Sentiment Analysis: a comparison of feature sets for social data and reviews

Thesis report

Laura Kreuk

Technische Universiteit Delft

TU Delft Delft University of Technology

Challenge the future

Sentiment Analysis: a comparison of feature sets for social data and reviews

Thesis report

by

Laura Kreuk

in partial fulfilment of the requirements for the degree of

Master of Science
in Computer Science
track Data Science and Technology with a specialisation in Information Architecture

at the Delft University of Technology
to be defended publicly on Friday November 2, 2018 at 3:00 PM.

Student number:	4152476
Supervisor:	dr. N. Tintarev
Thesis committee:	Prof. dr. ir. G. J. Houben (chair)
	Dr. N. Tintarev (supervisor)
	Dr. J. Urbano (external committee member)

TU Delft Delft University of Technology

Abstract

Consumers share their experiences or opinion about products or brands in various channels nowadays, for example on review websites or social media. Sentiment analysis is used to predict the sentiment of text from consumers about these products or brands in order to understand the tone of customers towards these products or brands. This thesis addresses sentiment analysis in the product domain on sentence level. In this thesis three data types are used which are collected by Unilever, review data which is text that contains the opinion of a customer towards a specific product. Social data, which can be tweets, Facebook messages, Instagram messages etc. and phone data which is a summary of a phone call of a customer about a specific product.

When conducting sentiment analysis one solution is to extract features from the data which can be given to a machine learning algorithm together with sentiment labels given by human annotators. The machine learning algorithm will generate a classifier which can predict a label for sentences. In sentiment analysis literature it is often not clear why certain features are chosen or for which data type certain features will work well. In this research we compare the differences when using several feature sets for the different data types.

We propose three feature sets for review data and three feature sets for social data. We focus on two aspects, comparing the different feature sets and comparing the data types. In our results we do not find significant differences in performance between the feature sets. The results suggest there might be feature sets which can improve sentiment analysis specifically for the data type, but a general feature set with standard features can be comparable to that result.

Acknowledgements

First of all, I would like to thank my thesis supervisor Dr. Nava Tintarev for her guidance throughout the thesis. She pushed me to think critically towards my own work and steered me in the right direction when I needed it. Furthermore, I enjoyed drinking tea during every meeting and the different tea flavours she let me choose from. I would also like to thank my thesis committee members for assessing my work.

Furthermore, I would like to thank Linda Hoeberigs for the guidance within Unilever and the help throughout the whole progress of the thesis. Besides that, I would like to thank Linda for the opportunity to conduct my thesis within Unilever and finding a project within Unilever. Moreover, I would like to think Liselotte Keizer for helping me through the process and checking continuously if I needed anything. In addition, she pushed me to learn the most from my internship at Capgemini and gave ideas on how to connect with more people. On top of that, the whole team of Capgemini working at the Unilever office that you for helping me with the manual annotation of the sentences and listening to my work throughout the whole process.

My sincere gratitude to the members of Epsilon, for their feedback and inspiring presentations. I think these meetings helped me improve my thesis.

Most importantly, I would like to thank my family and especially my parents for always believing in me and teaching me the value of education. Finally, I would like to thank Anton-Jan for always encouraging me and being there for me.

Laura Kreuk
Delft, October 2018

Contents

List of Figures

List of Tables

1

Introduction

1.1. Motivation

Consumers share their experiences or opinion about products or brands in various channels nowadays. This data can be exploited by companies to gather information about the sentiment of consumers when they talk about products or brands. Sentiment analysis is used to predict the sentiment of text from consumers about these products or brands.

In this thesis we focus on the features used for sentiment analysis of different data types (i.e. reviews, social data and phone data) in the product domain. Sentiment analysis tries to predict the polarity of a piece of text. In this thesis we focus on sentence level sentiment analysis with three classes, namely positive, negative, and neutral.

This thesis is a collaboration between TU Delft, Capgemini and Unilever, where Capgemini and Unilever have the role of client. When searching for a topic to write this thesis about I went to Capgemini to talk about the possibilities of writing my thesis there. Capgemini found an assignment for me at one of their clients, which is Unilever. Thus, within this thesis Unilever is the problem owner and my daily company supervisor is working at Unilever. This also means Unilever has provided the data used within this thesis of which parts are confidential.

Unilever uses a lot of data to answer questions from their business, mostly marketing related. Sentiment analysis can help answer some of the questions. When applying sentiment analysis over Twitter data this is a fast and effective method to identity the publics' feelings towards their brands [42]. Sentiment analysis can also be used to estimate the rate of acceptance of a product among customers and decide upon strategies to improve the quality of a product [40].

The data collected by Unilever includes social data, which can be tweets, Facebook messages, Instagram message etc. This social data is retrieved by using a tool called Brandwatch, that scrapes all the data from the Web based on a query that contains the search terms required. Furthermore, the collected data also includes reviews about products from different review sites which includes Amazon. This review data is retrieved with a different tool called Clavis. Finally, they also collect data directly from customers which is called internal data in this thesis, which can be via phone, email, form on a website of their brands, chat etc. From this internal data we selected only the phone data, which Unilever directly gathers from their customers, this is a summary of a phone call about a specific product. All these sources need to be processed in order to translate this into some useful insights for the business. This internal data is confidential and can be used in the project but it can not be shared with other parties than the ones involved in this project.

Unilever has built a machine learning based sentiment analysis algorithm. This model is trained separately for the different data sources but the methods used are the same for all of them. However, since the data of the sources is very different in structure and meaning Unilever wants to know if a more fitted solution can improve their sentiment analysis method. Social data consists often of short messages with slang and sometimes emoticons, links, hashtags etc. while reviews are longer texts

with more formal language.

