UNIVERSITY OF BERGEN

Department of Information Science and Media Studies

MASTERS THESIS

Sentiment Analysis in Scandinavian Languages: Systematic Review and Evaluation

Author: Eirik Sjøvoll

Supervisor: Andreas Lothe Opdahl

November 29, 2019

Abstract

Natural Language Processing has seen a tremendous boost in popularity following the widespread use of the World Wide Web, and emergence of machine learning tools. The specific problem of sentiment analysis has become a popular topic with the availability of user generated content, from micro-blogs and the likes. But these data dependent problems have seen a larger jump in popularity in the international field, compared to low-resource languages, due to the availability of language specific data. This thesis seeks to delve into the problem of sentiment analysis research within some of these low-resource languages, specifically those of mainland Scandinavia, which are closely related languages. We perform a literature review to uncover popular research topics within this language specific field, and seek to find practical and theoretical work as well as resources within this field. Furthermore we perform experiments adapting international tools for these low-resource languages, and compare our results to that of the research, in order to further contribute to the language specific research field.

Contents

List of Figures

List of Tables

List of abbreviations

NLP Natural Language Processing

SA Sentiment Analysis

UGC User Generated Content

WE Word Embedding

POS Part-of-Speech

ML Machine Learning

NN Neural Network

AI Artificial Intelligence

RNN Recurring Neural Network

D2V Doc2Vec

TF-IDF Term Frequency–inverse Document Frequency

SVM Support Vector Machine

LR Logistic Regression

MLR Multinominal Logistic Regression

OLS Ordinary Least Squares

CBOW Continuous bag-of-words

PV-DBOW Distributed Bag-of-Words version of Paragraph Vector

PV-DM Distributed Memory Version of Paragraph Vector

VS Vector Size

NS Negative Sampling

MC Minimum Count

Chapter 1

Introduction

Sentiment Analysis, sometimes referred to as *opinion mining* or *emotion analysis*, is the concept of extracting sentiment or opinions about an entity, from natural language (Liu (2012)). It can be considered one of many problems within Natural Language Processing (NLP), and it is concerned with the problem of teaching machines to autonomously classify text sequences by sentiment polarity. Autonomously extracting sentiment from natural text can be immensely useful in a range of domains. For example, we can autonomously extract opinions about certain subjects, products or services, or even map geographical sentiment based on social media for emergency management.

The NLP field is quite vast, and has seen further expansion with the emergence of Machine Learning (ML) technologies. Techniques that utilize ML often rely on large amounts of example data, which can be hard to come by in relatively small languages. Even though the the NLP field is large, the technologies and discoveries are often language dependent, meaning that less spoken languages have fewer resources available. This is why we want to explore the current state of sentiment analysis tools, techniques and resources for lesser spoken languages, specifically for those in mainland Scandinavia.

Seeing as the Scandinavian languages, specifically the *Mainland Scandinavian languages* that include Danish, Swedish and Norwegian, are quite similar and are adapta-

tions of the Germanic language (Holmberg and Platzack (2005)), we think that many of the resources within these languages can be interchangeable. Therefore we are going to perform a systematic review of relevant literature, in order to construct an overview of the sentiment analysis field within Scandinavian research. Our study is inspired by the popular Systematic Literature Review (SLR) method, and systematic review in the software engineering, and information systems field (Fink (2019), Kitchenham et al. (2009), Petersen (2008), Petersen (2015)).

Furthermore, depending on our finds we want to experiment with some of these techniques alongside the resources available in order to compare approaches and results to existing work. The specific techniques we want to experiment with depend on the results of our literature review, but we will be focusing on tools and techniques surrounding ML, as this has been popularized lately, and have shown very impressive results.

1.1 Problem Formulation

The field of sentiment analysis has had a rapid growth alongside machine learning technologies, and the results from these techniques are getting progressively better alongside better tools and resources. The best results we have been able to find, report error rates as low as 1.55% for a binary text classification task (Yang et al. (2019)). But the results from Yang et al. (2019) are achieved using immense amounts of English language data, generated by users on review sites such as Yelp[1] and IMDB[2], among others. Seeing as these data are essentially generated by users, the same amount of data is simply not available for the lesser spoken languages, such as those of mainland Scandinavia. We therefore think that there is a need for more research on these low resource languages, which is why we seek to create more oversight of the field to further encourage related research.

Additionally, there exists many tools and techniques for NLP research and develop-

[1] https://www.yelp.com/dataset
[2] https://www.imdb.com/interfaces/

ment, which often come with a steep learning curve. This makes it hard for beginners to understand what everything is and how it differs. Overall, it makes NLP research confusing and hard to grasp, especially now that ML has become deeply integrated in the field. We want to make it easier for everyone with a general interest to try and experiment with these technologies, and hopefully make a meaningful contribution. In order to do so, we need more structure and oversight in the field.

One way of gaining such oversight, is to create a systematic review of relevant existing research, which requires considerable effort and supporting literature. Seeing as the mainland Scandinavian languages can be considered low-resource languages, there might also be few reported experiments in Scandinavian research. We will therefore also be conducting some experiments of our own with different tools and techniques, alongside Scandinavian language resources, to further contribute to the field of sentiment analysis within mainland Scandinavian languages.

1.1.1 Research questions

We have created some research questions that we think cover our problem decently. Due to our two approaches to the problem, we have made two main research questions with further sub-questions to clarify.

RQ1 What is the state of the art of sentiment analysis in Scandinavia?

- RQ1.1 What are the most common research topics in Scandinavian sentiment analysis?

RQ2 Are English sentiment analysis techniques suited for use on Scandinavian languages?

- RQ2.1 Do the results improve the state of the art in Scandinavian sentiment analysis?

1.1.2 Tasks

The main objectives in this Masters thesis are to create an overview of existing resources, techniques and methods for sentiment classification in mainland Scandinavian languages, and experiment with English tools and techniques to create sentiment classifiers for these languages using language specific resources. This work is likely not able to cover all there is within the field, but we hope it can serve as good grounds for further research to take place. In addition to the literature review, we will be conducting an evaluation of existing technologies and how they fare in mainland Scandinavian languages. The main objectives for this thesis is therefore to:

1. Find Scandinavian research and resources for sentiment analysis.
2. Create an overview of popular research topics within the sentiment analysis field in Scandinavia.
3. Test and evaluate techniques to further expand upon the overview.
4. Discuss findings and results from the review and experiments.

1.1.3 Contributions

Through this thesis, we will make an empirical contribution to the research field of sentiment analysis in mainland Scandinavian languages, and hopefully make research easier and less daunting for others to build upon. We will also be contributing by experimenting with, and evaluating some popular techniques that have and have not been addressed in Scandinavian research, in order to get a broader coverage. Whatever practical work takes place during the thesis, will be made public and open-source using the popular code hosting platform GitHub[3].

The main contribution of this thesis, is therefore an empirical review of research, tools, techniques and resources for sentiment analysis in mainland Scandinavian languages. Backed up by practical experiments and evaluation of popular international techniques for sentiment analysis, using Scandinavian language resources.

[3] https://github.com/

1.1.4 Inspiration

The main inspiration for the thesis stems from a course at the University, specifically about big data for emergency management. In this course, students were expected to develop some sort of application that could be used for emergency management, based on available big data resources. We decided as a team to create a simple web application that classifies tweets with a lexical approach, specifically the sentiment lexicon from Nielsen (2011), which gave intriguing results. The vision we had was to create a web-dashboard with a geographical map, divided into municipalities, coloured by the average sentiment of tweets from the area. This idea was not something we managed to finish, but the idea stuck with us as a very interesting topic, seeing as all the data required for the project was freely available on the web.

But as we delved more into the topic of sentiment analysis, we realize that for local coverage, we would have to adapt to the local languages, which are lacking in resources at the time of this writing. Therefore we have decided to take a deeper look into the field of sentiment analysis, specifically for the mainland Scandinavian languages, as there seem to be lacking compared to the more resource-rich languages such as English.

1.2 Challenges

Since the thesis focuses on empirical research, one of the limitations is the extensive work required to perform it. Additionally, having no real experience in the field beforehand, and little experience with ML in general, experiments with advanced technologies can be tough. We therefore acknowledge that our efforts could have been better given more background knowledge in the field, and with these advanced technologies in general. We will discuss this further in chapter 6, section 6.3

1.2.1 What Remains to be Done?

As mentioned previously, this thesis is likely not enough to get a deep and thorough overview of the current state of sentiment analysis in Scandinavian languages, but it is

a good start. We believe that the work we perform in this thesis can be useful for the general field, and provides good grounds for further work, which we will be discussing further in chapter 6, section 6.4.

1.3 Method and Approach

If we wish to gain good oversight of the current state of sentiment analysis research in mainland Scandinavia, we will have to conduct a systematic review of related work, and include our own contributions from experiments and evaluation. This thesis is therefore in two parts, but the parts are related. The second part is dependent on finds from the literature review in order to evaluate and compare approaches and techniques. We use different methods and approaches for each part of the thesis, where the first part is the literature review, and the second part is experimentation and evaluation.

1.3.1 Literature Review

The first part of this thesis is to perform a literature review of relevant work within the field of sentiment analysis within mainland Scandinavian languages. We base our methods on the Systematic Literature Review (SLR) approach, which is a thorough approach to gaining good knowledge in a field (Fink (2019)). A more detailed explanation of methods and results for the literature review will be discussed in chapters 3 and 4.

1.3.2 Experiments and Evaluation

In addition to the review, we will be conducting some experiments with technologies and resources, that requires some technical effort. The technologies and data we are employing requires some manual preparation as well as parameter fine tuning. For the programming aspects of our experiments, we use Git and GitHub[4] for cloud storage and version control. The repository was made private during the working period since

[4] https://github.com/

we also use it to store notes, but will be made public after the thesis is finished and evaluated. An open repository removes the need for source code in the appendix.

We decided to focus our attention on Python based frameworks and tools, due to its simplicity, its wide use in scientific computing (Pedregosa et al. (2011)), and the fact that we are already familiar with the programming language reduces the learning curve and workload significantly, so we can focus on other aspects. A more detailed explanation of technologies, techniques and experiments will be discussed in chapters 3 and 5.

Chapter 2

Background

Throughout this thesis, we will be referring to different terms, tools and techniques. This chapter therefore contains descriptions and explanations of these underlying terms and techniques, so that we can focus on the topic in later chapters.

2.1 Sentiment Analysis

Sentiment analysis, often referred to as *opinion mining* or *emotion analysis*, is a field of study concerned with extracting sentiment polarity, or emotions towards an entity (Liu (2012)) from natural language. The field has seen an explosive interest after the 2000s, as there has been an increase in applications and industrial uses for the technique, following the expanse of the web (Liu (2012)).

These sentiments in text can be classified as negative or positive, which can further be used for a range of different applications. It can be considered closely related to the Natural Language Processing (NLP) task of text classification, which is concerned with automatically classifying text. In this thesis, we will be looking at sentiment analysis as a text classification problem. Using the text classification task, we can use different techniques and approaches to autonomously classify a text as positive or negative, which can be useful in a range of different applications such as emergency surveillance of social media, autonomous product or service enhancement based on user reviews,

toxic language and hate speech detection in social media, or even propaganda and false news detection in social media.

There are many interdisciplinary challenges within the sentiment analysis field, due to its focus on natural language understanding. While linguistics and computer science are the most prominent, one could argue that even psychology and sociology has a role to play when seeking to understand natural language. This in turn raises the complexity of the field to include linguistic problems specific to the language, and cultural sociology problems in regards to language understanding. In this thesis we will be focusing on the computer science related problems, seeing as we are not linguists nor psychologists.

2.1.1 Natural Language Processing

Natural Language Processing (NLP) is a collection of techniques regarding information retrieval from natural language. There are many different tasks within NLP, such as entity extraction, machine translation, text classification, and the list goes on. But what they all have in common, is the task of autonomously extracting information from human spoken and written language. Even though humans do not have many issues with understanding natural text, machines lack the capability of comprehending the meaning behind text, which is where NLP comes in. With the recent boom of machine learning techniques and technologies, many NLP tasks have seen a tremendous boost in efficiency and usability (Goldberg (2017)). And alongside the steadily increasing amount of User Generated Content (UGC) on the Word Wide Web (WWW), there is an increased need for autonomous information retrieval. One NLP task is concerned with autonomously classifying text as one of several proposed classes, where classes can be whatever you specify. For example, a text classification task can be concerned with classifying a text as either 0 or 1, or negative or positive, which would be considered a binary text classification task due to it only having two classes. There can also be several classes for a classification task, which is often referred to as multi-class classification.

2.1.2 Text classification

As mentioned previously, the field of sentiment analysis can be closely related to the NLP task of text classification. This means that we can use techniques on natural language to extract the sentiment of a linguistic resource such as text or speech. The text classification task can be topic and genera specific, meaning we can classify document topics or document genres (Ikonomakis et al. (2005)). In our case, we will be looking at extracting sentiment from written text, where all the texts are some sort of review. There are many different approaches to the task, some of which utilize binary classification, meaning they classify as one of two classes, or regression which is more concerned with which of several classes is the most similar. Specifically for the problem of sentiment analysis, there are many different approaches for classifying text, many of which utilize sentiment lexicons, some newer techniques use word embedding for semantic composition, and some of the newest use transformer based language modelling.

Additionally, according to Liu (2012), sentiment analysis problems are often divided into tasks of sentence level classification and document level classification. Sentence level classification would be to classify shorter text sequences, often sentences from social media, while document level classification is more concerned with classifying whole documents at a time, which can be useful for a range of tasks, for example autonomous library text categorization, and autonomous review classification on user reviews in regards to a product or service.

2.1.3 Sentiment lexicon

Sentiment lexicons, often referred to as *affective word lists* (Nielsen (2011)), is an approach to extracting sentiment polarity from text. This approach is quite simple, does not rely on machine learning tools and has been around for some time. We have found many different sentiment lexicons during the writing of this thesis, some from 2010 and 2011(Rosell and Kann (2010), Nielsen (2011), Bai et al. (2014), Hammer et al. (2014), Rouces et al. (2018a)). These lexicons work as a word dictionary with an applied label

for each word consisting of a value that represents the polarity of the given word. For example the word "bad" would have a label closer to -1, while the word "good" would have a label closer to +1. The lexicon is then used to calculate the average score of a sentence, based on the words in the sentence that also exist in the lexicon. Though this approach is a simple way of extracting general sentiment polarity from a texts, it poses some linguistic problems and it would not be able to correctly classify texts snippets such as "not bad" for instance. This is why the word embedding approach has become increasingly popular, since the embedding is capable of mapping word semantics in a vector space.

2.1.4 Word embedding

Even though neural network technologies and word vectors are considered new by many, they date as far back as Rumelhart et al. (1988)). *Word embeddings*, often referred to as *neural embeddings* (Levy and Goldberg (2014)), is essentially a technique where we train a machine learning model to create vector representations of words, in a matrix (Levy and Goldberg (2014), Mikolov et al. (2013)). When the vectors are created, words with similar meaning are mapped in the same vector space, so that we can group the meaning, or semantics, of words. The embedding can be used in several NLP tasks, including but not limited to, text classification.

2.2 Machine learning

Goodfellow et al. (2016) describes machine learning as Artificial Intelligence (AI) systems with the ability to gain knowledge by extracting patterns in data. Using these systems, we can automate any information retrieval task if there are enough training examples available for the method to learn from. These methods have become increasingly popular after the 2000s, since the training data available is growing alongside the adoption of the web (Liu (2012)). Though, as for most machine learning application, the data has to be structured correctly in order to be usable, and the results depend on the representation of data they are given (Goodfellow et al. (2016)).

