NTNU
Norwegian University of
Science and Technology

Brand sentiment analysis of the Norwegian banking sector

Anders Lien

Master of Science in Computer Science
Submission date: June 2017
Supervisor: Jon Atle Gulla, IDI

Norwegian University of Science and Technology
Department of Computer Science

Preface

The submission of this thesis marks the conclusion of my Master of Science degree in Computer Science. The work has been conducted at the Department of Computer Science at the Norwegian university of science and technology under the guidance of Prof. Jon Atle Gulla and in cooperation with Sparebank 1 SMN.

Acknowledgements

I would like to thank my supervisor Jon Atle Gulla for his guidance and support while I worked on this thesis as well as in the specialization project conducted in the fall of 2016.

I would also like to thank the group of financial analysts and social media marketers at Sparebank 1 SMN in Trondheim, Norway. Our discussions and their refunding of my expenses during the work has been very helpful.

Thanks to Cristina Marco who supplied her part-of-speech-tagger which was used in the early stages of the project.

Finally, I want to Nicole Løvik for her efforts in annotating a large number of bank reviews.

Abstract

For the last two decades, the world wide web has become a social arena where people can express themselves. People write opinionated texts on social media towards various targets, which is read by other people. The readers are often influenced by what is written. Companies holding brands often keep a close eye on what people write about them, but it is a burdensome task to monitor the World Wide Web. The use of sentiment analysis for automatically analysing brands has been on the rise in recent years for this reason. Norwegian banks are increasingly considering themselves as brands, and they monitor their online reputation accordingly.

This master thesis describe a system performing sentiment analysis on Norwegian reviews of the banking sector. The system use three different classifiers, Naive Bayes, Support Vector Machines and Maximum Entropy, to classify the polarity of reviews in the Norwegian banking sector. A custom made mapping between a Norwegian wordnet and the sentiment lexicon SentiWordNet aids the sentiment analysis with cross-lingual lookup, as Norwegian sentiment resources are limited. A set of unigrams and bigrams for the banking domain is also used as input to the classifiers, as well as textual features. The system also utilize knowledge of semantic structures in an attempt to extract sentiments on sub-aspects mentioned in the reviews.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter will focus on the motivation behind brand sentiment analysis and introduce the structure, goals and results of the thesis.

1.1 Background

In the recent decades, the Internet has changed the world and become a necessity in people's lives. An increasing amount of people are connected through it and the information flow is ever increasing. People all over the world post large amounts of opinionated text to different media services on the World Wide Web. Marketing teams and others take great interest in what customers say and feel towards them and their products.

Monitoring this is burdensome if the company has thousands or millions of customers. Automating this process is therefore of interest to all companies interested in their reputation and customer satisfaction. This has sparked interest in the research area sentiment analysis.

Sparebank 1 SMN is a Norwegian bank with headquarters in Trondheim, Norway. The bank is part of Sparebank 1-gruppen, an alliance of 16 independent banks cooperating under the brand Sparebank 1. The last years it has become easier than ever for people to switch banking provider in Norway. By using the BankID-technology for identification, it is possible to create a savings account in a new bank through online banking services alone. The simplicity of change leads to increased competition in the market, banks therefore need be aware of their brand's position and reputation. Opinions are also important as input to marketing. In the event of negative publicity a bank should be able to respond to it quickly, but for that to happen opinion detection systems are needed. A way to do this is through brand sentiment analysis which analyse text written about the bank.

1.2 Research Questions

The following questions arise in conjunction with this thesis.

- How can lexical sentiment resources be used in brand reputation analysis?
- How can semantic resources about group structures improve the analysis of brand reputation?
- What characterizes the average positive or negative statements about brands?

1.3 Approach

In this thesis a system performing sentiment analysis on reviews of the banking sector is presented. The system makes use of machine learning algorithms and multiple lexical resources in order to recognize the polarity of reviews in the bank domain. A general sentiment ontology tree structure of banks is used to classify sentiments of the correct aspect of the bank.

1.4 Results

One of the contribution from this thesis is a custom mapping between the lexical resources Norsk ordvev and SentiWordNet. This mapping was created using Google Translate, a popular machine translation service operated by Google. The implemented sentiment analysis system use this mapping, a custom lexicon of the bank domain and textual features to classify bank reviews with an accuracy of up to 76.4% with a Maximum Entropy classifier.

1.5 Report structure

The thesis starts with an introduction to the topic of sentiment analysis (SA) which you are reading now. Following this, chapter 2 introduce the theoretical foundation behind SA together with a snapshot of the current state-of-the-art of sentiment analysis in chapter 3. Chapters 4 and 5 describe resources and third-party programs that are used in the implemented sentiment analysis system. Chapter 7 describes the implementation before results and conclusions are drawn in chapters 8 and 9.

Chapter 2

Theoretical background

This chapter will focus on the theoretical foundation behind brand sentiment analysis.

2.1 Subjectivity

Subjectivity is a philosophical concept and one of the foundations of sentiment analysis. It denote the fact that people have their own perspectives, feelings and beliefs that influence their thinking, speech and behaviour (Oxford dictionaries online (2017)). As an example, a teacher grading homework based on his mood or his liking of a student is subjective. He lets his current personal feelings dictate which grade to set. Subjectivists believe that all truths are dependent on the perspectives of each individual. This leads to problems in society because people fundamentally dislike injustice and discrimination which would occur in a subjectivistic society.

The opposite of subjectivity is objectivity, in which there is solid truths or facts, independent of what people think. An objective person attempts to tell truth even if he disagree with what it says. If the teacher mentioned earlier was objective, he would set the grade based only on the work in front of him. Even if he dislikes the student, it should be possible for that student to receive the top grade.

Subjectivity is present in history, even though most people agree history and time are objective matters. The quote "history is written by the victors" (Sacco, 2016) illustrate this. Historians and text-book writers can be influenced directly by governments, or indirectly by their own patriotic feelings, to have a more positive point of view towards their home country. This shows the importance of objectivity. Having an objective mindset is just as important in scientific research, journalism and in the justice system.

Subjectivity is part of being human. People let their opinions guide them through life, and without them people wouldn't know what to do in any situation.

2.2 Sentiment analysis

Sentiment analysis is a field in Artificial intelligence (AI) which concern analysis of people's opinions, sentiments, attitudes and emotions towards entities such as products, organisations, events according to Liu (2012). Because people's opinions affect their behaviour in many ways, many companies are investing in sentiment analysis to improve customer satisfaction and gain insights into markets. The sentiment analysis is usually performed on text written in natural language and the goal is to discover the general attitude of the author towards the entities in the text.

The term *opinion mining* is often used interchangeably with sentiment analysis. Although an opinion is not equivalent to a sentiment, the two represent the same study. The terms originate from two separate articles, Nasukawa and Yi (2003) and Dave et al. (2003). This thesis will consistently use the term *sentiment analysis* to reference this study.

Sentiment analysis use techniques from the fields Natural language processing (NLP) and computational linguistics. Research has been conducted in both of these fields since the 1950s Jones (1994), when the United States took interest in automatically translating Russian to English. Sentiments and opinions, on the other hand, was not researched until the early 2000s. The true potential of sentiment analysis lies in automatic analysis of content from the World Wide Web.

Subjectivity is an important concept in context of sentiment analysis. In objective text there normally exist no opinions, therefore a purely objective text stating only concrete facts should have a polarity of zero. However, authors might hold strong feelings towards what they write about and sentiment might be present, but not written explicitly. For instance, the sentence "Banken doblet renten dagen etter jeg tok opp lån" (*The bank doubled the interest after giving me the loan*) is purely objective since it denote only facts. However, this sentence imply a negative sentiment since it appear the bank scammed the customer.

2.2.1 Classification levels

Liu (2012) define 3 levels of analysis which differ in how thoroughly they extract opinions.

Document-level

Sentiment analysis performed at the document level is the simplest approach. The classification is performed on a document which is a collection of sentences (e.g. a review). The output of document-level-SA is a single value describing the presence, or absence, of sentiment in the document. On the document-level only a single, non-mixed, sentiment towards a single entity might be extracted.

Sentence-level

The document-level often fail to capture all sentiments expressed in a text. The sentence-level analyse each sentence and extracts a single sentiment per sentence. The entities targeted by the sentiment might differ, and only a single sentiment can be extracted per sentence. For instance, this review of a restaurant shows the advantage of a deeper level of analysis.

> Vi fikk servert en god forrett, og en krabbe som smakte utmerket til hovedrett. Dessverre smakte husets vin forferdelig og ødela smaksopplevelsen.
> *We were served a nice appetizer, and a crab for the main course which tasted deliciously. Unfortunately, the house wine tasted terrible and ruined the tasting experience.*

This review communicate a positive sentiment towards the appetizer and main course. However, a negative sentiment towards the house wine is present in the second sentence. Sentence-level-SA is able to represent this as opposed to the document-level which would output a neutral polarity since the positive sentiment balance out the negative. Nevertheless, there is no semantic difference between document-level-SA and sentence-level-SA. The latter is equivalent to doing the former with each sentence as a document.

Entity- and aspect-level

The entity- and aspect-level is the most thorough level of sentiment analysis. Liu (2012) contain the following definitions of entity, aspect and opinion.

Definition 1 *An entity e is a product, service, topic, issue, person, organization, or event.*

Definition 2 *An aspect a_e is a part or an attribute of an entity e.*

Definition 3 *An opinion is a quintuple*

$$(e_i, a_{ij}, s_{ijkl}, h_k, t_l) \tag{2.1}$$

where e_i is the name of an entity, a_{ij} is aspect j of entity i, h_k is the opinion holder, t_l is the time the opinion is stated and s_{ijkl} is the sentiment expressed towards a_{ij} of e_i by the opinion holder h_k at time t_l.

The goal of sentiment analysis on this level is to extract a list of all opinion-quintuples present in a document d. It allows reviews to contain multiple points of view in the same review, as well as varying sentiments held at different times. Consider the following review:

> Vi dro til banken i går for å ta opp et lån. Kundebehandleren som møtte oss var hyggelig, men min kone syntes de hadde dårlig kundeservise. Da vi dro tilbake i dag mente hun de hadde bedre kundeservise, da de ga oss lån.
> *We went to the bank yesterday to take out a loan. The representative who met us was nice, but my wife felt they had bad customer service. When we went back the next day she thought they had better customer service, since they gave us a loan.*

From this the following opinions can be extracted:

(bank, representative, pos, author, yesterday)
(bank, customer service, neg, author's wife, yesterday)
(bank, customer service, pos, author's wife, today)

2.2.2 Applications

One of the reasons sentiment analysis has become one of the most active research areas in natural language processing is due to its numerous real world applications. Both governments, marketing teams, company shareholders and others, take interest in what is said about them. With the expansion of the World Wide Web (WWW), everyone has access to an almost unlimited amount of data. The amounts are so large that individuals cannot analyse sentiments manually because of the enormous workload that would entail. Automatic sentiment analysis is therefore a necessity to stay up to date.

One of the applications listed by (Bannister, 2015) is gauging the popularity of announcements. In fact, the Obama administration sentiment used it to monitor and gauge public opinion to policy changes and campaign messages ahead of the 2012 presidential election. According to (Bea, 2012), 31.7 million election-related tweets were published on the social media platform Twitter during the 2012 election night. With such numbers, extracting sentiment automatically is necessary to gain any insights.

Brand analysis is one of the most prominent applications of sentiment analysis. Brands are strong influencers of people's purchasing habits and they are therefore important to their owners. Measuring the popularity of a brand, or even a company, is one of the primary uses of sentiment analysis. Another possible use is in recommender systems where the system might recommend products based on what it knows about the current user and what other similar users have said about it.

2.2.3 Process

Extraction of sentiments from text is no easy task. It requires knowledge of natural language processing (NLP) as well as specialized knowledge of the language being analysed. Most tasks today in NLP, including SA, is utilizing statistical methods, this is also known as machine learning.

The first step of creating a system performing sentiment analysis is collecting large amounts of data. We use the term document to denote a data instance (e.g. a review). The data should be in the same format, e.g. text or speech, and will be used to train and evaluate a machine learning model. The collected document must then be manually annotated[1] as either subjective or objective. It must also be assigned polarity (either negative, neutral or positive) which is what we will teach a machine learning algorithm to predict.

After obtaining an annotated dataset the data must be preprocessed. This might involve steps such as removing objective sentences since they contain no opinions. Furthermore, feature sets must be extracted.

A feature is a numerical measure which is extracted from all documents. For instance, in the context of sentiment analysis, counting the occurrences of the words "good" and "bad" can count as two features. Creating a good feature set is the most critical part of creating a good sentiment classification model. The goal of engineering a feature set is to find metrics that describe the data in the best way in context of the task at hand. The feature set is the input that a machine learning model use to predict a label (output class).

2.2.4 Challenges

There are many challenges in sentiment analysis. Natural language is very complex, and the same words might have different meanings dependent on position. It might contain ambiguity, which makes it very challenging to understand and interpret a sentence. A famous example is:

[1] The annotation is done on either the entire document, each sentence or on the entity/aspect-level as described in section 2.2.1

I saw a man on a hill with a telescope.

This very ordinary sentence have at least 5 possible distinct meanings (credits to (Byrd, 2015)):

- There's a man on a hill, and I'm watching him with my telescope.
- There's a man on a hill, who I'm seeing, and he has a telescope.
- There's a man, and he's on a hill that also has a telescope on it.
- I'm on a hill, and I saw a man using a telescope.
- There's a man on a hill, and I'm sawing him with a telescope.

The last meaning does not make any sense, but a computer program without information about the correct use of a telescope would still consider it. In Norwegian, the word "tre" have multiple meanings: it can refer to a plant (*tree*), a material (*wood*), a number (*three*), or a verb (*to thread (a needle)*). In the context of brand sentiment analysis one might see this comparison of two competing brands:

Pepsi Max er bedre enn Coca-Cola. Den smaker forferdelig.
Pepsi Max is better than Coca-Cola. It tastes awfully

In this example we cannot be really sure about which brand "it" refers to, although human intuition tells us that it must refer to Coca-Cola since the second sentence is negative and the first expressed a preference of Pepsi Max over Coca-Cola.

Another challenge in NLP is detection of irony and sarcasm. Oxford dictionaries provide the following definitions:

Definition 4 *Irony*[2] *- The expression of one's meaning by using language that normally signifies the opposite, typically for humorous or emphatic effect.*

[2] https://en.oxforddictionaries.com/definition/irony

Definition 5 [3] *Sarcasm - The use of irony to mock or convey contempt.*

Irony and sarcasm are common literary effects in speech and also appear in written language. They make sentiment analysis difficult because the author mean to convey the opposite of what is really written. A sarcastic review about the movie *The Dark Knight Rises* could be as follows:

> I just love how it takes 2 hours and 45 minutes for Batman to rise.

The author of this review literary states that he loves that Batman does not rise until the end of the movie. The true, hidden meaning though, is that the author felt the movie was boring. This is difficult for a sentiment analysis system to grasp, and it often results in classifications errors.

Humans often disagree about things because of their subjectivity. This leads to disagreement between annotators of datasets as well. When annotating text from the general domain, different annotators agree 70% - 80% of the time at best (Mozetic et al., 2016). If the performance of a sentiment analysis system is how well it agrees with human judges, this could be considered an upper limit of the system's performance.

2.3 Brand analysis

One of the prominent applications of sentiment analysis is brand sentiment analysis. A brand is an abstract symbol that easily identifies a seller's product. Brand analysis is the process of analysing which state a brand is in, this is typically done by marketing teams. The process involves analysing customers and their buying habits, competitors and the market in which the brand is positioned. A brand can be very valuable as seen in company acquisitions. The bidding company often pay large amounts of money to acquire the brand. Goodson (2012) in fact claim that brands are among the most valuable assets a company holds.