Unilever had a few questions regarding their sentiment analysis method which might lead to improvements to their current method. The question which has been chosen to research after identifying the research available in this topic is to compare the differences when using several feature sets for the different data types. For each data type different features can chosen which reasonably will work well for that data type. By comparing these feature sets for different data types we also try to identify the differences between the data types. It was identified that in literature no studies exist yet that looks at the differences for sentiment analysis regarding the features extracted from the input data. Furthermore, it is often not clear why certain features are chosen in a sentiment analysis method in literature. There are papers which present their improvement of sentiment analysis for a specific data type, for example reviews, news articles or Twitter data but to our knowledge no study identifies the differences between these data types when using feature extraction methods.

1.2. Research objectives

In this thesis we focus on the features used for sentiment analysis for different data types (i.e. reviews, social data and phone data). By using different feature sets when applying a sentiment analysis method we try to identify which features can improve sentiment analysis compared to the method for sentiment analysis Unilever uses currently.

This process consists of two steps in order to investigate which features work best for a data type. First, we propose we can construct different feature sets which consist of reasonable features such that we can compare these feature sets upon the data types. We suggest we can construct different feature sets for which we suspect from the information in literature that they will work well for a specific data type. With this reasoning we constructed two research questions:

RQ1 Can we construct feature sets with reasonable features for review data for which we can identify differences in performance when using pair-wise comparison?

RQ2 Can we construct feature sets with reasonable features for social data for which we can identify differences in performance when using pair-wise comparison?

This leads to three feature sets proposed for review data and three feature sets designed for social data. Since the goal of this research is to identify the differences between the data types when using different feature sets, we compare the best performing feature set for each of the data types with which we can answer the main research question:

Main research questions: Can Sentiment Analysis be improved when constructing feature sets specifically for the data type?

1.3. Contributions

This thesis investigates the influence of various features on the performance for different data types. Our experiments uses three different feature sets designed for review data and three different feature sets designed for social data. The results from our experiments show the effect of different feature sets on the performance for different data types. We conduct two experiments to answer the research questions which leads to a few contributions of this thesis.

First of all, we study different feature sets and different ways to construct these feature sets. Since in literature about sentiment analysis it is often not clear why features are chosen, we look into ways to identify selecting relevant features and study the different feature sets resulting from this.

Secondly, we identify the influence of different factors, such as data quality and imbalanced data, on the performance when constructing feature sets for specific data types for sentiment analysis. Since the datasets we use are imbalanced we need to deal with imbalanced data, therefore some insights are found during this thesis about imbalanced data for sentiment analysis.

Also, we identified for each data type there might be different features which can influence the performance of sentiment analysis for that data type. We studied the different characteristics of several data types in this thesis and features which might work well in combination with these characteristics for sentiment analysis.

Finally, we identified there might be feature sets which can improve sentiment analysis specifically for the data type, but a general feature set with standard features can be comparable to that result.

1.4. Thesis outline

Chapter 2 First a general introduction into the concepts related to sentiment analysis is given. Then, the work related to this thesis is discussed and the boundaries of the sentiment analysis within this thesis are highlighted. The literature about review data is analysed afterwards, followed by the literature about social data. Finally, we point out the research gaps which lead to our research questions.

Chapter 3 The context of this project is explained, this includes the relation between this research and Unilever and how this has formed the research questions and methodology of this thesis.

Chapter 4 The data used within this research is explained. Both the already existing dataset retrieved from Unilever and the data which is added to that dataset within this thesis are defined. Furthermore, the data processing steps are clarified and the data is visualised.

Chapter 5 The methodology used to answer the research questions is demonstrated is this chapter. This includes a general overview of the sentiment analysis pipeline and an introduction of the different feature sets. Furthermore, the details of the features used are defined and the feature weighting methods. Finally, the classification method is explained.

Chapter 6 This chapter explains the experiment which is set up to answer RQ1 and RQ2. For both review and social data three feature sets have been used to compare upon their corresponding data type. We use hypotheses to make pair-wise comparison and identify the differences between the features sets. Finally, we discuss the results and answer the research questions.

Chapter 7 This chapter uses the best performing feature set for each data type to compare the differences between the data types. We use a hypothesis for each of the data types to compare which feature set performs better for the data type. We discuss these results to answer the main research question. Finally, we present additional results which we identified during our experiments which are also worth mentioning.

Chapter 8 The conclusion presents the answer to each of the research questions based on what we found during the experiments and the gained knowledge during this thesis. After that, the limitations and future research are presented in a final analysis.

2

Related work

2.1. Introduction

This chapter surveys the literature in the field of sentiment analysis related to this thesis. It describes the state of the art of this topic and tries to identify which methods to use in the further research within this thesis. *The focus of this research is the feature extraction when applying sentiment analysis for different data types.* Since the focus of the research is which features to select for sentiment analysis, the features extraction methods are an important part of this literature review.

- First, a general introduction into sentiment analysis is given, the problem is explained. Also, a general introduction into the feature methods, approaches and evaluation which can be used for sentiment analysis are explained. The reason to introduce these topics is to understand the literature which is studied in the other sections.
- Second, several problems which are closely related to detecting sentiment analysis are discussed. The basic idea of this is to show that these areas exist and a short intro on how it works. This section focuses more on sketching the boundaries of this research.
- Third, challenges on sentence level sentiment detection are explained to gain better insights what to be aware of during this research.
- Then, for the two main data sources in literature, reviews and social data, the features, methodology and evaluation methods used in literature are explained. Based on this, a choice can be made for the methods which should be used in the implementation phase.
- Finally, the research gaps and questions which follow from the literature are pointed out. The research gaps which will be studied during this research will be translated into the research questions of this thesis.

The main goal of this chapter is to introduce the research related to feature extraction when applying analysis for different data types. It also gives a high-level overview of the state-of-the-art of sentiment analysis. Section 2.1.1 introduces what sentiment analysis is and why it is useful. After that, the three main levels of sentiment analysis are discussed in Section 2.1.2. Then, the features which can be used to capture the relevant aspects related to sentiment analysis from the input text are explained in Section 2.1.3. There exist different techniques to build a sentiment analysis method which are explained in Section 2.1.4. Finally, the evaluation methods are explained in Section 2.1.5.