2.2.1 Classification and Regression

Classification is an ML problem concerned with learning to predict a categorical value for an input, based on examples. Compared to regression, classification is focused on outputting a specific category, while the regression outputs a number (Alpaydin (2009)), which is not a fixed category. Both classification and regression are regarded as *supervised learning* methods (Alpaydin (2009)), which is one of the most common forms of ML (LeCun et al. (2015)) that utilize feature examples with an associated *label* or *target* (Goodfellow et al. (2016)).

Support Vector Machine (SVM), Logistic Regression (LR), and Multinominal Logistic Regression (MLR) are all examples of supervised learning techniques for classification (Goodfellow et al. (2016), Durgesh and Lekha (2010), Starkweather and Moske (2011)), that we will utilize during the thesis.

2.2.2 Overfitting and underfitting

Goodfellow et al. (2016) mentions the term *generalization*, which is a ML systems ability to perform well on previously unseen examples. Overfitting is a problem within ML techniques that happen when an algorithm learns its examples too well, including the potential noise in the data (Alpaydin (2009)), and is thus unable to generalize well. Underfitting is a similar issue that occurs when the generalization error, the rate of which the algorithm manages to correctly predict unseen examples, is too low (Goodfellow et al. (2016)).

2.3 Pre-processing

All supervised machine learning applications require data, or examples, to learn from (Ikonomakis et al. (2005)). In order to create useful sequential text data for word embedding and sentiment analysis, the data has to be cleaned and prepared for the task at hand. The cleaning process can vary between NLP tasks, and some tasks require a more fine grained process than others. For the task we are attempting, which is binay-

and multi-class sentiment classification, there are many approaches to cleaning the data, depending on the technique you are using to train a classifier.

Some of the techniques include tokenization, one-hot encoding, lemmatization, part-of-speech tagging, stop word removal, case normalization, symbol removal, dependency parsing, among others. We will explain some of these pre-processing techniques in this chapter, since many are referenced throughout the literature review and practical work.

2.3.1 Tokenization and one-hot encoding

Tokenization is the process of breaking up sentences into singular words, or "tokens" (Schütze et al. (2008)). The tokenization process can vary between tasks and languages. For instance, when tokenizing reviews we often stumble upon names with special symbols integrated, which poses problems for a tokenizer that only splits whitespaces.

When tokens are individual and mapped in a corpus, they often get encoded into binary sequence representations, often referred to as "One-hot encoding" to make machine training easier. The name "one-hot" comes from the fact that only one bit is true at a time (Harris and Harris (2010)), which gives us large sequences containing 1's and 0's, instead of text. This is because String values are often larger in size and would slow down the training process notably. Therefore the sentences are constructed anew using these encoded tokens to create sentences of word IDs. This makes it less readable for humans, but the point is not to train a human. Later in the process, the tokens are mapped in a matrix using their integer IDs, but can be coupled with the word String for human readable purposes.

2.3.2 Stemming, Lemmatisation and Part-of-speech

Words can exist in different states, and word lemma is just the canonical form of a word, for instance 'play' is the lemma of 'playing' and 'played'. Stemming and lemmatisation is the process of converting words to their lemma form (Schütze et al. (2008)). Both stemming and lemmatisation refers to the same task, but the approaches differ. Stem-

ming is a simple process of splitting words to achieve their lemma form, which does not always work. While lemmatisation uses grammatical attributes for a more complex approach with better results (Schütze et al. (2008)). The problem of tagging words with such metadata can be very tricky, and is often language dependent, so we find many mentions of the problem in the literature we review. Though we include work focusing on these problems, we will not personally focus on these problems since they are more on the linguistic side of the research.

2.4 Classification and Evaluation

There exists many approaches for classification of items. In this thesis, we are focusing on binary- and multi-class classification tasks with two or three classes to classify. The items we are classifying are text documents, and the classes are *Positive*, *Neutral* and *Negative* texts. A classification task does not have to be classifying text into polar classes, it could also be classifying fruit into color classes for instance, but we are focusing on text classification in this thesis. There exists many different ways of evaluating effectiveness of classification and prediction, but the most common metrics in such classification tasks are precision, recall and accuracy (Ikonomakis et al. (2005)). An additional metric we will be mentioning is F-score, which is related to the aforementioned metrics.

2.4.1 Confusion Matrix

A *Confusion Matrix*, often referred to as *Contingency Table*, is a matrix that holds values for determining the correctness of predictions. It contains values for *True Positive*, *True Negative*, *False Positive* and *False Negative* predictions. These values are further used to calculate different metrics such as *Precision*, *Recall* and *F-Score*. There also exist other metrics for evaluation, but these are the metrics we will be using in later chapters, as they seem to be the most common (Ikonomakis et al. (2005)). In a binary classification task, this matrix would be a two by two matrix, but with more classes come more prediction options, so the confusion matrix grows. In a multi-class classification task with

three classes, we would get a three by three confusion matrix. We use pre-made python modules from scikit-learn library[1] to calculate metrics in all of our experiments.

2.4.2 Accuracy

The accuracy metric is often used to depict model success, but can be misleading when the evaluation sets are skewed (Ikonomakis et al. (2005)). This metric is calculated as $A_i = \frac{(Tp_i+Tn_i)}{(Tp_i+Tn_i+Fp_i+Fn_i)}$, where Tp is true positives, Tn is true negatives, Fp is false positives and Fn is false negatives. This metric is essentially calculated by having all the correct predictions divided by the total examples, which does not change much from binary- to multi-class classification tasks. We use several metrics alongside the accuracy to better represent results.

2.4.3 Precision

Among the metrics we will be using in our experiment evaluation, is Precision. This is essentially a way of calculating results from a classification task based on a confusion matrix, to learn. The precision score is calculated as $\pi_i = \frac{Tp_i}{(Tp_i+Fp_i)}$ for each class, where Tp is true positives, and Fp is false negatives (Ikonomakis et al. (2005)). For a multi-class task with three classes, the precision would be calculated with all the correct predictions of a class, divided by all the predictions of the said class.

2.4.4 Recall

The recall metric is similar to that of the Precision metric, but is calculated with different contingency values. The recall is calculated as $\rho_i = \frac{Tp_i}{(Tp_i+Fn_i)}$ for each class, where Tp is true positives, and Fn is false negatives (Ikonomakis et al. (2005)). For a multi-class task, the score would be calculated using the number of correct predictions of a class, divided by the number of actual instances of the said class.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html

2.4.5 F-score

Another metric that builds upon precision and recall, is often referred to as *F-score*, *F1-score* or *F-measure*. This uses both the precision and recall calculations to get a better picture of the classification results. It can be calculated as $F_\beta = \frac{(\beta^2+1)\pi\rho}{\beta^2\pi+\rho}$, where π is the precision, ρ is the recall, and β represents the goal of the task (Ikonomakis et al. (2005)). The standard approach for calculating F1-scores for multiple classes, is to arithmetically sum the class specific F-scores and divide by the total number of classes, which is also referred to as the macro-average.

2.4.6 Macro-, Micro-, and Weighted-averages

There are different ways to calculate the average of each metric type. For the F-score, the macro-average we explained in the former section is the most common, but it is also possible to calculate the micro- and weighted-averages for both precision, recall and F-score. These calculations vary slightly, where the macro-averages are simple arithmetic averages where we add all of the scores, and divide by total number of classes. The weighted-averages are representative when using skewed evaluation sets, and is calculated by dividing each class specific score with the total number of examples, adding these scores and dividing by the total number of class examples used in the evaluation set. To calculate the micro-average of precision, we simply take all the true positive predictions, and divide by all the false positive predictions. The micro-average recall is calculated by taking all the true positive predictions, and divide by all false negative predictions. The micro-average F-score is a simple addition of the micro-precision and micro-recall divided.

Chapter 3

Methods

In this chapter we will be going over our methods used for both parts of the thesis, the literature review and the experiments.

3.1 Literature Review

In order to perform a review, we need literature to review. To find relevant sources, we start by querying Google Scholar[1] so we can find out if there are many relevant entries in the combined databases. Furthermore, we query scientific databases, like Web of Knowledge[2], I3E[3], ScienceDirect[4], and Springer[5]. We use several different search terms, since we want to find literature about sentiment analysis in Scandinavian language which could also be published in their respective native languages. Therefore we construct search terms that include all languages, as well as individual terms for the respective language.

[1] https://scholar.google.com/
[2] https://app.webofknowledge.com/author/#/search
[3] https://ieeexplore.ieee.org/Xplore/home.jsp
[4] https://www.sciencedirect.com/
[5] https://www.springeropen.com/journals

3.1.1 Search and evaluation

We start by searching for keywords, and make a collection of articles that might seem relevant at first glance. Thereafter we read through abstracts, some introductions and conclusions. If they still seem relevant after assessing, we read it more thoroughly and make notes of content. The information gathered is plotted into an excel table, to get an overview of relevant literature, including metadata, summaries and keywords. Then when they have been read properly, we rate them by relevance for ease of use when performing cascading searches.

Furthermore, we use the literature we have found to perform cascading searches. By performing cascading searches, we look through sources cited by our sources. This approach to gathering papers provides quite a lot of relevant work, which is to be expected. The literature found through this approach, went through the same process of assessing and reading, as the first approach. After this process, we can recognize many titles and authors which are frequently cited, many of which are mentioned in the next chapter.

3.1.2 Inclusion criteria

For our search, we have set in place some criteria for literature to be included. Our criteria include date, language, peer review, publication type, and content.

Our inclusion criteria consist of:

1. Due to the fact that we are exploring literature about the mainland Scandinavian languages, we need to include literature written and published in their native Scandinavian languages. This includes Norwegian, Swedish and Danish.

2. We include literature that is directly related to the search terms we describe in the following section 3.1.3.

3. Additionally, we will include some literature we were on the fence of excluding if they are significant contributions to the Scandinavian NLP field, or can be useful in the Scandinavian SA field.

Exclusion criteria consist of:

1. Though there might be much relevant self-published content on blogs and the likes, we will not be including any sort of self-published work, with the possible exception of some masters theses, as it is somewhat lacking validity due to not being peer reviewed as research.

2. Even if something seems relevant at first glance, we will be evaluating the contents of the work to make sure it is not more in the linguistic side of the field. Seeing as NLP is a very interdisciplinary field, research can be focused on a specific discipline making it less suited for the information science aspect of NLP, which we will be focusing on.

3. Research published before year 2000 will be excluded, since we are mainly interested in up and coming technologies and techniques utilizing machine learning approaches. The sentiment analysis field has seen few contributions before the 2000s, due to the increased used of the internet (Liu (2012)), so research from before this is likely not up to the current standard.

3.1.3 Search terms

The following table includes the main search terms we used in our research. We also made some permutations from the terms with little effect, so the safest option to not miss out on relevant literature was to make the terms general.

3.2 Research categorization and mapping

In order for us to create a useful overview, we need to categorize contributions. This is done to be able to map scientific work in a graph to get an overview of work in certain domains. We do not use many of the same categories as Petersen (2008), due to the specific domain we are exploring. Instead, we create some categories ourselves, which we believe covers most of the resource types available for the SA task. We de-

"sentiment analysis" "scandinavian" "sentiment analysis" "norwegian" "sentiment analysis" "swedish" "sentiment analysis" "danish"
"opinion mining" "scandinavian" "opinion mining" "norwegian" "opinion mining" "swedish" "opinion mining" "danish"
"känsla analys" and "skandinavien" "känsla analys" and "svenska" "känsla analys" and "norska" "känsla analys" and "danska"
"sentimentanalyse" and "skandinavia" "sentimentanalyse" and "norsk" "sentimentanalyse" and "svensk" "sentimentanalyse" and "dansk"
"følelsesanalyse" and "skandinavien" "følelsesanalyse" and "dansk" "følelsesanalyse" and "svensk" "følelsesanalyse" and "norsk"

Table 3.1: Search terms

note 'Data' for contributions that consist mainly of making datasets publicly available for conducting research, 'Software' for contributions that proposes software or tools for SA research, 'Evaluation' for contributions that perform tests and experiments with existing approaches and report results, 'Method' for contributions that proposes methods and techniques for SA, and 'Others' for contributions we have found that are not directly inline with our categories, but is too significant of a resource to ignore. Note that some of the literature we review may contain several types of contributions, as some choose to create their own data and make it available, while mainly focusing on evaluating techniques.

3.2.1 Graph

For the overview itself, we decide to use a simple bar chart since this seems to be a popular choice for data visualization within systematic reviews (Petersen (2015)). In the graph we include publication year, amount of publications, and contribution category. We considered creating one column for each category, for each year, but we

Category	Description
Data	Contributions consisting mainly of making data suited for SA publicly available.
Software	Contributions that create and provide software or technology for SA research.
Evaluation	Contributions that test and experiment with existing software, data and methods, and provide results.
Method	Contributions consisting mainly of testing and/or proposing methods and techniques for SA.
Others	Contributions we could not fit within the other categories, but are too significant to ignore.

Table 3.2: Contribution categories.

decided to keep it in five year intervals to avoid creating a convoluted and confusing graph. The graph will be presented in the results of chapter 4.

3.2.2 Review results

Finding the state of the art (SOTA) results for sentiment analysis within mainland Scandinavian languages, would require extensive knowledge and overview of the field, including self-published results. Therefore, we cannot be absolutely certain that the results we find are in fact the current SOTA results within the field. Regardless, we will, to the best of our ability, find baselines to compare our results in chapter 5.

3.3 Experiments and development

As mentioned previously, this thesis can be considered a two-part assignment. We start by reviewing literature within the specified field, which we use to determine options for our experiments. During our literature review we have found many tools and resources that can be useful for the task at hand, which consists of creating and experimenting with sentiment classifiers. Additionally, the results we find in our review will, to the best of our ability, include the current state of the art results within sentiment analysis for mainland Scandinavian languages, which we compare our own results to.

3.3.1 Development methods

For the development aspects of the thesis, we do not follow a specific development methodology to the dot, but our methods are heavily inspired by agile methodologies. We have adopted the main ideas of the *Kanban* methodology (Kniberg and Skarin (2010)), by developing our experiments in increments and creating tasks. Seeing as we are a team of one person developing the experiments, this is not strictly necessary, but due to our interest in programming, and the development required for our experiments, we decided to get to know this methodology a little better for our own personal gain.

Furthermore we use GitHub[6] and the Git[7] tool for command line based code commits, cloud storage and version control. This gives us a better overview of the project, reduces the risks of losing valuable work, and makes it easier to perform the experiments on different machines. Since this is commonly used in larger development teams, we wanted to further get to know the tool.

[6] https://github.com/
[7] https://git-scm.com/

Chapter 4

Literature Review

In this chapter we will be going through the main literature finds, what they contribute, and how they are related. Though we will be focusing on the main literature finds, we will also be looking at literature that might not be directly in line with our inclusions and exclusions, but we still consider them valuable contributions.

4.1 Main literature

What we consider main literature finds, are generally the contributions we find that are inline with our inclusions criteria. Here we will be summarizing these main finds, discuss why they are included and how they contribute to the field.

4.1.1 Constructing a Swedish general purpose polarity lexicon

Rosell and Kann (2010) describe the creation of a sentiment lexicon, from seed words, using the Free Dictionary of Synonyms from Kann and Rosell (2006). This is the earliest work we have found describing the creation of a sentiment lexicon for any of the mainland Scandinavian languages. Seeing as a sentiment lexicon can be used for a lexical approach to sentiment analysis, we consider it an approach for SA and categorize it as a method contribution to the field.