Distility (2010) list 3 sub-analyses which constitute brand analysis:

[3] https://en.oxforddictionaries.com/definition/sarcasm

- **Customer analysis** - customer groups, key influencers, trends, needs
- **Competitor analysis** - analysing competing brands' market position and strengths and weaknesses
- **Current brand audit** - what the brand currently express, values, market position, threats

One of the reasons brand analysis is prioritized by companies is that current customers can be very influencing of other potential customers. People usually have no idea of which is better when looking for an item, or a service. Earlier, people based their choices largely on the experiences (and opinions) of people in their lives, like relatives or neighbours. Now, with the expansion of the World Wide Web (WWW), people can read reviews and be influenced by anyone with an Internet connection across the world. Companies know this, and keep a close eye on what is written about them and their brand. One negative review about them can be the difference of a purchase and the customer choosing the opponent's product.

An example of a strong brand is the Apple iPhone. The typical owner of an iPhone often emphasize usability and a great user interface as important qualities in a phone, and is often willing to pay extra for it. This would be included in a customer analysis. iPhone has a very strong brand reputation which promises the qualities mentioned above are present in every new iPhone model. Because of this promise, customers return for new iPhones whenever they need a new phone.

The competitor analysis is an important part of brand analysis. It can be seen as conducting a customer analysis on competing companies, thereby obtaining a new understanding of why people choose the other brand. Competitor analysis and current brand audit is somewhat related, the goal of both is to pinpoint the brand's market position. The difference is that the current brand audit focus on the company in question rather than the competition. Each brand has its own strengths and weaknesses, and brand analysis can be used to manoeuvre a brand in the right direction.

The output of a brand analysis must be used in a brand strategy to be of any use. That is, brand analysis is only the first step of branding.

2.3.1 Analysis techniques

Smith (2016) lists the following techniques as traditional ways of analysing brands:

Surveys

Conducting surveys is the most common way to gather quantitative customer opinions towards brands. Surveys can be conducted in person, over the phone or online. The answers are usually in the form of multiple choice.

Work groups and focus groups

Work groups and focus groups give more qualitative insights than the ones gathered from a survey. The questions are more open-minded and the answers are more detailed (not multiple choice).

Employees

All employees who interact with customers can supply insights of product reception. These insights are important to consider and can be collected internally in the organization.

2.3.2 Automatic brand analysis

Manual analysis techniques in brand analysis are time- and resource consuming, especially for small companies with many customers. Many companies now see the value of automatic brand sentiment analysis with data obtained from the Internet. All the analysis-techniques mentioned above benefit from sentiment analysis, since a good SA-system can highlight positive and negative aspects of both the company analysing its brand, as well as what is being expressed about the competition. The feedback from a review author is also relevant for the customer analysis since it shows what qualities he or she emphasizes in a product or service.

An SA system could either monitor the web or just run occasionally by marketing teams conducting brand analysis. Monitoring micro-blogs such as Twitter, as well as review sites and discussion forums can be used as a threat monitor, by issuing an alert to customer service representatives when finding criticism online. The representatives can then attend to the problem by communicating with the customer. Companies are expected to have a present and attentive customer service and when this is not the case, customer satisfaction often decreases. Companies have been following the advances of sentiment analysis closely, and some buy services from analysis providers which monitor the web.

2.4 Linguistics

This section will discuss the field of linguistics, which is the study of languages.

2.4.1 Part-of-speech

Most languages group the words of their language into categories called Part-of-Speech (PoS). The categories contain words with similar grammatical properties and they often hold the same function in a sentence. Finding the PoS of a word is therefore a step towards understanding text.

The Norwegian language contains 10 parts-of-speech according ordnett.no[4], an online language service by the Norwegian publishing company Kunnskapsforlaget.

- **Noun** - names and things
- **Verb** - a performed activity or process
- **Adjective** - describing a noun
- **Adverb** - modify the meaning of a verb, adjective or adverb
- **Pronoun** - substitution of nouns
- **Conjunction** - joins words together
- **Preposition** - specify locations or points in time
- **Interjection** - words signifying emotions

[4] https://www.ordnett.no/spr%C3%A5kverkt%C3%B8y/spr%C3%A5kvett.ordklassene

- **Determinativ** - possessives or demonstratives
- **Subjunction** - words that allow a clause to be inserted into another sentence

The 2 latter parts-of-speech are not included as parts-of-speech in the English language.

2.4.2 Lemmatizaion

Most languages have different forms of each word called inflected forms. For instance, English contains the words child and walk, which has inflected forms children (plural) and walked (past tense). Inflection of words can alter the time an action was performed (verb), or it can alter nouns to refer to multiple instances of the original word. Mostly, the alteration is the addition of a suffix to the word. Lemmatization is the process of determining this original word, called a lemma, given an inflected word. In order to keep lexical resources such as dictionaries as small and concise as possible, only lemmas can be used for lookup. Lemmatization is closely related to stemming, the difference being that stemmers use algorithmic patterns to remove suffixes, while lemmatizers use lexical resources. For this reason, lemmatizers perform much better in terms of accuracy since they handle irregular inflections such as lemmatizing "am" to the word "be".

2.4.3 Syntax

The syntax of a language is the study of the structure of a language, especially sentence structure. It is essentially a set of rules which govern how to write or speak a language correctly. Norwegian and English have different syntax, which makes translation of text difficult. In Norwegian, the negation clause *ikke* is positioned after the verb *liker*. This is in contrast to the English language, in which the negation clause *not* is placed before the verb *like*.

2.4.4 Valence shifters

Certain words called valence shifters can alter how strongly to express a sentence. Valence shifters consist of intensifiers, diminishers and negations. Intensifiers increase the meaning of a sentence, which can be both negative and positive. Examples are "veldig", "sykt", "ekstremt" (very, insanely, extremely) which e.g. can be followed by adjectives "bra" (good) or "dårlig" (bad). The phrase "very bad" imply a more negative sentiment than just "bad", thus, the polarity of "bad" has been increased (intensified). Correspondingly, diminishers can decrease the polarity of words. "Litt" and "noe" (a little, somewhat) decrease, or soften, the polarity of words. "Han er smart" contain a stronger sentiment than "Han er litt smart".

Both intensifiers and diminishers alter the polarity of words, but they cannot change positive words to have a negative polarity. Valence shifters with this property is called negators. These include "ikke", "knapt" and "aldri" (not, hardly, never).

2.5 Natural language processing

Natural language processing (NLP) is a field in artificial intelligence (AI) a which concerns interaction between natural (human) language and computers. More specifically, tasks in NLP involve some degree of understanding or generation of natural language. Humans started communicating with spoken language 100.000 years ago, and 7000 years ago we started writing texts (Russell et al., 2010). Although other species such as chimpanzees and dolphins communicate with vocabularies of about a 100 words, humans speak languages with hundreds of thousands, or millions, of words!

Natural language is unstructured data which is difficult to interpret. Humans continuously learn language through their lifespan, especially in the earliest years. For computers, this is very difficult because of natural language's size, complexity and ambiguity. A computer has no idea what a particular word means, and must be aided with human-annotated resources. All languages have their own syntax, and meanings can be altered just by switching the position of two words. Some prominent tasks are:

- **Natural language understanding**
- **Expert systems**
- **Named entity recognition**

- **PoS-tagging**

2.5.1 Named-entity recognition

Another part of understanding text is recognizing entities mentioned in it. Entities are normally nouns, and an accurate recognition is necessary to understand who the subject of the sentence is about. In the sentence "Sparebank 1 SMN er bedre enn DNB" (*Sparebank 1 SMN is better than DNB*) one must recognize the two entities "Sparebank 1 SMN" and "DNB" in order to understand that the author favour the former over the latter.

2.5.2 N-grams

One of the most popular features in sentiment analysis is the presence of n-grams. N-grams are word sequences of length n which occur in order. For instance, the bigrams (n=2) present in the sentence "Jeg leker med ballen" are: "Jeg leker", "leker med" and "med ballen". The set of unigrams (n=1) are the individual words. In domain-dependent sentiment analysis, the presence of unigrams and bigrams is a popular feature which was used in the pioneering work of (Pang et al., 2002).

2.6 Lexical resources

Computers executing NLP-tasks often rely heavily on lexical resources. Computers work exclusively on numbers (signals) at the most basic levels. Text (strings) is defined using an encoding mapping numbers to characters, but a computer cannot interpret the meaning (semantics) of words even though it can represent them. A lexical resource is a collection of lexical items (words) that contain linguistic information.

2.7 Machine learning

This section will provide an introduction to machine learning, one of the most applicable fields in *artificial Intelligence*.

2.7.1 Introduction to machine learning

The field of machine learning is a sub-field of artificial intelligence concerning the development of algorithms which change behaviour and adapt when exposed to data. Machine learning tasks are usually divided into three types based on how they receive feedback (Russell et al., 2010):

- **Unsupervised learning** where the algorithm should find patterns in data without receiving any feedback. The most common unsupervised task is data clustering.
- **Supervised learning** where the algorithm is exposed to multiple pairs of input sets and labels, and tries to generalize the connection between input and labels into a function.
- **Reinforcement learning** where an agent (a program) performs a set of actions, and occasionally receives punishments or rewards based on its performance. The agent then tries to learn to perform in order to maximize rewards and minimize punishments.

Sentiment analysis uses supervised learning in order to predict text polarity, given previously seen texts and their polarities. Because sentiment analysis and NLP-tasks mostly utilize supervised learning, learning will for the rest of this thesis refer to supervised learning, unless explicitly stated otherwise.

In order to evaluate the performance of a learning algorithm, the dataset is commonly divided into a training set and a testing set. The algorithm is then trained on the training set, in which it tries to find a relationship between a feature set (input) and a label (output). Often, this is done by adjusting an internal model for each new sample it receives. After lots of adjusting, it can perform better when receiving a sample where the input is similar to a sample it has seen before. It can then assume the two samples belong to the same class, based on this similarity.

After the algorithm has been trained, learning is stopped, by ceasing to adjust the model. The model can then be applied to the testing set by outputting a predicted class based on the input set. The performance of the model can then be evaluated by comparing the prediction with the correct value. Evaluation metrics are used to summarize this performance as described in section refsec:evaluation-metric.

2.7.2 Overfitting

One challenge in all supervised learning algorithms is avoidance of overfitting. Overfitting refer to a problem where a model is overly complex compared to the actual function underlying the data. Figure 2.1 illustrates this, where two functions predict the value based on points it has seen. It is easy to see that the complex, curved function probably is not the true function, even if it fits the data perfectly. The straight line is a much better fit in general, when testing the data on previously unseen data. The reason behind the error is probably some kind of noise (e.g. sensory noise).

Overfitted models often let features unrelated to result have an impact, making the model detect coincidental patterns. For instance, a model predicting the eyes of a dice might learn that a specific result is due to a person crossing his fingers. A model will also overfit if the training data is completely random. Most widely deployed learning algorithms are subject to, and have methods to combat, overfitting. Mainly, these methods involve keeping the model as simple as possible. Another way of avoiding overfitting is by increasing the size of the training set, either by allocating a bigger proportion of the dataset, or by increasing the size of the dataset.

2.7.3 Naive bayes

A Naive Bayes classifier uses a probabilistic learning model, specifically a Bayesian classifier. The model is organized as a tree where the root node is the class, and it has leaves representing each feature we use in the classification. All the different nodes take certain values, and the values of the leaf nodes are dependent on the value of the root node. This is illustrated in figure 2.2. Naive Bayes is based on Bayes theorem:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \tag{2.2}$$

The goal is to classify bank reviews, thus it can be formulated as:

$$P(c|r) = \frac{P(r|c) * P(c)}{P(r)} \tag{2.3}$$

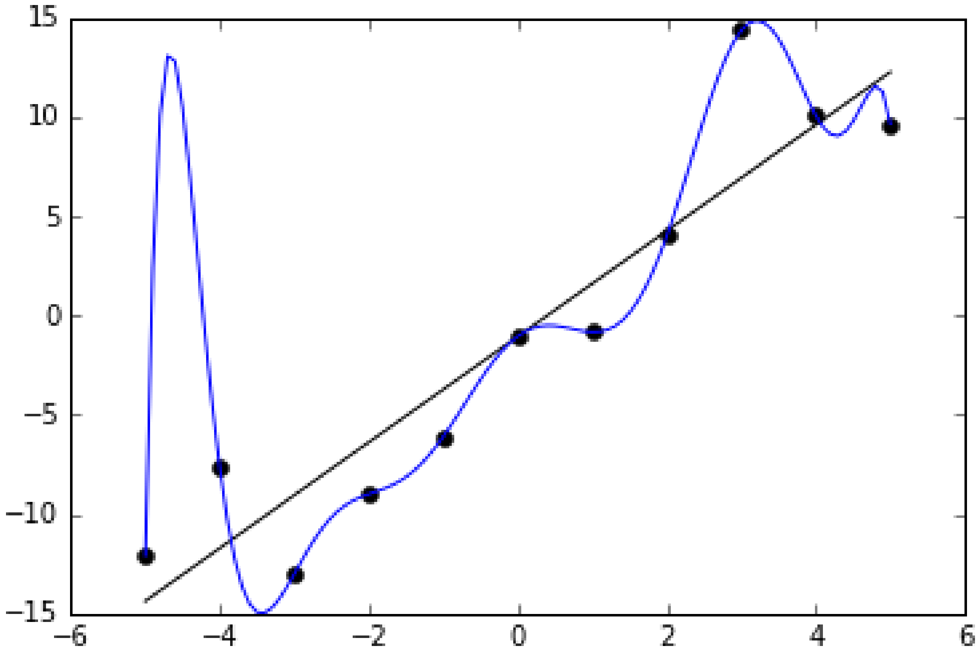


Figure 2.1: Two functions approximating a dataset. *By Ghiles [CC BY-SA 4.0], via Wikimedia Commons*

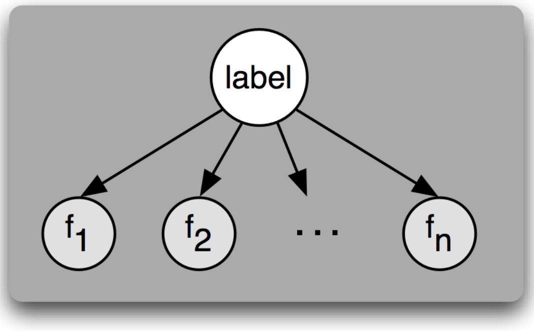


Figure 2.2: Naive Bayes classifier with two classes, black and white

where r is a review to be classified, and c is a predicted class.

The classifier is naive in the sense that it assumes conditional independence between leaf-nodes (features), given the root node (the class). This is an incorrect assumption in the real world, but for modelling purposes it works reasonably well. The assumption simplifies the calculation a great deal as there is no need to calculate how each feature affects every other feature. By using the joint probability model, and (naively) assuming $P(x_i|x_1, x_2, ...x_n, C) = P(x_i|C)$ where $\mathbf{x}$ is a feature vector derived by r, we obtain:

$$P(c|\mathbf{x}) = P(c)\prod_i P(x_i|c) \tag{2.4}$$

The classifier assigns the label with the highest probability to each instance:

$$\hat{C} = \underset{k}{\operatorname{argmax}} P(k|\mathbf{x}) \tag{2.5}$$

where $\hat{C}$ is the predicted class.