2.1.1. What is sentiment analysis?

Sentiment analysis is the field that tries to analyse the opinion, feeling, emotion, attitude of a person with respect to some entity, such as a situation, event, product, organisation, person, issue or topic. This opinion is often expressed as unstructured text which has to be processed using natural language processing (NLP), text analysis or computational linguistics. There are a lot of terms used when talking

about sentiment analysis which in some contexts mean the same such as opinion mining, emotion analysis, sentiment mining, opinion extraction.

Sentiment analysis tries to define the polarity at document, sentence or aspect level, the meaning of each of these levels is explained in Section 2.1.2. This categorisation of sentiment can be binary, positive and negative, or multi-class which means on a n-point scale (e.g. very good, good, neutral, bad, very bad). Beyond polarity there are also different emotional states which can be detected such as sad, angry or happy. However, in literature about sentiment analysis mostly the classes are positive and negative or positive, neutral, negative.

When large amounts of data are collected from which the tone is useful to analyse, it is not possible to do this by hand anymore. This is where sentiment analysis comes in, where a classifier is constructed which automatically can predict the tone for a piece of text. This classifier does this by predicting the possible label based on the knowledge it has about the data which is previously learned to the classifier.

Research in sentiment analysis mainly started early 2000, in the early years most of the research is about product reviews. However, the latest years social or Twitter data has gained more popularity. The reason for this is the enormous growth of social media, which includes for example Twitter, Facebook, reviews, forums and posts in social network sites. The main reason to start doing research in the sector is the enormous amount of data.

Generally, there are two types of methods for sentiment analysis, lexicon-based and machine-learning-based. The lexicon-based approach often uses a dictionary where each word is associated to a specific sentiment. Machine learning is often supervised, which means that a labelled dataset is needed. For this dataset, features can be extracted which depict the text and can be given to the machine learning algorithms which learns a model based on the features of a text instance and the labels associated to that. However, there also exist methods that combines both approaches, the different methods used to predict the sentiment of text are explained in more detail in subsection 2.1.4.

2.1.2. Levels of sentiment

Different research problems occur in the field of sentiment analysis. This is also related to the fact that different levels exist in which sentiment can be detected. These three levels are document level, sentence level and aspect or feature level. This research will focus on sentence level sentiment analysis dividing them in positive, neutral, or negative classes. The reason for this is the dataset used within this thesis contains sentences where each sentences has been assigned the label positive, negative, or neutral. Also, sentence-level sentiment analysis is more in depth than document-level sentiment analysis as explained in subsection 2.1.2.

Document level

At document level the task is to identify the polarity of the whole document, thus whether the full text is positive or negative [38]. This level assumes that the whole document is one information entity, this means that when multiple opinion are stated in a document only the overall sentiment is stated. It is not possible to compare multiple entities within the document. Most of the research papers describing this level are about online reviews [11, 38]. An example is the following review about tea which would be rated to be negative: ”This tea is inexpensive, but the taste is very weak. For a tea lover such as myself, it was a disappointment.”

Sentence level

The sentence level classification goes a level deeper and determines per sentence if the tone is positive, negative or neutral. Usually neutral means the sentence has no opinion in it. In this level thus the sentence is seen as one document, thus this level is fundamentally the same as document level analysis. A lot of papers describe a solution for determining the sentiment in sentences [9, 34, 47, 48]. An example of sentence level sentiment is the following sentence which would be negative: ”For a tea lower such as myself, it was a disappointment.”

A problem which is said to be closely related to sentence classification is subjectivity classification, which tries to identify if a sentence is subjective or objective. It is said that objective sentences contain factual information while subjective sentences contain opinions and views [25]. This problem related to sentence level sentiment analysis is explained in more detail in 2.3.1.

Aspect level

Sentence level for some applications might not be detailed enough. At sentence and document level the opinion targets are not identified, even if the document or sentence talks about one entity the author can still have a different opinion about the different aspects of this entity [25]. For example, if the author has a positive opinion about an entity, this does not mean that the author is positive about all aspects of this entity. In order to give a complete analysis the aspects need to be discovered and for each aspect the sentiment should be determined. An example to explain aspect-based level sentiment analysis is: ”This tea is inexpensive, but the taste is very weak.” In the first part of this sentence the aspect is ’price’ which is positive and the second part of the sentence has the aspect ’taste’ which is negative.

2.1.3. General features

In literature a lot of different feature methods are used in combination with sentiment analysis. There are even different type of methods to preprocess the data and to extract, select or smooth the features which eventually will be given to the machine learning algorithm that determines the polarity of the text. Even though there are a lot of different feature methods, it is often still unclear which features work best or why certain methods are used.

When choosing features there are a few things to take into consideration. First of all, the size of the dataset affects the performance for some of the techniques. It has been proven that using a small set of features works better for smaller datasets, while using a lot of features is too selective for smaller datasets. Furthermore, the domain of the text in the dataset is also of significant influence. There are certain features that work better either for a specific domain or in a non-domain dependent dataset [31]. Finally, the classification or regression technique can also perform better or worse in combination with certain features.

Preprocessing

Preprocessing is performed in order to clean or change the text before extracting some features from it. There are a lot of different preprocessing methods which are used before certain features can be extracted.

First of all, tokenization can be applied either on sentence level or on word level. This divides the text on sentence level into a sentence per token and for word level every token represents a different word. Tokenization usually is applied in combination with POS tagging or word frequency techniques which are explained in Section 2.1.3.

Another preprocessing method is stemming, which is used in [8] and [18]. This method reduces words to its morphological root. For example, ”stems”, ”stemmer”, ”stemming”, ”stemmed” would be changed when stemming to its root ”stem”. However, when comparing stemming it doesn’t improve the accuracy [31]. However, there are a lot of different stemming methods which depending on the method have a different influence on the performance. Some stemming methods may work better than others, which makes it harder to determine whether to use stemming and if so which stemmer is suitable.

