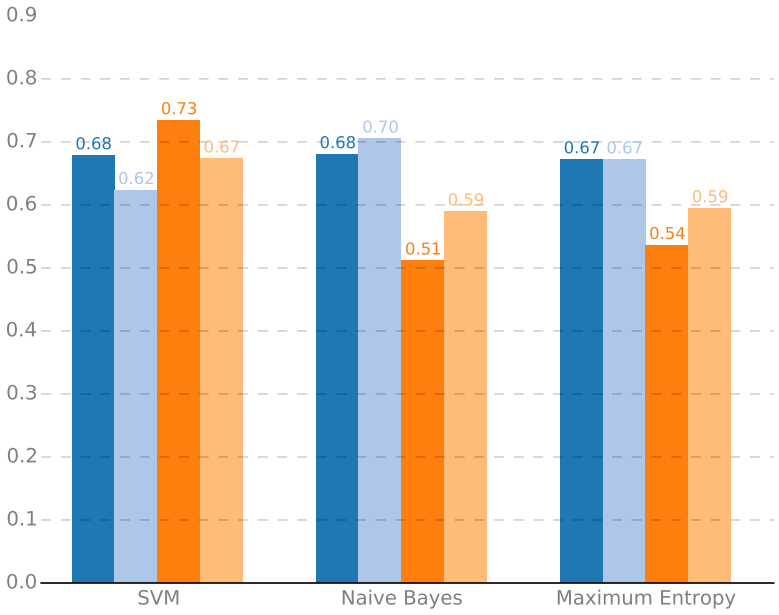


Figure 7.1: Accuracy, Precision, Recall, and F1-Score for feature set *SA*

SUBSECTION 7.3.1

Subjectivity Classification

The performances for subjectivity classification of the three classifiers for feature set SA can be seen in figure 7.1. As previously mentioned, feature set *SA* consists of only word tokens. This is clearly reflected in the performance of the classifiers.

As expected, these results show that the SVM classifier is the clear winner with a feature set consisting of only word tokens. As can be seen in 7.1 - the classification results for feature set *SA* - the SVM classifier outperforms both MaxEnt and NB with nearly 10 points higher F1-score, but with accuracy tied for all three classifiers. The performance is not very surprising considering that among these three SVM is widely recognised as the best text classifier.

The poorer performances of the NB classifier and MaxEnt classifier are most likely due to the lack of contextual features in this feature set. Since feature set *SA* has been stripped away of emoticons, exclamation marks, etc., features that can be important in an informal context like Twitter, it can be hard to get good results.

In figure 7.2 we can see the scores of the classifiers using a richer set of features with feature set *SB*. This feature set includes additional metadata, including data such as exclamation mark counts, number of emoticons, and POS-tag counts using the POS-tagger. The feature set can be seen in detail in table 7.4.

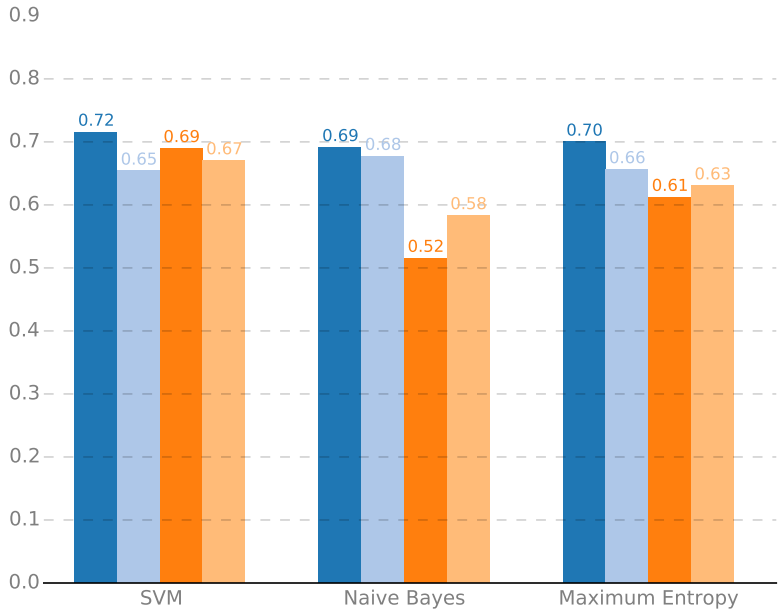


Figure 7.2: Accuracy, Precision, Recall, and F1-Score for feature set *SB*

When compared to the results in figure 7.1, we can see that we now have a MaxEnt classifier which has improved upon its F1-score with four points. It seems that the MaxEnt classifier utilises the extra set of features well. We also see four points of improvement in the accuracy of the SVM classifier, with its other performances unchanged. The NB classifier however shows no change in its performance.

In figure 7.3 and figure 7.4 we can see the perfomances of the classifiers using the full set of features extracted for subjectivity classification - feature set *SC*. The difference from feature set *SB* is that these results are obtained using features including sentiment scores given from using translation and lookup in the SentiWordNet sentiment lexicon.

The performances shown in figure 7.3 were obtained using the *Bing Translator* method as it is described in section 6.1, where translation was performed on single words with subsequent lookup in SentiWordNet in order to obtain sentiment values. These results are surprising, as they show mostly the same results for all classifiers when compared to feature set *SB*, when the expectation - or perhaps the hope - for these results was that we would see an improvement in the classifiers. The only change that can be seen is a slight decrease in the performance of the SVM classifier.

In addition to the *Bing Translator*, a method using *Google Translate* was done, which used translation of entire tweet texts before performing lexicon lookup. The details of this method can be fond in section 6.1. These results are obtained using the same feature set - feature set *PC* - but the sentiment

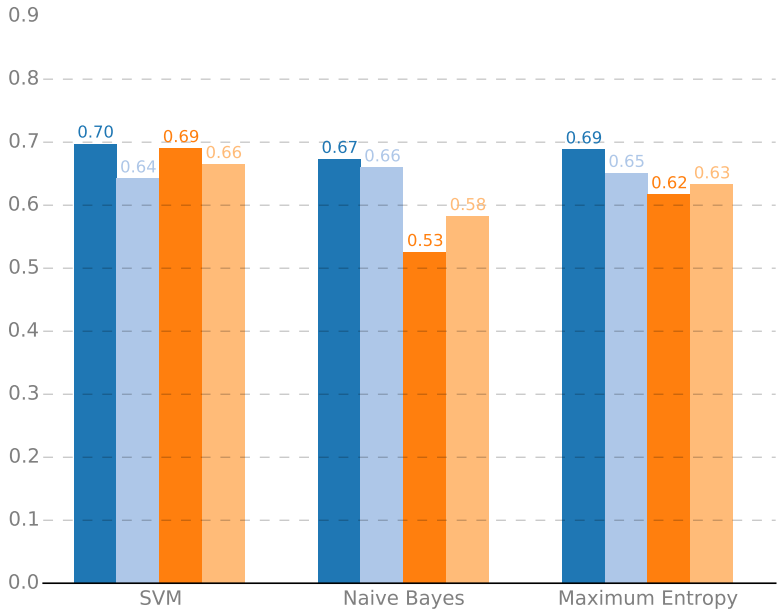


Figure 7.3: Accuracy, Precision, Recall, and F1-Score for feature set *SC* with Bing Translator

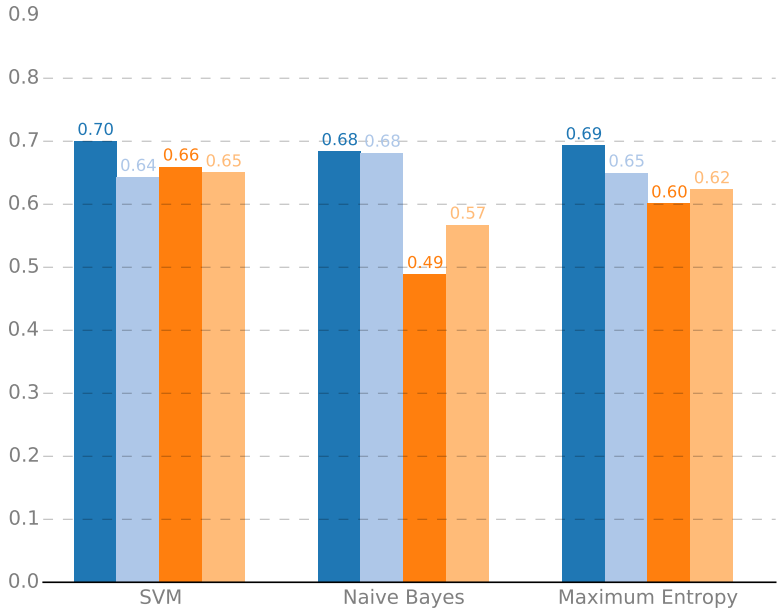


Figure 7.4: Accuracy, Precision, Recall, and F1-Score for feature set *SC* with Google Translate

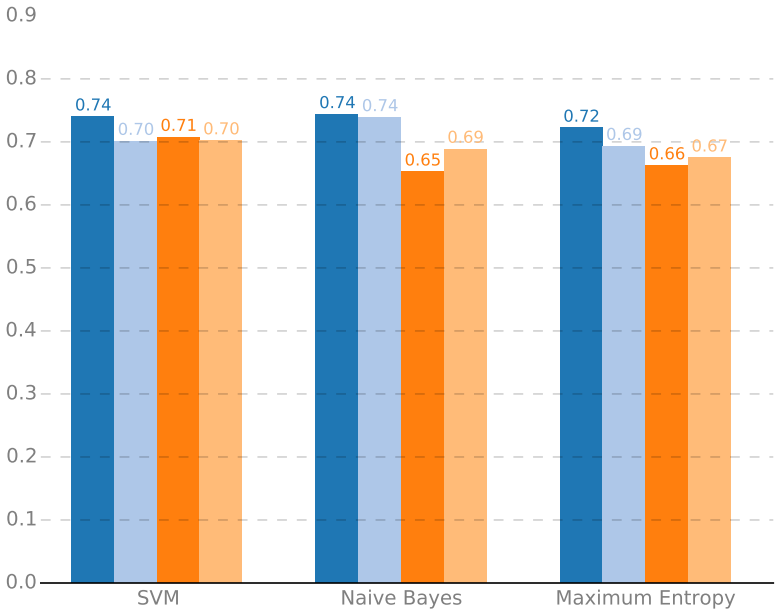


Figure 7.5: Accuracy, Precision, Recall, and F1-Score for feature set *PA*

values obtained may be different as a result of the different translation method. This resulted in the performances that can be seen in figure 7.4. As can be seen, the results are mostly unchanged from the previous feature set. The only difference that is shown is a slight decreases of about 1 point for each of the three classifiers, which is arguable negligible.