2.7.4 Maximum entropy classifier

Maximum entropy classifiers, also known as multinomial logistic regression classifiers, are based on the principle of maximum entropy. Maximum entropy classifiers use the logistic function to estimate probabilities that an observed instance belongs to a class, before choosing the most likely class in the same way as a Naive Bayes classifier. As opposed to Naive bayes, maximum entropy classifiers do not assume features are conditionally independent, and are therefore more computationally expensive to calculate. This may however lead to better results if the features are in fact dependent.

$$P(c|\mathbf{x}) = \frac{e^{\mathbf{w}_c \cdot \mathbf{x}}}{\sum\limits_{j \in C} e^{\mathbf{w}_j \cdot \mathbf{x}}} \tag{2.6}$$

where w_j is a weight vector associated with class j, C is the set of classes, and x is a feature set.

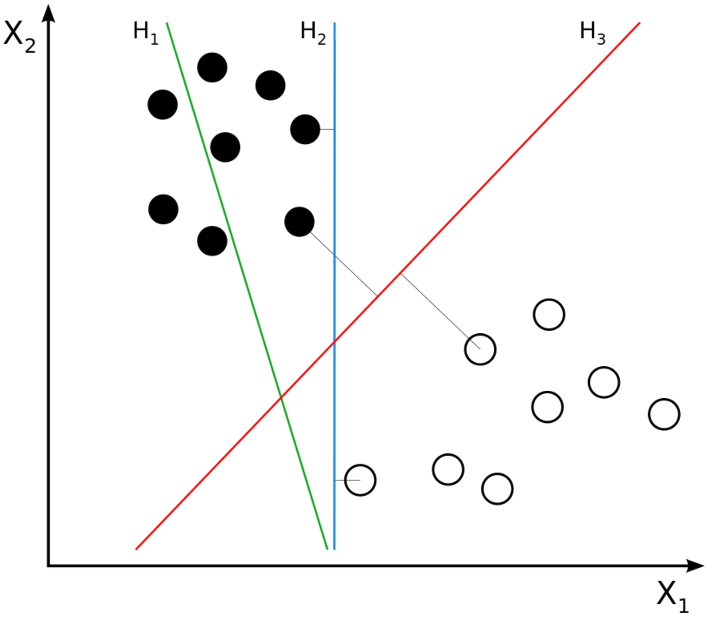


Figure 2.3: 3 functions attempting to separate the dataset. *By ZackWeinberg [CC BY-SA 3.0], via Wikimedia Commons*

2.7.5 Support vector machines

Support vector machine (SVM) attempts classify instances to a class by separating data points with a function h. The training of an SVM-model consists of varying h to approximate the true function t. The training algorithm tries to maximize the margin to the closest data points, while correctly separating the instances. This way the model correctly classifies similar instances to the same class.

In figure 2.3 three suggestions of h that are plotted. The goal of h is to separate data points of different class. As we see, H1 fails to classify them correctly, but H2 and H3 are both valid. We notice that if a black instance was inserted at the top right, just to the right of the H2 border, it would be wrongly classified, even though it is closest to the other black instances. Intuitively, H3 would be a more probable approximation to t because of a greater margin to the closest nodes. A support vector classifier tries to find the separation (line) that maximize this margin. For this reason, SVMs are also called maximum-margin classifiers.

Predicted value True value	True	False
True	True Positive	False Negative
False	False Positive	True Negative

Table 2.1: Confusion matrix

Predicted value True value	Negative	Neutral	Positive
Negative	True Negative	False Neutral	False Positive
Neutral	False Negative	True Neutral	False Positive
Positive	False Negative	False Neutral	True Positive

Table 2.2: Confusion matrix of sentiment polarity classification

2.8 Evaluation metrics

Evaluation is very important in science. It gives us the tools to compare the results of multiple works by e.g. checking which one obtained the highest accuracy. Given that all works give honest evaluation scores, people can easily choose to implement the best systems, giving better results. Dishonest values might result from errors (intentional or unintentional), where testing is performed with the same data that was used in training, or by running the algorithms lots of times, reporting only the best evaluation score.

Before defining various evaluation metrics, the concept of a confusion matrix is introduced. A confusion matrix is a statistical visualization of the number of correct and wrong classifications in supervised learning. A confusion matrix for a two-class classification problem (e.g. subjectivity classification) is illustrated in table 2.1. In a three-class classification problem (e.g. sentiment polarity classification) the matrix is extended as seen in table 2.2. The value of cell (i,j) in a confusion matrix is the number of times the model classifies a document as belonging to class j when its true class is i.

Accuracy is the simplest evaluation metric. It is the proportion of results which are correctly classified. For a 2x2 confusion matrix

$$Accuracy = \frac{\#correct}{\#total} = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{2.7}$$

where T_p, T_n, F_p and F_n is the number of true positives, true negatives, false positives and false negatives, respectively. Extended to a 3x3 confusion matrix, this extends to

$$Accuracy = \frac{\#correct}{\#total} = \frac{M_{neg,neg} + M_{neu,neu} + M_{pos,pos}}{\sum\limits_{i \in C} \sum\limits_{j \in C} M_{i,j}} \tag{2.8}$$

where $M_{m,n}$ denote the number of times a document of class m was classified as belonging to n, and C is the set of classes.

Precision and **Recall**, are measures originating in information retrieval (like Internet search), but has been applied to general classification as well. Both are metrics in context of a given class, that is, we calculate precision and recall of each class of document polarity.

Precision (or *positive predictive value*) of class i, is defined as the proportion of documents classified as i which truly belongs to i:

$$Precision(i) = \frac{M_{i,i}}{\sum\limits_{j \in C} M_{j,i}} \tag{2.9}$$

Recall (or *sensitivity*) of class i, is defined as the proportion of documents of class i which is classified as i:

$$Recall(i) = \frac{M_{i,i}}{\sum\limits_{j \in C} M_{i,j}} \tag{2.10}$$

2.8.1 Inter-annotator agreement

Sentiment analysis usually require an annotated dataset to perform well. Annotators of that dataset might let their subjective judgement affect the annotation, or have different personal thresholds for what constitutes as positive or negative. By adding noise to this mix of reasons, a dataset might have low annotation quality. This can be combated by adding more annotators.

Evaluation of annotator agreement can be done with Krippendorff's alpha, which is a generalization of several statistical measures. A coincidence matrix, similar to a confusion matrix, is used to count the number of times annotators annotate an instance. With two annotators, each cell (i,j) in a k by k matrix, similar to a confusion matrix, is used to count the number of times the first annotator annotates an instance as class i, while another annotates the same instance as class j. (Mozetic et al., 2016) then calculate Krippendorff's alpha as follows:

$$\alpha = 1 - \frac{D_o}{D_e} \tag{2.11}$$

where D_o is the observed annotator agreement and D_e is the disagreement expected by random selection. They are calcultated as follows:

$$D_o = \frac{1}{N} \sum_{c,c'} N(c,c') * \delta^2(c,c') \tag{2.12}$$

$$D_e = \frac{1}{N(N-1)} \sum_{c,c'} N(c) * N(c') * \delta^2(c,c') \tag{2.13}$$

where N(c), N(c') refer to sums and N(c,c') refer to the value of cells in the coincidence matrix. δ refer to a difference function which explains the relationship of the classes (if they are ordered or not).

Chapter 3

State of the art and related work

3.1 Sentiment analysis

One of the first attempts at sentiment analysis was (Pang et al., 2002) in which Pang, Lee and Vaithyanathan attempted to classify movie reviews from Internet Movie Database (imdb) as negative, neutral or positive. An accuracy of 82.9% was obtained by training a support vector classifier on a bag-of-words model of 16165 unigrams. The authors of the IMDb-reviews had labeled the review with either a 1-5 star rating or another numerical value, which was used as a target value for supervised learning algorithms. Pang and Lee later published a paper Pang and Lee (2004) proposing a two-step classification process, extracting only subjective sentences before classifying polarity. They obtained 86.4% accuracy, as opposed to 82.8% without extraction.

3.2 Obtaining sentiment target by topic detection

In 2015, (Øye, 2015) completed his master thesis where he performed sentiment analysis on Norwegian Twitter messages. The goal of his thesis was to perform general sentiment analysis on 3 separate datasets fetched from social media platiform Twitter, as well as topic detection to determine the sentiment target. The datasets were one general set of Norwegian tweets, one about the Norwegian prime minister Erna Solberg, and one set of tweets about the Norwegian football team Rosenborg. Øye performed the analysis using the two-step approach as described by (Pang and Lee, 2004) and obtained an accuracy of up to 80% on the polarity classification,

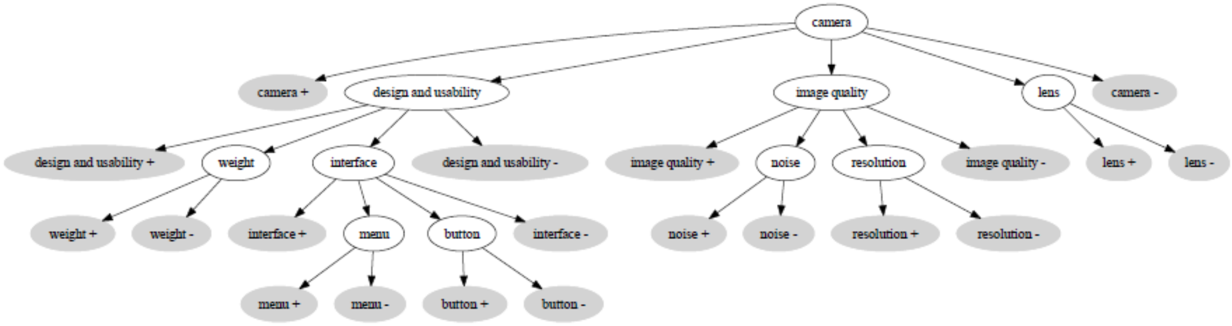


Figure 3.1: Wei and Gulla's camera ontology

and up to 76% when combining subjectivity detection with polarity classification.

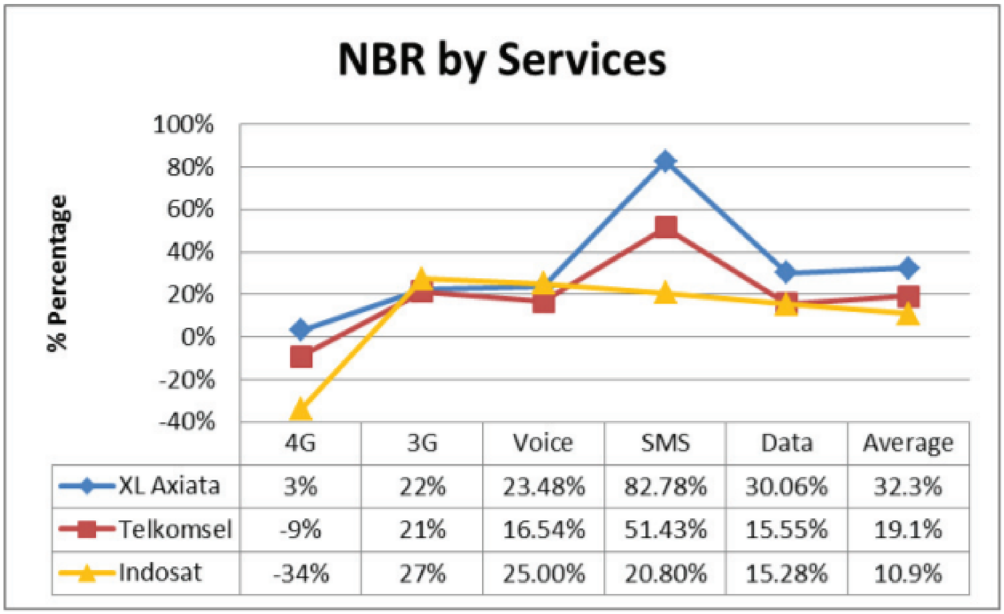
3.3 Hierarchical sentiment classification

(Wei and Gulla, 2010) published a paper describing an approach to sentiment learning aided by sentiment ontology tree (SOT) structures. The SOT contains different aspects, and sub-aspects, with an entity as root, organized in a hierarchical manner. Their example SOT was the camera ontology in figure 3.1, and they used this knowledge of the camera ontology structure to extract which entity or aspect was targeted by a review. A hierarchical learning algorithm was applied to the problem to learn threshold- and weight-vectors. After training the algorithm, it would e.g. correctly classify a review mentioning the interface as targeting the "camera", "design and usability" and "interface", but not necessarily the "menu" and "button" which are children of "interface".

3.4 Using sentiment analysis to measure brand reputation

(Vidya et al., 2015) used Twitter data to create a sentiment analysis system for mobile phone providers in Indonesia. The goal was to analyse brand reputation by extracting sentiments towards five products: 3G, 4G, SMS, voice services and internet services. A metric, net brand reputation (NBR), was used to calculate customer satisfaction per service, and was calculated as follows:

$$NBR = \frac{Positivementions - Negativementions}{Positivementions + Negativementions} * 100 \tag{3.1}$$



	4G	3G	Voice	SMS	Data	Average
◆ XL Axiata	3%	22%	23.48%	82.78%	30.06%	32.3%
■ Telkomsel	-9%	21%	16.54%	51.43%	15.55%	19.1%
▲ Indosat	-34%	27%	25.00%	20.80%	15.28%	10.9%

Figure 3.2: NBR per service of Indonesian mobile phone providers

The NBR metric was compared to the Net promoter score (NPS) which is commonly used in computing customer satisfaction (Satmetrix, 2017). NPS has several weaknesses, like only asking current customers, and being expensive to measure if using a large sample size. Net brand reputation, on the other hand, gathered sentiments towards the provider's products by conducting social media listening. This gave the mobile phone providers a larger quantity of opinions for their analysis, and gave new insights. One of the providers, XL Axiata, had a much higher NBR-score than the two others. Axiata's customer service and marketing divisions had understood the importance of engaging with social media followers, leading to increased positive mentions of their service brand.

Chapter 4

Lexical resources

4.1 POS-taggers

The first step of understanding text is to identify the individual words. By identifying the Part-of-Speech (PoS) of each word we start seeing the structure of the sentence in question. For instance, consider the sentence "Gutten sparker ballen" (the boy kicks the ball). Here, we have the noun *gutten*, the verb *sparke* and a second noun *ballen*. Since verbs denote actions we easily infer that one noun is performing an action with the second noun. PoS-taggers are usefull tools both when understanding semantics and syntax, but also when analysing sentiments because the number of each part-of-speech might be used as a feature.

4.1.1 Oslo-Bergen tagger

The Oslo-Bergen tagger (OBT) was produced in a joint project between the University of Oslo and Uni Computing in Bergen. The result was a robust morphological and syntactic tagger for the Norwegian language. It consists of three modules:

- A preproccessor performing multitagging and compound analysis.
- A grammar module for morphological and syntactic disambiguation.
- A statistical module filtering out remaining morphological ambuiguity.

The tagger starts by collecting hits associated with a word in the Norwegian lexicon Norsk Ordbank, which is a database of inflected-forms with morphological information. This often results in multiple hits because of word ambiguity in the Norwegian language. The grammar module then use constraint based rules to filter the unintended hits. A statistical module has been created for bokmål which further filters the results.

These modules are implemented in three different processes which are piped (chained), which means the output of the first module is directed to the input of the second module, and the output of the second module is directed to the input of the third module.

In addition to performing PoS-tagging, OBT also returns the associated lemma of the word. This was the main reason that OBT was used in the work associated with this thesis.

4.1.2 Marco's POS-tagger

Marco (2014) developed a simple PoS-tagger based on two lexical resources: Norsk ordbank and Norwegian Dependency Treebank[1]. The latter is a large, manually annotated corpus of text from newspapers. Her work resulted in a PoS-tagger yielding accuracy close to state-of-the-art taggers such as the Oslo-Bergen tagger.