Statistical substitution tries to replace certain tokens with a more generic label. For instance, number in a text can be replaced with the label NUMBER. Also, all tokens that are an instance of a product name can be replaced with the label _productname [8]. This method does not necessarily improve performance but it does help to prevent misclassifications.

Stopword removal refers to the most common words in a language. Usually they do not characterise what is being said with the text which makes them meaningless in the text. Because they take up quite some part of the text and the are not that useful it is common to remove these. This can be done either by counting occurrences of words are removing the words occurring above a certain threshold or by using a predefined list based on words occurring a lot in the language of the text.

Finally, spelling checking can be applied to the text to correct the mistakes made by the writer of some text. This can make feature extraction methods perform better. For example, when using a dictionary it would not be possible to find words in the dictionary with spelling mistakes, by using the spelling checking this problem is tackled.

Usually the preprocessing methods are chosen based on the feature extraction methods which are chosen. Based on the feature extraction methods there might be preprocessing needed, which will be identified when choosing the feature extraction methods.

Feature extraction methods

Feature extraction methods can be separated into three main big groups, NLP-based methods, lexicon-based method which need manual annotation (explained in **??**) and automatic methods which are based upon statistical functions. Each of these methods are explained in the following paragraphs. Although, it is important to note there are a lot more feature extraction methods than explained in these paragraphs. There are a lot of feature extraction methods which makes it impossible and useless to explain all of them in this survey. For this reasons, only the most occurring methods used in the literature explained in this survey are described.

NLP features Usually when extracting features, extracting some text features is one of the first steps. These are specific characteristics of text which can be used to describe the text.

One of these methods is POS tagging which tries to identify the part of speech of words within a sentence [18, 33]. These can be for example nouns, adjectives, adverbs, interjections etc. Adjectives are known to often state an opinion and thus can be useful to extract from a text. An example of POS tagging can be seen in figure 2.1.

```
>>> text = word_tokenize("And now for something completely different")
>>> nltk.pos_tag(text)
[('And', 'CC'), ('now', 'RB'), ('for', 'IN'), ('something', 'NN'),
('completely', 'RB'), ('different', 'JJ')]
```

Figure 2.1: Example of POS tagging, the first word is a coordinating conjunction (CC), followed by an adverb (RB), preposition or subordinating conjunction (IN), noun (NN), an adverb again and an adjective (JJ).

Secondly, negation detection methods also are often used [8, 38, 39]. The reason for this is the occurrence of negation words might change the semantic orientation, for example not bad is equal to good.

Also, n-grams is an often used technique for feature extraction, which means the text is divided into different combinations of consecutive words with their length based on the size of n. For example, when n is two the n-gram of the sentence ”I am really happy with this product!” is as follows:

- I am
- am really
- really happy
- happy with
- with this
- this product

When n is three, the n-grams are:

- I am really
- am really happy
- really happy with
- happy with this
- with this product

After the n-grams are created they can be counted using a formula to represent the relative importance of the features in the document [31]. The formulas which are used the most for representing the n-grams are term presence and term frequency which either look at the occurrence of certain words or at how many times it occurs in a sentence. Sometimes also the most occurring or least occurring terms are extracted or removed.

Statistical based features Point-wise Mutual Information (PMI) models between two terms the degree of statistical dependence:

$$PMI(term_1, term_2) = log_2\left(\frac{Pr(term_1 \wedge term_2)}{Pr(term_1)Pr(term_2)}\right)$$

In this formula, $Pr(term_1 \wedge term_2)$ represents the co-occurrence probability of the two terms together, and $Pr(term_1)Pr(term_2)$ the co-occurrence of the terms if they are independent. This formula can thus be used to see how correlated certain words are, for example when picking a positive word like good for the second word can be seen how correlated to good it is and thus compute the association.

Chi-square is a method to represent the ratio between a word and a class. In this formula $p_i(w)$ represents the conditional probability that a word w is occurring in a class i. F(w) represents the fraction of documents or sentences the word appears in.

$$\chi_i^2 = \frac{n \cdot F(w)^2 \cdot (p_i(w) - P_i)^2}{F(w) \cdot (1 - F(w)) \cdot P_i \cdot (1 - P_i)}$$

Other statistical feature selection methods are information gain (IG), occurrence frequency, log likely-hood and minimum frequency threshold [2].

Lexicon-based feature methods A sentiment lexicon is a dictionary of words with a label assigned which indicates the polarity. When using sentiment lexicons as a feature, there are several options how to use it as a feature. For example, in a sentence the number of positive words and the number negative words in a sentence can be counted or the number of positive divided by the number of negative words in a sentence etc. For this it is possible to use an already existing sentiment lexicon or construct a sentiment lexicon yourself based on the data. Sentiment lexicons can either be manually or automatically created. For a manual created lexicon a human annotates all the instances with a label. While for automatic lexicon construction is done by computing a score by using a linguistic resource [16].

Choudhury et al. [7] use a sentiment lexicon after preprocessing their data. They use the SentiWordnet lexicon which describes each term with a triple of positive, neutral and negative of which the sum of the three equals one. When there are multiple words in one entry they average the score of each class to normalise the entry. One of their findings about sentiment lexicons is that they are more suitable for news, blogs, product reviews and movie reviews than for social data because the linguistic syntax is more formal. Other papers that use a lexicon as a feature are [30], [33], [39], [18], [8], [31], [35], [4], [7] and [15].

Feature reduction

When a lot of features are used the dimensionality of these features can be reduced by representing them as a function of the original set of features. A method to do this is called Latent Semantic Indexing (LSI) which is one of the most popular transformation methods, which changes the original word feature with a linear combination [29].

Principle component analysis (PCA) is another linear technique for dimensionality reduction [29]. It uses a linear mapping to represent the data in a lower dimensional space while keeping the variance in this lower dimensional representation maximised.