SUBSECTION 7.3.2

Polarity Classification

The results in figure 7.5 are from polarity classification using only word tokens. As we can see from the results the SVM and ME classifiers with fairly high accuracies, and moderate F1-scores. The surprising numbers from these results are the ones from the NB classifier, here we that the NB classifier actually performs on par with the SVM and MaxEnt classifiers.

When compared to the subjectivity classification task, we see here that the SVM classifier is no longer the outperforming algorithm it was earlier. When it comes to polarity classification both the NB and MaxEnt classifiers perform on par with the SVM classifier.

In figure 7.6 we can see the scores of the classifiers using a richer set of features with feature set *PB*. This feature set is tailored for polarity classification, containing features that are supposed to give information towards the given tweet being positive or negative. This includes textual information such as the count of happy emoticons and number of sad emoticons. The entire feature set

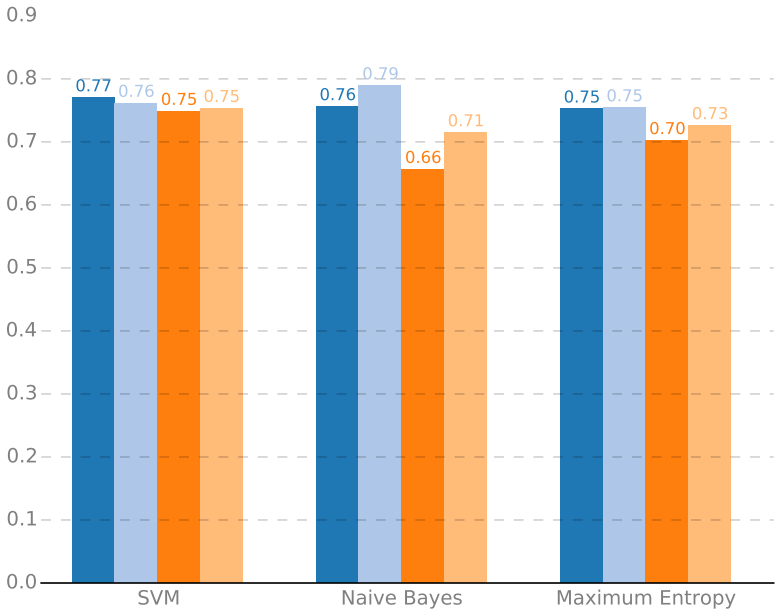


Figure 7.6: Accuracy, Precision, Recall, and F1-Score for feature set *PB*

is described in table 7.5.

When comparing performance with feature set *PA* and performances with feature set *PB* the results show both the SVM and MaxEnt classifiers with higher F1 scores using this richer feature set; MaxEnt gaining six points and SVM gaining five points. The NB classifier also shows a slight increase in overall performance. All three classifiers show approximately a two point increase in accuracy.

In figure 7.7 and figure 7.8 we can see the performances of the classifiers using the richest set of features extracted for classification - feature set *PC*. In addition to the features in the previous sets this set includes sentiment scores from a sentiment lexicon, metadata given using translation and SentiWordNet lookup, such as polarity scores and subjectivity scores.

Figure 7.7 shows feature set *PC* obtained using the *Bing Translator*. In these results we see all three classifiers with either the same or worse results when compared to earlier. The SVM and NB classifiers show both slightly worse accuracies and F1-scores, and the MaxEnt classifier showing a drop of four and five points in accuracy score and F1 score respectively.

The results using feature set *PC* obtained with the *Google Translate* method is shown in 7.8. From these results we can clearly see an increase in all performances. The SVM classifier reaches the highest score for polarity classification nearing 0.80 for both accuracy and F1 score. This is an increase of four and five points for accuracy and F1 score respectively when compared to using the *Bing Translate* method.

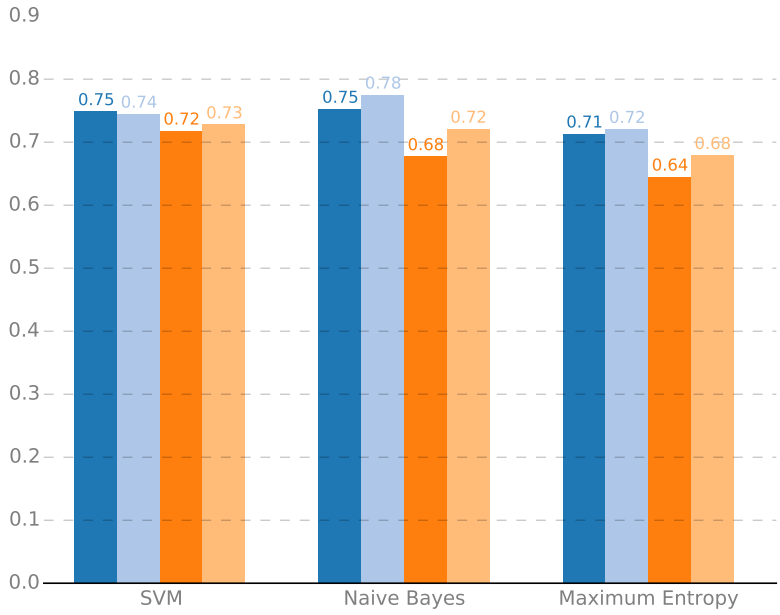


Figure 7.7: Accuracy, Precision, Recall, and F1-Score for feature set *PC* with Bing Translator

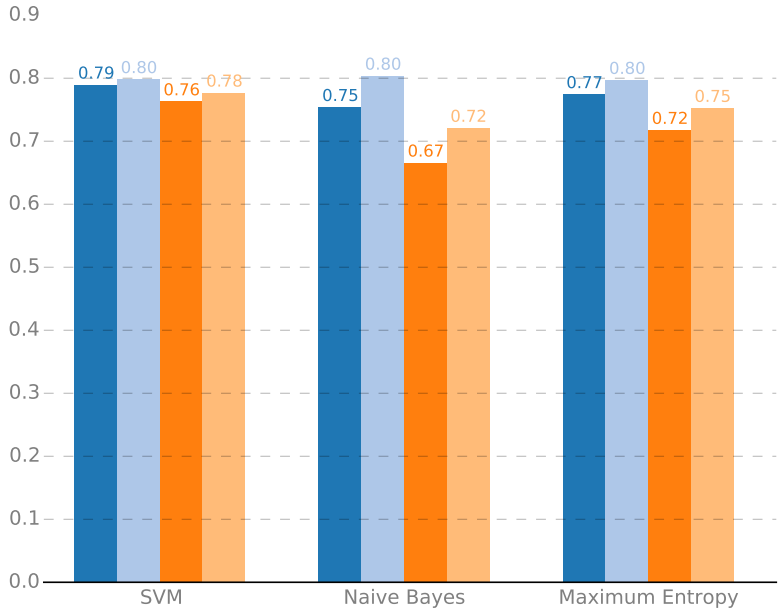


Figure 7.8: Accuracy, Precision, Recall, and F1-Score for feature set *PC* with Google Translate

SECTION 7.4

Combined Performances

The combined performance of the classifier is evaluated by using different combinations of the three classifiers and feature sets for subjectivity and polarity classification tasks. The combined results can be seen in table 7.9. These results show the *accuracy*, *precision*, *recall*, and *F1-score* for 5 different combinations of *Classifier(Feature Set)* chosen on the basis of their performances.

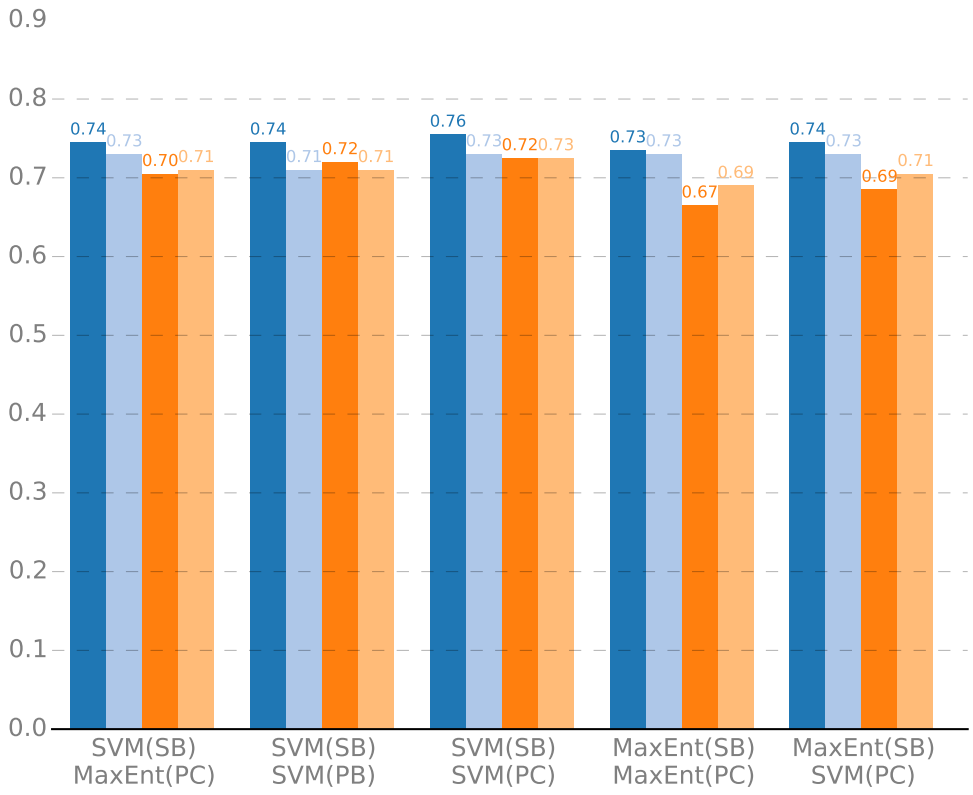


Figure 7.9: The results from the combinations yielding best performances

We can see that three of the combinations yielded the decidedly best performances. The combination of the *SVM* classifier using the *SA* feature set and the *NB* classifier using the *PB* feature set showed the best F1-score with 0.61. Combinations of *SVM* using *PA* and *MaxEnt* using *PB*, and *SVM* using *SA* and *SVM* using *PB*, are tied for the best accuracy on 0.70