4.1.2 Constructing sentiment lexicons in Norwegian from a large text corpus

This paper from Bai et al. (2014) has constructed a sentiment lexicon for Norwegian words. They state that they could not find any Norwegian sentiment lexicons, previous to this work, which was their motivation for doing so. They describe a technique to autonomously construct sentiment lexicons from existing text data, using the Pointwise Mutual Information (PMI) technique, slightly modified. They use data provided by the National Library of Norway (NLN), which consists of Norwegian newspapers from the period 1998-2011. They also scrape online forums for user generated content (UGC). Furthermore they attempt to translate the lexicon from Nielsen (2011), which yielded surprisingly good results. The approaches and corpora created during this work have been made publicly available on GitHub[1]. Since they make the sentiment lexicon publicly available, we include this work as a method contribution.

4.1.3 Building sentiment Lexicons applying graph theory on information from three Norwegian thesauruses

In this paper from Hammer et al. (2014), they claim there were no publicly available sentiment lexicons for Norwegian at the time the work took place, but a similar approach was used in Bai et al. (2014) and the lexicons were made public afterwards. We think the work from Bai et al. (2014) was performed at the same time as that of Hammer et al. (2014), and the lexicons was made available after publishing. Regardless, they propose several sentiment lexicons automatically created using two different approaches. In the first approach they try to autonomously create a lexicon from seed words, and extend it by crawling three thesauruses and extracting synonyms and antonyms for the seed words. Then they are labeled with the Label Propagation algorithm from Zhu and Ghahramani (2002). The second approach was to machine-translate the well known AFINN lexicon from Nielsen (2011), and manually evaluate and fix the the translated results, since many slang words are not directly translatable.

[1] https://github.com/aleksab/lexicon

The results show that the machine-translated and manually tampered lexicon, provided better results than the lexicons constructed from thesauruses, and indicates that lexical resources can be translated and retain much of their usefulness. We also include this work as a method contribution.

4.1.4 Robust cross-domain sentiment analysis for low-resource languages

This paper from Elming et al. (2014) explores the feasibility of adapting domain specific sentiment classification, in the Danish language, to work with a different genre of text data. They argue that there have been no attempt at Sentiment Analysis using Danish language before this work. The approach they choose is based on Mohammad et al. (2013) where they use Support Vector Machines (SVM) to classify Twitter messages. They focus the work on the problem of domain adaptation (DA), which is to adapt to a specific domain, or genre of data. We include this work as both Method and Evaluation contribution, as they evaluate existing technologies for their domain while simultaneously describing an approach for the SA task using existing Danish language resources.

4.1.5 Building a sentiment lexicon for Swedish

Nusko et al. (2016), as the title implies, creates a sentiment lexicon for the Swedish language. They base their experiment on automatically expanding a lexicon based on seed words, using a publicly available Swedish lexical resource called *SALDO*, which we also mention later in our section for secondary literature 4.2. The main contribution of this work is the sentiment lexicon, which they make publicly available. We therefore include this work as a method contribution, seeing as the sentiment lexicon would be used as an approach for lexical sentiment analysis tasks.

4.1.6 A Sentiment model for Swedish with automatically created training data and handlers for language specific traits

This paper from Ludovici and Weegar (2016), explores the possibility of automatic binary sentiment labeling, and sentiment classification using Support Vector Machines (SVM). The data they use was provided by MittMedia[2], which consisted of newspaper articles from 2002-2015. They narrow it down a bit by category filtration, and pre-processes it with different techniques, including but not limited to tokenization and lemmatization. Thereafter they combine two methods for automatically labeling the data, which starts of by machine translating sentences from Swedish to English, and using the Stanford Recurring Neural Network (RNN) classification algorithm from Socher et al. (2013). The second method was to classify the sentences using a publicly available sentiment lexicon, using the Naive Bayes technique of polarity calculation (Maron (1961)). The final set consisted of the sentences where both methods could agree on the classification, the sentences they did not agree on were dropped. Finally the set was split into three categories, training, validation and test. The test set is manually annotated by native Swedes, as being positive, neutral or negative. They used SVM and the Term-Frequency - Inverse Document Frequency (TF-IDF) algorithm (Jones (2004)), and evaluate using the precision, recall, F-score and accuracy metrics. Had they made their data publicly available, we could have included the work as a data contribution, but we could not find any mentions of data publication. We consider this work as an evaluation contribution, as they experiment with, and evaluate technologies with Swedish text data. We also refer to this work in our own evaluation and comparison.

4.1.7 NoReC: The Norwegian Review Corpus

This main contribution of this work from Velldal et al. (2017), is the creation of a large text corpus. The corpus contains about 36 thousand full text documents, and every token is annotated with word form, lemma or stem of word form, universal part-of-

[2]https://www.mittmedia.se/

speech tag, language-specific part-of-speech tag, list of morphological features from the universal feature inventory, head of token, universal dependency relation to the head, enhanced dependency graph, among other attributes. The data also includes a metadata tag for the rating given by the original author, in the form of a dice rating (1-6), which can be used as sentiment labels. All the documents have been provided by different media groups operating in Norway. This dataset was constructed complete with document level sentiment annotations, and thus fits perfectly into our dataset contributions category even though the main work was not focused on the sentiment analysis task. We therefore include it as a data contribution. We will revisit this work in chapter 5.

4.1.8 SenSALDO: Creating a Sentiment Lexicon for Swedish

This paper from Rouces et al. (2018b) describes the creation of a sentiment lexicon based on a paper from Rouces et al. (2018a) where they use publicly available data to generate a sentiment lexicon. They use three different approaches, including a word embedding approach using the Word2Vec model from Mikolov et al. (2013) and Logistic Regression. The resulting sentiment lexicon has been made public from The Swedish Language Bank at the University of Gothenburg (Rouces et al. (2018b))[3]. The lexicon is created from a dataset we will mention in the next section about secondary literature. We include this work as a method contribution since they create and publish a sentiment lexicon.

4.1.9 Twitter Sentiment Analysis of New IKEA Stores Using Machine Learning

The work from Li and Fleyeh (2018), describes experiments with different sentiment analysis techniques to uncover opinions about the opening of a new IKEA store. They create classifiers for both English and Swedish data, but use an established sentiment lexicon based technique for the English sentiment classification. For the Swedish data,

[3]https://spraakbanken.gu.se/eng/resource/sensaldo

they crawl Twitter to extract tweets in an area containing the word "IKEA", and label the data based on emoticons. They base their emoticon labeling technique on previous work, and argue that manual labeling would be too ineffective. They try different pipelines with different models and algorithms, and compare the results before fine-tuning the best option for the classification task. They experiment with a total of six different techniques before settling on the Elastic net model from Friedman et al. (2009). Among the techniques they experiment with are, Logistic Regression, Neural Networks, Support Vector Machine, Random Forest and Naïve Bayes (Li and Fleyeh (2018)). They also refer to the previously mentioned Ludovici and Weegar (2016) in regards to Swedish text data and its challenges. We include this work as and evaluation contribution, as they seek to evaluate different approaches for autonomous sentiment analysis.

4.1.10 Sentiment classification of Swedish Twitter data

This work by Palm (2019) is one of the few exceptions to our inclusion and exclusion criteria, as it is a masters thesis. They start with the same motivations as this thesis, the lack of non-English NLP resources for sentiment classification. The fact that there are few resources available for performing such a task in Swedish, impedes research within the language specific NLP domain. They want to counteract this fact by designing a sentiment classifier with available Swedish resources, and come to the conclusion that the results can be compared to the international work within the sentiment analysis domain. They also experience that pre-processing of Swedish data has to be handled differently than its English equivalent.

They use manually annotated Twitter data classified into three classes. They use different text cleaning approaches due to the fact that the text sequences are essentially user generated content, which is prone for grammatical errors and fluff. They then use the popular word embedding technique from Google's Mikolov et al. (2013), and an SVM based approach for classification. They report some results that we will revisit in chapter 5. We include this work as both evaluation and method, as they evaluate approaches for sentiment analysis.

4.2 Secondary literature

Secondary literature are papers we have found that is not directly inline with our criteria, but can be considered contributions to the Scandinavian NLP domain. Some of these works are not directly related to sentiment analysis, but can be considered useful resources for the general NLP domain, and might be useful for the sentiment analysis problem. We do not include these papers in our primary graph, but we include them in a secondary graph which will be described more closely in section 4.3.2. Even though we describe the papers as being included as some category of contribution, they are not included in the primary graph.

4.2.1 A Constrained-Based Tagger for Norwegian

In this paper, from Hagen et al. (2000), an automatic morphosyntactic tagger is created. This is essentially a software contribution, that is capable of disambiguating and returning grammatical information about any given Norwegian word. This work proves the feasibility of morphological and syntactic disambiguation of Norwegian words (Hagen et al. (2000)). The approach for a constraint grammar based tagger, is inspired by Karlsson et al. (1995). This contribution is not directly linked to the topic of SA, and is not directly inline with our criteria, but provides grounds for much related research, and is cited by many. We therefore include this work as a software contribution.

4.2.2 Named Entity Recognition for the Mainland Scandinavian Languages

This paper from Johannessen et al. (2005) covers a research project from different universities in Scandinavia, performing experiments with the NLP task of Named Entity Recognition. Several different methods are used during the paper, and one of the techniques is the software contribution from Hagen et al. (2000), which was not the best performing experiment in the paper. In all, they try six different approaches with vary-

ing results. This is not directly related to the task of sentiment analysis, but can be considered a useful contribution within the NLP field for mainland Scandinavian languages, which is why we include it as an evaluation contribution for our secondary graph, even though it is not evaluating sentiment analysis approaches.

4.2.3 OBT+Stat: Evaluation of a combined CG and statistical tagger

This paper from Johannessen et al. (2011) describes the creation of a part-of-speech tagger called The Oslo Bergen Tagger (OBT), which is based on the Constraint Grammar approach from Karlsson et al. (1995), and the previous work from Hagen et al. (2000). Though this is not directly related to the field of sentiment analysis, this tagger is further used by many of the publications included in our study. We therefore think it is a valuable software contribution for the NLP field within mainland Scandinavian languages.

4.2.4 Building gold-standard treebanks for Norwegian

This paper from Solberg (2013) describes the process of creating a Norwegian Dependency Treebank in collaboration with the National Library of Norway. The work is not directly SA related, but provide important structured data that can be used for SA purposes, among other NLP tasks, and is referenced by some of the work we cover in our review. We therefore include this as a data contribution.

4.2.5 SALDO: a touch of yin to WordNet's yang

Borin et al. (2013) describes the creation of a lexical resource for Swedish Natural Language Processing (NLP) applications. This dataset is not created for the Sentiment Analysis (SA) task, but can be useful for the task regardless, as seen in the previously mentioned work from Rouces et al. (2018a) and Nusko et al. (2016). Seeing as it has already been used for Scandinavian SA purposes, we include this as secondary literature, and as a data contribution for the secondary graph.

4.2.6 The Norwegian Dependency Treebank

This paper from Solberg et al. (2014) goes further into detail about the treebank created by Solberg (2013), describing the creation of a large Norwegian dataset created at The National Library of Norway, which includes features such as syntactic and morphological annotation. At the time of the papers writing, they claim there was no previously created treebank for the Norwegian language. The dataset is referred to as "Språkbanken's Gold Standard Corpus" or simply "The Norwegian Dependency Treebank" (NDT), which we find mentions of in later research. The dataset is manually annotated by linguists and was made publicly available on the web.[4] The work in the paper is not a direct SA contribution, but can be used in SA research. But seeing as this paper is essentially an extension of the paper from Solberg (2013), we will not include it in the secondary graph.

4.2.7 An open source part-of-speech tagger for Norwegian: Building on existing language resources

This paper from Marco (2014) describes the creation of an open source part-of-speech tagger, created from existing resources. The approach uses one of the same corpora as Bai et al. (2014), specifically the 'Gullkorpus', or 'Gold standard corpus', created by the National Library of Norway (Solberg (2013), Solberg et al. (2014)). Furthermore, they use the dictionary 'Norsk ordbank' created by IBM Norway[5]. They also base their work on the previously mentioned work from Hagen et al. (2000) and the Oslo Bergen Tagger (OBT) from Johannessen et al. (2011). The results were close to the state-of-the-art taggers at that time. This research essentially falls under our category of software contributions, where a new part-of-speech tagger is created and evaluated.

[4] https://www.nb.no/sprakbanken/show?serial=sbr-10
[5] https://www.nb.no/sprakbanken/show?serial=oai%3Anb.no%3Asbr-5&lang=en

4.2.8 Supersense Tagging for Danish

This paper from Alonso et al. (2015) describes the creation of a tool for autonomous supsersense tagging. Though this is an NLP task, it is not directly related to the task we are exploring, and is not inline with our criteria. Regardless, we think it can potentially be used for the said task, and we will include it as a software contribution.

4.2.9 The Swedish Culturomics Gigaword Corpus

This paper from Eide et al. (2016) presents a dataset consisting of over a billion Swedish words, from different categories of literature. They made sure to include a good mix of different genres, collected from various text from the period 1950 to 2015. Genres include newspapers, legal texts, web forums, and more. The contribution is not directly related to the task of sentiment analysis, but has seen some use in the creation of sentiment lexicons, specifically that of Rouces et al. (2018a). We will include this as a data contribution in our secondary graph.

4.2.10 WordNet extension via word embeddings: Experiments on the Norwegian WordNet

In this paper from Sand et al. (2017) they create an approach to autonomously extend an existing WordNet based on existing Norwegian corpus resources. The main contribution of this work falls under the category for data contributions, since the WordNet is used as a lexical resource. This is not a direct contribution to the sentiment analysis filed, but the resources created can be used for sentiment analysis purposes in mainland Scandinavian languages.

4.2.11 Optimizing a PoS Tagset for Norwegian Dependency Parsing

This paper from Hohle et al. (2017) describes experiments with part-of-speech tagsets and performance with syntactic dependency parsing. They utilize the aforementioned

Norwegian Dependency Treebank from Solberg et al. (2014), and improve parsing accuracy significantly. This is not directly related to the sentiment analysis field, but the techniques can be utilized for a sentiment analysis task, so we include it as a software contribution.

4.2.12 Joint UD Parsing of Norwegian Bokmål and Nynorsk

This paper from Velldal et al. (2017) tackles the problem regarding the two official language variants of Norwegian. They experiment with isolated and combined pipelines for word vector creation in Norwegian Bokmål and Nynorsk, and found that the combined pipelines provide much higher accuracy at the cost of lexical size. Even though the two variants are quite similar, the lexical difference is big. But combining the two makes it possible to learn all the common words, as well as the variant specific words. We would categorize this research as a method contribution, but they have also created a Norwegian Nynorsk variant of the Norwegian Dependency Treebank from Solberg et al. (2014) and released all data and models[6], so we also include it as a data contribution. Though this is not directly related to the SA task, we think it is significant for the Scandinavian NLP field, since all Scandinavian languages have language variants.

4.2.13 Danish Resources

This paper from Nielsen (2018) is a collection of publicly available Danish language resources. The collection is intended for use in autonomous natural language tasks, and includes many data-sets and tools specifically for NLP in the Danish language. The resources are not specifically for a sentiment analysis task, but NLP tasks in general. The author of this work also created a very well know English sentiment lexicon (Nielsen (2011)), and has been cited many times in the previously mentioned literature. Though it is not directly related to SA, we think this is a significant contribution within the field, that will be cited by much future work. It also does not fall into any of or specific contribution categories, which is why we include it as *other* contributions.