Smoothing

Before assigning a value to a term frequency they can be smoothed using a smoothing method. According to Dave et al. [8] Laplace smoothing performs best and gives a higher performance compared to the baseline method they use. With Laplace smoothing the following formula is used to give a value to words.

$$p(w_i) = \frac{count(w_i) + 1}{\sum_{i=1}^{M} count(w_i) + M}$$

Other methods which can be used for smoothing are Witten-Bell, Good-Turing and add-one smoothing [8]. Smoothing of features was only seen in one of the papers analysed in this literature study

which can either mean that is generally does not work well or that because it is not used that others are not using it for that reason.

2.1.4. General methodology

There are a lot of different techniques used for sentiment analysis. Before explaining what techniques are used in specific papers about reviews or social data a clear distinction should be made regarding to the choices which can be made. These different options are explained and reasons for choosing an option are also discussed.

- *Machine learning*, lexicon-based or hybrid
- *Supervised learning* versus unsupervised learning
- *Classification* versus regression

Since this thesis focuses on the different feature sets for sentiment analysis, the italic text in the list above represents the choices made regarding to the classification part of the methodology. Also, to set the classification method the *support vector machine (SVM)* is used in this research for reasons that the focus is features and the SVM is mostly used in literature and said to perform well.

Machine learning, lexicon-based or hybrid

After the features are selected a sentiment classification method or regression model can be used. There are different groups of classification which can be seen in figure 2.2 as well as the most popular classification methods. As an addition to this figure regression models are also possible to use instead of classifiers. Some of these regression models are for example SVM regression, linear regression, ordinal regression etc. The main groups these of methods to detect sentiment can be divided into are Machine Learning, Lexicon-based and the hybrid approach. Where the hybrid method combines both the machine learning and lexicon-based approaches.

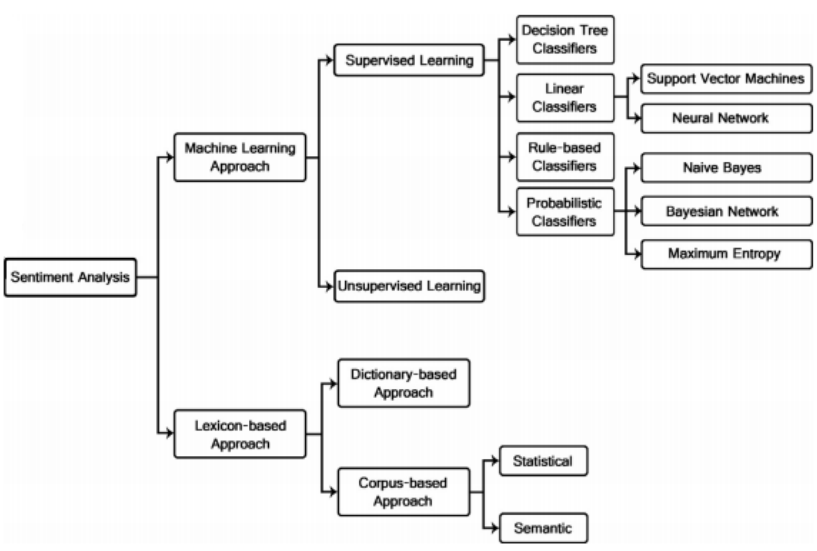

Figure 2.2: Overview sentiment analysis classification techniques [29]

Lexicon-based The lexicon-based method uses a collection of words and terms which are known to have a certain sentiment. This approach can be divided into dictionary-based approach and corpus-based approach, where the dictionary-based approach uses a list of words which is enlarged by finding the synonyms of the words using for example thesaurus. The Corpus-based approach uses a statistical or semantic approach when it tries to find opinion words within a certain context. It tries to find patterns that occur or syntactic patterns [29].

Machine learning The machine learning approach uses known machine learning algorithms to classify the data as text into different sentiment classes or to fit the data to a regression model. A training set is used in which each record is linked to a certain class. Based on the different features these records have, the classification or regression model is made. Then, for an instance of an unknown class the class it belongs to is predicted based on the underlying features of the instance. As for the regression model the data is fitted to a trend/line of values. Within the machine learning approach the solutions can also be divided into two classes, the supervised and unsupervised learning methods. An overview of how the machine learning method works is given in figure 2.4.