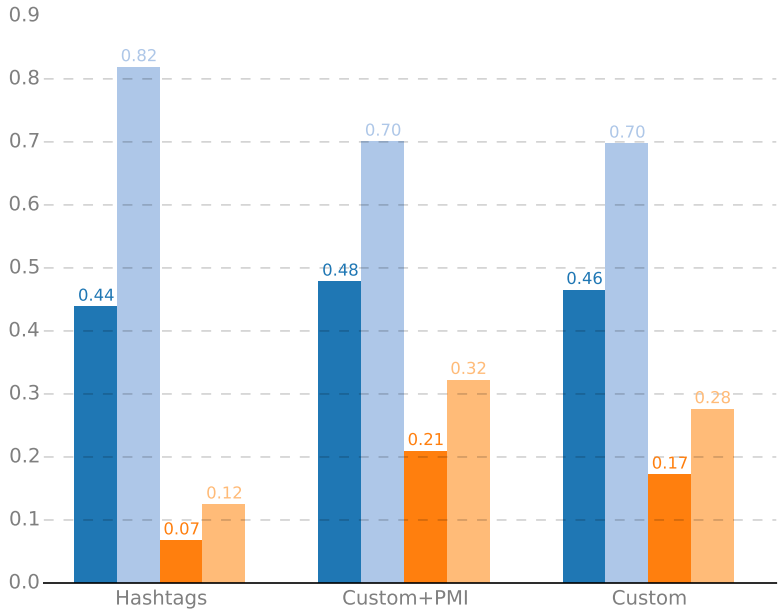


Figure 7.10: Accuracy, Precision, Recall, and F1-Score for topic detection

SECTION 7.5

Sentiment Topic Detection

The performance of the sentiment topic system was evaluated using the annotated subset of the *rosenborg* dataset. This subset consisted of only true positives from the subjectivity classification of this dataset. This was used in order to evaluate the systems ability to correctly identify the sentiment topic given a already correct classification of an existing sentiment. Figure 7.10 show *accuracy*, *precision*, *recall*, and *F1-score* for the hashtag-entities, the *Custom* method described in section 6.3, and the *Custom+PMI* method.

We see that the custom extraction method shows better at both accuracy and F1-score. The hashtag extraction showing a good precision score but an abysmal recall, which can most likely be explained by the often lack of any hashtag at all in a tweet. The high precision shows however, that hashtags are relatively often true sentiment topics when used, but they miss out on several tweets which also may include sentiment towards it.

The *Custom+PMI* is an attempt to disambiguate remaining possible topics using PMI, after using the *Custom* elimination method. We can see a slight improvement in both accuracy and F1 score using this method for sentiment topic detection.

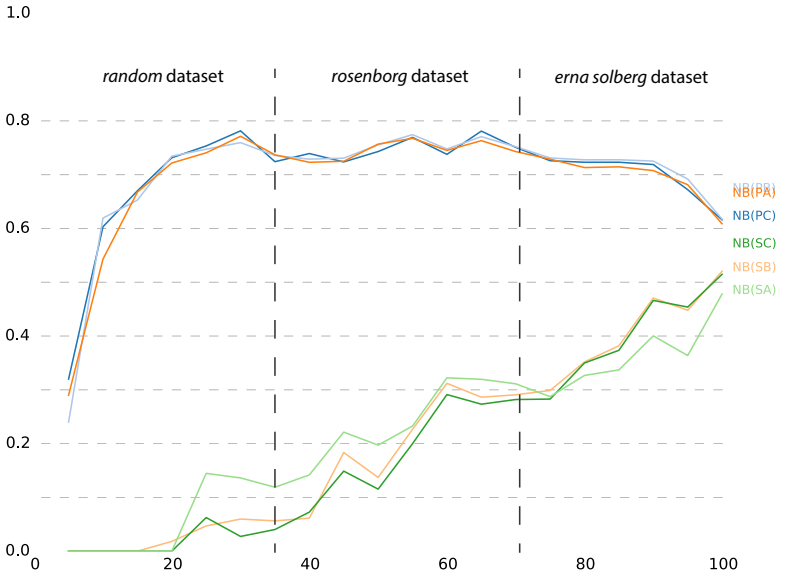


Figure 7.11: F1-scores scores for the incremental dataset-size runs for NB

SECTION 7.6

Incremental Dataset-Size Analysis

In order to view how performances of the different algorithms and feature sets relate to the size of the dataset used, an analysis was done of their performances using incremental portions of the acquired dataset. These results were obtained by starting with 5% of the dataset, performing full analysis with all algorithms and feature sets, storing the results, and then incrementing the size of the dataset to 10% and performing analysis again, storing the results, and so on. This was done with an incremental of 5 percentage points for a total of 20 runs for each algorithm and feature set combination.

F1 scores from the runs can be seen in figures 7.11, 7.12, and 7.13 for NB, SVM, and MaxEnt, respectively. Since three different datasets were used in classification, the datasets were sequentially added in the order seen in the figures. Starting with the *random* dataset, then after about 1/3 into the increments it starts using the *rosenborg* dataset, and then at 2/3 increments it starts using the *erna solberg* dataset. The performances are presented for each of the three classifiers combined with all the six different feature sets. The performance of all classifiers in polarity classification can be seen rising for the first thirds of the increments, followed by a general decline in polarity performance. This can be seen in the polarity performance for all three classifiers. As can also be seen in the figures, this decline starts at approximately the same time as the start of each of the two additional datasets - the *rosenborg* dataset

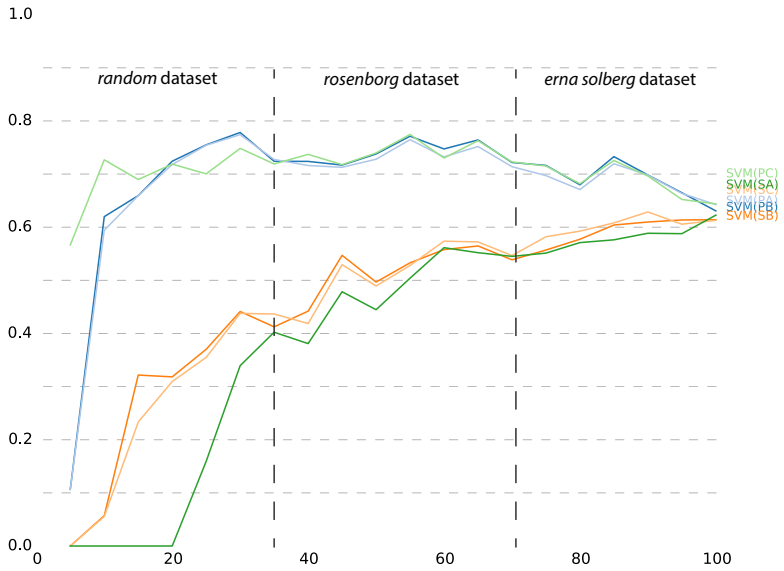


Figure 7.12: F1 scores for the incremental dataset-size runs for SVM

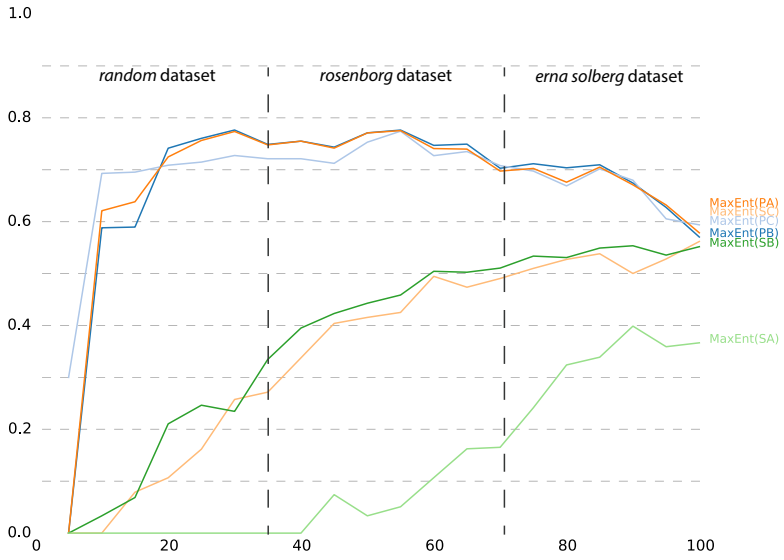


Figure 7.13: F1-scores scores for the incremental dataset-size runs for MaxEnt

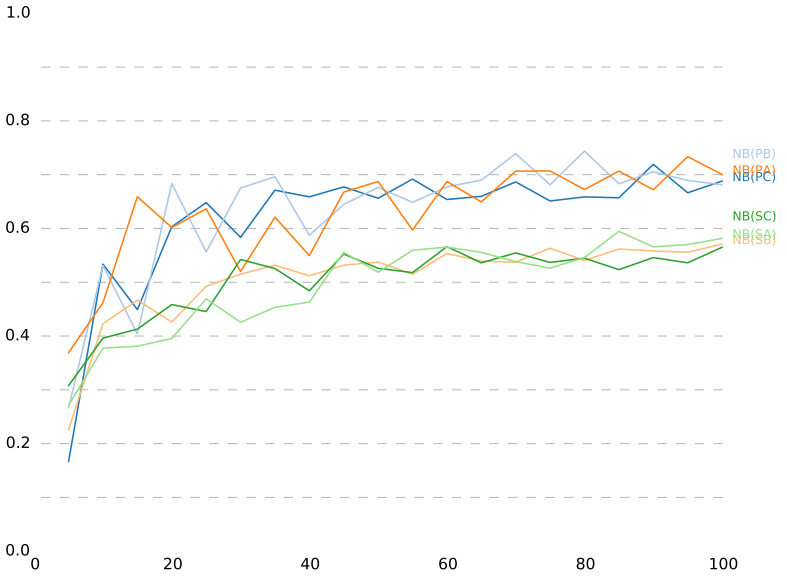


Figure 7.14: F1-scores scores for the incremental dataset-size runs for NB

and the *erna solberg* dataset.

Results from similar incremental runs can be seen in figures 7.14, 7.15, and 7.16, for NB, SVM, and MaxEnt, respectively. These results were however obtained by shuffling the three different datasets together. When comparing it to the previous incrementation runs we see a more general increase throughout the entire incrementation process. We see that even close to the end, where the nearly the entire size of the dataset is used, the performance is still increasing at a steady rate.

In addition to F1 scores from these incremental runs, accuracy scores were also measured. The accuracy scores can be seen in appendix F where they can be viewed in detail if it pleases the reader. These results show mainly similar scores for all combinations, and show a stable accuracy score for all increments of the dataset.

SECTION 7.7

Aggregated Sentiments

Temporal aggregations of both subjectivity scores and polarity scores were visualised in order to see how the aggregated sentiment values behaved. Topically aggregating sentiment values in a temporal perspective allow the comparison of these values and their changes in time to real world events. One can view the general response to certain events by viewing an increase or decrease of values on a given topic for a specific date.