[6] https://github.com/erikve/bm-nn-parsing

4.2.14 The Lacunae of Danish Natural Language Processing

Similarly to Nielsen (2018) the goal of the paper by Kirkedal et al. (2019) is to get a coarse overview of Danish specific NLP resources, as they admit it is not an exhaustive survey. They argue the same as us, that the Danish language (Scandinavian in our case) is a less privileged language when it comes to NLP resources than English, and want to get an overview of NLP models, tasks and data-sets specifically for the Danish language. This does not include resources found on the web, but rather scientifically published work. Though Danish also has several language variants, they focus on the official standard Danish.

They mention that the Danish government has sponsored much work in regards to Danish NLP, which has resulted in much data production from Dansk Sprogværn [7] and CLARIN DK [8]. They find contributions within several different NLP topics, like Part-of-speech tagging (PoS), Dependency Parsing, Named Entity Recognition and Senses, Machine Translation, Speech Technology, Speech Synthesis, and Sentiment ExtractionKirkedal et al. (2019). Within Sentiment Extraction, the well known AFINN tool from Nielsen (2011) for lexical sentiment analysis, is mentioned. Other than that there are a few multilingual approaches that are mentioned, mainly a full-text annotation system from Elming et al. (2014), and Alexandra Institutes model[9] based on Facebook's LASER[10] multilingual sentiment tool.

They conclude by stating that there is a lack of large datasets for the Danish language, something only the privileged languages have obtained, which a major part of NLP research relies on. As that of the similar work from Nielsen (2018), this does not fall under any of our specific contribution categories, which is why we include it under *other* contributions.

[7] https://dsn.dk/
[8] https://clarin.dk/clarindk/forside.jsp
[9] https://github.com/alexandrainst/danlp
[10] https://github.com/facebookresearch/LASER

4.3 Review and Overview

Originally we found 33 papers we thought were very relevant, but the number was reduced to 21 after comparing to our criteria. Thereafter we split them into main and secondary literature, resulting in 12 and nine respectively. As mentioned in chapter 3, we chose not to exclude work that can be considered relevant to the SA task but are not directly inline with our criteria, due to the potential usefulness of these entries. We call these entries secondary literature since they are not exactly what we were looking for, but can be considered important contributions to the Scandinavian NLP field, and can be useful for the Scandinavian SA field. These secondaries are not included in the primary graph due to them not being inline with our criteria. Instead, we create a secondary graph that include these papers.

4.3.1 Primary Graph

Following our review, we have created a graph to visualize the categorical finds, divided in five year intervals. The reason we chose to display the finds in five year intervals, is simply due to the size of the graph, as we think it should be easily readable. Though we found no primary literature between the years 2000 and 2010, we included these time periods in the graph to visualize the research trends over the years. These results gives an indication of the research trends in these Scandinavian countries following the year 2000. It seems as though little research was conducted in the field of Scandinavian sentiment analysis before the year 2010. We also found much work on the subject of sentiment lexicons, which accounts for five of the six entries in the method category.

4.3.2 Secondary graph

We also made a graph visualizing the finds of the main and secondary literature combined. This graph gives further indication that little research was conducted in the field before 2010.

The results of the review surprised us somewhat, as we did not expect so much research

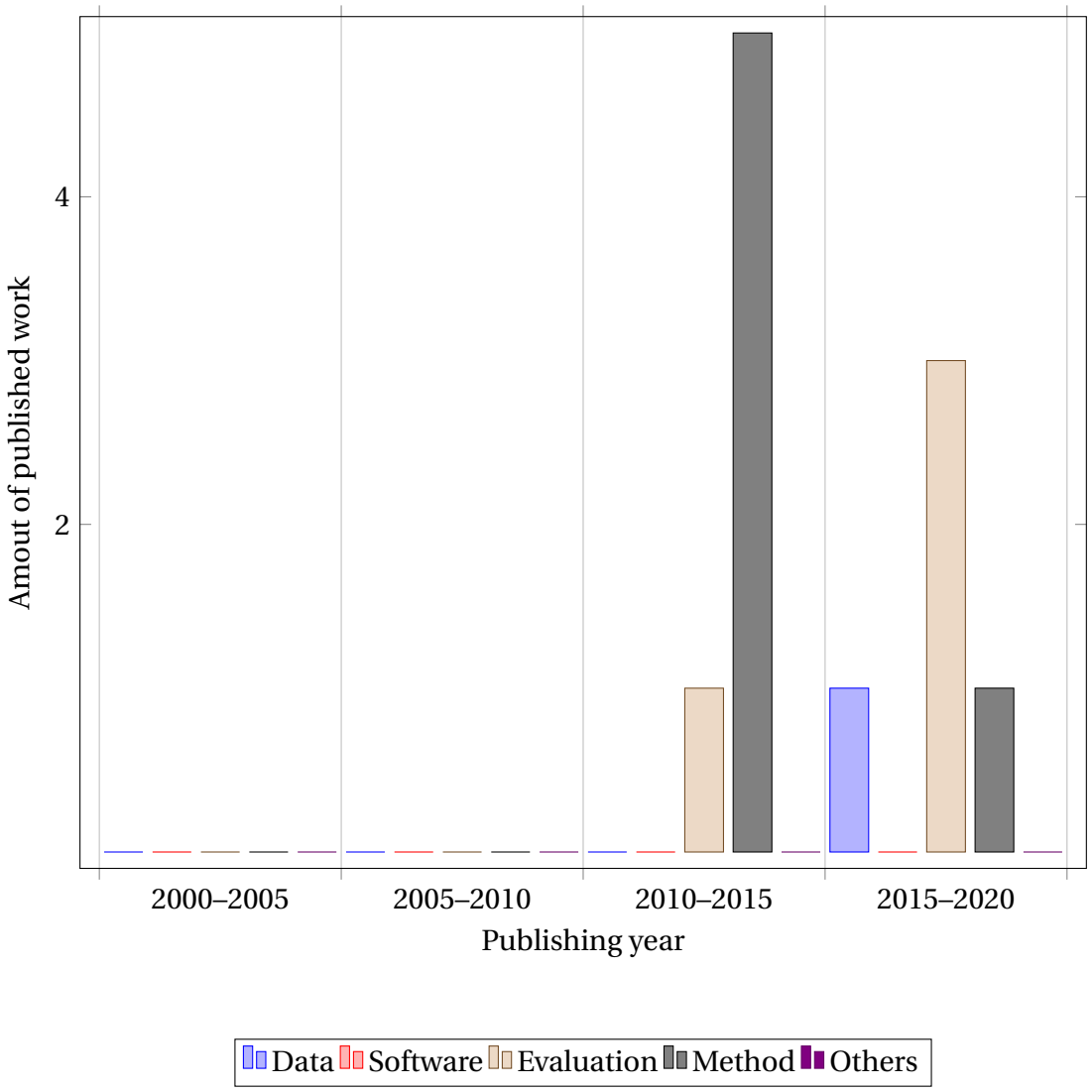

Figure 4.1: Main literature bar chart

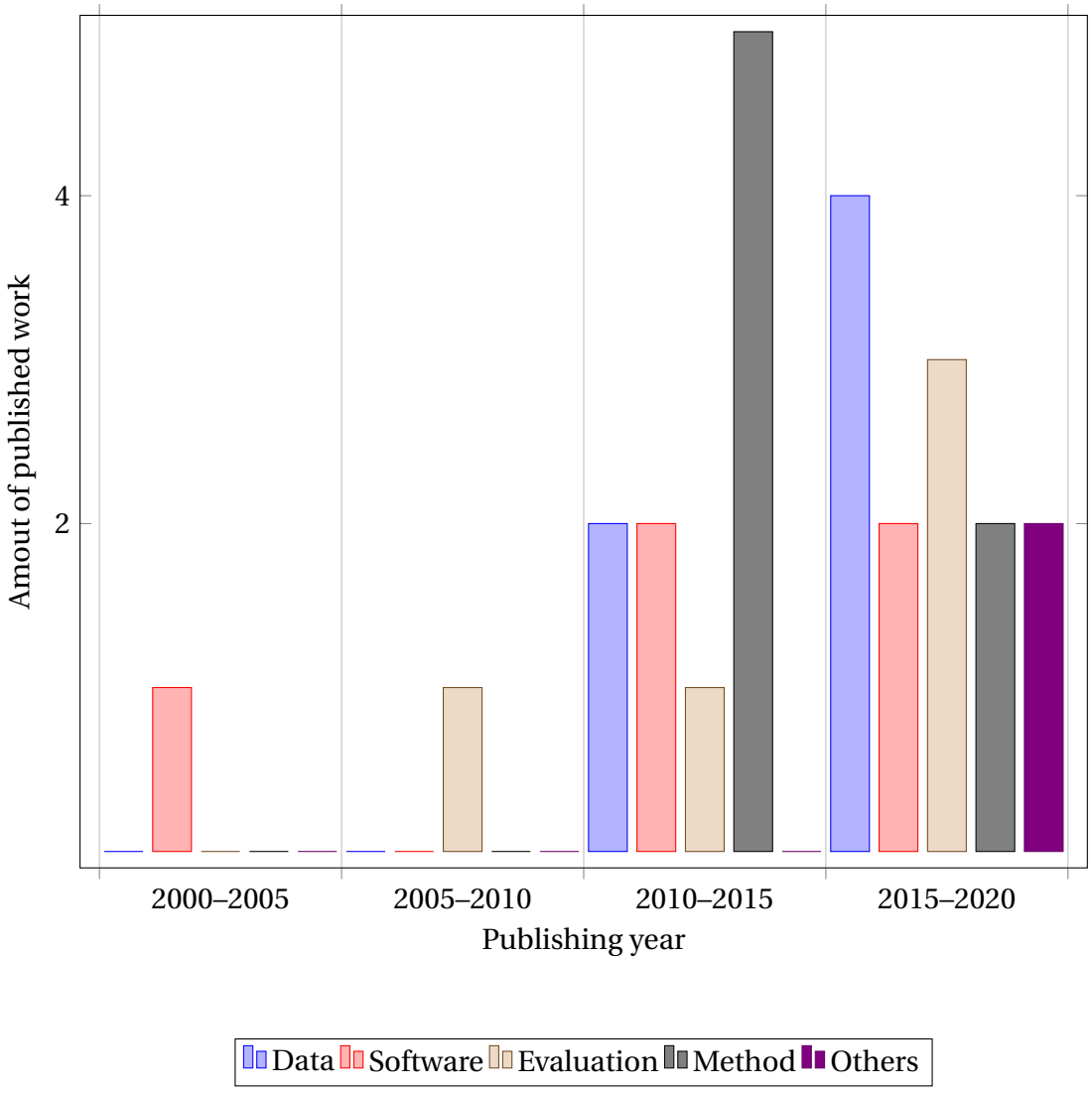

Figure 4.2: Combined literature bar chart.

on the topic of sentiment lexicons, the latest of which is from Rouces et al. (2018b). We thought that the international sentiment analysis field was more focused exploring and expanding deep learning for sentiment classification, and thought that the Scandinavian research fields would follow. But it seems as though the Scandinavian field only recently started looking at these deep learning techniques.

Chapter 5

Development and Evaluation

In this chapter we will be going over the experiments we have conducted. We chose our approaches based on the literature review we performed in chapter 4, and included results from papers we found to compare with our own results. Additionally, we did some minor testing with adaptation of newer approaches for English sentiment analysis.

5.1 Planning and Development

During the literature review we found resources and experiments we can test and compare, and decided on techniques and approaches based on what we found in the review. The goal of our experiments is to create general purpose sentiment classifiers for the mainland Scandinavian languages, based on existing resources, and compete with the state of the art results found through the literature review. If we are not able to beat the best performing approaches, we will hopefully still be able to produce a viable approach for the task in Scandinavian languages, using freely available resources and techniques.

5.1.1 Data choice

Seeing as the original inspiration for the research included sentiment mapping of User Generated Content (UGC), we went ahead to look for endpoints where UGC was gen-

erated for the Scandinavian languages. First we considered Twitter, seeing as their endpoints include attributes for geography, but users have a tendency to use the English language on the web, and there is a limit to how much historical data one can retrieve. In addition, this data would not be ready and available for sentiment classification tasks due to the lacking sentiment labels, which we would have to generate ourselves. Thus the search for data continued until we found a perfect candidate from the University of Oslo.

The data in question is the Norwegian Review Corpus (NoReC) from Velldal et al. (2017), which we describe in chapter 4. To recap, the data consists of around 36 thousand full-text reviews of different products, separated by categories, and including sentiment labels in the form of the authors dice-rating based review. This data is labeled for document-level sentiment analysis, which is not suited for models requiring a smaller fixed-length input. Furthermore, they have preemptively prepared sub samples of the data for evaluation, where 10% of the available data is saved as a test, or evaluation, set. With the publication of the work, they also created a Python module for easily accessing and manipulating the data. This in turn, made it easier for us to use the data as we are already familiar with the Python programming language, which has become one of the most popular programming languages for scientific computing due to its library ecosystem (Pedregosa et al. (2011)).

5.1.2 Model choices

As we want to compare our results to that of the literature we have found, we decided to experiment with some of the reoccurring techniques found in the literature from our review. One of which being the Support Vector Machine (SVM) technique, which we mention in chapter 2.

In addition to comparing our work to that of the review, we also want to experiment with techniques we have found few mentions of. Originally we looked at some popular word embedding techniques from Facebook and Google, mainly the FastText method from Joulin et al. (2016) and the Word2Vec method from Mikolov et al. (2013). These are both dependent on fixed length text sequences, and are therefore more suited for

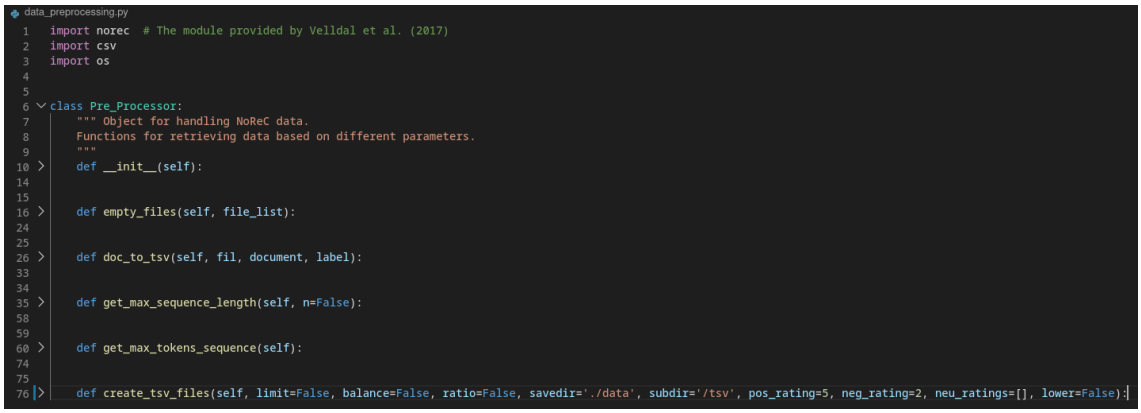

Figure 5.1: The data pre-processing program

analyzing short messages like UGC from social media. We therefore decided to have a look at Le and Mikolov (2014), where they make an adaptation of the Word2Vec model for document vectors, with the imaginative name Doc2Vec.