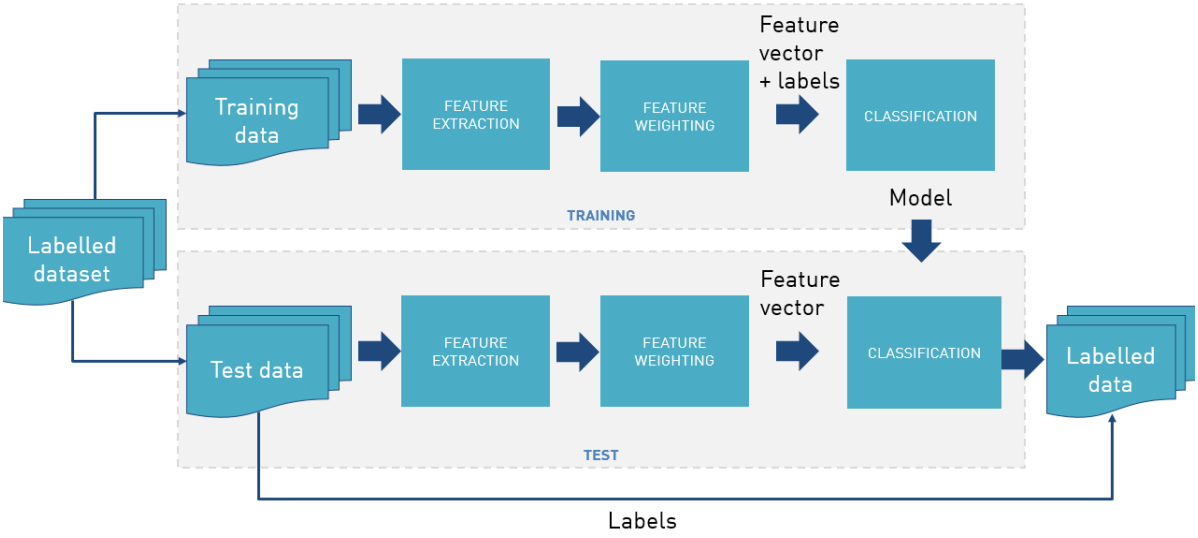

Figure 2.3: Overview of supervised sentiment analysis method

Supervised and unsupervised learning

When using a supervised method a large set of labelled training data should be available. Labelled data means that a set should be created with instances of text (document, sentence or phase level) with a label assigned based on their polarity. Within the supervised method this is done by using human annotators to give the labels to the instances of the text. This is the only way to know if a given label is correct. However, for some instances giving a label might not be that straightforward since people have different understandings of what might be positive in a certain domain for example. This makes the task more difficult and this is why often three annotators per text instance are used and a majority vote is determining the final polarity of the instance.

There are a lot of different classifiers that can be used for sentiment analysis, decision tree classifiers, linear classifiers, rule-based classifiers and probabilistic classifiers. The probabilistic classifier assumes that each class is a component of a model that is called the mixture model. For each of these classes a probability can be computed of sampling a term into that class. Some of the common known classifiers which are probabilistic are Naïve Bayes, Bayesian Network and Maximum Entropy. Linear classifiers use the normalised document word frequency and linear coefficients with the same dimensionality as feature space to predict the outcome. There are a lot of linear classifiers of which the Support Vector Machine (SVM) is the most common known. The SVM has proven to work really well for text data and is often said to be one of the best performing algorithms for sentiment analysis. The decision tree classifier translates the training data into a tree-like structure using conditions on the attribute values to divide the data. This condition can be the absence of one or more words and the division is build recursively where the leaf nodes need a minimum number of records. The records of each leaf are then used for the purpose of the classification. Finally, rule based classifiers use a set of rules to model the data space. When using these rules a rule based classifier is really close to using a decision tree classifier. The main difference however is the fact that a decision tree needs to be divided strictly hierarchical over the data space, while the decision space for a rule-based classifier

allows overlaps. Since the problem requires a multiple-class solution also regression models should be taken into account [29]. There are a lot of different regression algorithms as well, they are used to learn the relationships between real-valued attributes. These regression models are Support Vector Regression, linear regression, ordinal regression etc.

Sometimes, it is hard to create labelled training documents. However, unlabelled documents are usually more easily to gather. The unsupervised learning method tries to categorise these unlabelled documents with other methods than human annotation by finding patterns in the data, for example using a keyword list and sentence similarity measure. There are many methods proposed in different papers which try to categorise the semantic orientation using the unsupervised learning method [29].

Besides supervised and unsupervised learning methods there also exist semi-supervised learning methods. With these semi-supervised methods there is some labelled and unlabelled data. This labelled data can then be used to give the unlabelled data also a label. For example, using a sentiment lexicon (a dictionary with sentiment labels assigned to each word) or the method by Wang et al. [48] explained in subsection 2.3.1.

Regression or classification

When conducting sentiment analysis this can be from a two-class problem (positive vs. negative) until infinite-class (real-valued). Performing sentiment analysis on a real-valued scale means that a score is given on for example a scale from 1 until 100. Of course, 100 are a lot of values which is why regression would be used for this problem. However, there is separation for the scale from which the problem instead of classification becomes regression. Drake et al. [11] use different classification and regression algorithms and different numbers of classes to see for which number of classes it would be more efficient to use regression instead of classification. They find out that for a two-class case the best result comes from classification algorithms, while in the three and four class cases the results are mixed and beyond four classes the regression algorithms perform better.

2.1.5. Evaluation

In order to understand how evaluation works a few terms should be introduced. Since the method chosen within this thesis is a supervised approach, a labelled (human annotated) training set will be used to train a machine learning model, the main method to evaluate the performance of this constructed method is to use a test set and a metric which can compute this performance. Thus, to give a effective overview of what is needed to evaluate a sentiment analysis method an overview is given:

- Gold standard labelled data, for sentiment analysis, when using a supervised method, human annotated data is seen as the gold standard. This means that data to train and test the model is labelled (e.g. positive, negative or neutral) by humans. However, even for a human it can be quite hard for some instances to see what the sentiment of a piece of text is. Part of this data is then used to train the machine learning algorithm and the other part is used as a test set to validate how well this trained model is performing.
- Baseline method, the baseline method is used as a bare minimum which the sentiment method should proceed. This is used as a point of reference to identify the effectiveness of the algorithm. The performance, which should be shown with an evaluation metric, of the proposed sentiment analysis method should at least proceed the baseline algorithm.
- Evaluation metric, this metric is a way of presenting in a number how well the proposed method is performing. There are different type of metrics which are explained later in this section.
- Cross-validation, this divides the data based into k-folds, where k is a number, into k different sets. This helps with the problem of splitting a labelled dataset into a training and test set. The test set should be as big as possible to improve the validation of the model and the training set should also be as big as possible to reach the best learning result. Usually, one of the k sets is used as test set and the others as training set. With this method a dataset is split into k mutual exclusive subsets which all have almost the same size. The model is then train and tested k times with these sets, where each time the test set is a different set and the model is tested on data

minus test set [21]. This method limits the problem of overfitting which is especially useful when the amount of labelled data is not that big.

Metrics

There are a few metrics which are commonly used when checking the performance of a machine learning model. These commonly used metrics will be explained first.