4.2 Wordnets

One important concept in natural language processing is the fact that different words can have the same meaning. This is called synonymity and a set of words with the same meaning is called a synset (synonym set). A wordnet is a collection of synsets and this can help algorithms understand text better. For instance, an algorithm which groups documents of the same topic together, would fail to put an article of plants and another about flora in the same group. In the context of sentiment analysis, it is also important to understand this in order to interpret text correctly.

[1] https://www.nb.no/sprakbanken/show?serial=oai%3Anb.no%3Asbr-10&lang=en

4.2.1 Princeton Wordnet and SentiWordNet

The Princeton wordnet (PWN) was the first wordnet to be developed and currently consists of over 117 000 distinct synsets for the English language. PWN contain synsets of the PoS nouns, verbs, adjectives and adverbs.

SentiWordNet (SWN) by Baccianella et al. (2010) is a lexical resource based on PWN, which assign polarity values to each synset which is useful for sentiment analysis. It is developed at Istituto di Scienza e Tecnologie dell'Informazione, an institute of the National Research Council of Italy. A semi-supervised learning algorithm is used to generate the sentiment scores based on the "gloss" (brief description of the term) associated with the synset. The sentiment values in SWN are automatically generated, therefore errors might occur. Among curious values are synset 00642725-a (uncivil, rude) which is considered 87.5% positive, and synset 02512922-a (nonviolent) which is 37.5% negative. Such errors might affect the classification in a sentiment analysis system.

4.2.2 Norsk ordvev

The Norwegian Wordnet Norsk ordvev[2] is the largest wordnet created for Norwegian. It was developed by Kaldera Språkteknologi AS on behalf of The National Library of Norway and is based on the Danish Wordnet DanNet. Norsk ordvev contains roughly 300.000 synsets where 250.000 are proper nouns.

Open Multilingual Wordnet

The Open multilingual wordnet (OMW) is a collection of wordnets in which all individual synsets have been mapped to the Princeton wordnet (PWN). It, as well as an extended version, is developed at Nanyang Technological University, Singapore. Since SWN is derived from PWN, a lookup in the OMW can directly access polarity values which is useful for sentiment analysis. The original OMW include a wordnet, called NorNet (Fjeld and Nygaard, 2009), for Norwegian Bokmål and Norwegian Nynorsk, and they contain 4455 and 3671 synsets respectively. The

[2] http://www.nb.no/sprakbanken/show?serial=sbr-27

wordnets are based on what is called the core word senses of PWN, which is the (approximately) 5000 most frequently used word senses.

4.3 Automatic translation service

Most approaches to sentiment analysis has targeted the English language. This is due to English being the most spoken language in the world. Most other languages have limited lexical resources available, forcing developers to find other ways to understand text. One solution is to utilize automatic translation services to access lexical resources in English.

4.3.1 Google Translate

Google Translate[3] is the most popular machine translation service in use today. It was introduced in 2004[4] and it was integrated with other Google services. Since November 2016, it has used a neutral machine translation engine, which replaced the statistical translation service it had used since its inception.

[3] https://translate.google.no/

[4] https://www.blog.google/products/translate/ten-years-of-google-translate/

Chapter 5

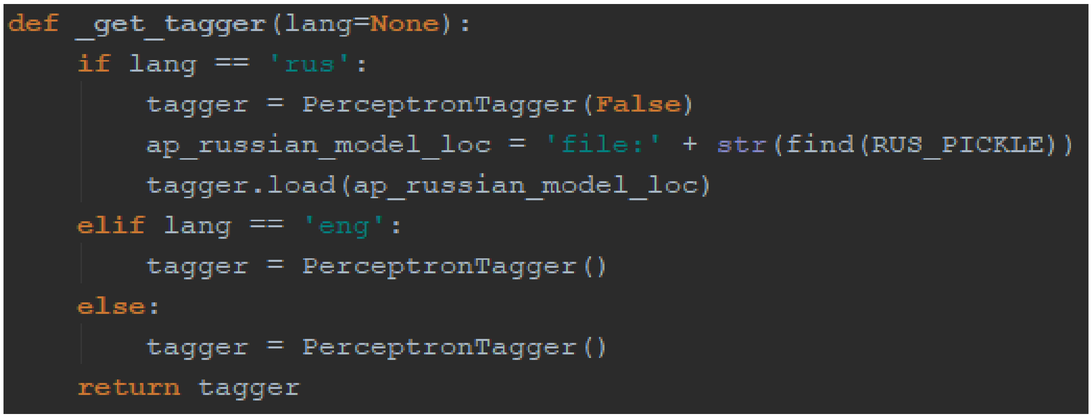
Programming language and libraries

The implementation created alongside this thesis is written in the Python programming language. Python is a popular high-level, general-purpose programming language, and it was chosen for this thesis implementation because of its simplicity, the large amount of third-party packages available and personal preference.

5.1 NLTK

Natural language toolkit (NLTK)[1] is a popular package for executing NLP tasks. It is a python package with many utility methods, machine learning classifiers and various corpus. One shortcoming with NLTK is its lacking support of other languages than English due to hard-coding as seen in figure 5.1. However, NLTK still contains lots of utility methods that are useful, e.g. unigram- and bigram-extraction. NLTK also contain some machine learning algorithms, but most of them are now deprecated in favour of algorithms in the scikit-learn library which is presented in the following section.

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



```
def _get_tagger(lang=None):
    if lang == 'rus':
        tagger = PerceptronTagger(False)
        ap_russian_model_loc = 'file:' + str(find(RUS_PICKLE))
        tagger.load(ap_russian_model_loc)
    elif lang == 'eng':
        tagger = PerceptronTagger()
    else:
        tagger = PerceptronTagger()
    return tagger
```

Figure 5.1: Example of hard coding. This code loads a PoS-tagger, but support only Russian and English. This is internal code which is called by other methods, therefore, enabling support for the Norwegian language would require sub-classing many different classes.

5.2 Scikit-learn

Scikit-learn (also called *sklearn*)[2] is a Python library initially developed in a Google Summer of Code[3] project and is currently being maintained by a team of volunteers (Pedregosa et al., 2011). It provides a framework for a large number of machine learning algorithms within the areas of classification, regression and clustering with tunable parameters. Sklearn also contain tools for dimensionallity reduction, model selection and preprocessing.

5.3 YAML

YAML is a data serialization language which uses indentation to specify data structures. Some of the resources used in the implementation were in the YAML format, which resembles the Python syntax. Parsing of the files in question was performed using the PyYAML library which returns a dictionary of the data structure.

[2] http://scikit-learn.org/stable/#

[3] https://developers.google.com/open-source/gsoc/

Chapter 6

Dataset

In this chapter the dataset used in the implementation of the master thesis will be discussed.

6.1 Data type

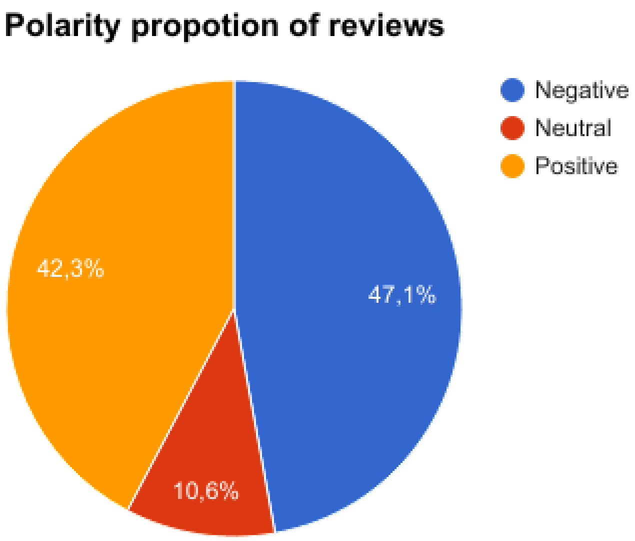
To be of any use, the implemented sentiment analysis system require a large amount of data. The goal of this implementation is to create a model that correctly classifies Norwegian text as negative (-1), neutral (0) or positive (1). In order to create a good model for this problem, all documents are reviews of the bank domain. The reviews used in this implementation are all found on the World Wide Web (WWW). The authors of the reviews are bank customers in Norway and all reviews are written in the Norwegian language.

6.2 Data sources

The reviews were mostly fetched from three sources:

- Review sites
- Social media
- Discussion forums

Review sites are the easiest to extract sentiment from, because reviews are rated with a 1-5 star rating. The stars indicate the polarity of the review, where 1 is very negative and 5 is very



positive. Although the dataset in this thesis was manually annotated, this could allow for sentiment analysis training without annotation. This could be done by considering 1 and 2 stars as negative, 3 stars as neutral and 4 and 5 stars as positive.

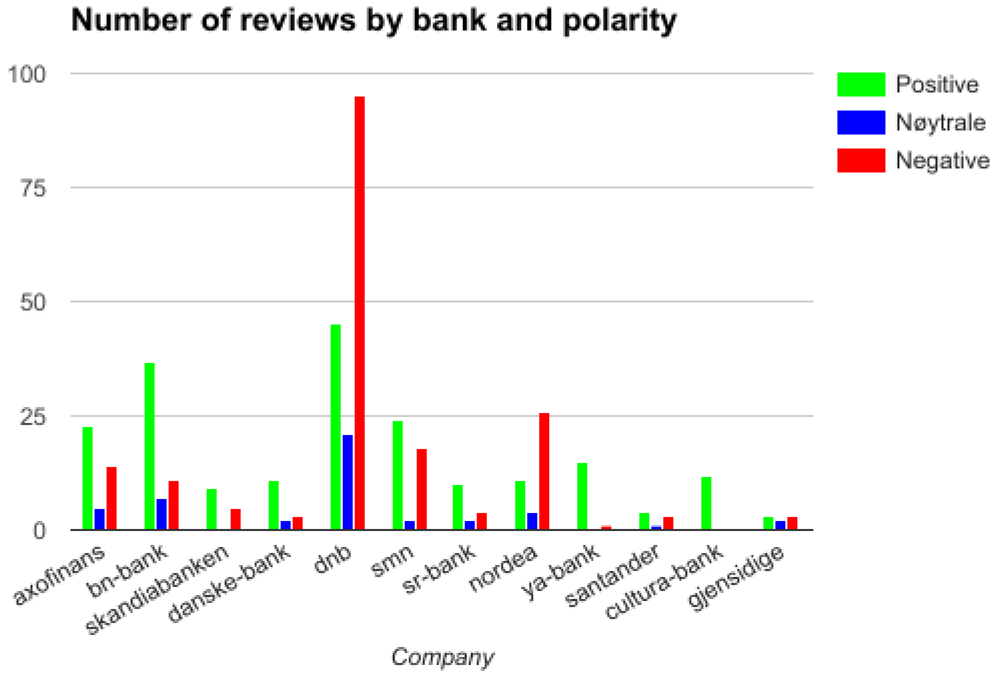
Posts from Facebook and Twitter were also fetched and added to the dataset. Pages on Facebook sometimes contain a review column where people can leave reviews. These reviews also contained a star-rating, therefore the same star-to-polarity mapping as for review sites were performed before adding to the dataset. Other Facebook posts in addition to all Twitter posts contained no such rating. These texts had to be manually annotated.

The website https://bytt.no specialises in publishing user reviews of banks (among others). A total of 165 reviews were fetched by scraping the site with javascript commands.

Finally, a few threads on Norwegian discussion forums were retrieved. The discussion topics were user experiences about various banks, their offered conditions and their customer service.

6.3 Dataset statistics

A total of 431 reviews were retrieved from the mentioned sources. Of these, 204 (47%) were positive, 181 (42%) were negative and 46 (11%) were neutral.



Looking at the polarity proportion we see that there are very few neutral reviews. Most earlier sentiment analysis research do not follow this trend, traditionally, the majority of data has been neutral. In figure 6.1, the average frequency of parts-of-speech per polarity class is shown. It is easy to see a pattern that long reviews are usually negative (all frequencies in the negative section are much larger than in the other polarities). As in most natural language, the most common parts-of-speech are nouns and verbs in most polarity categories. We see that the positive category is an exception to this, it actually contains more adjectives than verbs. This might be due to short reviews such as "bra bank" (*good bank*), "super kundeservice" (*excellent customer service*).

The dataset contained a low amount of sarcasm. This might be domain dependent because people want their opinions to be clearly visible in a review. Of the 433 reviews, only one was considered sarcastic:

> Skjønner det er lukrativt i bankbransjen om dagen når dere takker nei til nye kunder bare fordi vi bor i Holmestrand. Det er greit det! Skal bare betale renter på boliglånet de neste 20 årene..
>
> *I see there is a luxury in the banking business when you reject new customers just*

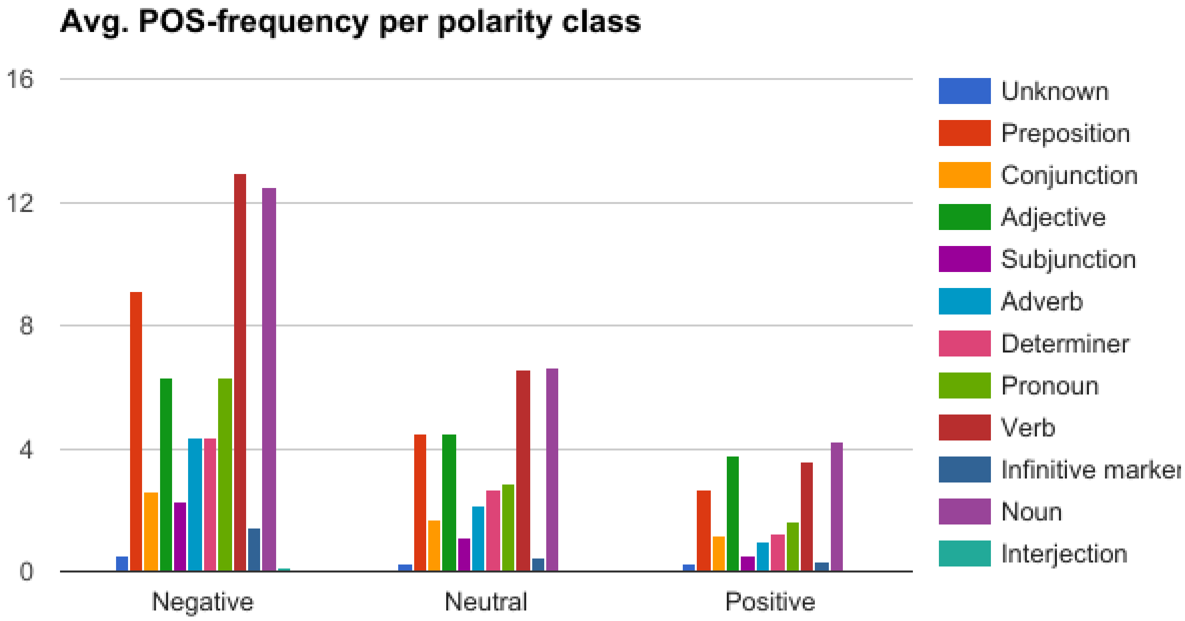


Figure 6.1: Average frequencies of Part-of-Speeches per polarity class

> *because we live in Holmestrand. That's just fine! We're just going to pay interest on our morgage the next 20 years..*

The author is complaining about not being accepted as a customer, claiming its because they live in the town Holmestrand. This is followed by the sarcastic remark "That's just fine!", indicating dislike, even though positive words are used.

Chapter 7

Approach

In this chapter the sentiment analysis implementation created in this master project will be described.