The Doc2Vec approach comes with two models, namely Distributed Memory version of Paragraph Vector (PV-DM), and Distributed Bag of Words version of Paragraph Vector (PV-DBOW), which are extensions of the Skip-gram model and Distributed Bag-of-Words model used in Mikolov et al. (2013).

5.2 Data pre-processing

Even though Velldal et al. (2017) has done much of the work for us by making the data available through a Python module, we have to further process it to make the data fit our approaches. Since we will be experimenting with different pipelines, we want to easily be able to adjust the parameters of the pre-processing stage, so we make our own Python program to manipulate the data utilizing the module provided by Velldal et al. (2017). This gives us the ability to create a simple interface for adjusting pre-processing parameters for each experiment.

5.2.1 Parameters

Among the parameters available in the pre-processing pipeline are document limit, class distribution balancing, class distribution ratio, and options for selecting which ratings fall into which class. The document limit parameter is mainly used for experimenting with less data. The class distribution balancing parameter is an option to balance the input classes to counteract the problem of skewed class distributions. This parameter can also be adjusted with a distribution ratio, where we have the option to, for example, use a 1:2 distribution of negative and positive documents. The final parameters are options to select which ratings fall into which class, for example all documents with a rating of five or more is categorized as positive, while all documents with a rating of two or less is categorized as negative. By default these are set to five and two respectively.

We also implemented the option to use a third class, for neutral documents, in the classification task. This can be selected by specifying two rating values, where the documents with that rating, and the ratings in between, would be considered neutral. By default these are set to the ratings three and four, since these are on the mid range of the scale.

When we divide classes based on ratings, we reduce the amount of training data quite a bit. Originally the dataset contains around 36 000 documents (Velldal et al. (2017)), which gets reduced drastically as it seems there are more reviews on the mid-range ratings. The full specifications of how many documents are in each class based on rating divisions can be seen in table 5.1. When using a third class for neutral documents, we can choose to use all documents if we also specify the negative class to include documents of rating two or lower, and the positive to include ratings of five and higher. We experiment with different scenarios.

Table 5.1: Data class distribution specifications.

Ratings	**Train Neg**	**Train Pos**	**Test Neg**	**Test Pos**
P<=6 N>=1	353	1 692	25	204
P<=5 N>=2	2 326	11 597	231	1 559
P<=4 N>=3	7 387	20 771	807	2 706

5.3 Experiments

For our experiments, we chose three slightly different approaches. Several of the papers we have reviewed include some experimentation with Support Vector Machines (SVM) (Palm (2019), Ludovici and Weegar (2016), Li and Fleyeh (2018)). Therefore, we have also decided to do some experiments with this classification technique. We also found experiments using logistic regression for classification, which we also decided to experiment with. Most of the approaches we found, utilize different data and feature extraction, so we experiment with two different techniques here as well. Additionally we perform experiments with both binary- and multi-class classification using these techniques. In the following sections we will be describing each of these experiments, as well as present our results and compare to that of the research.

5.3.1 Data

For all our experiments, we use the same data as we describe in section 5.2. This include the same evaluation set. We use slightly different pre-processing pipelines in most of our experimental iterations, which is described in tables following the description of each approach.

5.4 Doc2Vec embedding based classification using LogReg

Our first experiment utilizes the Doc2Vec model from Le and Mikolov (2014) to create word embeddings for a classification task with logistic regression.

5.4.1 Pre-training vectors

There are many parameters to assess before starting a training process. As the model seeks to learn semantic representations of words, we also have to consider what words are worth embedding since some words are seldom used, some words are concatenations, some words are bound with symbols, and some words are a mixture of lower-

and upper-case letters. Some words may even come as both capitalized and lower case if they appear in the start of the sentence, such as 'Some' in this sentence. This we can affect by using the data parameters for word count and lower case.

The data specifications for our experiments are depicted in table 5.2, model training parameters can be seen in table 5.3, and the results can be seen in table 5.4. We did not include all results in table 5.4 since some of the results did not manage to predict a single instance of a class. The full tables can be found in the appendix A.1.

Table 5.2: D2V and logistic regression data specifications.

Experiment	**Pos/neg**	**Neu**	**Balance**	**Ratio**	**Lowercase**
D2V1	P<=5 N>=2	–	False	–	False
D2V2	P<=5 N>=2	–	False	–	True
D2V3	P<=5 N>=2	–	True	1:1.3	False
D2V4	P<=5 N>=2	–	True	1:1.0	True
D2V5	P<=6 N>=1	–	False	–	True
D2V6	P<=6 N>=1	–	False	–	False
D2V7	P<=4 N>=3	–	False	–	False
D2V8	P<=5 N>=2	–	False	–	True
D2V9	P<=5 N>=2	–	False	–	True
D2V10	p<=5 N>=2	–	False	–	True
MD2V1	P<=5 N>=2	3-4	False	–	False
MD2V2	P<=5 N>=2	3-4	False	–	False
MD2V3	P<=5 N>=2	3-4	False	–	True
MD2V4	P<=5 N>=2	3-4	True	1:1	True
MD2V5	P<=5 N>=2	3-4	True	1:1	True
MD2V6	P<=6 N>=1	3-4	False	–	True
MD2V7	P<=6 N>=1	3-4	True	1:1	True
MD2V8	P<=5 N>=2	3-4	True	1:1.3	True

We made several test cases, as depicted in these tables. The data itself was provided from Velldal et al. (2017), which they have made publicly available as a Git repository[1]. Our Python based experiments utilize the Pandas module[2] for easily creating dataframes for use with the Doc2Vec model provided by Gensim[3], then we used the Logistic Regression model from scikit-learn[4].

[1] https://github.com/ltgoslo/norec
[2] https://pandas.pydata.org/
[3] https://radimrehurek.com/gensim/models/doc2vec.html
[4] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

5.4.2 Training parameters

To create the actual classifier, we train and evaluate the model using Logistic Regression (LR), which we mention in chapter 2. We also had an idea of using the Multinominal Logistic Regression (MLR) algorithm for multic-class experiments, but we landed on the Ordinary Least Squares (OLS) regression model, because it was available in the same scikit-learn module, as opposed to MLR. There are a number of parameters available for tuning, which all affect the results of the training. We have done several experiments with different parameters and gotten different results. The parameters can be seen in 5.3 with the respective results in 5.4. Though, as mentioned earlier, the full result tables are not shown in this section, and can be found in full in the appendix A.1.

Table 5.3: D2V and logistic regression parameter specifications.

Experiment	Model	Vector size	Neg sampling	Min frequency	Alpha	Epochs
D2V1	PV-DM	500	10	1	0.065	50
D2V2	PV-DM	500	10	1	0.065	50
D2V3	PV-DM	500	10	1	0.065	50
D2V4	PV-DM	500	10	1	0.065	50
D2V5	PV-DM	500	10	1	0.065	50
D2V6	PV-DM	500	10	1	0.065	50
D2V7	PV-DM	500	10	1	0.065	50
D2V8	PV-DBOW	500	10	1	0.065	50
D2V9	PV-DBOW	500	10	2	0.065	100
D2V10	PV-DBOW	500	20	2	0.065	100
MD2V1	PV-DM	500	10	2	0.065	50
MD2V2	PV-DBOW	500	10	2	0.065	50
MD2V3	PV-DM	500	10	2	0.065	50
MD2V4	PV-DM	500	10	2	0.065	50
MD2V5	PV-DBOW	500	10	2	0.065	50
MD2V6	PV-DM	500	10	2	0.065	50
MD2V7	PV-DBOW	500	10	2	0.065	50
MD2V8	PV-DM	500	20	2	0.065	50

The results for our binary-classification experiments look very good, but we wonder if this has been overfitted by pre-training the word vectors on the same data it is learning to classify. Regardless, the evaluation set is unique from the training set during fine-tuning for classification, though both were used for pre-training the embedding. The results from our multi-class classification experiments did not turn out great. We have

Table 5.4: D2V and logistic regression results.

Experiment	Metric	Class			Accuracy
		Pos	Neu	Neg	
D2V1	Precision	0.95	–	0.62	90.5%
	Recall	0.94	–	0.66	
	F-score	0.95	–	0.64	
D2V2	Precision	0.96	–	0.67	91.8%
	Recall	0.95	–	0.73	
	F-score	0.95	–	0.70	
D2V7	Precision	0.86	–	0.66	82.6%
	Recall	0.92	–	0.52	
	F-score	0.89	–	0.58	
D2V8	Precision	0.98	–	0.81	95.5%
	Recall	0.97	–	0.86	
	F-score	0.97	–	0.83	
D2V9	Precision	0.98	–	0.84	**95.9**%
	Recall	0.98	–	0.85	
	F-score	0.98	–	0.84	
MD2V1	Precision	0.78	0.67	0.78	68.4%
	Recall	0.70	0.74	0.03	
	F-score	0.67	0.70	0.06	
MD2V3	Precision	0.71	0.67	1.00	69.0%
	Recall	0.73	0.75	0.03	
	F-score	0.72	0.71	0.07	
MD2V8	Precision	0.66	0.69	0.29	62.1%
	Recall	0.74	0.52	0.59	
	F-score	0.70	0.59	0.39	

some thoughts on why the results were much poorer in the multi-class experiments, which we discuss further at the end of this chapter, as well as in chapter 6.

5.5 Term frequency and Support Vector Machine

Several of the papers we have reviewed include some experimentation with Support Vector Machines (SVM) (Palm (2019), Ludovici and Weegar (2016), Li and Fleyeh (2018)). Therefore, we have also decided to do some experimentation with this technique.

5.5.1 Training

For training vectors before the classification step, we utilize the SGDClassifier model[5] from the Scikit-learn Python library[6], which we also utilized in the previous experiments described in section 5.4. The first approach utilizes a count-vectorizer[7] and the Term Frequency - Inverse Document Frequency (TF-IDF) model[8] for feature creation. These features were then used to classify using the hinge loss function for a linear SVM approach. We experiment with different parameters as shown in tables 5.5 and 5.6, where the experiments from this approach are prefixed with "TF-".

We experiment with both binary- and multi-class classification using this approach, where the multi-class approach is prefixed with "TF-MSVM". We also experiment with class distribution due to the vast differences in the available classes, as can be seen in table 5.1. Using the linear SVM approach for classification we experienced very poor classification results while using an unbalanced distribution, which is why we experiment with different balance ratios to find a goldilocks zone for class distribution.

5.5.2 Evaluation and Results

For evaluation, we use a subset of the NoReC (Velldal et al. (2017)) data, which has been sub sampled as mentioned in section 5.2. We also have the option of randomising test data, by combining training and evaluation data, and randomly sub sampling data for a new evaluation set for each experiment, but we choose to use the pre-made test set for all test cases, which we think is a better way of representing the results since they are based on the exact same evaluation. The results can be seen in the following table 5.7. For results that did not manage to predict any instances of a certain class, most often the negative class, we did not include the accuracy as we think the scores could seem misleading.

[5] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.htmlsklearn.linear_model.SGDClassifier

[6] https://scikit-learn.org/stable/index.html

[7] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

[8] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

Table 5.5: TF-IDF and SVM data specifications.

Exp	Lower	Balance	Ratio	Pos/Neg	Neu	Neg docs	Neu docs	Pos docs
TF-SVM1	False	False	–	P>=5 N<=2	–	2 326	–	11 597
TF-SVM2	False	True	1:1	P>=5 N<=2	–	2 326	–	2 326
TF-SVM3	False	True	1:1.1	P>=5 N<=2	–	2 326	–	2 558
TF-SVM4	False	True	1:1.2	P>=5 N<=2	–	2 326	–	2 789
TF-SVM5	False	True	1:1.3	P>=5 N<=2	–	2 326	–	3 020
TF-SVM6	False	True	1:1.4	P>=5 N<=2	–	2 326	–	3 251
TF-SVM7	False	True	1:1.3	P>=5 N<=2	–	2 326	–	3 020
TF-SVM8	False	True	1:1.3	P>=5 N<=2	–	2 326	–	3 020
TF-SVM9	False	True	1:1.3	P>=5 N<=2	–	2 326	–	3 020
TF-SVM10	False	True	1:1.3	P>=5 N<=2	–	2 326	–	3 020
TF-SVM11	False	True	1:1.3	P>=5 N<=2	–	2 326	–	3 020
TF-SVM12	True	True	1:1.3	P>=5 N<=2	–	2 326	–	3 020
TF-MSVM1	False	False	–	P>=5 N<=2	3-4	2 326	14 235	11 597
TF-MSVM2	False	True	1:1.3	P>=5 N<=2	3-4	2 326	3 024	3 020
TF-MSVM3	True	False	–	P>=5 N<=2	3-4	2 326	14 235	11 597
TF-MSVM4	False	True	1:1	P>=5 N<=2	3-4	2 326	2 327	2 327
TF-MSVM5	True	False	–	P>=6 N<=1	3-4	353	14 235	1 692
TF-MSVM6	True	True	1:1	P>=6 N<=1	3-4	353	354	354
TF-MSVM7	True	False	–	P>=5 N<=2	3-4	2 326	14 235	11 597

5.6 Doc2Vec embeddings with SVM classification

We also did a third experiment where we use techniques from both previous approaches. Here we use Doc2Vec to create word embeddings, and use these embeddings to classify with an SVM approach. We perform experiments with both binary- and multi-class classification. These experiments are prefixed with "D2_".

5.6.1 Preparation and runtimes

The data pre-processing stage is the same as described in the former section 5.2 of this chapter, as this is used for all experiments. Here we also train the word vectors in several epochs for best possible semantic word representation. This is also why there are fewer experiments using the Doc2Vec word vectors. At most, we spent about six hours for one experiment where we used as much as 500 epochs for the pre-training stage and 10 000 for the classification stage, but on average they do not take more than half

Table 5.6: TF-IDF and SVM parameter specifications.

Experiment	Loss	Penalty	Learning rate	Epochs
TF-SVM1	Hinge	l1	1e-3	1000
TF-SVM2	Hinge	l2	1e-3	10
TF-SVM3	Hinge	l2	1e-3	10
TF-SVM4	Hinge	l2	1e-3	10
TF-SVM5	Hinge	l2	1e-3	10
TF-SVM6	Hinge	l2	1e-3	10
TF-SVM7	Hinge	l2	1e-3	50
TF-SVM8	Hinge	l2	1e-3	500
TF-SVM9	Hinge	l2	1e-3	1000
TF-SVM10	Hinge	l1	1e-3	1000
TF-SVM11	Hinge	elasticnet	1e-3	1000
TF-SVM12	Hinge	l2	1e-3	1000
TF-MSVM1	Hinge	l1	1e-3	100
TF-MSVM2	Hinge	l1	1e-3	100
TF-MSVM3	squared_hinge	l2	1e-3	100
TF-MSVM4	squared_hinge	l2	1e-3	100
TF-MSVM5	squared_hinge	l2	1e-3	100
TF-MSVM6	squared_hinge	l2	1e-3	100
TF-MSVM7	squared_hinge	elasticnet	1e-3	100

an hour when using 50 epochs for the pre-training, and 1000 epochs for the classification which is the less time consuming of the two stages. Furthermore, the training times also differ based on class distribution as there are many more training examples when using ratings 5-6 and 2-1 as positive and negative, than there are when using only 6 and 1 respectively.

5.7 Comparisons

Out of all the research we have found that include experimentation with sentiment classification, none share an identical approach. Different data and pipelines are used between all the experiments, but we could not, to the best of our ability, find any identical experiments. Since all the approaches and data differ in each experiment, we cannot definitely say which approach is the absolute best. Regardless, we create a table to compare our results to that of the research we have found. The results and comparisons can be seen in in table 5.11.