Precision and recall Precision and recall are commonly used to identify the measure of relevance. Precision is the fraction of relevant instances of the retrieved instances and recall the fraction of retrieved instances of the total amount of instances. Thus precision is the measure that determines how many times the sentiment rating was correct. Recall shows how many documents with sentiment where also classified with having sentiment instead of being neutral. For classification tasks the following formulas are used to calculate precision and recall:

$$Precision = \frac{tp}{tp + fp} \tag{2.1}$$

$$Recall = \frac{tp}{tp + fn} \tag{2.2}$$

(Where tp is true positive, fp is false positive and fn is false negative)

Accuracy Accuracy, which can also be called precision, looks at how often the sentiment prediction model is correct. In a lot of the sentiment analysis method only accuracy is used to validate the model, however this does not explain the full evaluation of the method. When only the accuracy is known it's hard to determine if the model really works that well as proposed. For example, when a class is imbalanced (90% positive and 10% negative) and all the instances are identified as positive the accuracy is still high but it is not good if the model only identifies positive instances. Also, in this literature review most of the papers discussed use only accuracy as a measure in combination with cross-validation, which makes it unclear how well their really perform.

F-measure or F1-score To measure the performance the F-measure can be used. The following formula explains the F-measure [36]:

$$F = (1 + \beta^2)\frac{precision \cdot recall}{(\beta^2 \cdot recall) + recall} \tag{2.3}$$

Precision, recall and F-measure (also F1-score with β is 1 or F-score) are most used when measuring the results of sentiment analysis. However, when using the values from the confusion matrix (true positive, false positive and false negative) there might be different opinions about the number of false positives and false negatives which are different for different problems and thus are not a clear metric to see how well the model performs. Also, precision, recall and F1-score lack robustness to imbalance [3]. Also, these metrics work better for two-class classification, but since the problem is divided into more categories it is more important to see if the predicted sentiment is close to the real sentiment [11].

ROC curve and AUC Another commonly used metric is the ROC curve, which plots the results from the most positive to the most negative result. The area under the curve (AUC) is used here as well with imbalanced datasets, when it is run once it is known as balanced accuracy. ROC gives the full range of true positives and false negatives [44].

$$ROC = \frac{P(x|positive)}{P(x|negative)} \tag{2.4}$$

$$AUC_b = (sensitivity + specificity)/2 \tag{2.5}$$

Error rate Drake et al. state that when there are more then 2 classes it is more important that the predicted sentiment is close to the true sentiment. This is where the error rate is used, of which different versions are possible to represent this. When talking about just the error rate for two classes this is equal to 1-accuracy.

In the paper of Drake et al. [11] the mean squared error is used as can be seen in equation 2.6 to compare different regression and classification outcomes. As this is a clear solution my preference at the moment would be to use this metric. Also, according to [3] MSE works best for systems that are related to reviews.

$$MSE = \frac{1}{R}\sum_{r \in R}(score(r) - prediction(r))^2 \tag{2.6}$$

A measure often used for regression problems is the root mean squared error, which can be seen below.

$$RMSE = \sqrt{\frac{\sum(predicted_i - actual_i)^2}{N}} \tag{2.7}$$

Where N is the total number of observations

There are some other metrics which are closely related to these ones such as mean zero error, mean error, mean percentage error etc.

2.2. Closely related problems

In this section some closely related topics in literature are mentioned. This is used as an overview of the field and to show that these exist but are not chosen for the research. Also, the reason why these problem are different from the problem considered in this research is explained. This section focuses on sketching the boundaries of this research.

2.2.1. Cross-domain SA

Sentiment analysis can be performed either on generic data or domain specific. It is important to make this distinction because it has been shown that sentiment classification is highly sensitive to the domain of the training data [25]. For domain specific data this means when a classifier is trained on data from one domain and this classifier is used to predict labels for data from another domain it will perform poorly. Furthermore, certain words can be positive in one domain while it has a negative meaning in another domain. For example, in the domain of products 'lasts long' is usually positive because it means the product will stay good longer, while in a movie review this might suggest that the movie is too long which can be boring.

Multiple papers describe solutions for cross-domain sentiment analysis. The reason for this is that for some domains there might not exist labelled datasets. Labelled datasets are entities of data, which can be a document, sentence or phrase based on the sentiment level of the entity, with a polarity assigned to the entity. For example this can be sentences of reviews, with for each sentence either positive, neutral of negative assigned to it ideally by a human annotator. For building a classifier on multiple domains the original domain with the training data is often called the source domain and the new domain which is used for tested has no or a small set of test data which is called the target domain. Because of the lack of labelled data for the target domain there is the need of domain adaption or transfer learning [25]. most methods either propose a method which requires a small training set with labelled data for the new domain or no data for new domain.

Summarising this problem there are two requirements for this problem. First, the need for a classifier on multiple domains. Second, the existing datasets do not contain enough data for all the different domains. This can either mean that for the different domains there is a small set available or for certain domains there does not exist a labelled dataset. Because the data used in this research is mainly on one domain for which labelled data is available or created cross-domain research is not part of the further research.

2.2.2. Opinion summarising

Opinions state what a single person thinks about a topic and usually when one opinion is stated this does not lead to an action. However, when a lot of people state the same opinion this is something that could require an action. Opinion summarising is related to summarising multiple documents which is called text summarising. Although, the difference between the two is the fact that for summarising opinions the aspects and entities of the text are the factors important for these opinions and their sentiment. While for document summarising some important sentences of this document are selected. Thus, opinion summarising covers the topics is the text and tries to find the sentiment of those important entities, which is a more in-depth summary than for text summarising [25].

There are different type of summarising problems from which aspect-based summarising is one. A certain topic can contain a lot of different aspects of which people give their opinion on. For example, when multiple reviews are written about a product there can be several aspects of the product that can be rated. When summarising the opinion about the product there would then be a list of aspects and for every aspect there can be different opinions which have a quantity indication how many people have the same opinion about the aspect. This example is an aspect-based summarising which is identified by Hu and Liu [18].

They perform this task in three steps, the first step is to mine the product features that users have been commenting on. Secondly, for each review the opinion sentences are identified. At least one of the product features from the first step must appear in the opinion sentences. The opinion of each sentences is extracted by first looking at the set of adjective words, which are seen as opinion words. Then, the semantic orientation for each opinion word is checked. Finally, for the complete sentence the semantic orientation is determined. When all the opinion sentences are given a semantic orientation the summarising is constructed as describe above.