7.1 Annotating dataset

The dataset described in chapter 6 was annotated as a first step towards creating a sentiment analysis system targeting Norwegian bank reviews. The annotation process consisted of assigning whether the reviews are subjective, and in that case assign polarity values. This was done manually by two separate annotators.

7.2 Subjectivity extraction

Some researchers of sentiment analysis have adopted the approach by (Pang and Lee, 2004) where only subjective sentences are extracted. This is not the case of this thesis due to of multiple reasons. Firstly, absence of subjectivity does not necessarily imply absence of sentiment. A customer who has bought a brand new car, and writes in the review that the engine broke down the first week, implies sentiment.

Secondly, a two-step approach is error prone. Normally, the approach involves setting the polarity as zero if it is detected as objective. An error in the subjectivity classification propagates to the polarity classification, and worsens the accuracy of the entire system combined.

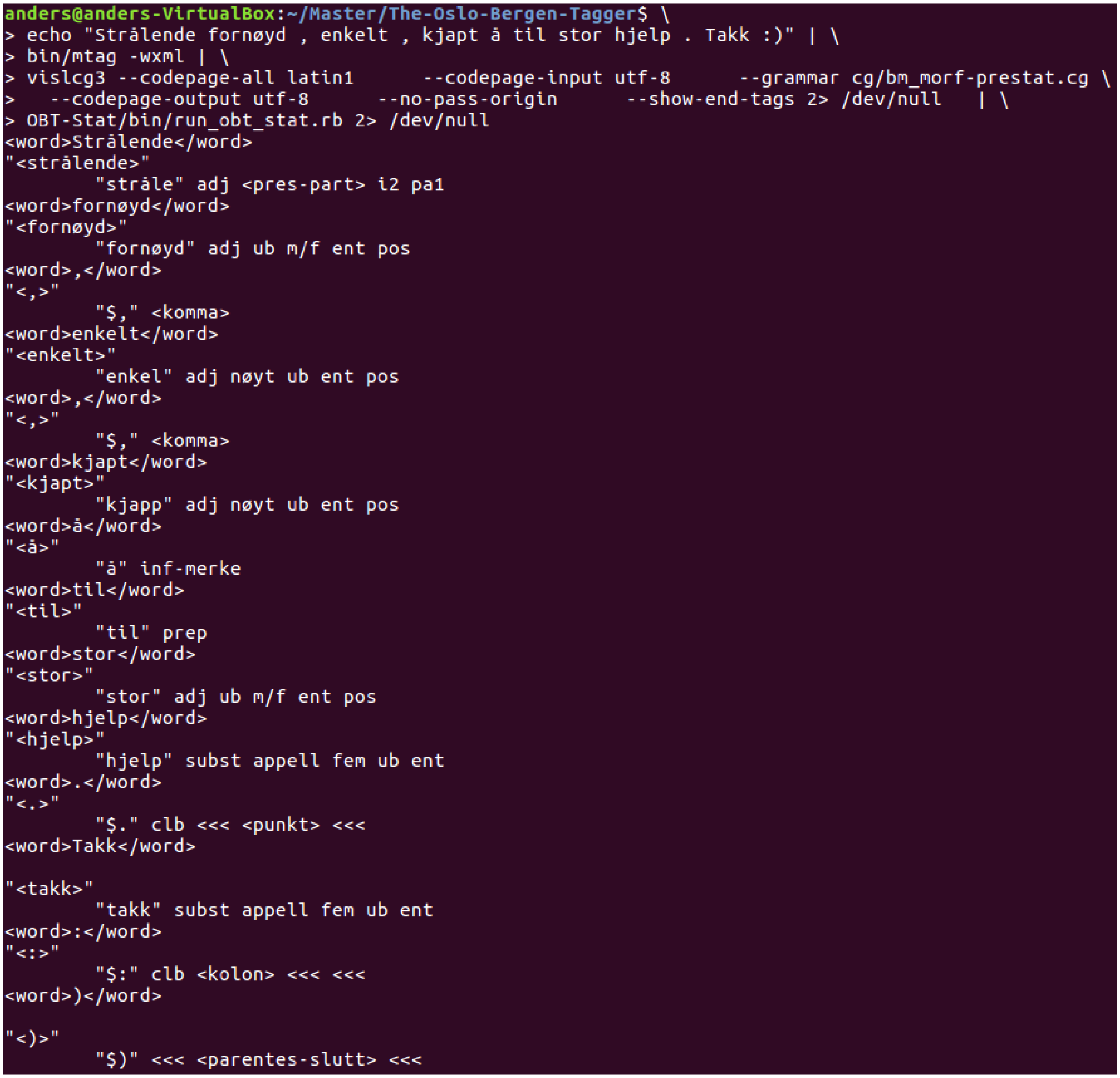
The dataset contains only 16 objective reviews, thus the accuracy of the subjectivity detector would probably be sub-optimal. Therefore, implementing a two-step process would be counter-productive in this system.

7.3 Feature sets

A feature set should try to capture the essence of a text by converting it to a format that is easy for machines to analyse. The common machine readable formats are numeric and boolean values. Optimally, a feature set should represent exactly the same information as natural text and it should be possible to generate the source natural language from the features. However, as optimal conditions rarely apply, such a feature set would include an almost unlimited set of features. Therefore, some information loss is to be expected when extracting features from natural language.

Multiple feature sets are considered when performing feature extraction on the dataset in question. One of the first features tested was the frequency of each part-of-speech. The first PoS-tagger that was used was Marco's tagger (see section 4.1.2). This was due to a difficult setup-process of the Oslo-Bergen tagger which later superseded Marco's tagger. Marco's tagger consist of a simple perceptron tagger with an averaged perceptron as model. These are implemented as classes in the NLTK-library, and provide a simple interface for PoS-tagging.

The OBT was later used to obtain both the lemma and the PoS of words. OBT is a robust morphosyntactic tagger which perform lookup in the Norwegian Wordbank as opposed to Marco's tagger, which was trained on annotated corpora. OBT also return morphological information, like which (inflected) form the word was stated in. Figure 7.1 shows output from OBT to a Linux terminal.



```
anders@anders-VirtualBox:~/Master/The-Oslo-Bergen-Tagger$ \
> echo "Strålende fornøyd , enkelt , kjapt å til stor hjelp . Takk :)" | \
> bin/mtag -wxml | \
> vislcg3 --codepage-all latin1      --codepage-input utf-8      --grammar cg/bm_morf-prestat.cg \
>   --codepage-output utf-8      --no-pass-origin      --show-end-tags 2> /dev/null   | \
> OBT-Stat/bin/run_obt_stat.rb 2> /dev/null
<word>Strålende</word>
"<strålende>"
        "stråle" adj <pres-part> i2 pa1
<word>fornøyd</word>
"<fornøyd>"
        "fornøyd" adj ub m/f ent pos
<word>,</word>
"<,>"
        "$," <komma>
<word>enkelt</word>
"<enkelt>"
        "enkel" adj nøyt ub ent pos
<word>,</word>
"<,>"
        "$," <komma>
<word>kjapt</word>
"<kjapt>"
        "kjapp" adj nøyt ub ent pos
<word>å</word>
"<å>"
        "å" inf-merke
<word>til</word>
"<til>"
        "til" prep
<word>stor</word>
"<stor>"
        "stor" adj ub m/f ent pos
<word>hjelp</word>
"<hjelp>"
        "hjelp" subst appell fem ub ent
<word>.</word>
"<.>"
        "$." clb <<< <punkt> <<<
<word>Takk</word>

"<takk>"
        "takk" subst appell fem ub ent
<word>:</word>
"<:>"
        "$:" clb <kolon> <<< <<<
<word>)</word>

"<)>"
        "$)" <<< <parentes-slutt> <<<
```

Figure 7.1: Screenshot from running OBT on the text
"Strålende fornøyd , enkelt , kjapt å til stor hjelp . Takk :)"
(*Very pleased, easy, fast and very helpful. Thanks :)*)

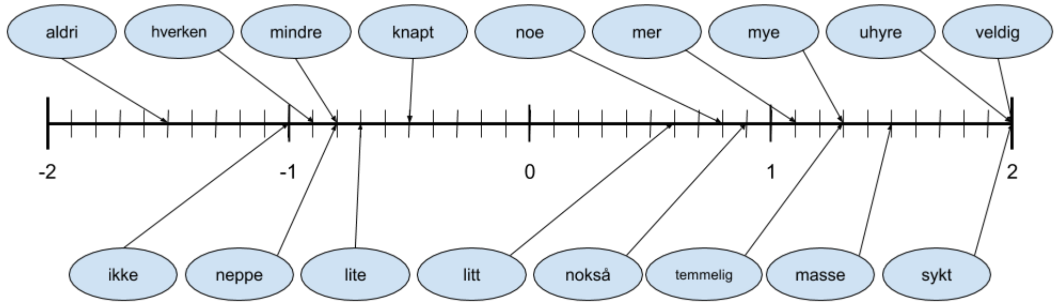


Figure 7.2: Valence shifters

7.4 Computational linguistics

Natural languages are difficult to model and interpret. Syntax can impact a sentence and change its meaning when moving a word to another position. Since sentiment analysis is a restricted area in NLP, we simplify the linguistics very much. However, some rules have to be taken into consideration since meaning can be altered. Valence shifters are important to consider as they can alter the polarity of a sentence. In the sentiment analysis system, valence shifters have been implemented as a dictionary (also called a map in other programming languages) where a valence shifter is a key mapping to a real value. Negators are mapped to negative real numbers, intensifiers are mapped to values larger than one and diminishers are mapped to values between zero and one. When finding a valence shifter in the text, the valence of the entire sentence (initially zero) is multiplied with the value. The mapping is shown in 7.2.

7.4.1 The problem of Norwegian sentiment analysis

The main challenge of doing Norwegian sentiment analysis is the lack of lexical resources. English sentiment analysis systems can utilize large corpora, which are annotated with sentiment values. Another useful resource is sentiment lexica, such as *SentiWordNet*[1], which explicitly state the positivity, negativity or lack thereof for a given word. Unfortunately, there exist no widely employed gold standard annotated corpus for use in Norwegian sentiment analysis. The following section explain an approaches to deal with this problem.

[1] http://sentiwordnet.isti.cnr.it/

7.4.2 Creating mapping to SentiWordNet

The attempted approach tried to utilize the sentiment lexicon SentiWordNet. It appears to be a good aid in a sentiment analysis system since sentiment values have been computed beforehand. (Øye, 2015) used SWN in his master thesis by translating text through the Bing translation service (currently named *Microsoft Translator*). After translation, lookup in SWN was performed, thereby obtaining cross-lingual lookup.

This thesis propose a slightly different approach for obtaining sentiment values. Since SWN is based on Princeton wordnet[2], a mapping from Norwegian words to PWN would suffice to determine sentiment values for Norwegian words. Such a mapping already exists in the OMW described in section 4.2.2, where 5883 words has been mapped to 4455 synsets in the Princeton wordnet. An extended open multilingual wordnet (Bond and Foster, 2013) has also been added to the set, resulting in a total of 10074 Norwegian words that are directly applicable for lookup in SentiWordNet.

The Norwegian language contains much more than 10 074 words though. The Language Council of Norway (Hovdenak, 2009) claims there is at least 300 000 Norwegian words. This is however dependent on the definition of a word. For instance, in Norwegian there is a word, *sykkelsete* which means bicycle seat. Over 60 "words" in Norwegian dictionaries start with the word sykkel, it is therefore hard to count the number of words since it is hard to capture all compound words.

In order to extend the initial set of words, the Norwegian wordnet *Norsk ordvev* was consulted. Norsk ordvev contains about 50 000 synsets of nouns, verbs, adjectives and adverbs in addition to 250 000 synsets of proper nouns. The synsets were translated through the Google Translate Python Client before an extra lemmatization step was performed on the translated words due to Google Translate sometimes returning an inflected form, even though the input was uninflected.

After translating a synset from Norsk Ordvev, the corresponding PWN-synset was found by lo-

[2] http://wordnet.princeton.edu/

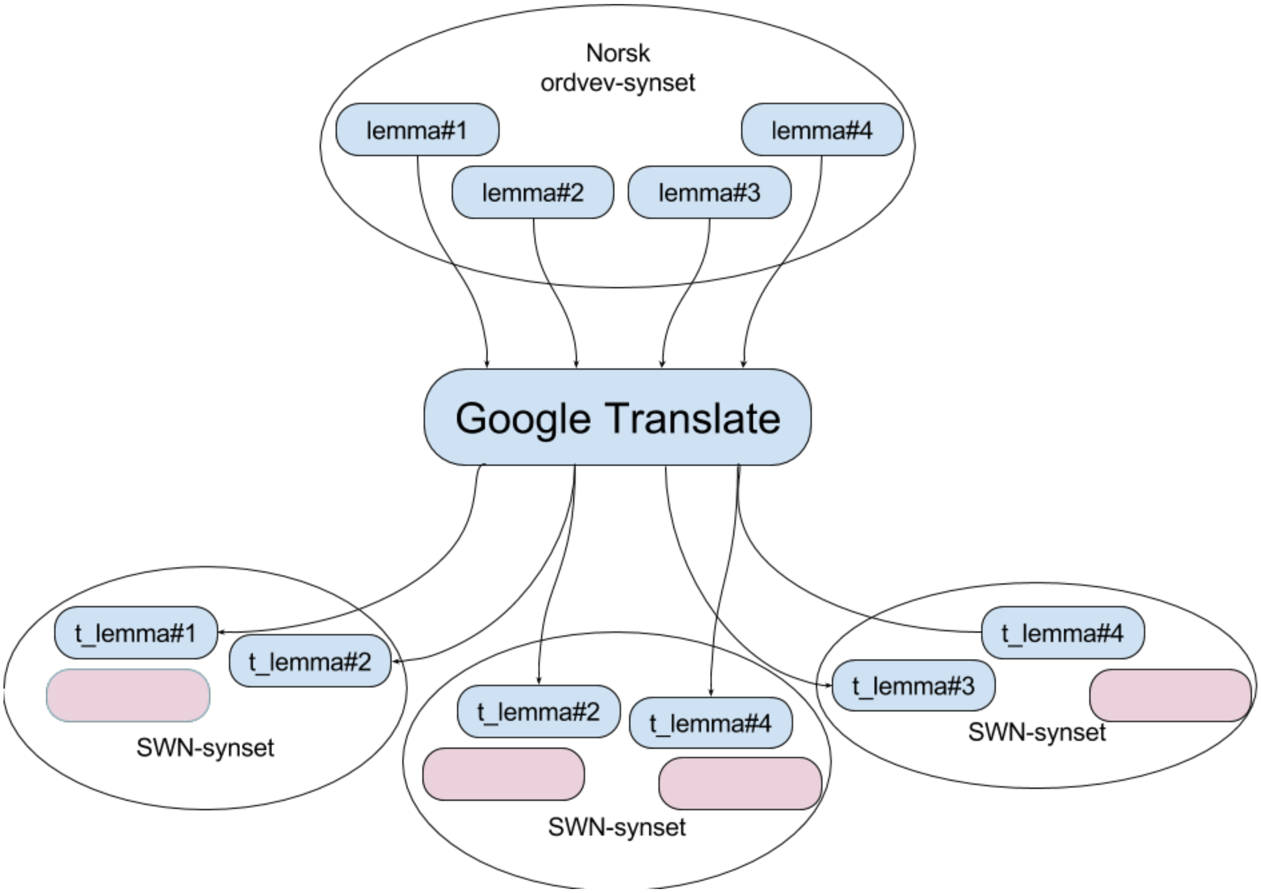


Figure 7.3: The mapping process between Norsk ordvev and SentiWordnet. The selected synset is the one where the translated (blue) lemmas have the lowest average sense number.

cating all PWN-synsets containing at least one of the translated lemmas (see figure 7.3). The synset with the lowest average sense-number of the translated lemmas was then chosen as the corresponding PWN-synset. This process is performed for each synset in Norsk ordvev.