Table 5.7: TF-IDF and SVM results.

Experiment	Metric	Class			
		Pos	Neu	Neg	Acc
TF-SVM5	Precision	0.93	–	0.79	91.7%
	Recall	0.98	–	0.50	
	F-score	0.95	–	0.61	
TF-SVM7	Precision	0.93	–	0.76	**91.8%**
	Recall	0.98	–	0.53	
	F-score	0.95	–	0.63	
TF-SVM8	Precision	0.93	–	0.76	91.6%
	Recall	0.98	–	0.52	
	F-score	0.95	–	0.62	
TF-SVM12	Precision	0.93	–	0.75	91.3%
	Recall	0.97	–	0.49	
	F-score	0.95	–	0.59	
TF-MSVM1	Precision	0.65	0.50	0.00	–
	Recall	0.06	0.97	0.00	
	F-score	0.11	0.66	0.00	
TF-MSVM2	Precision	1.00	0.49	0.14	49.00%
	Recall	0.00	1.00	0.00	
	F-score	0.00	0.66	0.01	
TF-MSVM4	Precision	0.64	0.73	0.27	60.3%
	Recall	0.80	0.41	0.69	
	F-score	0.71	0.53	0.39	
TF-MSVM6	Precision	0.21	0.97	0.08	54.9%
	Recall	0.86	0.51	0.72	
	F-score	0.34	0.67	0.14	

Palm (2019) have done some experimentation with SVM using two different kernels, linear and Radial Basis Function kernel (RBF). This work does not present results from a binary classification task, but rather a classification task using three classes. The results they present is that of multi-class classifications, but they do not report what type of average they include in the results, so we are assuming they use the macro-average which we describe in chapter 2. Note that we only assume that they use the macro-average, which might not be the case. In addition, this work is a masters thesis, which we were on the fence of including, but since their experiments are interesting and similar to our own, we will include it in the comparisons. These experiments are labeled with the prefix "Palm" in the comparison table 5.11.

Table 5.8: D2V and SVM data specifications..

Exp	Lower	Balance	Ratio	Pos/Neg	Neu	Neg docs	Neu docs	Pos docs
D2_SVM1	False	False	–	P>=5 N<=2	–	2 326	–	11 597
D2_SVM2	False	False	–	P>=5 N<=2	–	2 326	–	11 597
D2_SVM3	False	True	1:1	P>=5 N<=2	–	2 326	–	2 326
D2_SVM4	True	True	1:1.3	P>=5 N<=2	–	2 326	–	3 020
D2_SVM5	True	True	1:1.3	P>=6 N<=1	–	353	–	459
D2_MSVM1	True	False	–	P>=6 N<=1	3-4	353	14 235	1 692
D2_MSVM2	False	False	–	P>=6 N<=1	3-4	353	14 235	1 692
D2_MSVM3	True	False	–	P>=5 N<=2	3-4	2 326	14 235	2 326
D2_MSVM4	True	True	1:1	P>=5 N<=2	3-4	2 326	2 327	2 327
D2_MSVM5	True	True	1:2	P>=5 N<=2	3-4	2 326	4 653	4 653
D2_MSVM6	False	False	–	P>=5 N<=2	3-4	2 326	14 235	11 597
D2_MSVM7	True	False	–	P>=6 N<=1	3-4	353	14 235	1 692
D2_MSVM8	True	True	1:1.3	P>=6 N<=1	3-4	353	459	459

Table 5.9: D2V and SVM parameter specifications.

Exp	Loss	Penalty	D2V Ep	SVM Ep	Model	VS	NS	MC
D2_SVM1	Hinge	l2	10	1000	PV-DBOW	500	10	1
D2_SVM2	Hinge	l2	10	1000	PV-DM	500	10	1
D2_SVM3	Hinge	l2	10	1000	PV-DM	500	10	1
D2_SVM4	Hinge	l2	10	1000	PV-DM	500	10	2
D2_SVM5	Hinge	l2	10	1000	PV-DBOW	500	10	2
D2_MSVM1	squared_hinge	l2	100	10000	PV-DBOW	500	10	1
D2_MSVM2	squared_hinge	l2	100	10000	PV-DBOW	500	10	1
D2_MSVM3	squared_hinge	l2	100	10000	PV-DBOW	500	10	1
D2_MSVM4	squared_hinge	l2	100	10000	PV-DBOW	500	10	1
D2_MSVM5	squared_hinge	l2	100	10000	PV-DBOW	500	10	1
D2_MSVM6	squared_hinge	l2	100	10000	PV-DM	500	10	1
D2_MSVM7	squared_hinge	l2	100	10000	PV-DM	500	10	1
D2_MSVM8	squared_hinge	l1	100	1000	PV-DM	500	10	2

Ludovici and Weegar (2016) report experiments with an SVM model, but do not include many details about the experiments. They only include the results of their best performing experiment, but does not include all the same level of metrics as our own experiments, and that of Palm (2019). The metrics they do not include, are class specific metrics, so we have to assume this is the average metrics of the classes, but they do not report which type of average they use, so once again we assume it is the macro-average which we describe in chapter 2. They also experiment with a binary classifi-

Table 5.10: D2V and SVM results.

Experiment	Metric	Class			
		Pos	Neu	Neg	Acc
D2_SVM1	Precision	0.87	–	0.00	–
	Recall	1.00	–	0.00	
	F-score	0.93	–	0.00	
D2_SVM3	Precision	0.95	–	0.30	73.9%
	Recall	0.74	–	0.77	
	F-score	0.83	–	0.43	
D2_SVM4	Precision	0.88	–	0.39	86.5%
	Recall	0.98	–	0.08	
	F-score	0.93	–	0.13	
D2_MSVM1	Precision	0.25	0.91	0.12	78.3%
	Recall	0.43	0.83	0.12	
	F-score	0.31	0.87	0.12	
D2_MSVM3	Precision	0.56	0.56	0.25	55.4%
	Recall	0.60	0.58	0.07	
	F-score	0.58	0.57	0.11	
D2_MSVM6	Precision	0.58	0.57	0.27	56.0%
	Recall	0.60	0.56	0.25	
	F-score	0.59	0.57	0.26	
D2_MSVM8	Precision	0.27	0.95	0.04	64.3%
	Recall	0.71	0.64	0.40	
	F-score	0.39	0.76	0.07	

cation task, so they do not include the "neutral" class. Since they do not include class specific metrics, nor the neutral class, we can only place their results in the "Avg" (Average) column of 5.11. Their results are labeled with the prefix "Ludo" in table 5.11.

Li and Fleyeh (2018) experiments with different techniques for analysing sentiment polarity in tweets, in regards to a new IKEA store opening. They also experiment with SVM based techniques, as well as logistic regression, which uses automatically annotated twitter data based on emoticons. Though they perform a binary classification task, they only report the average F-score of the experiments, but they do not specify which type of F-score average they use, so we once again assume they use the most common type which is macro-average, as mentioned in chapter 2. We include two of their experiments in table **??** with the prefix "Li".

Rouces et al. (2018b) has performed two experiments with multi-class classification and reported their results using precision, recall and accuracy metrics. They perform an experiment where they use word embeddings created with the Word2Vec model from Mikolov et al. (2013), and classify with an SVM approach, and one where they classify using logistic regression. Quite similar to our own experiments, tough we use the Doc2Vec model which is a document level variant of the Word2Vec model, as mentioned in 5.1.2. Their results are prefixed with "Rouces" in table 5.11.

Finally, we include our own experiments with the same prefixes as in previous tables. Though due to the size of our tables, we only include the best results from each approach. Note that we cannot be sure of the average metric used in some of these papers, as they have failed to mention it. We therefore assume all the averages are macro-averages, being the most common. Because we are uncertain of the metrics reported in these papers, we denote the uncertain results with a "*" suffix.

5.8 Results

As seen in table 5.11, our binary classification approach yielded quite good results, but our results with multi-class classification is considerably worse. We did not manage to beat any of the multi-class classification results found in the literature, which means that there exist some good approaches within the research material, for the Scandinavian specific field. We did however manage to beat the results from the only reported binary-class experiment in the table, which is that of Ludovici and Weegar (2016), but due to the uncertainty of the result we cannot for sure claim our results to be a bigger success. Though these experiments all use different data and approaches, we are not able to definitely announce one as being the best approach, but we can determine which have had better results with their respective approach. From the literature we have reviewed, the experiments from Rouces et al. (2018b) have achieved the best performances, with Ludovici and Weegar (2016) at a close second for binary-classification, though we do not have the full details of their results.

We would like to consider our experiments with binary-classification as a success, though

Table 5.11: Result comparisons including our own experiments.

Experiment	Metric	Class				
		Pos	Neu	Neg	Avg	Acc
Palm SVM Linear	Precision	0.78	0.60	0.66	0.68*	68.2%
	Recall	0.74	0.62	0.68	0.68*	
	F-score	0.76	0.61	0.67	0.68*	
Palm SVM RBF	Precision	0.83	0.52	0.70	0.70*	70.9%
	Recall	0.71	0.65	0.69	0.69*	
	F-score	0.79	0.57	0.69	0.70*	
Ludo SVM RBF	Precision	–	–	–	0.895*	–
	Recall	–	–	–	0.824*	
	F-score	–	–	–	0.858*	
Li LogReg	F-score	–	–	–	0.724*	–
Li SVM	F-score	–	–	–	0.721*	–
Rouces W2V SVM RBF	Precision	0.65	0.92	0.65	–	**89.0%**
	Recall	0.46	0.96	0.44	–	
Rouces W2V Logit	Precision	0.37	0.93	0.46	–	84.0%
	Recall	0.54	0.88	0.52	–	
D2V8	Precision	0.98	–	0.81	–	**95.5%**
	Recall	0.97	–	0.86	–	
	F-score	0.97	–	0.83	–	
MD2V5	Precision	0.67	0.72	0.24	–	60.5%
	Recall	0.77	0.46	0.65	–	
	F-score	0.71	0.56	0.35	–	
TF-SVM7	Precision	0.93	–	0.76	–	91.8%
	Recall	0.98	–	0.53	–	
	F-score	0.95	–	0.63	–	
TF-MSVM3	Precision	0.71	0.67	0.00	–	68.6%
	Recall	0.72	0.75	0.00	–	
	F-score	0.71	0.70	0.00	–	
D2_SVM4	Precision	0.88	–	0.39	–	86.5%
	Recall	0.98	–	0.08	–	
	F-score	0.93	–	0.13	–	
D2_MSVM1	Precision	0.25	0.91	0.12	–	78.3%
	Recall	0.43	0.83	0.12	–	
	F-score	0.31	0.87	0.12	–	

we do not have much research to compare our experiments to. Regardless, the classification results of our binary-class experiments managed to differentiate negative and positive reviews quite satisfactory, but became considerably less accurate when introducing a third class for neutral documents. This might be due to the language used in neutral reviews where both praise and criticism is likely given, which could cloud the

classification judgement.

Finally, we had a goal of finding and competing with state of the art results within Scandinavian specific sentiment analysis, which we hoped would give more definite results. We found the best results, to the best of our ability, but we think there is more research to uncover and assess, as well as more experiments outside the academic domain. The SOTA results we found are included in the comparisons table 5.11 as bold text.

Chapter 6

Discussion and further work

In this final chapter we will be discussing some of the findings from our work.

6.1 Literature findings and discussion

During the literature review we had a goal of constructing an overview of popular research topics within the field of sentiment analysis for mainland Scandinavian languages. We also came across several interesting topics, that we did not explore to its fullest due to it not being in the scope of the thesis. In this section we will be discussing some of the results from our review, and discuss some of the more interesting findings that we think hold potential for further research.

6.1.1 Popular research topics

The literature review let us create an overview of popular research topics witing the field of sentiment analysis in mainland Scandinavian languages. We made a primary and secondary graph to get an overview of popular research topics in certain time periods. The primary graph consists of the research inline with our criteria, while the secondary graph contain both the main findings as well as secondary findings we were on

the fence of including. These graphs give us an idea of the most popular research topics, and gives an indication that little research was conducted before 2010. We believe we managed to complete our goal to a certain degree, but we believe it can be further extended. There exists more literature in the field than we previously imagined, which we consider a pleasant surprise, for the sake of the research field.

6.1.2 Language similarity

In Velldal et al. (2017), they experiment with different pipelines for creating tree banks for Norwegian Nynorsk and Bokmål, and attempted to combine the two in one pipeline. The results show that the combined pipeline provides a much higher word representation accuracy than that of the individual pipelines, at the cost of lexical size. They speculate that this is because the language variants are very similar, and that the main differences are minor spelling differences. Since Nynorsk is an adaptation of Bokmål, and Norwegian Bokmål, Danish and Swedish are all mainland Scandinavian languages and are quite similar, we wonder if it would be possible to create Scandinavian multilingual word vectors. Though Velldal et al. (2017) gained good results with this technique, Kirkedal et al. (2019) did the opposite. According to Kirkedal et al. (2019), there are several different language variants in Danish as well, which they chose to ignore for ease of use. This is something we believe could be interesting to explore further, to find out if the approach and results from Velldal et al. (2017) is applicable to other Scandinavian languages.

6.1.3 Translating existing resources

In several of the papers we read, we find experiments with machine translation of existing resources. Hammer et al. (2014) and Bai et al. (2014) experimented with translating the *AFINN* sentiment lexicon from **?**, and they report that it provided surprisingly adequate results. The lexicon was in both cases machine translated using Google Translate[1], with some manual correction. The main problems caused were due to very

[1] https://translate.google.com/

specific slang-words, which could not be machine translated properly. This also means that the lexicon will not understand slang words specific to the target language, unless they are extracted from someplace else or manually included. This gives an indication that translating existing English resources can still be very useful, increasing the availability of possible resources more than previously thought. We think this topic could be worth exploring further.

6.2 Experiments and results

We made quite a few experiments as described in chapter 5, gaining both good and bad results which we compared to the results we found in the literature review in chapter 4. We think that these results prove the feasibility of using English techniques and approaches with Scandinavian, or at least Norwegian, language resources, which was one of our research questions in section 1.1.1 of chapter 1. Furthermore, we think it proves the feasibility of using full-text reviews, with simple ratings and minor cleaning, as sentiment labeled training data for a supervised task. This means that we can potentially use more of these published reviews to train adequate autonomous sentiment classifier.

Our multi-class classification experiments yielded significantly worse results than the binary-class experiments. In our comparison, others were able to get much better results for the multi-class classification task, but none of them used the same types of data as us. Since we used full-text review documents for our features, and got significantly lower results for the multi-class compared to the binary-class, we think that this could have been due to the contents of the neutral documents, as they contain both criticism and praise, which might cloud our models judgement. We therefore think better results could be gained by using sentences as input sequences, though that would require another dataset.

We also had a goal of finding the SOTA results for sentiment classification in mainland Scandinavian languages, which we did to some degree. As mentioned in chapter 5, we found different experiments and results within the language specific domain, but

few of the results are directly comparable due to the differences in approaches. We believe that these results give a good indication of the current state of sentiment analysis results for Scandinavian languages, but we hope others want to expand on these experiments with similar approaches for each language within mainland Scandinavia.

6.2.1 Data specifications

Though we have performed many experiments, we believe there are more approaches to experiment with. The dataset from Velldal et al. (2017) contains many different attribute for each documents, and we did not have the capability to test all the possibilities with these data, due to lacking time and resources. As mentioned in section 5.2 in chapter 5, the data contains many different attributes in the *CoNLL-U* format, which could be used for several different approaches. For instance, we could use different word forms such as *lemma* or *part of speech*, instead of using the default unaltered words which we did.