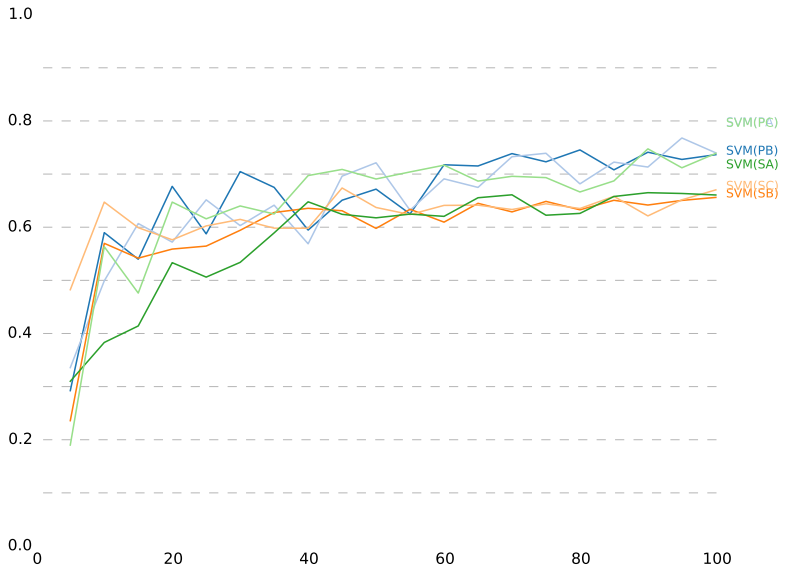


Figure 7.15: F1 scores for the incremental dataset-size runs for SVM

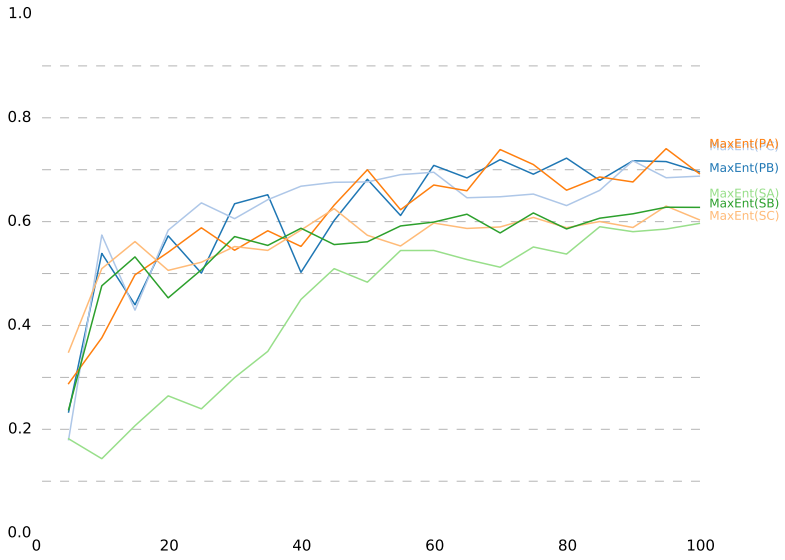


Figure 7.16: F1-scores scores for the incremental dataset-size runs for MaxEnt

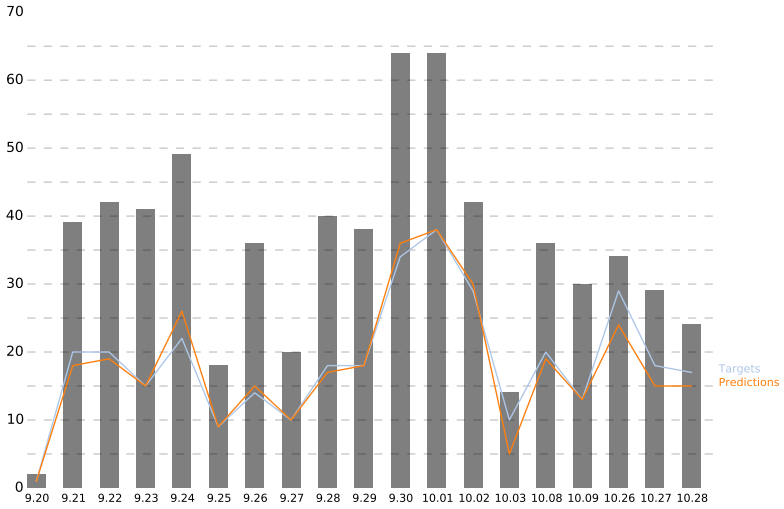


Figure 7.17: Aggregated subjectivity targets and predictions

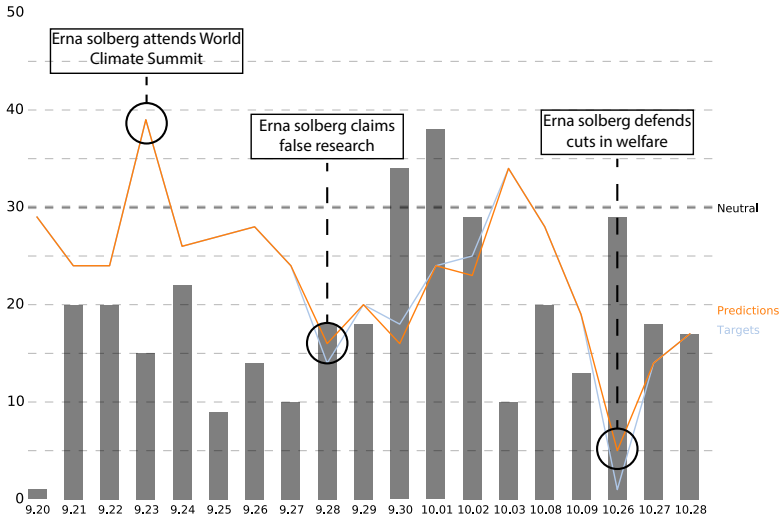


Figure 7.18: Aggregated polarity targets and predictions

Figure 7.17 shows the aggregated subjectivity for the *erna solberg* dataset in a time period spanning 19 non-consecutive days from 20th of September to 28th of October 2014. I.e. the figure shows the aggregation of correct targets and the actual predictions of the SVM subjectivity classifier. The greys bars show the actual tweet frequency per day. The aggregated targets and predictions values have not been weighted, therefore the values presented are the frequencies of sentimental tweets each day.

In figure 7.18 we can see the polarity difference from the same dataset and during the same time period as the previous figure. This figure shows the aggregated difference between all positive tweets and all negative tweets per day. The dashed line at the 30 point mark representing a positive-negative difference of zero. In addition this figure displays three events which were empirically evaluated to be associated with the three extremes - one top and two bottoms - that can be seen in the figure. The top point is associated with Erna Solberg attending the World Climate Summit 2014[1] which in general seemed to yield positivity among tweeters in anticipation of the outcome of the meeting. The first bottom is associated with a news event of Erna Solberg having falsely, or erroneously, used supporting arguments from research which, as it showed, did not exist[2]. The last and lowest bottom is associated with Erna Solberg defending proposed cuts to the Norwegian welfare system, which garnered negative remarks on Twitter[3].

[1] www.norad.no/no/aktuelt/nyheter/klimatoppmøte-i-new-york-et-steg-på-veien-til-paris-2015

[2] www.vg.no/nyheter/innenriks/solberg-regjeringen/erna-skroet-paa-seg-stoette-fra-professor-forstaar-ikke-hvor-hun-tar-det-fra/a/23323936/

[3] www.aftenposten.no/nyheter/iriks/politikk/Dette-er-tallene-Erna-Solberg-forsvarer-uforekuttene-med-7754971.html

PART IV

Conclusions

CHAPTER 8

Discussion

This chapter will elaborate on the research questions presented in section 1.3 with respect to the contributions of this thesis, and discuss whether or not the goals of this thesis have been met. Several topics regarding different aspects of the contributions will also be discussed. At the end of this chapter, several critiques of this thesis and its contributions will be presented and discussed.

SECTION 8.1

Summary of Work

The main research questions of this thesis as presented in section 1.3 were as follows:

RQ1 How can we proficiently extract sentiment from Norwegian Twitter messages?

RQ2 How can we accurately find the topics towards which the sentiment is directed?

RQ3 Can Norwegian-English translation be used in order to apply an English sentiment lexicon for augmenting sentiment analysis on Norwegian tweets?

From these question I derived three goals:

G1 Perform a review into the theoretical components and the state of the art.

G2 Create a sentiment classification engine with topic detection for Norwegian tweets using cross-lingual sentiment lexicon lookup.

G3 Perform an evaluation of the sentiment system and it's components.

In order to complete this task - in accordance with goal **G1** - and to develop the needed system I performed an elaborate review into the theoretical components needed for this thesis in chapter 2, where I presented theory on everything from different aspects of Twitter to the details of word classes and idioms. In chapter 3 I presented an encompassing review into the state of the art, giving good topical coverage of the currents of the relevant academic fields.

In response to goals **G2** and **G3**, I developed a batch system using Python for retrieval and analysis of Twitter corpora, classification of the sentiment in tweets, and evaluation of the algorithms and components used. For these contributions the motivations were - as explained in section 1.1 - the relatively large amount of sentiment presented in an informal context such as Twitter, and the potential value of such sentiment information for several various venues of business and academia, especially if such information can be connected to specific entities with some measure of certainty. The system uses translation tools in order to perform cross-lingual sentiment lookup in SentiWordNet, in an attempt for this to increase the performance of the classifiers.

The various performances and evaluations - in order to meet goal **G3** - of the system and its components were visualised and presented in chapter 7. I performed a parameter search in order to find optimised parameters for each of the algorithms used, and presented performances of the algorithms using different feature sets.

The component performing **cross-lingual sentiment lookup** was developed in order to meet part of goal **G2**. This component used two different ways of translation with two different translation tools. The performances of both can be seen in section 7.3. Performance evaluation was done in accordance with goal **G3**. The results of feature sets *SC* and *PC* show results using features obtained from performing the lexicon lookup, on subjectivity classification and polarity classification, respectively. A total of six different **feature sets** were devised in order to view differences in performances for different sources of information for classification. In general, they show reduced performances for the feature sets using sentiment lexicon features. The reason behind this performance reduction may be that both of the tools used in the lexicon lookup - the NTNU SmartTagger and the SentiWordNet lexicon - are both tools created with the intent of usage on formal corpora. The NTNU SmartTagger has been trained on a formal corpus, and SentiWordNet contains word entries obtained in formal contexts. This could result in low accuracy POS-tags, which then in turn are used for lookup and disambiguation in the lexicon. The information gained from these two steps may be lending more in the way of noise to the classifiers than they are helping the process. The reason behind the performance decrease can also simply be the results of a poor feature set creation. Performing translations of entire tweets however, instead of single words at a time, showed to increase polarity classification performance by a good amount. This is most likely due to the much higher accuracy of translation methods when context is included.