7.4.3 Custom lexicon for banking reviews

Reviews of banks have a tendency to resemble each other, the reviews are all from the same domain of banking reviews. For this reason, the proposed sentiment analysis system would probably not perform well on other sources like newspaper articles. The resemblance can, however, be exploited to achieve a higher accuracy by extracting words frequently associated with each polarity class.

The list of domain-dependent, polarized words is obtained by extracting unigrams and bigrams from the text. Before obtaining n-grams, however, stopwords and names of banks are removed from the text. Stopwords are common words of a language, that are usually removed before natural language processing. The NLTK-library contains a corpora of stopwords, including a list of 176 Norwegian stopwords. Neither stopwords nor names of banks, should be any indicator of the polarity. If a bank receives many positive reviews, n-grams would assume the name of the bank is an indicator of sentiment. This is not a desired outcome since we want to obtain words that indicate negativity or positivity, across the entire bank domain.

After removing stopwords, names and punctuation marks from strings, the most frequent unigrams and bigrams are extracted from the resulting text. A maximum entropy classifier from the NLTK-library is employed to classify classes based on presence of the n-grams. The classifier uses weights associated with each feature to classify instances. By finding the features with the highest weights we obtain a bank domain lexicon which is used in conjunction with other features.

7.4.4 Sentiment target detection

One of the challenges in sentiment analysis is locating the targeted aspect of a sentence. This problem consist of two parts: detecting which bank the review is targeting (recognizing entity), and detecting which aspect of the bank is being targeted. Banks are complex organisations, with different services which can obtain sentiments. (Wei and Gulla, 2010) performed classification with a sentiment ontology tree (SOT), which taught a hierarchical learning algorithm how to detect aspects targeted by reviews.

The correct detection of the targeted entity is a higher priority than correctly detecting aspects. A company monitoring the web for sentiments might miss a review if do not think it concerns them. The detection of the wrong aspect might misguide the focus away from what was positive or negative, but the overall polarity score of the bank would still be the same.

In order to detect sentiment targets, different names (spelling) of each bank are listed of each banking entity. Any occurrence of the following names are marked as Sparebank1 SMN's bank

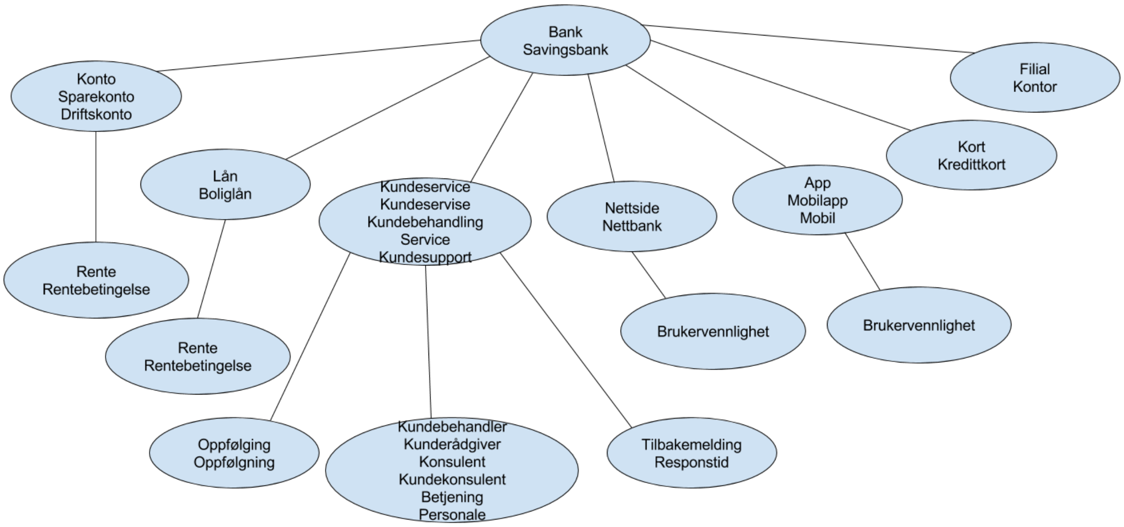


Figure 7.4: A sentiment ontology tree in the bank review domain

object.

- Sparebank 1 SMN
- Sparebank1 SMN
- Sparebank 1 Midt-Norge
- SMN

The position of all mentioned banks are logged, and all words of each sentence are marked as targeting the bank to which they are closest to in the sentence. If no banks are mentioned, the target is assumed to be the company which the review targets as a whole.

Aspects are listed in a hierarchical manner as seen in figure 7.4, with a banking entity as the root node. All banks are assumed to have the same aspect hierarchy, thus facilitating comparison of multiple banks' aspects (e.g. comparing the customer service of DNB and Sparebank 1 SMN).

The words closest to an entity in a sentence is marked as targeting that entity (if no bank is named, the review target is assumed). The words closest to a detected aspect is marked as targeting that aspect, until a noun is found. All aspects are nouns, and when a new noun is found, it is assumed the aspect is no longer the subject of following sentiment.

The system support three optional parameters affecting sentiment lookup:

–filter-banks

–aspect-features

–propagate-aspect-sentiments

The filter-banks parameter tells the feature extractor to ignore all sentiments obtained from words it believes is targeting other banks.

The aspect-features parameter tells the feature extractor to include each aspect as an entry in the feature set. This is useful as the classifier can learn which aspects that are most important to customers.

Finally, the propagate-aspect-sentiments parameter tells the extractor to propagate all aspect-targeting sentiment values up the aspect tree. That way, a sentiment towards an aspect is targeting all nodes between the aspect and the root (inclusive). This is intuitive, a sentiment towards responstid (response time) also indirectly targets kundeservice (customer service), as well as the bank itself.

7.5 Feature selection

Feature selection is the process of choosing which features to retain before applying machine learning algorithms. The goal of feature selection is to find the subset of the features yielding, the best results in terms of accuracy. Excessive features, without any contribution to the classification result, usually results in overfitting (see section 2.7.2).

In order to determine the best set of features, the features are grouped into the categories seen in tables 7.1, 7.2, 7.3, 7.4, 7.5 and 7.6. With this grouping, we try all possible feature set combinations of all sizes (except the empty set). This results in $\sum_{k=1}^{n} \binom{n}{k}$ combinations. With 6 feature set groups, this gives 63 distinct combinations [3].

[3] https://www.wolframalpha.com/input/?i=sum+from+k%3D1+to+6+binom(6,k)

Category	N-grams
Features	unigrams bigrams

Table 7.1: The n-gram feature category

Category	Emoticons
Features	numberOfPositiveEmoticons numberOfNegativeEmoticons

Table 7.2: The emoticon feature category

Category	Part-of-speech-frequencies
Features	numberOfNouns numberOfVerbs numberOfAdjectives numberOfAdverbs numberOfPronouns numberOfInfinitiveMarkers numberOfInterjections numberOfConjunctions numberOfSubjunctions numberOfUnknown

Table 7.3: The PoS-frequency feature category

Category	Terminal marks
Features	precenseOfExclamationSign precenseOfQuestionSign

Table 7.4: The presence of terminal marks feature category

Category	Text length
Features	numberOfWords averageSentenceLength averageWordLength numberOfNegations

Table 7.5: The text statistics feature category

Category	Text analysis (with SentiWordNet lookup)
Features	positiveScore (for each aspect) negativeScore (for each aspect)

Table 7.6: The sentiment-lookup feature category

7.6 Machine learning with scikit-learn

The final step of creating a model for sentiment classification is applying machine learning (ML). In the experimentation, multiple learning algorithms from the scikit-learn-library were tested with various parameters. The following algorithms were applied to this problem:

- Gaussian Naive Bayes classifier[4]
- Linear support vector classifier[5]
- Maximum entropy classifier (logistic regression)[6]

The testing of the models was conducted with 5-fold cross-validation to find the model which yields the best accuracies across the entire model.

[4] http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html

[5] http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

[6] http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Chapter 8

Results

In this chapter the results of the implemented sentiment analysis system will be presented and evaluated.

8.1 Inter-annotator agreement

As two individual people annotated the dataset, one should expect some disagreement between the two. (Mozetic et al., 2016) examined the limits of automatic sentiment analysis, and found that in their general domain dataset, inter-annotator agreement meassured by Krippendorff's alpha was less than 70%.

Krippendorff's alpha metric was employed to compute inter-annotator agreement between the annotators. It supports considering the annotations as either nominal (unordered) or ordered. The results were surprisingly good, inter-annotator agreement of bank-dataset achieved up to 95.5% agreement! The results are summarized in table 8.1. The inter-annotator agreement probably would not have been this correlated if the dataset was not domain-restricted. Re-

Included classes	Nominal	Ordered
Subjective/objective	66.1%	66.1%
Polarity	80.3%	92.5%
Both	86.9%	95.5%

Table 8.1: Krippendorff's alpha scores for the two annotators of the bank reviews

n	Feature classes	Best accuracy
1	Bank lexicon, Emojis, Terminal marks, Sentiment lookup	0,764 (MaxEnt)
2	Bank lexicon, Emojis, Terminal marks, Text lengths, Sentiment lookup	0,759 (MaxEnt)
3	PoS, Bank lexicon, Emojis, Terminal marks, Text lengths, Sentiment lookup	0,755 (MaxEnt)
4	PoS, Bank lexicon, Emojis, Terminal marks, Sentiment lookup	0,752 (MaxEnt & SVC)
5	PoS, Bank lexicon, Terminal marks, Sentiment lookup	0,752 (MaxEnt)
6	Bank lexicon, Terminal marks, Sentiment lookup	0,752 (MaxEnt)
7	Pos, Emojis, Terminal marks, Sentiment lookup	0,750 (SVC)
8	Bank lexicon, Emojis, Sentiment lookup	0,750 (MaxEnt)
9	PoS, Bank lexicon, Terminal marks, Text lengths, Sentiment lookup	0,750 (MaxEnt)
10	PoS, Bank lexicon, Sentiment lookup	0,748 (MaxEnt)
11	Pos, Emojis, Sentiment lookup	0,748 (SVC)
12	PoS, Bank lexicon, Emojis, Text lengths, Sentiment lookup	0,745 (MaxEnt)
13	PoS, Bank lexicon, Emojis, Sentiment lookup	0,743 (MaxEnt & SVC)
14	Bank lexicon, Emojis, Text lengths, Sentiment lookup	0,743 (MaxEnt)
15	PoS, Bank lexicon, Text lengths, Sentiment lookup	0,743 (MaxEnt)

Table 8.2: The 15 highest scoring feature set class combinations. The last column denote the best accuracy obtained with a classifier.

views of banks are often polarised, honest, direct and without sarcasm, leading to high levels of agreement.

8.2 Sentiment analysis evaluation

The implementation used features from 6 classes of features to predict polarity of reviews in the banking domain. The features differed in complexity, some only tested for presence of a character, while others were the result of natural language processing, using cross-lingual lookup in SentiWordNet.

8.2.1 Optimal feature selection

Figure 8.1 shows how the classification algorithms perform with the 15 feature class sets which obtained the highest accuracies (with any classifier). The feature classes are shown in table 8.2. It is easy to observe that the sentiment-lookup class had an impact on classification accuracy. We see in table 8.2 that all of the 15 highest scoring feature sets had taken sentiment values from SWN as input. Training with all feature classes results in a lower accuracy, than when

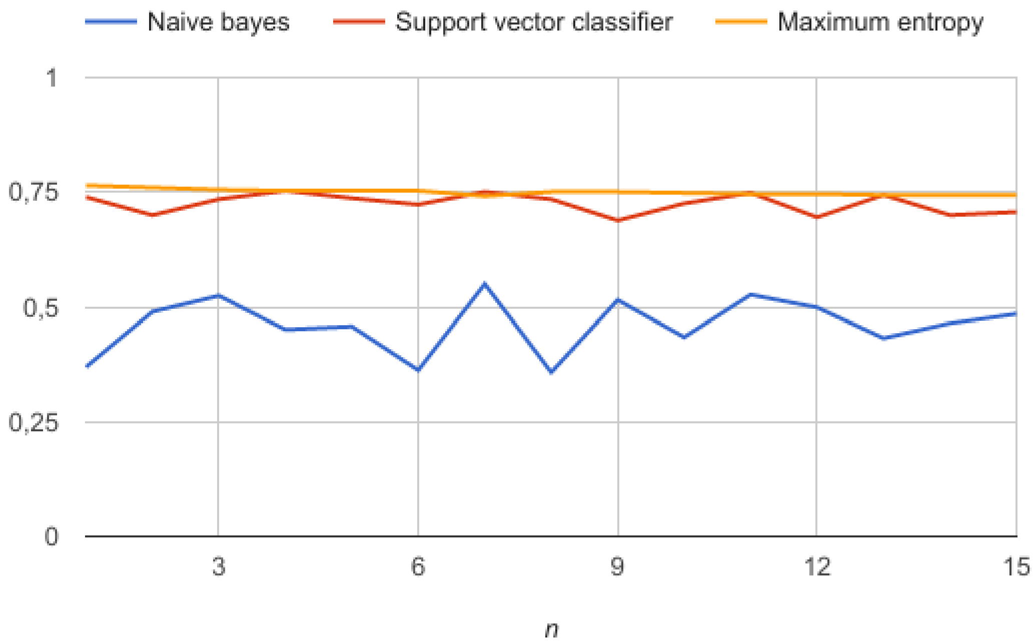


Figure 8.1: Feature sets obtaining highest accuracies

	Negative	Neutral	Positive
Negative	61	101	19
Neutral	13	30	3
Negative	11	126	67

Table 8.3: Confusion matrix for the naive bayes classifier

selecting a subset of the classes. This goes against intuition as additional information should not worsen the accuracy of a "proper" learner. However, it is probably due to the training data being overfitted to irrelevant features, thus decreasing the accuracy of the previously unseen testing data.