We also speculate that the approaches using word embedding could be overfitted due to us using the combined training and evaluation examples to pre-train the embedding, before classifying on the same training data. A different approach could be to use sub-samples of the data, or even separate datasets for the pre-training and classification steps.

Additionally, since we only used the raw document reviews without any linguistic attributes, we think it could be possible to expand the dataset by crawling the web for more reviews, without the need of processing them and tagging with these linguistic attributes. This could expand the training data considerably, which might also increase the results from similar approaches to that we used.

6.2.2 Other model considerations for experiments

Due to the fact that the techniques we chose for our experiments, in chapter 5, have been around for several years, there have probably been made similar attempts. But we could not, to the best of our ability, find published research on the topic. Therefore,

we decided to look at some of the newer international techniques and how they fare. Among others, we looked at the new *XLNet* model from Yang et al. (2019), which has increased the English state-of-the-art (SOTA) in many NLP tasks (Yang et al. (2019)). The technique is closely related to that of Devlin et al. (2018), with their Bidirectional Encoder Representations from Transformers (BERT) model. Unfortunately, the XLNet model has few pre-trained models available, those that are available are trained on English text. This model requires extensive resources to pre-train, as the creators describe they use the following specifications: "We train XLNet-Large on 512 TPUv3 chips for 500K steps with an Adam optimizer, linear learning rate decay and a batch sizeof 2048, which takes about 2.5 days" (Yang et al. (2019)). These specifications are very demanding, and require much available resources, and would ideally not take document level input sequences.

We also looked at the popular BERT model from Devlin et al. (2018), which is based on the same technology as the XLNet model from Yang et al. (2019), but was released a year prior. Both models utilize bi-directional language modelling, making them very resource demanding. But due to the fact that the BERT model has been available for much longer, there are also more available pre-trained models, including multilingual language models trained on over a hundred different languages[2]. We made an attempt at fine-tuning the multilingual model to classify document strings as positive or negative, based on the NoReC data from Velldal et al. (2017). We trained it with three epochs, a sequence length of 500, training and evaluating in batches of four, on a Virtual Private Server (VPS) provided by UH-IaaS, taking approximately 9 hours to complete. The results were nothing short of a failure, managing to predict 2705 True positives, 0 True negatives, 231 false positives, and one false negative. Following the Matthews Correlation Coefficient (MCC) metrics for evaluation (Matthews (1975)), this would not be better than guessing randomly. Furthermore, we found an adaptation of the BERT model for document level input sequences, called *DocBERT*, created by Adhikari et al. (2019). This model is dependent on a pre-trained BERT model, which means it should be possible to use the pre-trained multilingual BERT model.

[2]https://huggingface.co/transformers/pretrained_models.html

6.3 Limitations

In hindsight, there are always things that could have been done better. In the case of our literature review, we think we could have gotten more results. Originally we started looking for papers using search term permutations that gave some, but few results. If we had spent more time constructing accurate search terms, we could possibly have spent less time reading through titles and abstracts of irrelevant papers, which in turn could have given us more results and increased the usefulness of this review. Additionally, we regret not having participated in ML courses during the writing period, as we believe this could have eased the steep learning curve of the experimentation part of the thesis. We also think that participating in more discussions with student peers could have given us a better understanding of many technical concepts. Regardless, we are pleased with the results we have presented.

6.4 Recommendations for Further Work

Though we have explored much of the research within the field of sentiment analysis in mainland Scandinavian languages, we think there is much more to be done. Compared to the international scene, the Scandinavian specific research is lacking in research and resources. We have therefore included some points we think should be explored further, that we did not have the capability to explore.

6.4.1 Other models

As mentioned in the previous section 6.2.2, we did some minor experimentation with newer, more advanced techniques. These techniques have yet to see considerable use in the international scene, thus the learning curve for these techniques are a also bit steeper than that of which we have already experimented with. Originally we wanted to experiment more with these techniques, but the XLNet option from Yang et al. (2019) required considerable resources which we did not have available. We therefore did some minor testing with the BERT model from Devlin et al. (2018), which as been avail-

able for longer and thus has more available resources to flatten the curve. The *Huggingface* module[3] for the Python programming language, has made several pre-trained models available for use through the module, including some trained on many different languages. These multi-lingual models has also been trained on both Norwegian language variants, Swedish and Danish. We therefore believe this requires further testing, as they can potentially be very beneficial for the Scandinavian sentiment analysis field.

6.4.2 Other data considerations

As mentioned in section 6.2.1, we believe there are more techniques to experiment with using the same dataset we did. We also believe that the training data for our techniques can be expanded using simple web-crawlers, as we only used the raw reviews and the rating they were given by the author. This is not something that requires much cleaning before it can be used, as described in our section about pre-processing in chapter 5. We also believe that we can experiment with different approaches using the different linguistic attributes applied in the dataset from Velldal et al. (2017). We therefore believe there is much more to be done with this data, and we hope others want to further experiment with it, using some knowledge from our own work, to adjust their own approaches.

We also noticed, at the time of this writing, that the people behind the NoReC dataset (Velldal et al. (2017)), from the Language Technology Group at the University of Oslo, has started work on a version of this dataset with fine-grained sentiment annotations[4]. We still do not know for sure what they mean by fine-grained, as there has not been published much in this new repository yet, but we believe it provides more opportunities for experimentation with sentiment classification in the future.

[3] https://huggingface.co/transformers/pretrained_models.html
[4] https://github.com/ltgoslo/norec_fine

6.4.3 Language similarity

Seeing as the mainland Scandinavian languages are all Germanic based langauges, they are also quite similar (Holmberg and Platzack (2005)). This means that they share much of the same grammar and many of the same words with minor differences, and that the people speaking one of these languages often times understand each other. These languages also have internal language variants, which means they are not identical. For the Norwegian language variants called *Nynorsk* and *Bokmål*, Velldal et al. (2017) has made attempts at combining the two in one pipeline with good results.

In our own experiments we had the option to filter texts based on language variants, but following the research from Velldal et al. (2017) we specifically chose not to, in order to create sentiment classifiers that work with Norwegian text regardless of language variations. Though since the data we used was grammatical texts from established media platforms, we did not experiment with the many Norwegian dialects, which vary in almost every municipality within Norway. But we did experiment with Nynorsk and Bokmål in a combined pipeline with quite decent results.

We therefore think that the same can be done with the other mainland Scandinavian language variants, or even combining all the mainland Scandinavian languages in one pipeline to create Scandinavian multilingual word representations for different Natural Language Processing (NLP) tasks. This is something we did not find mentions of in the Scandinavian research, but hope to see in the future.

6.4.4 Machine translation of existing resources

As mentioned in section 6.1, we have also found a few instances where experiments are made by machine translating existing English resources for use in Scandinavian languages. There has been several attempts at translating sentiment lexicons for use in a lexical sentiment analysis approach for Scandinavian languages. Many of these have reported good results with the use of these translated resources, but they have had to do some manual correction of the translations.

We therefore think that this can be a viable approach to gaining language specific re-

sources even for low-resource languages such as those of mainland Scandinavia. This approach still has some problems since they all have to manually fix some of these translations, since not all words have a direct translation. We think that for future research, there should be made attempts at creating a specific machine translation method for languages that is able to also disambiguate slang words for their respective language, and thus not be dependent on manual correction. This way there many more resources can be made available for NLP resources in these low-resource languages.

6.5 Summary and Conclusions

Throughout this thesis, we have made some effort to get more oversight, and contribute to the field of sentiment analysis within Scandinavian research. We have performed two main tasks within the thesis, one empirical and one practical. The first part of the thesis was concerned with uncovering and gaining oversight into the field of sentiment analysis in Scandinavian research, while the second part was more concerned with making a practical contribution to the said field.

In the first part of our thesis, we performed a literature review inspired by the Systematic Literature Review (SLR) approach, where we searched for and read research surrounding sentiment analysis within research in mainland Scandinavian languages. The goal of this review was to gain a good oversight of popular research topics, state of the art results from practical contributions, and an overview of tools, techniques and resources tested and made available for sentiment analysis in Scandinavian languages. We made a visualization of our findings, categorized by contributions and time of publishing to gain a good visual of the most popular topics within the field.

The second part of the thesis was more focused on making a practical contribution by testing and evaluating tools and techniques for sentiment mining of text data. During the first part, we were able to gain a decent overview of the field, which we used to make an educated selection of tools, techniques and resources to use for the creation of sentiment classifiers. We used a relatively new dataset from the Language Technology

group at the University of Oslo (Velldal et al. (2017)), and a selection of different pre-training and classification techniques we found, and did not find during the literature review. Thereafter we compared the results from our experiments to that we found in the Scandinavian research. Our results were not able to beat what we believe to be the SOTA results in the field, but provided decent results.

There were some tools and techniques we wanted to experiment with further, but we did not have enough time or resources to perform all of these experiments. There were also some topics we stumbled upon during our review that we though was quite interesting. We have therefore also made some examples for further research that we hope others might take in to consideration.

References

Adhikari, A., A. Ram, R. Tang, and J. Lin (2019). Docbert: Bert for document classification. *arXiv preprint arXiv:1904.08398*.

Alonso, H. M., A. Johannsen, S. Olsen, S. Nimb, N. H. Sørensen, A. Braasch, A. Søgaard, and B. S. Pedersen (2015). Supersense tagging for danish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA 2015, May 11-13, 2015, Vilnius, Lithuania*, Number 109, pp. 21–29. Linköping University Electronic Press.

Alpaydin, E. (2009). *Introduction to machine learning*. MIT press.

Bai, A., H. Hammer, A. Yazidi, and P. Engelstad (2014). Constructing sentiment lexicons in norwegian from a large text corpus. In *2014 IEEE 17th international conference on computational science and engineering*, pp. 231–237. IEEE.

Borin, L., M. Forsberg, and L. Lönngren (2013). Saldo: a touch of yin to wordnet's yang. *Language resources and evaluation 47*(4), 1191–1211.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Durgesh, K. S. and B. Lekha (2010). Data classification using support vector machine. *Journal of theoretical and applied information technology 12*(1), 1–7.

Eide, S. R., N. Tahmasebi, and L. Borin (2016). The swedish culturomics gigaword corpus: A one billion word swedish reference dataset for nlp. In *Digital Humanities 2016. From Digitization to Knowledge 2016: Resources and Methods for Semantic*

Processing of Digital Works/Texts, Proceedings of the Workshop, July 11, 2016, Krakow, Poland, Number 126, pp. 8–12. Linköping University Electronic Press.

Elming, J., B. Plank, and D. Hovy (2014). Robust cross-domain sentiment analysis for low-resource languages. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 2–7.

Fink, A. (2019). *Conducting research literature reviews: From the internet to paper*. Sage publications.

Friedman, J., T. Hastie, and R. Tibshirani (2009). glmnet: Lasso and elastic-net regularized generalized linear models. *R package version 1*(4).

Goldberg, Y. (2017). Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies 10*(1), 1–309.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Hagen, K., J. B. Johannessen, and A. Noklestad (2000). A constraint-based tagger for norwegian. *ODENSE WORKING PAPERS IN LANGUAGE AND COMMUNICATIONS* (1), 31–48.

Hammer, H., A. Bai, A. Yazidi, and P. Engelstad (2014). Building sentiment lexicons applying graph theory on information from three norwegian thesauruses. *Norsk Informatikkonferanse (NIK)*.

Harris, D. and S. Harris (2010). *Digital design and computer architecture*. Morgan Kaufmann.

Hohle, P., L. Øvrelid, and E. Velldal (2017). Optimizing a pos tagset for norwegian dependency parsing. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pp. 142–151.

Holmberg, A. and C. Platzack (2005). The scandinavian languages. *The Oxford handbook of comparative syntax*, 420–459.

Ikonomakis, M., S. Kotsiantis, and V. Tampakas (2005). Text classification using machine learning techniques. *WSEAS transactions on computers 4*(8), 966–974.

Johannessen, J. B., K. Hagen, Å. Haaland, A. B. Jónsdottir, A. Nøklestad, D. Kokkinakis, P. Meurer, E. Bick, and D. Haltrup (2005). Named entity recognition for the mainland scandinavian languages. *Literary and Linguistic Computing 20*(1), 91–102.

Johannessen, J. B., K. Hagen, A. Nøklestad, and A. Lynum (2011). Obt+ stat: Evaluation of a combined cg and statistical tagger. *Constraint Grammar Applications*, 26–34.

Jones, K. S. (2004). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Joulin, A., E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov (2016). Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Kann, V. and M. Rosell (2006). Free construction of a free swedish dictionary of synonyms. In *Proceedings of the 15th Nordic Conference of Computational Linguistics (NODALIDA 2005)*, pp. 105–110.

Karlsson, F., A. Voutilainen, J. Heikkilä, and A. Anttila (1995, 01). *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*.

Kirkedal, A., I. Copenhagen, B. Plank, L. Derczynski, and N. Schluter (2019). The lacunae of danish natural language processing. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pp. 356–362.

Kitchenham, B., O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman (2009). Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology 51*(1), 7–15.

Kniberg, H. and M. Skarin (2010). *Kanban and Scrum-making the most of both*. Lulu. com.

Le, Q. and T. Mikolov (2014). Distributed representations of sentences and documents. In *International conference on machine learning*, pp. 1188–1196.

LeCun, Y., Y. Bengio, and G. Hinton (2015). Deep learning. *nature 521*(7553), 436–444.

Levy, O. and Y. Goldberg (2014). Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pp. 2177–2185.

Li, Y. and H. Fleyeh (2018). Twitter sentiment analysis of new ikea stores using machine learning. In *2018 International Conference on Computer and Applications (ICCA)*, pp. 4–11. IEEE.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies 5*(1), 1–167.

Ludovici, M. and R. Weegar (2016). A sentiment model for swedish with automatically created training data and handlers for language specific traits. In *Sixth Swedish Language Technology Conference (SLTC), Umeå, Sweden, 17-18 November, 2016*.

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

Maron, M. E. (1961). Automatic indexing: an experimental inquiry. *Journal of the ACM (JACM) 8*(3), 404–417.

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure 405*(2), 442–451.

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119.

Mohammad, S. M., S. Kiritchenko, and X. Zhu (2013). Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.

Nielsen, F. Å. (2011). A new anew: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*.

Nielsen, F. Å. (2018). Danish resources. http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/6956/pdf/imm6956.pdf.

Nusko, B., N. Tahmasebi, and O. Mogren (2016). Building a sentiment lexicon for swedish. In *Digital Humanities 2016. From Digitization to Knowledge 2016: Resources and Methods for Semantic Processing of Digital Works/Texts, Proceedings of the Workshop, July 11, 2016, Krakow, Poland*, Number 126, pp. 32–37. Linköping University Electronic Press.

Palm, N. (2019). Sentiment classification of swedish twitter data.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research 12*(Oct), 2825–2830.

Petersen, K., F. R. M. S. . M. M. (2008). Systematic mapping studies in software engineering. *Ease 8*, 68–77.

Petersen, K., V. S. . K. L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Softare Technology 64*, 1–18.

Rosell, M. and V. Kann (2010). Constructing a swedish general purpose polarity lexicon random walks in the people's dictionary of synonyms. In *Proceedings of Swedish language technology conference*, pp. 19–20.