In section 7.6 I presented results showing how the performances of different algorithms and feature sets changed with an incremental change in size for the dataset. Results were also shown using a sequential incrementation of the three different datasets as well as when incrementing a shuffled version of the datasets. The differences in the results show that when incrementing sequentially, the performances of the classifiers decrease when encountering the new datasets. This decrease is probably due to the difference in context between random tweets, tweets of a political nature, and tweets regarding a football team. When the datasets were shuffled we saw a general increase along the whole incremental process, even towards the end. This may show that a larger dataset may contribute towards improved results in some of the algorithms and feature sets.

Part of the developed system was the component designed to identify **sentiment topics**, with the aid of metadata from the NTNU SmartTagger and the SentiWordNet lexicon. Results from experiments with this component were presented in section 7.5. I developed a simple custom system of topic detection, using grammatical metadata and simple similarity measures and augmented with PMI methods to disambiguate between any remaining potential topics. Results showed it to outperform selecting hashtags as sentiment topics.

SECTION 8.2
Criticism

There are several points of this thesis and the developed system which are worthy of scrutiny. I will delve into a few of them below.

Firstly, the developed system is not a running application, it is only presented as a batch system. And as such it has little practical value besides the value of academic results. This also means that any visualisation graphics presented are static and do not display any dynamic change in information.

In addition to this the classifiers are trained on data from a fixed point in time in a context that is constantly changing, which means that the results of the classifiers will diminish over time. This in and of itself greatly reduces the practical value of the system. However it also somewhat diminishes the use in a continuous system. Even if I were to develop a continuous sentiment analysis system, it would diminish over time unless the algorithms were retrained regularly, which is time-consuming considering the need to perform manual annotation for supervised learning.

I have no performance results on the NTNU SmartTagger. I have little background knowledge in general of this tagger, except that it was developed at NTNU and is trained on formal corpora. Neither have I done performance testing of it myself. In order to draw more solid conclusions regarding the performance results, a performance test of the tagger should have been done.

CHAPTER 9

Conclusions

In this thesis I have tested three different machine learning classifiers: Naive Bayes, Maximum Entropy, and Support Vector Machines. From the experiments I can conclude that of these three SVM is the one showing best performance in general, as can be seen in section 7.3. SVM appears to perform well even with relatively poor information. MaxEnt showed promising increases in results for subjectivity classification when more informative features were added. This leads me to predicting high performances with MaxEnt if better feature sets were to be constructed.

Context is an important factor when classifying in an informal context. We see that while MaxEnt showed improved results from using POS features, the actual performances using this metadata did not meet the expected performances. This may be attributable to the context of the POS-tagger. Since there are few Norwegian POS-taggers, getting a tagger for the social media context would mean creating it myself, which is out of the scope of this thesis. This task is however suggested and elaborated upon in chapter 10,

The results presented regarding hashtags for topic detection leads me to tentative conclusions when it comes to hashtags and their reliability for topic markers in regards to sentiment. Several earlier systems have utilised hashtags as topic markers, and shown aggregated results of sentiment values. I believe an assumption of hashtag unreliability is necessary if one is to use such a method, since clearly the hashtags of a tweet are not necessarily reliable to being the target of a given sentiment expressed in the tweet.

As discussed in chapter 8, the results from the incremental runs show the importance of context in a machine learning classifier. We saw a decline in performances when the additional datasets were introduced, i.e. we see that the change in context from tweets concerning politics to tweets concerning a football team has an negative impact on classifier performance. I would argue that considering context on several levels of granularity - not only the informal-versus formal context level - can be useful when performing machine learning text classification. The results also lead me to conclude that performances

could have been increased by obtaining a larger dataset, as well as by obtaining a more randomised dataset.

The improvement in the results from feature set *PA* to *PB* shown in section 7.3.2 suggests that while emoticons are scarce they can be useful for polarity classification. The dataset analysis performed in chapter 5 showed an emoticon count of 39 in a total of 1847 tweets. In addition to this, POS tags such as interjections and comparative adjectives can be important for polarity classification, as was suggested by the dataset analysis performed in section 5.4.

The large improvement when translating entire tweet texts instead of single words show the importance of contextual translation. The results in section 7.3.2 show a good increase in performance when using feature set *PC* through *Google Translate*, when compared to single-word method using *Bing Translator*. This should not however reflect a performance difference between the two tools, as the difference in performance is most likely due to the method and not the tools used. These results also show the inability for this cross-lingual method in aiding with sentiment topic detection. Since word by word translation is too inaccurate it is hard to get word-specific sentiment values when using a sentiment lexicon in a different language.

The accuracy scores presented in appendix F of the incremental dataset runs shows how accuracy in itself is an unreliable measure of performance of a classifier. Given a smaller test dataset, accuracy may show deceivingly high values.

CHAPTER 10

Further Work

> "We can only see a short distance ahead, but we can see plenty that needs to be done."
>
> -Alan Turing

This chapter will identify and elaborate on several venues of further work. The number of actual venues for improvement are most likely tenfold or more, this chapter identifies some of them. Firstly the applicability of the system will be explored. Then the venues of further improvement will be described.

SECTION 10.1
Applicability

While the system in itself does not have much practical applicability, I can see several venues for using the mechanics explored in this thesis.

A **news recommendation** system can be built around creating a user sentiment profile regarding news entities. If augmented with proper entity modelling and identification with a focus on news entities such a system can be able to accurately recommend news articles based on a user's statements in Twitter messages. E.g. recommending news articles concerning Prime Minister Erna Solberg if a user has several positive remarks regarding her in their tweet history. If combined with a news sentiment system, recommendation can be made based on whether the sentiment in articles correspond to the sentiment of the user, keeping the user blissfully ignorant believing everybody thinks the same as they do.

If integrated into the Twitter web application, a sentiment system could be used in order to make more accurate **following recommendations** for twitter users. An integrated sentiment tool scanning users for sentiment on

different topics can create sentiment profiles, and perform recommendations for a user to follow other users which have the same sentiment profiles.

Detecting whether a source is overly enthusiastic regarding their own opinions and are **spamming** others with it can be useful in order to moderate output from such sources.

Using temporal perspectives of aggregated sentiment one can identify trends in users as **signs of depression**. Tracking the general mood of a user can lead to insight into the development of the users state of mind, and thus help with psychoanalysis or therapy.

SECTION 10.2
Venues of Improvement

There are several ways in which one could improve upon the system in this thesis.

SUBSECTION 10.2.1
Creating A Norwegian Sentiment Lexicon

Using manual annotation aided with automated methods, most popular of which is the dictionary method and corpus method, one can create an open lexicon of sentiment values for the Norwegian language. Given the interest in and this thesis focus on social media, the lexicon could contain several features specifically collected for the social media context. Such a lexicon could probably significantly improve performances especially in subjectivity classification in the social media context.

SUBSECTION 10.2.2
Contextual Part-Of-Speech Tagger for Norwegian

As previously mentioned, context is important when performing POS-tagging in a corpus from an informal source such as a social media site. Creating a contextual POS-tagger trained on corpora from social media sites could lead to higher performance for tasks such as tweet sentiment classification. Contextual taggers have shown improves results reaching 90% accuracy. Having a good proper noun disambiguation in a tagger would also be a great value for efficient and accurate topic detection.

APPENDIX A

Acronyms

AI Artificial Intelligence

AMT Amazon Mechanical Turk

ANN Artificial Neural Network

ARM Association Rule Mining

API Application Programming Interface

EWGA Entropy Weighted Genetic Algorithm

HL Hierarchical Learning

IR Information Retrieval

JSON JavaScript Object Notation

K-NN K-Nearest Neighbour

LDA Latent Dirichlet Allocation

MaxEnt Maximum Entropy

MC Minimal Cuts

MPQA Multi-perspective Question Answering Opinion Corpus[27]

MIC Maximum Information Coefficient

MT Machine Translation

NB Naive Bayes

NLP Natural Language Processing

NTNU Norwegian University of Science and Technology

OM Opinion Mining

PMI Pointwise Mutual Information

POS Part-of-Speech

REST Representational State Transfer

RQ Research Question

RT Re-tweet

SA Sentiment Analysis

SMO Sequential Minimal Optimization

SOT Sentiment Ontology Tree

SVM Support Vector Machines

TWA Tweets With Attitude

US United States

WBSN Web-Based Social Network

APPENDIX B

Glossary

Blog Truncation for *Weblog*.

Emoticon Short for *emotion icon*. A pictorial representation of a facial expression, often written in text. E.g. ":)".

Hashtag A form of topic labels used in Twitter messages.

Meme An idea or behaviour that spreads from person to person within a culture.

Microblog Broadcast medium where users express themselves in the form of small elements of content. E.g. Twitter.

Tweet Short for *Tweet messages*. A unit of text with metadata posted on Twitter[105]. These messages have a maximum length of 140 characters.

REST API A software architecture style for creating scalable web services.

Retweet A repost of a Tweet, often sent by one user to confirm, debunk, or share the opinion of the original Tweet. Retweets are marked in Twitter with the "RT" label.

Weblog Informal discussion site published on the World Wide Web.

APPENDIX C

User Manual

This chapter will give an introduction into usage of the batch system implemented in this thesis. The different processes in this batch system can be used to replicate the results presented in chapter 7.

SECTION C.1
Prerequisites

The batch system builds upon several other existing frameworks. The following frameworks are needed in order to run the system:

NumPy NumPy is a fundamental package for scientific computing with Python[88]. Files for download can be found here[1]. Can also be installed using *pip install numpy*.

SciPy Scipy is a collection of tools for mathematics, science, and engineering[89]. Files for download can be found here[2]. Can also be install using *pip install scipy*.

Scikit Learn Scikit Learn is an open source machine learning framework supporting several machine learning algorithms[87]. Download and instructions can be found here[3]. Can also be installed using *pip install scikit-learn*.

SECTION C.2
Using the System

The system is interfaced using the *classifier.py* module along with the terminal commands presented in figure C.1.