The most interesting feature set is the one yielding the highest accuracies which is probably the best model. The best set consist of the feature classes "bank lexicon", "emojis", "terminal signs" and "sentiment lookup". The confusion matrices of these feature sets are shown in tables 8.3, 8.4 and 8.5 yielding the presicion, recall and f-measure in figures 8.2, 8.3 and 8.4

The annotated dataset contain 431 reviews, but only 46 of these are neutral (either subjective or objective). As seen in figures 8.2, 8.3 and 8.4 the neutral polarity class has a low f-measure compared to the other classes. This trend applies to all three classifiers. The low f-measure suggests the classifier is either bad at classifying truly neutral reviews as neutral (recall), detecting

	Negative	Neutral	Positive
Negative	129	8	44
Neutral	18	12	16
Negative	21	2	181

Table 8.4: Confusion matrix for the support vector classifier

	Negative	Neutral	Positive
Negative	135	2	44
Neutral	18	8	20
Negative	18	1	185

Table 8.5: Confusion matrix for the maximum entropy classifier

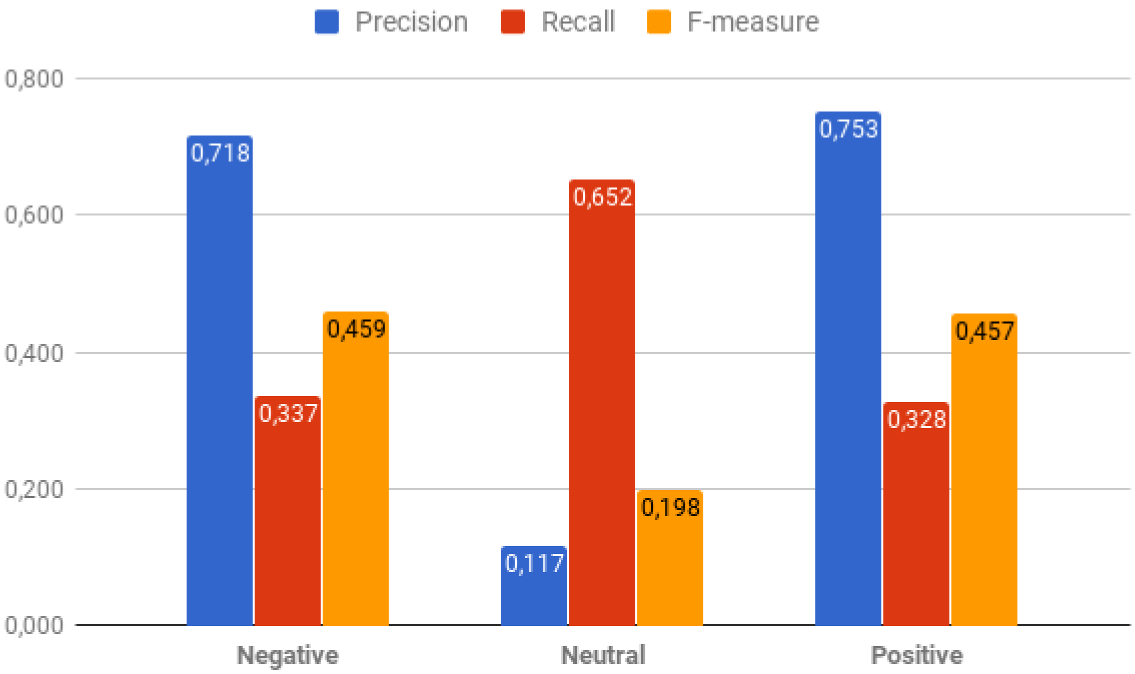


Figure 8.2: Precision, recall and f-measure values for the naive bayes classifier

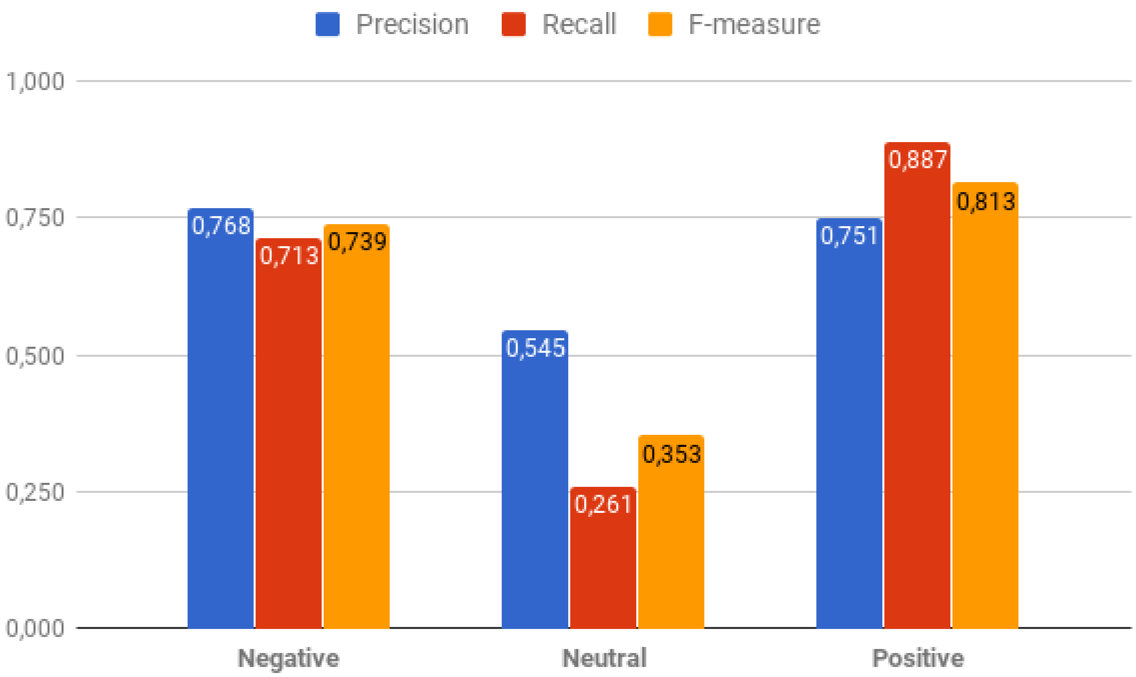


Figure 8.3: Precision, recall and f-measure values for the support vector classifier

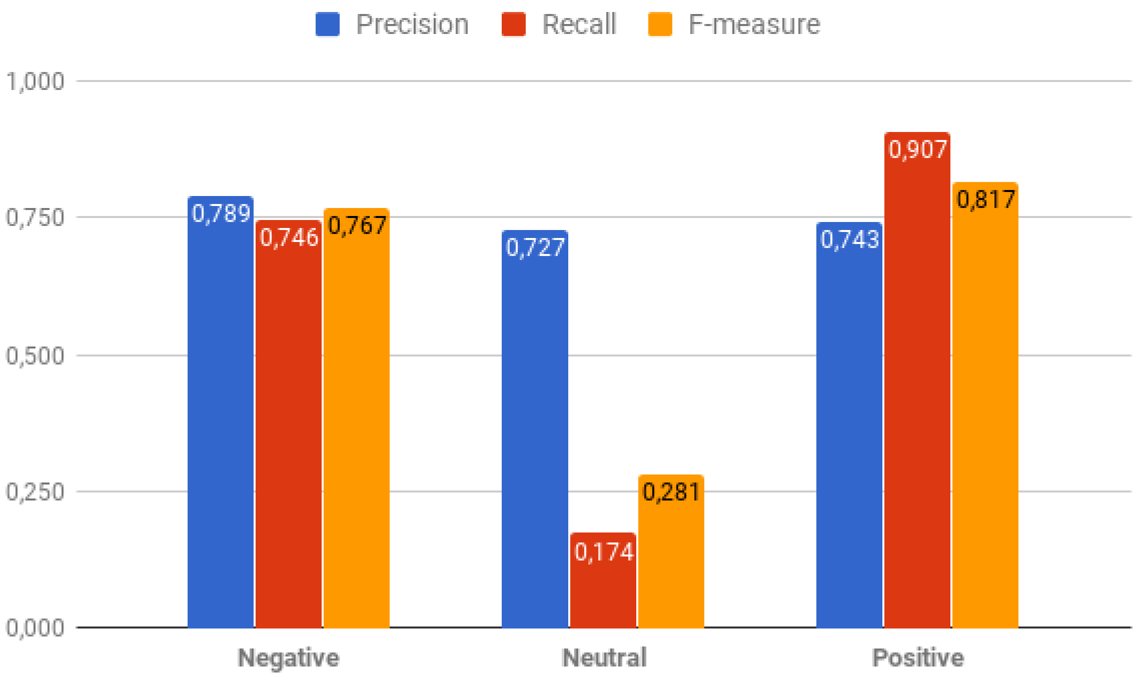


Figure 8.4: Precision, recall and f-measure values for the maximum entropy classifier

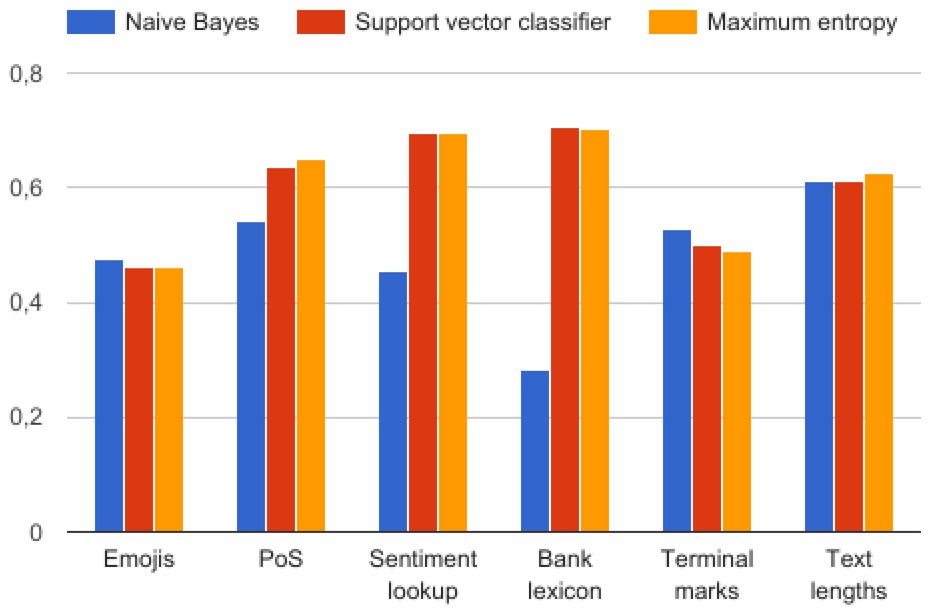


Figure 8.5: Evaluation metrics with single feature set classes

that a review is not neutral (precision), or both. The naive bayes classifier has a high recall of neutral reviews, but this is because it has predicted that there is 257 neutral reviews. The f-measure is still low here because such a small proportion of the predicted reviews were neutral.

Figure 8.5 show the accuracy of each feature class alone.

8.2.2 Effect of N-grams

Section 7.4.3 described a process for extracting words associated with sentiment in the already obtained dataset. The unigrams and bigrams obtained are shown in table 8.6.

Presence of unigrams and bigrams were a surprisingly effective indicator for capturing sentiment. When restricting the feature set to only presence of n-grams, the accuracy reaches 70.3%, which is comparable to that of sentiment lookup. The n-grams are not annotated, it is the responsibility of the machine learning algorithm to decide if they should be associated with polarity.

Unigram	**Bigram**
gebyrer	ny kunde
konto	bytter bank
del	søkte lån
lenger	super kundeservice
nettbanken	andre banker
flere	bedre tilbud
kommer	ringe tilbake
få	betale regninger
fått	god oppfølging
alt	bedre betingelser
nettbank	lang tid
hos	enkel nettbank
rask	dine erfaringer
godt	helt ok
penger	få tak
aldri	god service
erfaring	bytte bank
igjen	åpne konto
rente	tok kontakt
mer	dag 1
andre	hatt behov
god	raskt svar
fikk	nytt kort
betale	aldri opplevd
dårlig	fikk avslag
får	grei nettbank
hatt	veldig dyrt
mulig	del dine
hjelp	aldri problemer
må	veldig fornøyd
kunde	hatt kundeforhold
gode	
komme	
ting	
kunder	
veldig	
raskt	
tar	

Table 8.6: The n-grams in the bank lexicon

	GaussianNB	LinearSVC	Maximum entropy
Bank filtering, aspect-features and propagation	0,457	0,680	0,708
Aspect-feats and propagation	0,464	0,684	0,703
Aspect-features	0,490	0,677	0,687
Bank filtering, aspect-features	0,503	0,677	0,687
Bank filtering and propagation	0,643	0,696	0,701
Propagation only	0,636	0,689	0,703

Table 8.7: Accuracies with various parameters for sentiment-lookup

8.2.3 Target detection

Optionally, the system can detect the target of each sentiment and ignore sentiments directed at targets of no interest, as described in section 7.4.4. This is important in brand sentiment analysis because sentiments towards other entities/competitors can clutter the results we actually seek. To measure the impact of filtering sentiments not directed at the target, we train the machine learning algorithms with the sentiment-lookup feature sets, both with and without filtering. This is done with the "–filter-banks" and "–propagate-aspect-sentiments" parameters. Without the second parameter, all sentiments which do not target the bank-aspect explicitly, would be ignored.

To measure how the sentiment ontology tree of bank aspects affect the classification, the parameter "–aspect-features" is used, which includes all aspect nodes as entries in the feature set.

Figure 8.6 shows that the Naive Bayes and support vector classifiers, yield higher accuracies by applying bank filtering. They increased from 63.6% to 64.3% and 68.9 to 69.6%, respectively. Maximum entropy, however, showed a decrease in accuracy from 70.3% to 70.0%. This is unfortunate since maximum entropy is the overall best performing learning algorithm, thus bank-filtering alone does not increase the maximal accuracy achieved by the system.

The maximal accuracy is obtained when applying all parameters with a maximum entropy classifier, which gives an accuracy of 70.8%. Curiously, the accuracy of the Naive Bayes classifier drops dramatically when aspects are included as entries in the feature set. This might be due to the conditional independence assumption underlying the algorithm, as aspects are organised in a tree hierarchy and therefore dependent. The dependency increase when values