Rouces, J., N. Tahmasebi, L. Borin, and S. R. Eide (2018a). Generating a gold standard for a swedish sentiment lexicon. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Rouces, J., N. Tahmasebi, L. Borin, and S. R. Eide (2018b). Sensaldo: Creating a sentiment lexicon for swedish. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Rumelhart, D. E., G. E. Hinton, R. J. Williams, et al. (1988). Learning representations by back-propagating errors. *Cognitive modeling 5*(3), 1.

Sand, H., E. Velldal, and L. Øvrelid (2017). Wordnet extension via word embeddings: Experiments on the norwegian wordnet. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pp. 298–302.

Schütze, H., C. D. Manning, and P. Raghavan (2008). Introduction to information retrieval. In *Proceedings of the international communication of association for computing machinery conference*, pp. 260.

Socher, R., A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642.

Solberg, P. E. (2013). Building gold-standard treebanks for norwegian. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013); May 22-24; 2013; Oslo University; Norway. NEALT Proceedings Series 16*, Number 085, pp. 459–464. Linköping University Electronic Press.

Solberg, P. E., A. Skjærholt, L. Øvrelid, K. Hagen, and J. B. Johannessen (2014). The norwegian dependency treebank.

Starkweather, J. and A. K. Moske (2011). Multinomial logistic regression. *Consulted page at September 10th: http://www. unt. edu/rss/class/Jon/Benchmarks/MLR_JDS_Aug2011. pdf 29*, 2825–2830.

Velldal, E., L. Øvrelid, E. A. Bergem, C. Stadsnes, S. Touileb, and F. Jørgensen (2017). Norec: The norwegian review corpus. *arXiv preprint arXiv:1710.05370*.

Velldal, E., L. Øvrelid, and P. Hohle (2017). Joint ud parsing of norwegian bokmål and nynorsk. In *Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa, 22-24 May 2017, Gothenburg, Sweden*, Number 131, pp. 1–10. Linköping University Electronic Press.

Yang, Z., Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Zhu, X. and Z. Ghahramani (2002). Learning from labeled and unlabeled data with label propagation.

Appendix A

Appendix

A.1 Doc2Vec & LogReg experiments

This section contains the full tables of data specifications, training parameters and results from the Doc2Vec experiments using logistic regression classification. They are divided into three separate tables for data, parameter turning and results to make it easier to read.

A.1.1 Doc2Vec training parameters

These are the feature parameters used in said experiments.

A.1.2 Doc2Vec result table

A.2 TF-IDF and SVM experiments results

Here is the full table table of the SVM experiment results. We only included the best result in our comparison in **??**, due to the sheer size of the table.

Table A.1: Full D2V and LogReg data specifications.

Experiment	Pos/neg	Neu	Balance	Ratio	Lowercase
D2V1	P<=5 N>=2	–	False	–	False
D2V2	P<=5 N>=2	–	False	–	True
D2V3	P<=5 N>=2	–	True	1:1.3	False
D2V4	P<=5 N>=2	–	True	1:1.0	True
D2V5	P<=6 N>=1	–	False	–	True
D2V6	P<=6 N>=1	–	False	–	False
D2V7	P<=4 N>=3	–	False	–	False
D2V8	P<=5 N>=2	–	False	–	True
D2V9	P<=5 N>=2	–	False	–	True
D2V10	p<=5 N>=2	–	False	–	True
MD2V1	P<=5 N>=2	3-4	False	–	False
MD2V2	P<=5 N>=2	3-4	False	–	False
MD2V3	P<=5 N>=2	3-4	False	–	True
MD2V4	P<=5 N>=2	3-4	True	1:1	True
MD2V5	P<=5 N>=2	3-4	True	1:1	True
MD2V6	P<=6 N>=1	3-4	False	–	True
MD2V7	P<=6 N>=1	3-4	True	1:1	True
MD2V8	P<=5 N>=2	3-4	True	1:1.3	True

Table A.2: Full D2V and LogReg parameter specifications.

Experiment	Model	Vector size	Neg sampling	Min frequency	Alpha	Epochs
D2V1	PV-DM	500	10	1	0.065	50
D2V2	PV-DM	500	10	1	0.065	50
D2V3	PV-DM	500	10	1	0.065	50
D2V4	PV-DM	500	10	1	0.065	50
D2V5	PV-DM	500	10	1	0.065	50
D2V6	PV-DM	500	10	1	0.065	50
D2V7	PV-DM	500	10	1	0.065	50
D2V8	PV-DBOW	500	10	1	0.065	50
D2V9	PV-DBOW	500	10	2	0.065	100
D2V10	PV-DBOW	500	20	2	0.065	100
MD2V1	PV-DM	500	10	2	0.065	50
MD2V2	PV-DBOW	500	10	2	0.065	50
MD2V3	PV-DM	500	10	2	0.065	50
MD2V4	PV-DM	500	10	2	0.065	50
MD2V5	PV-DBOW	500	10	2	0.065	50
MD2V6	PV-DM	500	10	2	0.065	50
MD2V7	PV-DBOW	500	10	2	0.065	50
MD2V8	PV-DM	500	20	2	0.065	50

Table A.3: Full D2V and LogReg binary-classification results.

Experiment	Metric	Class		Accuracy
		Pos	Neg	
D2V1	Precision	0.95	0.62	90.5%
	Recall	0.94	0.66	
	F-score	0.95	0.64	
D2V2	Precision	0.96	0.67	91.8%
	Recall	0.95	0.73	
	F-score	0.95	0.70	
D2V3	Precision	0.96	0.63	91.0%
	Recall	0.94	0.71	
	F-score	0.95	0.67	
D2V4	Precision	0.96	0.64	91.1%
	Recall	0.94	0.72	
	F-score	0.95	0.68	
D2V5	Precision	1.0	1.0	100.0%
	Recall	1.0	1.0	
	F-score	1.0	1.0	
D2V6	Precision	1.0	1.0	100.0%
	Recall	1.0	1.0	
	F-score	1.0	1.0	
D2V7	Precision	0.86	0.66	82.6%
	Recall	0.92	0.52	
	F-score	0.89	0.58	
D2V8	Precision	0.98	0.81	95.5%
	Recall	0.97	0.86	
	F-score	0.97	0.83	
D2V9	Precision	0.98	0.84	**95.9**%
	Recall	0.98	0.85	
	F-score	0.98	0.84	
D2V10	Precision	0.97	0.81	95.3%
	Recall	0.97	0.83	
	F-score	0.97	0.82	

A.3 Doc2Vec and SVM experimental results

Here we present the full tables for our experiments using Doc2Vec embeddings and SVM classification.

Table A.4: Full Doc2Vec & LogReg multi-class results.

Experiment	Metric	Class			Accuracy
MD2V1	Precision	0.78	0.67	0.78	68.4%
	Recall	0.70	0.74	0.03	
	F-score	0.67	0.70	0.06	
MD2V2	Precision	0.75	0.64	0.00	–
	Recall	0.61	0.83	0.00	
	F-score	0.67	0.72	0.00	
MD2V3	Precision	0.71	0.67	1.00	69.0%
	Recall	0.73	0.75	0.03	
	F-score	0.72	0.71	0.07	
MD2V4	Precision	0.65	0.67	0.25	58.2%
	Recall	0.71	0.45	0.71	
	F-score	0.78	0.54	0.37	
MD2V5	Precision	0.67	0.72	0.24	60.5%
	Recall	0.77	0.46	0.65	
	F-score	0.71	0.56	0.35	
MD2V6	Precision	0.77	0.91	0.00	–
	Recall	0.31	0.99	0.00	
	F-score	0.44	0.95	0.00	
MD2V7	Precision	0.24	0.96	0.06	60.2%
	Recall	0.77	0.58	0.56	
	F-score	0.36	0.72	0.10	
MD2V8	Precision	0.66	0.69	0.29	62.1%
	Recall	0.74	0.52	0.59	
	F-score	0.70	0.59	0.39	

Table A.5: Full TF-IDF and SVM experimental results for binary-classification.

Experiment	Metric	Class			Acc
		Pos	Neu	Neg	
	Precision	0.87	–	0.00	
TF-SVM1	Recall	1.00	–	0.00	87.0%
	F-score	0.93	–	0.00	
	Precision	0.97	–	0.48	
TF-SVM2	Recall	0.87	–	0.80	86.0%
	F-score	0.92	–	0.60	
	Precision	0.95	–	0.60	
TF-SVM3	Recall	0.93	–	0.68	90.0%
	F-score	0.94	–	0.64	
	Precision	0.95	–	0.66	
TF-SVM4	Recall	0.95	–	0.64	91.1%
	F-score	0.95	–	0.65	
	Precision	0.93	–	0.79	
TF-SVM5	Recall	0.98	–	0.50	91.7%
	F-score	0.95	–	0.61	
	Precision	0.92	–	0.82	
TF-SVM6	Recall	0.99	–	0.38	90.9%
	F-score	0.95	–	0.52	
	Precision	0.93	–	0.76	
TF-SVM7	Recall	0.98	–	0.53	**91.8%**
	F-score	0.95	–	0.63	
	Precision	0.93	–	0.76	
TF-SVM8	Recall	0.98	–	0.52	91.6%
	F-score	0.95	–	0.62	
	Precision	0.93	–	0.76	
TF-SVM9	Recall	0.98	–	0.52	91.6%
	F-score	0.95	–	0.62	
	Precision	0.93	–	0.44	
TF-SVM10	Recall	0.90	–	0.51	85.2%
	F-score	0.91	–	0.47	
	Precision	0.90	–	0.76	
TF-SVM11	Recall	0.99	–	0.27	89.4%
	F-score	0.94	–	0.40	
	Precision	0.93	–	0.75	
TF-SVM12	Recall	0.97	–	0.49	91.3%
	F-score	0.95	–	0.59	

Table A.6: Full TF-IDF and SVM results from multi-classification.

Experiment	Metric	Class			Acc
		Pos	Neu	Neg	
TF-MSVM1	Precision	0.65	0.50	0.00	–
	Recall	0.06	0.97	0.00	
	F-score	0.11	0.66	0.00	
TF-MSVM2	Precision	1.00	0.49	0.14	49.00%
	Recall	0.00	1.00	0.00	
	F-score	0.00	0.66	0.01	
TF-MSVM3	Precision	0.71	0.67	0.00	68.6%
	Recall	0.72	0.75	0.00	
	F-score	0.71	0.70	0.00	
TF-MSVM4	Precision	0.64	0.73	0.27	60.3%
	Recall	0.80	0.41	0.69	
	F-score	0.71	0.53	0.39	
TF-MSVM5	Precision	1.00	0.88	0.00	–
	Recall	0.02	1.00	0.00	
	F-score	0.05	0.94	0.00	
TF-MSVM6	Precision	0.21	0.97	0.08	54.9%
	Recall	0.86	0.51	0.72	
	F-score	0.34	0.67	0.14	
TF-MSVM7	Precision	0.70	0.65	0.00	–
	Recall	0.67	0.76	0.00	
	F-score	0.69	0.70	0.00	

Table A.7: Full D2v and SVM data parameters specifications.

Exp	**Lower**	**Balance**	**Ratio**	**Pos/Neg**	**Neu**	**Neg docs**	**Pos docs**	**Neu docs**
D2_SVM1	False	False	–	P>=5 N<=2	–	2 326	11 597	–
D2_SVM2	False	False	–	P>=5 N<=2	–	2 326	11 597	–
D2_SVM3	False	True	1:1	P>=5 N<=2	–	2 326	2 326	–
D2_SVM4	True	True	1:1.3	P>=5 N<=2	–	2 326	3 020	–
D2_SVM5	True	True	1:1.3	P>=6 N<=1	–	353	459	–
D2_MSVM1	True	False	–	P>=6 N<=1	3-4	353	1 692	14 235
D2_MSVM2	False	False	–	P>=6 N<=1	3-4	353	1 692	14 235
D2_MSVM3	True	False	–	P>=5 N<=2	3-4	2 326	2 326	14 235
D2_MSVM4	True	True	1:1	P>=5 N<=2	3-4	2 326	2 327	2 327
D2_MSVM5	True	True	1:2	P>=5 N<=2	3-4	2 326	4 653	4 653
D2_MSVM6	False	False	–	P>=5 N<=2	3-4	2 326	11 597	14 235
D2_MSVM7	True	True	1:1.3	P>=6 N<=1	3-4	353	459	459

Table A.8: Full D2V and SVM parameter specifications.

Exp	**Loss**	**Penalty**	**D2V Ep**	**SVM Ep**	**D2V DM**	**VS**	**NS**	**MC**
D2_SVM1	Hinge	l2	10	1000	PV-DBOW	500	10	1
D2_SVM2	Hinge	l2	10	1000	PV-DM	500	10	1
D2_SVM3	Hinge	l2	10	1000	PV-DM	500	10	1
D2_SVM4	Hinge	l2	10	1000	PV-DM	500	10	2
D2_SVM5	Hinge	l2	10	1000	PV-DBOW	500	10	2
D2_MSVM1	squared_hinge	l2	100	10000	PV-DBOW	500	10	1
D2_MSVM2	squared_hinge	l2	100	10000	PV-DBOW	500	10	1
D2_MSVM3	squared_hinge	l2	100	10000	PV-DBOW	500	10	1
D2_MSVM4	squared_hinge	l2	100	10000	PV-DBOW	500	10	1
D2_MSVM5	squared_hinge	l2	100	10000	PV-DBOW	500	10	1
D2_MSVM6	squared_hinge	l2	100	10000	PV-DM	500	10	1
D2_MSVM7	squared_hinge	l1	100	1000	PV-DM	500	10	2

Table A.9: Full D2V and SVM results for binary-, and multi-class.

Experiment	Metric	Class			Acc
		Pos	Neu	Neg	
D2_SVM1	Precision	0.87	–	0.00	–
	Recall	1.00	–	0.00	
	F-score	0.93	–	0.00	
D2_SVM2	Precision	0.87	–	0.00	–
	Recall	1.00	–	0.00	
	F-score	0.93	–	0.00	
D2_SVM3	Precision	0.95	–	0.30	73.9%
	Recall	0.74	–	0.77	
	F-score	0.83	–	0.43	
D2_SVM4	Precision	0.88	–	0.39	86.5%
	Recall	0.98	–	0.08	
	F-score	0.93	–	0.13	
D2_SVM5	Precision	0.89	–	0.00	–
	Recall	1.00	–	0.00	
	F-score	0.94	–	0.00	
D2_MSVM1	Precision	0.25	0.91	0.12	78.3%
	Recall	0.43	0.83	0.12	
	F-score	0.31	0.87	0.12	
D2_MSVM2	Precision	0.23	0.91	0.00	–
	Recall	0.44	0.82	0.00	
	F-score	0.31	0.87	0.00	
D2_MSVM3	Precision	0.56	0.56	0.25	55.4%
	Recall	0.60	0.58	0.07	
	F-score	0.58	0.57	0.11	
D2_MSVM4	Precision	0.69	0.56	0.07	34.0%
	Recall	0.18	0.46	0.53	
	F-score	0.29	0.50	0.13	
D2_MSVM5	Precision	0.58	0.57	0.27	56.0%
	Recall	0.60	0.56	0.25	
	F-score	0.59	0.57	0.26	
D2_MSVM6	Precision	0.57	0.52	0.12	50.4%
	Recall	0.25	0.79	0.10	
	F-score	0.35	0.63	0.11	
D2_MSVM7	Precision	0.27	0.95	0.04	64.3%
	Recall	0.71	0.64	0.40	
	F-score	0.39	0.76	0.07	