[1] sourceforge.net/projects/numpy/files/
[2] sourceforge.net/projects/scipy/files/
[3] scikit-learn.org/stable/install.html

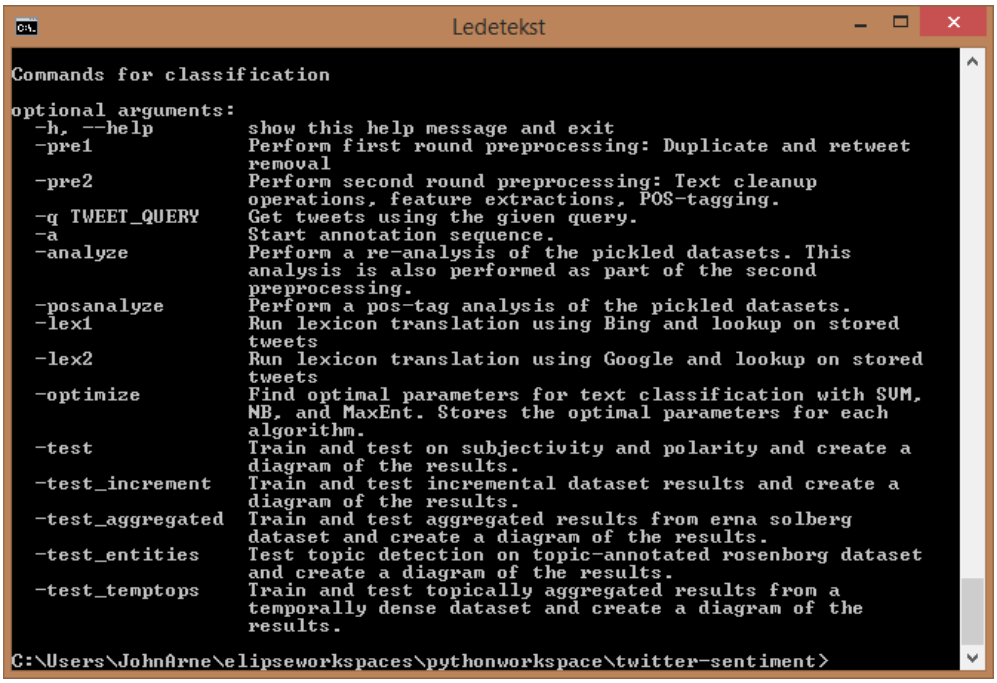


Figure C.1: Command arguments for *classifier.py*

Preprocessing can be done using the *-pre1* and *-pre2* commands. These commands perform preprocessing on the three datasets. The former command is intended for use before sentiment annotation, the latter for use after sentiment annotation. The annotation process can be startet with the *-a* command.

The commands *-analyse* and *-posanalyse* perform the analysis processes used to generate the results presented in chapter 5.

In order to perform translation and sentiment lexicon lookup, the commands *-lex1* and *-lex2* can be used to initiate the processes using *Bing Translator* or *Google Translate*, respectively.

The various *-test* commands can be used in order to produce various results presented in chapter 7.

APPENDIX D

Definition of Sentiment

The following is the definition of sentiment by the Merriam-Webster dictionary [106]. This definition was used in order to introduce and explain sentiment for subjects who were to perform annotation of sentiment on the datasets.

SECTION D.1
Full Definition of SENTIMENT

a: an attitude, thought, or judgement prompted by feeling.

b: a specific view or notion.

SECTION D.2
Examples of SENTIMENT

- His criticism of the court's decision expresses a **sentiment** that is shared by many people.
- An expression of antiwar **sentiments**.
- She likes warmth and **sentiment** in a movie.
- You have to be tough to succeed in the business world. There's no room for **sentiment**.

APPENDIX E

TypeCraft POS-Tagset

The following tables lists the entire Part-of-Speech tagset used in the NTNU SmartTagger. This tagset is developed as part of TypeCraft; a multi-lingual online database of linguistically annotated natural language text[107].

POS tag	**Tag description**
ADJ	adjective
ADJC	comparative adjective
ADJS	superlative adjective
ADV	adverb
ADVm	manner adverb
ADVneg	negative operator
ADVplc	place adverb
ADVtemp	temporal adverb
ART	article
AUX	auxiliary
CARD	cardinal numeral (e.g.4, sixty-five)
CIRCP	circumposition
CL	clitic
CLFnom	nominal classifier
CLFnum	numeral classifier
CN	common noun
COMP	complementiser
CONJ	conjunction
CONJC	coordinating conjunction (e.g. and, or)

POS tag	**Tag description**
CONJS	subordinating conjunction (e.g. when)
COP	copula
COPident	identity copula
COPloc	locative copula
COPneg	negative copula
DEM	demonstrative specifier
DET	determiner
EXPL	expletive pronoun
INTRJCT	interjection
IPHON	ideophone, onomatopoeia
MOD	modifier
N	common noun
Nbare	bare noun
Ncomm	noun with common gender (Norwegian
NDV	deverbal noun
NFEM	feminine noun
NMASC	masculine noun
NNEUT	neuter noun
NNO	noun neutral for number (e.g. data)
Np	Proper noun
Nrel	relational noun
Nspat	spatial noun
NUM	numeral
NUMpart	partitive numeral
ORD	ordinal
P	preposition
PN	personal pronoun
PNabs	absolute pronoun (Bantu)
PNana	pronominal anaphor
PNdem	demonstrative pronoun
PNposs	possessive pronoun
PNrefl	reflexive pronoun
PNrel	relative pronoun
PPOST	postposition
PREP	preposition
PREPdir	directional preposition
PREP/PROspt	hybrid locative category (Bantu)

POS tag	**Tag description**
PREPtemp	temporal preposition
PROint	interrogative pronoun
PROposs	possessive pronoun
PRT	particle
PRTexist	existential marker
PRtinf	infinitive marker
PRTint	interrogative particle
PRTn	nominal particle
PRTposs	possessive particle
PRTpred	predicative particle
PRTprst	presentational particle
PRTv	verbal particle
PTCP	participle
QUANT	quantifier
REL	relative clause marker
V	verb
V1	first verb in a SVC
V2	second verb in a SVC
V3	Third verb in a serial verb construction
V4	Fourth verb in a serial verb construction
Vbid	verbbid(Kwa)
Vcon	converb
Vdtr	Ditransiti ve verb
Vimprs	impersonal verb
Vitr	Intransitive verb
VitrOBL	intransitive verb with prepositional object
Vlght	light verb
Vmod	modal verb
Vneg	negative verb
Vpre	preverb
Vrefl	reflexive verb
Vtr	transitive verb
VtrOBL	transitive verb with a prepositional object
Vvec	vector verb
Vvector	vector verb
Wh	wh-word

APPENDIX F

Additional Results

This appendix presents several additional analysis results. These results may be of value, but were however placed in the appendix in order to be able to present the results in chapter 7 in a clear manner.

SECTION F.1

Incremental Accuracies

Figures F.1, F.2, and F.3 show accuracies for incremental runs with NB, SVM, and MacEnt, respectively. These runs were performed using a portion of the tweet dataset, starting with 5% and increasing with 5% each run up to 100%. Since three different datasets were used in classification, the datasets were sequentially added in the order seen in the figures. Starting with the *random* dataset, then after about 1/3 into the increments it starts using the *rosenborg* dataset, and then at 2/3 increments it starts using the *erna solber* dataset.

SECTION F.2

Shuffled Incremental Accuracies

Accuracies were also obtained performing incremental runs on shuffled datasets. This was done in order to see if performance would increase during the entirety of the increments. Figures F.4, F.5, and F.6 show the incremental runs using shuffled datasets for NB, SVM, and MaxEnt, respectively.