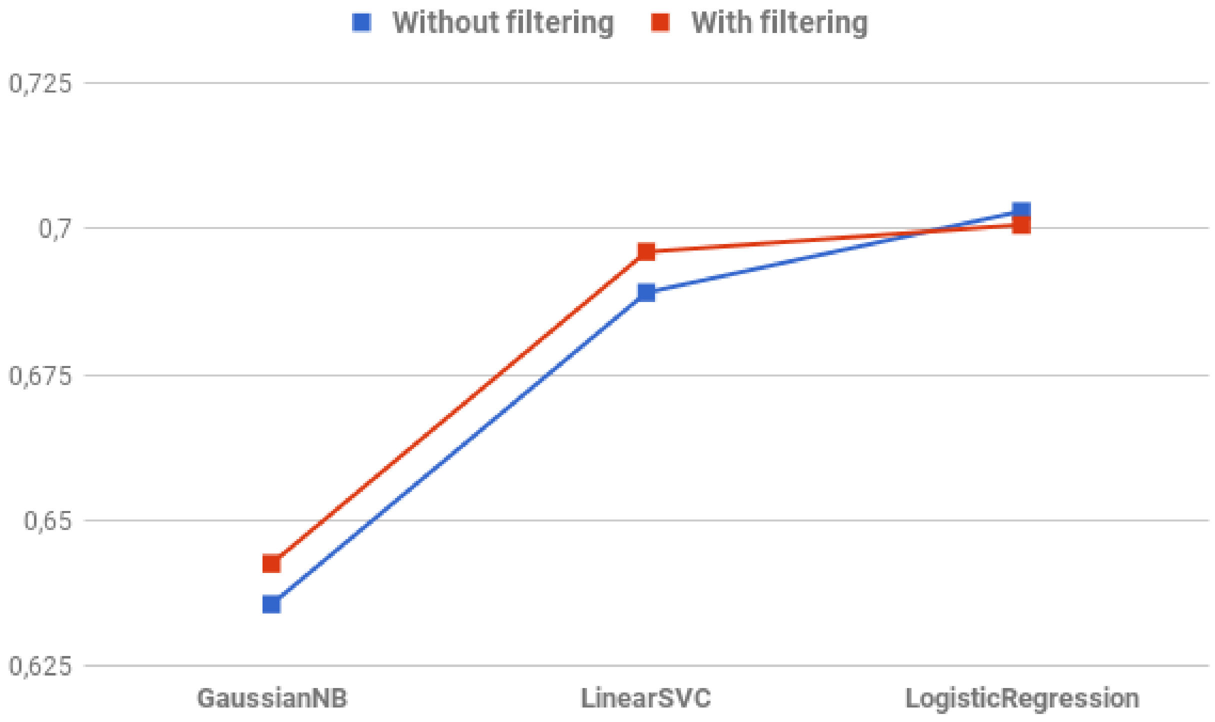


Figure 8.6: Accuracy before and after bank-filtering

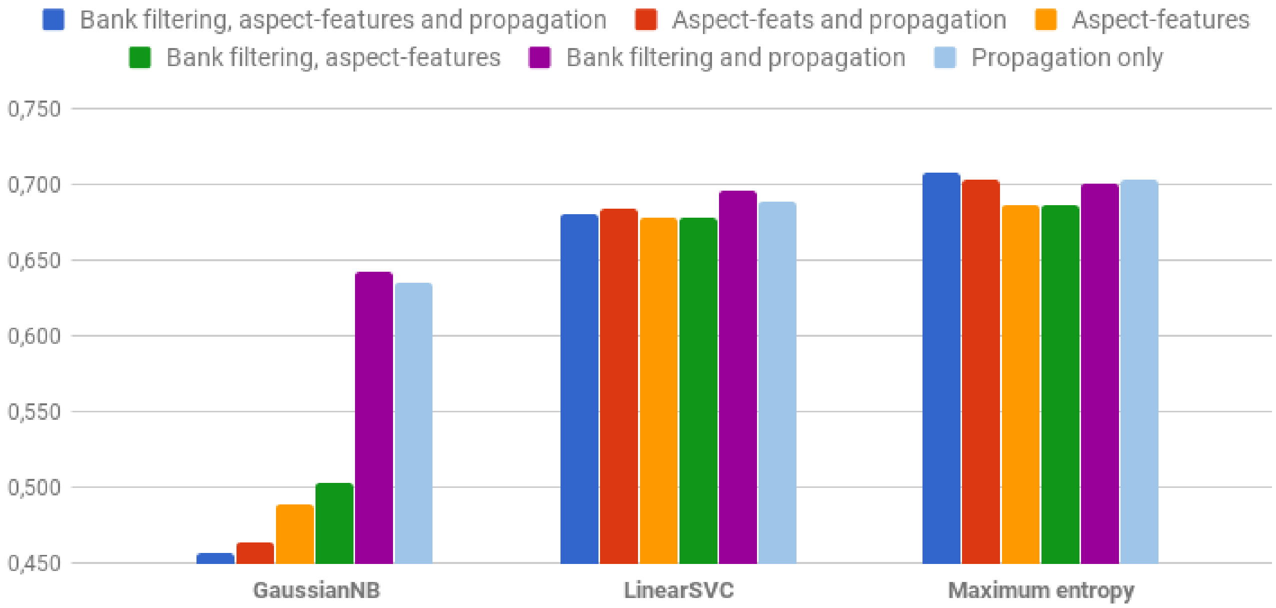


Figure 8.7: Accuracy with different parameter combinations

	GaussianNB	LinearSVC	Maximum entropy
Using only OMW	0,631	0,649	0,649
Using OMW with the new mapping	0,635	0,689	0,703

Table 8.8: Accuracy with and without the new mapping

are propagated, and we see that the combination of bank-filtering and aspect-features (without propagation) yields a higher accuracy.

8.3 Norsk ordvev to SentiWordNet-mapping

Obtaining sentiment scores from SentiWordNet has been a goal since the commencement of this thesis. This has indeed been proven difficult by the lack of Norwegian resources with a mapping to the corpus. Section 7.4.2 proposed a method for obtaining cross-lingual lookup in SentiWordNet. The custom mapping from Norsk Ordvev to SentiWordNet has gone through a series of refinements, and will now be evaluated by comparing it with the mappings in the Open multilingual wordnet.

Due to the ambiguous nature of natural language, selecting the correct synset is a challenge even in English sentiment analysis. The lemma *rude* occurs 3 times in SWN with both positive and negative sentiment values. Selecting which synset is the correct synset is therefore not trivial, and an erroneous selection will decrease the accuracy. Furthermore, the lemma *good* occurs 27 times (21 adjectives, 4 nouns and 2 adverbs). The Norwegian language also contain ambiguity. This makes sentiment analysis with such a mapping challenging, since uncertainty is present in whether the correct synset in Norsk ordvev has been chosen, and if the mapping is correct.

By executing the approach described in section 7.4.2 a set of 33 872 mappings were compiled. The effect the mapping had on sentiment-lookup compared to only using the mapping from OMW, is shown in table 8.8. The maximum entropy and support vector classifiers showed a significant increase in accuracy with the new mapping. Naive bayes on the other hand is nearly unaffected.

8.3.1 Mapping accuracy

To evaluate this mapping, it is compared to the existing mapping in the OMW, mentioned in section 4.2.2 using the Sørensen-Dice coefficient (SDC) which is equivalent to F-measure. The synsets (of same PoS) in the OMW is grouped together for each (lemma, pos) combination. These synsets are then compared with all synsets obtained from lemma-lookup in Norsk Ordvev combined with the custom mapping. The similarity of these two lists of SWN-synsets are then averaged across all words in OMW.

The average SDC-value was 32.2%. The might appear to be a poor result, but the fact that SentiWordNet is so large compared to Norsk Ordvev and the OMW might lead to the two mappings capturing separate segments. A manual inspection of the results also reveals that the various synsets obtained from a word had similar sentiment values inside the same Part-of-Speech. The custom mapping emphasized to capture the most popular sense of each lemma. As expected, the custom mapping has a lower average sense number in SentiWordNet compared to OMW's mapping (1.69 versus 1.90).

Chapter 9

Conclusions

Section 1.2 devised three research questions that will now be answered.

- How can lexical sentiment resources be used in brand reputation analysis?

One of the most challenging aspects in this project has been the lack of Norwegian lexical resources that can be utilized in a sentiment analysis context. There is no existing connection between the Norwegian and English wordnets except for a small subset. There is also no Norwegian sentiment resources available. A lot of the focus of this work has been directed towards solving this problem, and a custom mapping between Norwegian synsets and English synsets has been created. The correctness of such a mapping is hard to evaluate since the Princeton wordnet contain separate synsets for minor variations of the same thing. Some correlation was observed when comparing it to the already existing mapping in the OMW, but it was not very high, only 32.2% of the Norwegian synsets refered to the same PWN-synset.

- How can semantic resources about group structures improve the analysis of brand reputation?

In section 7.4.4, an approach for detecting sentiments towards aspects and sub-aspects present in the text. The approach was inspired by (Wei and Gulla, 2010) who used a sentiment ontology tree, but instead of teaching a hierarchical algorithm to recognize aspects, the aspect words manually labeled with lemmatized words and their synonyms.

Companies conducting brand analysis can benefit from this by labeling their brands as aspects. A brand might have sub-brands as described by (Marchak, 2015). An example is the iPhone which is a sub-brand of Apple's general brand. By organizing brands in the same way as this thesis has organized bank aspects, a sentiment analysis might be improved due to increased information of subjects in text, and the relationships between them. A sub-brand which is performing badly in a market might affect a parent brand, and monitoring this with sentiment analysis helps companies stay aware of their brand's reputation.

- What characterizes the average positive or negative statements about brands?

Statements about brands are often very clear, at least in reviews of the banking sector. The dataset used in the implementation contained very clear positive or negative sentiments, based on their experiences with the banks. The reviews are rarely neutral, reviewers are often either very pleased or very unhappy about their bank.

Negative statements in reviews do tend to be longer than positive reviews. Unhappy customers are often willing to write long reviews explaining their negative encounters with their bank in great detail. Happy customers, on the other hand, often keep their reviews short, e.g. "Kjempebra kundeservice!", "Beste banken:-)".

This project has required a lot of work, but there is still potential to improve the sentiment analysis. First of all, the feature extraction could be improved by incorporating more syntactic rules, thus increasing the understanding of sentence structures. The hierarchical sentiment ontology structure could also learn to weight sentiments that are propagated from lower levels since some aspects are more important than others.

Another improvement could be enlarging the dataset, possibly by including more data domains. However, for Norwegian sentiment analysis engine to be really successful, a Norwegian sentiment lexicon should be created. Lack of Norwegian lexical resources is the main bottleneck, preventing the widespread use of sentiment analysis.

Appendix A

Acronyms

AI Artificial intelligence

NLP Natural language processing

NLTK Natural language toolkit

OBT Oslo-Bergen tagger

OMW Open multilingual wordnet

PoS Part-of-Speech

PWN Princeton wordnet

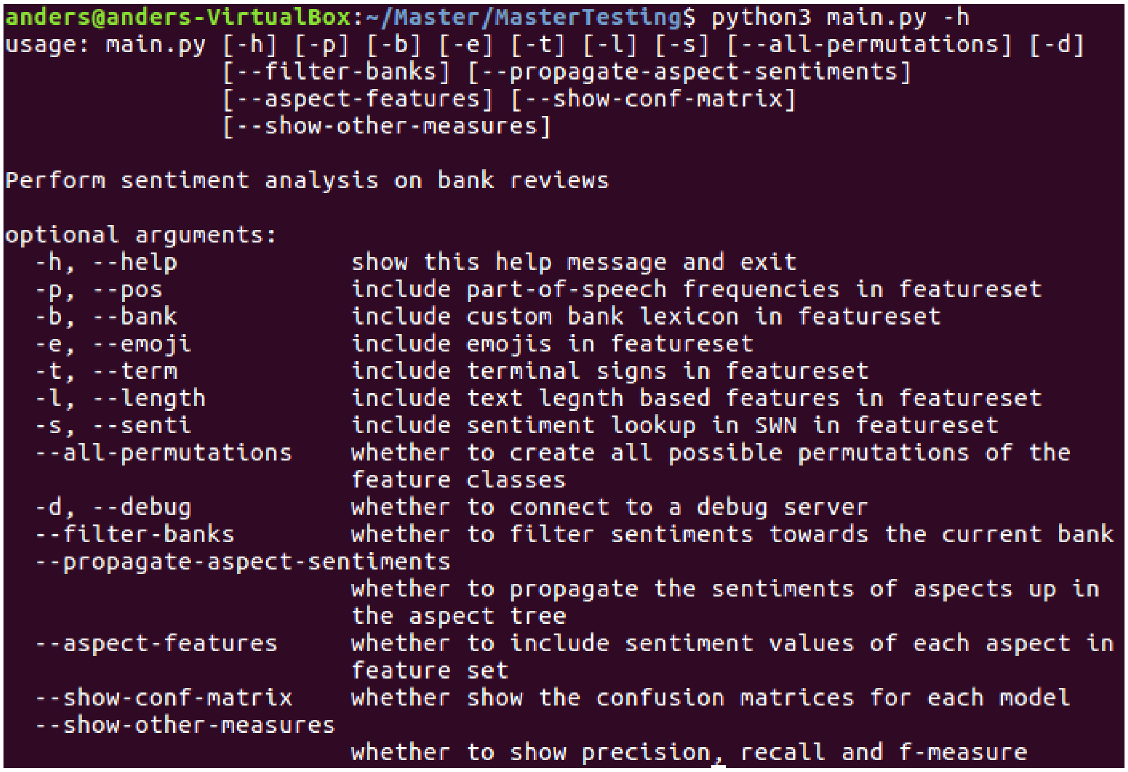
SVM Support vector machine

SWN SentiWordNet

Appendix B

The python script main.py starts the sentiment analysis process. It can be launched with several parameter combinations as seen in the following screenshot.

To run the program, the Oslo-Bergen-tagger must be installed one level above the project root as seen in the script "obt_tagger.py". An installation of Ruby and Python 3.5 is required with the python libraries PyYAML, Scikit-learn and NLTK. All libraries can be installed with pip.



```
anders@anders-VirtualBox:~/Master/MasterTesting$ python3 main.py -h
usage: main.py [-h] [-p] [-b] [-e] [-t] [-l] [-s] [--all-permutations] [-d]
               [--filter-banks] [--propagate-aspect-sentiments]
               [--aspect-features] [--show-conf-matrix]
               [--show-other-measures]

Perform sentiment analysis on bank reviews

optional arguments:
  -h, --help            show this help message and exit
  -p, --pos             include part-of-speech frequencies in featureset
  -b, --bank            include custom bank lexicon in featureset
  -e, --emoji           include emojis in featureset
  -t, --term            include terminal signs in featureset
  -l, --length          include text legnth based features in featureset
  -s, --senti           include sentiment lookup in SWN in featureset
  --all-permutations    whether to create all possible permutations of the
                        feature classes
  -d, --debug           whether to connect to a debug server
  --filter-banks        whether to filter sentiments towards the current bank
  --propagate-aspect-sentiments
                        whether to propagate the sentiments of aspects up in
                        the aspect tree
  --aspect-features     whether to include sentiment values of each aspect in
                        feature set
  --show-conf-matrix    whether show the confusion matrices for each model
  --show-other-measures
                        whether to show precision, recall and f-measure
```

Bibliography

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *LREC*, volume 10, pages 2200–2204.

Bannister, K. (2015). Sentiment Analysis: How Does It Work? Why Should We Use It?

Bea, F. (2012). Election night 2012 by the social media numbers.

Bond, F. and Foster, R. (2013). Linking and Extending an Open Multilingual Wordnet. In *ACL (1)*, pages 1352–1362.

Byrd, I. (2015). Ambiguous Sentences.

Dave, K., Lawrence, S., and Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International Conference on World Wide Web*, pages 519–528. ACM.

Distility (2010). Brand Analysis, Strategy, Systems: What do You Need?

Fjeld, R. V. and Nygaard, L. (2009). NorNet – a monolingual wordnet of modern Norwegian. In *Proceedings of the NODALIDA 2009 Workshop WordNets and Other Lexical Semantic Resources — between Lexical Semantics, Lexicography, Terminology and Formal Ontologies.*, volume NEALT Proceedings Series, Vol. 7, pages 13–16.

Goodson, S. (2012). Why Brand Building Is Important.

Hovdenak, M. (2009). Ord, ord og ord.

Jones, K. S. (1994). Natural Language Processing: A Historical Review. In Zampolli, A., Calzolari, N., and Palmer, M., editors, *Current Issues in Computational Linguistics: In Honour of Don Walker*, pages 3–16. Springer Netherlands.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining.* Number 16 in Synthesis lectures on human language technologies. Morgan & Claypool. OCLC: 855872441.

Marchak, E. (2015). The pros and cons of sub-branding and brand extension.

Marco, C. S. (2014). An open source part-of-speech tagger for Norwegian: Building on existing language resources. In *LREC*, pages 4111–4117.

Mozetic, I., Grcar, M., and Smailovic, J. (2016). Multilingual Twitter Sentiment Classification: The Role of Human Annotators. 11(5):e0155036.

Nasukawa, T. and Yi, J. (2003). Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd International Conference on Knowledge Capture*, pages 70–77. ACM.

Pang, B. and Lee, L. (2004). A Sentimental Education: Sentiment Analysis Using Subjectivity. In *Proceedings of ACL*, pages 271–278.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. 12:2825–2830.

Russell, S. J., Norvig, P., and Davis, E. (2010). *Artificial Intelligence: A Modern Approach.* Prentice Hall series in artificial intelligence. Prentice Hall, 3rd ed edition. LCCB: Q335 .R86 2010.

Sacco, N. (2016). Bad Historical Thinking: “History is Written By the Victors”.

Satmetrix (2017). What Is Net Promoter?

Smith, K. (2016). Building a Brand Research Strategy: How to Analyze Your Brand.

Vidya, N. A., Fanany, M. I., and Budi, I. (2015). Twitter Sentiment to Analyze Net Brand Reputation of Mobile Phone Providers. 72:519–526.

Wei, W. and Gulla, J. A. (2010). Sentiment Learning on Product Reviews via Sentiment Ontology Tree. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 404–413. Association for Computational Linguistics.

Øye, J. A. (2015). Sentiment Analysis of Norwegian Twitter Messages. Master's thesis.