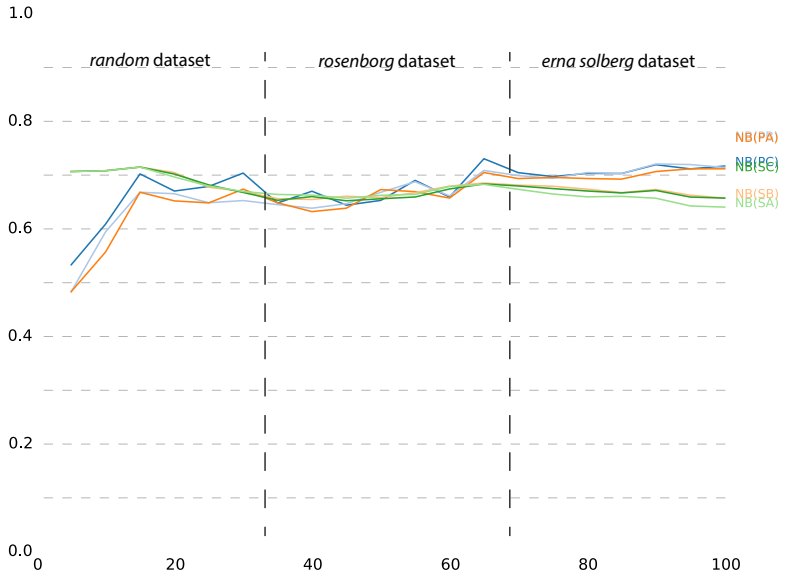


Figure F.1: Accuracy scores for the incremental dataset-size runs for NB

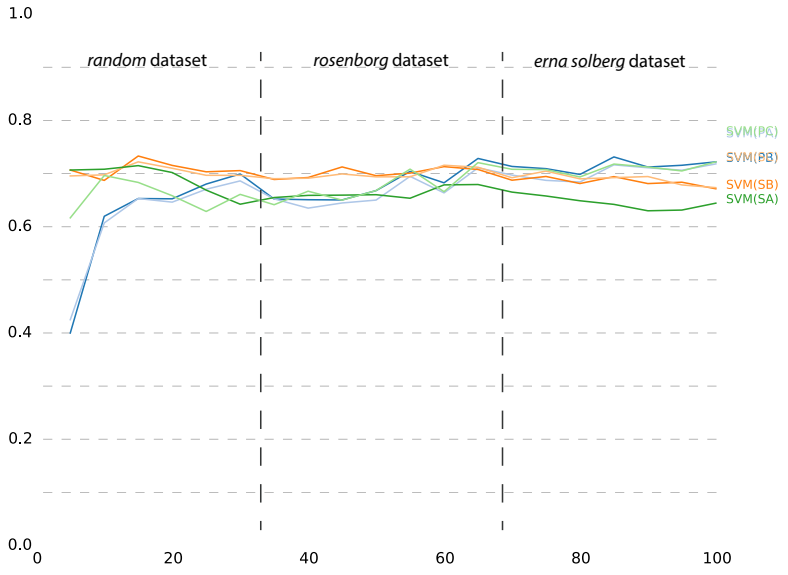


Figure F.2: Accuracy scores for the incremental dataset-size runs for SVM

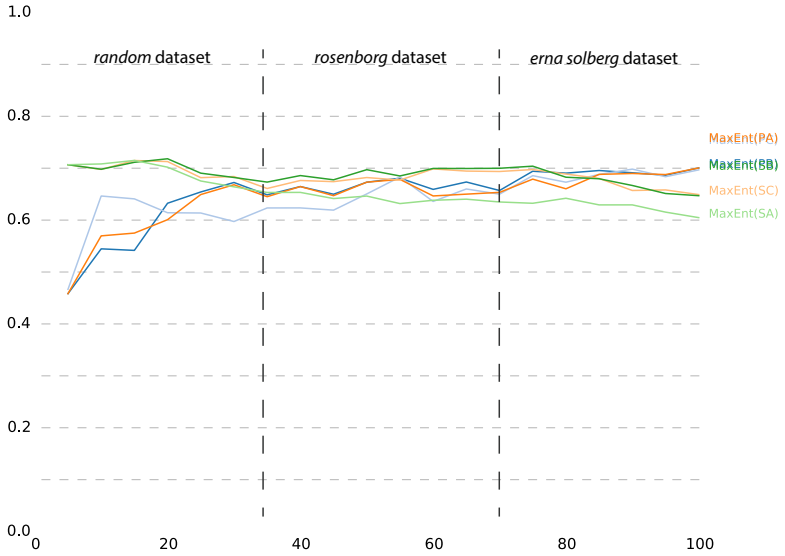


Figure F.3: Accuracy scores for the incremental dataset-size runs for MaxEnt

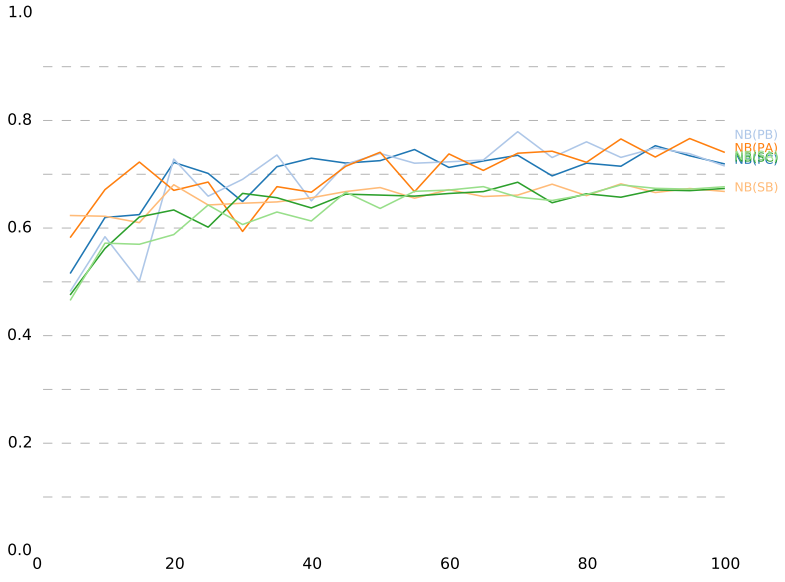


Figure F.4: Accuracy scores for the incremental dataset-size runs for NB

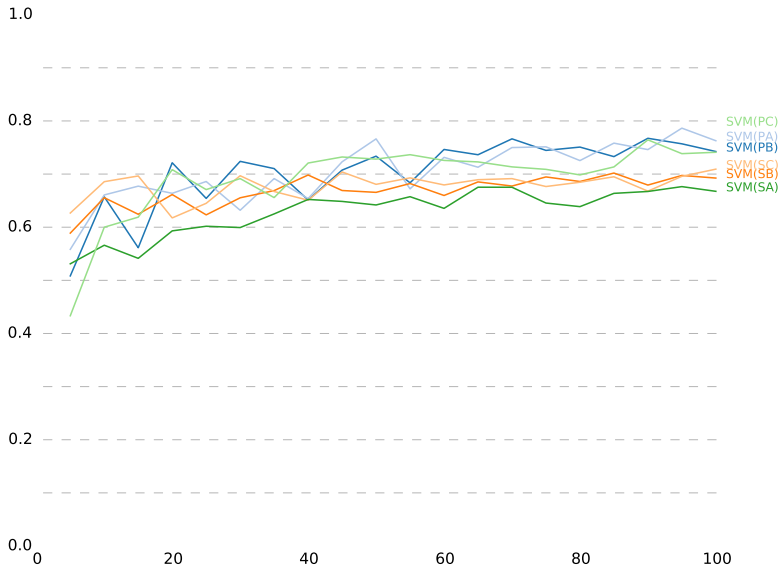


Figure F.5: Accuracy scores for the incremental dataset-size runs for SVM

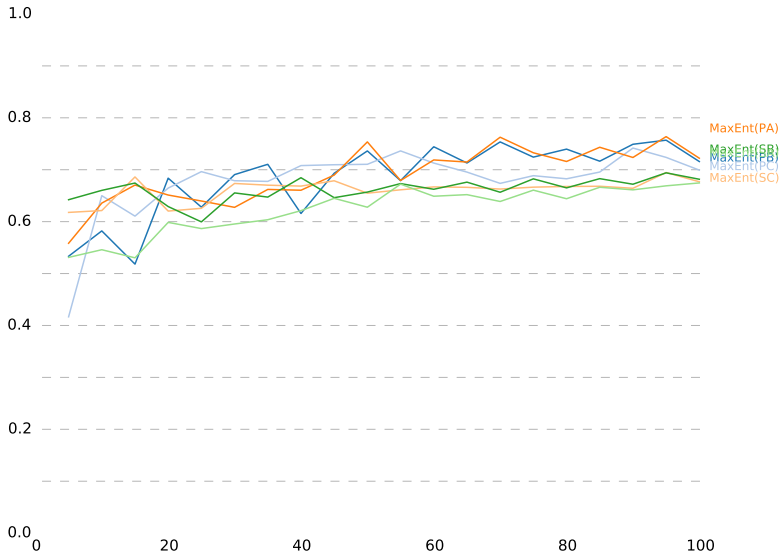


Figure F.6: Accuracy scores for the incremental dataset-size runs for MaxEnt

APPENDIX G

Articles in Submission

This chapter will present abstracts which elaborate on two different articles on subjects of this thesis. These two articles are in submission at the time of delivery of this thesis.

SECTION G.1

Article: Sentiment Topic Detection on Norwegian Tweets

Interest in mining sentiment from weblogs and social media have been growing rapidly the recent decade. While sentiment analysis in itself can be useful it is arguably insufficient for practical purposes if no specific topic is presented along with it. Jointly detecting sentiment and topic should be a useful function in a context such as Twitter, where the informal structure of the platform opens up for a good deal of targeted sentiment from the users. In order to perform relevant and practical sentiment mining and aggregation, identifying the topics of these sentiments is a must.

This paper explores the use of machine learning classification and part-of-speech tagging in an attempt at classifying sentiment and detecting sentiment topics in Norwegian tweets. This article shows the use of a Support Vector Machines text classifier for two-step binary classification - subjectivity- and polarity classification. The subjectivity classifier is then used in order to detect possible sentiment bearing words by classifying substrings of the tweet texts, from there possible sentiment topics are identified and disambiguated using Pointwise Mutual Information.

SECTION G.2

Article: Using Cross-Lingual Lexical Sentiment Look-Up to Improve Classification on Norwegian Tweets

Sentiment lexica can be useful in sentiment classification tasks given the fact that the sources of sentiment can often be identified from specific words. In Norwegian, adjectives like "bra", "fantastisk", and "vakkert" are words that are used to express positive sentiment, while words like "elendig" and "forferdelig" are used to express negative sentiment. While there can be found several sentiment lexica for English, sentiment lexica in the Norwegian language are scarce.

This paper will evaluate the use of a sentiment lexicon through cross-lingual lookup. We use the English sentiment lexicon SentiWordNet on a Norwegian tweet corpus, in order to augment sentiment classification using machine learning techniques. The translation process is performed using two different methods. The first method is performed by translating each word separately, then performing sentiment lexicon look-up utilizing the words' part-of-speech tags in order to disambiguate the lexicon entries. The second method translates the entire tweet texts at a time, before tokenizing and performing sentiment lexicon lookup, thereby using no disambiguation method on the lexicon entries.

Bibliography

[1] "Twitter engineering blog." blog.twitter.com/2013/new-tweets-per-second-record-and-how. Accessed: 12.01.15.

[2] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?," in *Proceedings of the 19th international conference on World wide web*, WWW '10, (New York, NY, USA), pp. 591–600, ACM, 2010.

[3] J. Bollen, H. Mao, and A. Pepe, "Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena.," in *ICWSM*, 2011.

[4] P. Njølstad, L. Høysæter, and J. A. Gulla, "Evaluating feature sets and classifiers for sentiment analysis of financial news," 2013.

[5] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79–86, Association for Computational Linguistics, 2002.

[6] A. Brew, D. Greene, D. Archambault, and P. Cunningham, "Deriving insights from national happiness indices," in *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pp. 53–60, IEEE, 2011.

[7] M. Bautin, L. Vijayarenu, and S. Skiena, "International sentiment analysis for news and blogs.," in *ICWSM*, 2008.

[8] "Alexa site overview: twitter.com." www.alexa.com/siteinfo/twitter.com. Accessed: 11.01.15.

[9] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[10] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity," in *Proceedings of ACL*, pp. 271–278, 2004.

[11] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining.," in *LREC*, 2010.

[12] A. Bermingham and A. F. Smeaton, "Classifying sentiment in microblogs: is brevity an advantage?," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 1833–1836, ACM, 2010.

[13] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval*, vol. 463. Pearson Education Limited, 2011.

[14] T. Joachims, "Making large scale svm learning practical," 1999.

[15] E. T. Jaynes, "Information theory and statistical mechanics," *Physical review*, vol. 106, no. 4, p. 620, 1957.

[16] C. Manning, "Maxent models and discriminative estimation," *CS 224N lecture notes, Spring*, 2005.

[17] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report, Stanford*, pp. 1–12, 2009.

[18] B. Liu and L. Zhang, "A survey of opinion mining and sentiment analysis," in *Mining Text Data*, pp. 415–463, Springer, 2012.

[19] J. H. Martin and D. Jurafsky, *Speech and Language Processing, Second Edition*. prentice hall, 2008.

[20] A. Ratnaparkhi *et al.*, "A maximum entropy model for part-of-speech tagging," in *Proceedings of the conference on empirical methods in natural language processing*, vol. 1, pp. 133–142, Philadelphia, PA, 1996.

[21] A. Kennedy and D. Inkpen, "Sentiment classification of movie reviews using contextual valence shifters," *Computational Intelligence*, vol. 22, no. 2, pp. 110–125, 2006.

[22] C. C. Aggarwal and C. Zhai, *Mining text data*. Springer, 2012.

[23] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.

[24] J. Carletta, "Assessing agreement on classification tasks: the kappa statistic," *Computational linguistics*, vol. 22, no. 2, pp. 249–254, 1996.

[25] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of systems and software*, vol. 80, no. 4, pp. 571–583, 2007.

[26] "Google scholar." scholar.google.com. Accessed: 26.11.14.

[27] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Language resources and evaluation*, vol. 39, no. 2-3, pp. 165–210, 2005.

[28] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources.," in *EMNLP*, vol. 4, pp. 412–418, 2004.

[29] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pp. 347–354, Association for Computational Linguistics, 2005.

[30] C. Whitelaw, N. Garg, and S. Argamon, "Using appraisal groups for sentiment analysis," in *Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 625–631, ACM, 2005.

[31] A. Esuli and F. Sebastiani, "Determining term subjectivity and term orientation for opinion mining.," in *EACL*, vol. 6, p. 2006, 2006.

[32] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *Proceedings of LREC*, vol. 6, pp. 417–422, 2006.

[33] H. Kanayama and T. Nasukawa, "Fully automatic lexicon expansion for domain-oriented sentiment analysis," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pp. 355–363, Association for Computational Linguistics, 2006.

[34] N. Godbole, M. Srinivasaiah, and S. Skiena, "Large-scale sentiment analysis for news and blogs.," *ICWSM*, vol. 7, 2007.

[35] R. McDonald, K. Hannan, T. Neylon, M. Wells, and J. Reynar, "Structured models for fine-to-coarse sentiment analysis," in *Annual Meeting-Association For Computational Linguistics*, vol. 45, p. 432, 2007.

[36] X. Ding, B. Liu, and P. S. Yu, "A holistic lexicon-based approach to opinion mining," in *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pp. 231–240, ACM, 2008.

[37] T. Nasukawa and J. Yi, "Sentiment analysis: Capturing favorability using natural language processing," in *Proceedings of the 2nd international conference on Knowledge capture*, pp. 70–77, ACM, 2003.

[38] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and trends in information retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[39] A. Abbasi, H. Chen, and A. Salem, "Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums," *ACM Transactions on Information Systems (TOIS)*, vol. 26, no. 3, p. 12, 2008.

[40] C. Lin and Y. He, "Joint sentiment/topic model for sentiment analysis," in *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 375–384, ACM, 2009.

[41] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis," *Computational linguistics*, vol. 35, no. 3, pp. 399–433, 2009.

[42] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining.," in *LREC*, vol. 10, pp. 2200–2204, 2010.

[43] B. Liu, "Sentiment analysis and subjectivity," *Handbook of natural language processing*, vol. 2, pp. 627–666, 2010.

[44] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[45] A. Go, L. Huang, and R. Bhayani, "Twitter sentiment analysis," *Entropy*, vol. 17, 2009.

[46] J. Sharma and A. Vyas, "Twitter sentiment analysis," *Indian Institute of Technology unpublished report (2010 http://home. iitk. ac. in/˜ jaysha/cs365/projects/report. pdf)*.

[47] A. K. Jose, N. Bhatia, and S. Krishna, "Twitter sentiment analysis,"

[48] A. Bifet and E. Frank, "Sentiment knowledge discovery in twitter streaming data," in *Discovery Science*, pp. 1–15, Springer, 2010.

[49] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," in *Proceedings of the Workshop on Languages in Social Media*, pp. 30–38, Association for Computational Linguistics, 2011.

[50] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent twitter sentiment classification," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 151–160, Association for Computational Linguistics, 2011.

[51] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!," *ICWSM*, vol. 11, pp. 538–541, 2011.

[52] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang, "Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 1031–1040, ACM, 2011.

[53] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack, "Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques," in *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pp. 427–434, IEEE, 2003.

[54] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai, "Topic sentiment mixture: modeling facets and opinions in weblogs," in *Proceedings of the 16th international conference on World Wide Web*, pp. 171–180, ACM, 2007.

[55] K. Cai, S. Spangler, Y. Chen, and L. Zhang, "Leveraging sentiment analysis for topic detection," in *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on*, vol. 1, pp. 265–271, IEEE, 2008.

[56] Y. Choi, Y. Kim, and S.-H. Myaeng, "Domain-specific sentiment analysis using contextual feature generation," in *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pp. 37–44, ACM, 2009.

[57] H. Saif, Y. He, and H. Alani, "Alleviating data sparsity for twitter sentiment analysis," CEUR Workshop Proceedings (CEUR-WS. org), 2012.

[58] C. Lin, Y. He, R. Everson, and S. Ruger, "Weakly supervised joint sentiment-topic detection from text," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 6, pp. 1134–1145, 2012.

[59] H. Saif, Y. He, and H. Alani, "Semantic sentiment analysis of twitter," in *The Semantic Web–ISWC 2012*, pp. 508–524, Springer, 2012.

[60] M. A. Hearst, "Direction-based text interpretation as an information access refinement," *Text-based intelligent systems: current research and practice in information extraction and retrieval*, pp. 257–274, 1992.

[61] J. M. Wiebe, "Tracking point of view in narrative," *Computational Linguistics*, vol. 20, no. 2, pp. 233–287, 1994.

[62] E. Riloff and J. Wiebe, "Learning extraction patterns for subjective expressions," in *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pp. 105–112, Association for Computational Linguistics, 2003.

[63] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives," in *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pp. 174–181, Association for Computational Linguistics, 1997.

[64] J. Brooke, M. Tofiloski, and M. Taboada, "Cross-linguistic sentiment analysis: From english to spanish.," in *RANLP*, pp. 50–54, 2009.

[65] X. Wan, "Using bilingual knowledge and ensemble techniques for unsupervised chinese sentiment analysis," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 553–561, Association for Computational Linguistics, 2008.

[66] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[67] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National academy of Sciences of the United States of America*, vol. 101, no. Suppl 1, pp. 5228–5235, 2004.

[68] I. Titov and R. McDonald, "Modeling online reviews with multi-grain topic models," in *Proceedings of the 17th international conference on World Wide Web*, pp. 111–120, ACM, 2008.

[69] K. Nigam and M. Hurst, "Towards a robust metric of opinion," in *AAAI spring symposium on exploring attitude and affect in text*, pp. 598–603, 2004.

[70] P. Subasic and A. Huettner, "Affect analysis of text using fuzzy semantic typing," *Fuzzy Systems, IEEE Transactions on*, vol. 9, no. 4, pp. 483–496, 2001.

[71] L. Lloyd, D. Kechagias, and S. Skiena, "Lydia: A system for large-scale news analysis," in *String Processing and Information Retrieval*, pp. 161–166, Springer, 2005.

[72] L. S. H. Pal Christian S. Njolstad, "Sentiment analysis for financial applications," 2014.

[73] W. Wei and J. A. Gulla, "Sentiment learning on product reviews via sentiment ontology tree," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 404–413, Association for Computational Linguistics, 2010.

[74] A. Aue and M. Gamon, "Customizing sentiment classifiers to new domains: A case study," in *Proceedings of recent advances in natural language processing (RANLP)*, vol. 1, pp. 2–1, Citeseer, 2005.

[75] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 36–44, Association for Computational Linguistics, 2010.

[76] D. M. McNair, M. Lorr, and L. F. Droppleman, *Profile of mood states*. Univ., 1971.

[77] L. Derczynski, A. Ritter, S. Clark, and K. Bontcheva, "Twitter part-of-speech tagging for all: Overcoming sparse and noisy data.," in *RANLP*, pp. 198–206, 2013.

[78] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith, "Part-of-speech tagging for twitter: Annotation, features, and experiments," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pp. 42–47, Association for Computational Linguistics, 2011.

[79] "Twitter rest api documentation." https://dev.twitter.com/overview/api. Accessed: 17.03.14.

[80] "Twitter developer rules of the road." https://dev.twitter.com/terms/api-terms. Accessed: 12.03.14.

[81] "Tweepy." www.tweepy.org. Accessed: 12.02.14.

[82] "Ntnu smarttagger." smarttagger.herokuapp.com/tag. Accessed: 10.09.14.

[83] "Requests: Http for humans." docs.python-requests.org/en/latest/. Accessed: 12.02.14.

[84] K. Hagen, J. B. Johannessen, and A. Nøklestad, "A constraint-based tagger for norwegian1," 2000.

[85] "Norsource tagger." regdili.hf.ntnu.no:8081/webtagger/tagger. Accessed: 11.06.14.

[86] C. S. Marco, "An open source part-of-speech tagger for norwegian: Building on existing language resources,"

[87] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[88] T. E. Oliphant, *A Guide to NumPy*, vol. 1. Trelgol Publishing USA, 2006.

[89] E. Jones, T. Oliphant, and P. Peterson, "Scipy: Open source scientific tools for python," *http://www.scipy.org/*, 2001.

[90] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 0090–95, 2007.

[91] "Prediction api client library for java." https://developers.google.com/api-client-library/java/apis/prediction/v1.6. Accessed: 06-02-14.

[92] "Apache mahout." mahout.apache.org. Accessed: 06-02-14.

[93] "Datumbox." www.datumbox.com. Accessed: 06-02-14.

[94] "Textblob: Simplified text processing." https://textblob.readthedocs.org/en/latest/index.html. Accessed: 12-02-14.

[95] T. Schaul, J. Bayer, D. Wierstra, Y. Sun, M. Felder, F. Sehnke, T. Rückstieß, and J. Schmidhuber, "PyBrain," *Journal of Machine Learning Research*, 2010.

[96] "Sentiwordnet." sentiwordnet.isti.cnr.it. Accessed: 10.11.14.

[97] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[98] "Learning lexical scales: Wordnet and sentiwordnet." compprag.christopherpotts.net/wordnet.html#sentiwordnet. Accessed: 10.11.14.

[99] "Natural language toolkit - nltk 3.0 documentation." www.nltk.org. Accessed: 11-02-14.

[100] "Microsoft translator." msdn.microsoft.com/en-us/library/dd576287.aspx. Accessed: 21.10.14.

[101] "Google translate." https://translate.google.com. Accessed: 03.12.14.

[102] "Google translate api." https://cloud.google.com/translate/docs. Accessed: 21.10.14.

[103] "Research shows twitter is driving english language revolution." http://www.brandwatch.com/2013/05/research-shows-twitter-is-driving-english-language-evolution/. Accessed: 13-02-14.

[104] J. R. Landis and G. G. Koch, "An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers," *Biometrics*, pp. 363–374, 1977.

[105] "Twitter." http://www.twitter.com. Accessed: 02.02.14.

[106] "Merriam-webster." www.merriam-webster.com/dictionary/sentiment. Accessed: 12.10.14.

[107] "Typecraft." www.typecraft.org/tc2wiki/Main_Page. Accessed: 06-02-14.