Most of the summarising methods are similar to the example explained above [25]. However, it is out of scope for this research.

2.2.3. Topic annotation

Since the model from Unilever besides extracting sentiment from the sentences also applies topic modelling which is explained in section 3, it is useful to look at some papers in literature related to this as well. Abstract ”topics” within bodies of text are found by topic models. These topics can be used to gain insights in the text. Topics in the text are also often the target of the sentiment.

Mullen and Collier [33] present a sentiment analysis model where they also provide the topic of the text if possible. They created several classes of features based on the position of phrases and their semantic values in relation to the topic of the text. They use secondary information available to describe the topic, however when this is not available entities are tagged. By tagging entities the try to represent the relationship between value phrases and topic entities.

2.3. Challenges of sentence level sentiment detection

There are some fields related to sentence level sentiment analysis which make it hard to detect the sentiment correctly in sentences. Three of these fields will be explained in the following sections.

2.3.1. Subjectivity classification

Subjectivity classification states that sentences can be divided into two categories, subjective and objective sentences. Where subjective sentences describe the view or opinion of a person, while

objective sentences contain factual information. Subjectivity classification at sentence level is more useful than at document level, the reason for this is document usually exist from subjective and objective sentences. It is important to state that subjective and objective sentences are not the same as the separation of positive and negative versus neutral. Positive, neutral and negative can all be subjective or objective, meaning that both objective and subjective sentences can contain sentiment. Furthermore, it is also possible for a objective or subjective sentence not to contain any sentiment. Examples of these different option can be found in table 2.1.

Table 2.1: Examples of subjective and objective sentences

Sentences	Sentiment	No sentiment
Subjective	"I like Lipton tea"	"I think he drank Lipton tea"
Objective	"The tea bags broke when putting them in hot water"	"Lipton is a Unilever brand"

Most of the sentence-level subjectivity classification are designed by using supervised learning approaches. However, since supervised learning methods require labelled training data which is usually manually collected this is difficult, expensive and time-consuming to obtain. Because of this reason, there is a lot of motivation to explore semi-supervised where the learning algorithm is trained on both labelled and unlabelled sentences [25].

Wang et al. [48] propose for sentiment subjectivity classification a semi-supervised learning approach which can classify sentences as either subjective or objective. For the classification part they compare decision tree models, namely C4.5, C.4.4 and Naive Bayes tree. To identify if a sentence is subjective or objective, they first train a classifier with a small number of labelled sentences with all the features. This classification model then can be used to classify unlabelled sentences. The just classified sentences are then selected based on their value computed with a selection metric which ranks these sentences and the sentences with the highest raking are selected. The labelled training set is expanded with these highest rated sentences and this procedure iterates until all the sentences are labelled or a certain limit of iterations is reached.

The selection metrics they use, are confidence degree and their own constructed selection metric called Value Difference Metric (VDM). This selection metric which is used to select the classified unlabelled sentences with the highest raking is very important to the performance of self-training. The confidence degree is the traditional metric used for self-training, which ranks a classified unlabelled entity by looking at the differences class membership probability. However, this metric doesn't work good for instances if the class membership probability have a low ranking performance. This is why the authors propose their own selection metric called VDM, which looks at the distance between feature conditional probability estimates for entities.

The data which is used is first pre-processed by tokenizing the data, assigning part-of-speech (POS) tags to all the words and stemming the words. After pre-processing the sentences they are some subjectivity clues which are used onto the stemmed words and their POS tags. They evaluate their method by using the Multi-Perspective Question Answering (MPQA)corpus, which is a publicly available manually annotated dataset, as test set. They use as metrics accuracy, the area under the ROC (AUC) and F-measure. They compare the results of these metrics for the decision tree classifiers of which NBTree performs best. When comparing the two selection metrics in their evaluation the VDM has performs better for all the metrics the authors use. Also, they compare their method to a supervised methods which they construct by training them on the initial 350 labelled sentences. This shows that their method achieves comparable performance to supervised learning models. However, it should be noted that they train their own model on model their 350 sentences and training a supervised method on that amount of sentences is very limited.

Barbosa and Feng [4] constructed a method where they first distinguish between subjective and objective messages and after that detect in the subjective messages positive and negative messages. This is a so-called two-step classification method. First they build a classifier to distinguish between subjective and objective messages. After that, they take the subjective set of messages and classify this into positive and negative messages.

Even though the difference between subjective and objective seems like it is the same as breaking down the set into positive and negative versus neutral, this is not the same. Although, it might make

sense for certain purposes to only look at the sentiment of subjective or objective sentences this problem is different from looking at sentiment in general.

2.3.2. Conditional sentences

Narayanan et al. are the first to look into the problem of conditional sentences. They basically say that conditional sentences are a special kind of sentence which means they need a special treatment. Different kind of sentences express sentiment in a different way which means there is no one-technique-fits-all solution according to Narayanan et al. [34].

Conditional sentences describe a hypothesis and its result. The authors say, a conditional sentence contains two parts which are dependent on each other, the condition clause and the consequent clause. The relation between the condition clause and consequent clause has influence on weather the sentence has negative or positive sentiment. These conditional sentences often have a different sentiment structure than a non-conditional sentence would have. For example, "If you do not like this Dove shower foam, you should try one from Fa.". This sentence contains of the first part which does not express any sentiment and the second part is positive about the Fa part. Whereas in a non-conditional sentences this might look like the following "You should try the shower foam from Fa!". Since the structure of conditional and non-conditional sentences is very different, using non-conditional sentences method for this conditional sentence might not perform well [25]. Most of the conditional sentences can be detected by certain conditional connective, which in most cases is 'if'.

Narayanan et al. use POS tags to identify conditional sentences. To solve this problem the authors use a support vector machine (SVM) with a set of linguistic features. The features the use are sentiment words, POS tags, words indication no opinion, tense patterns, special characters, conditional connectives, length of conditional and consequent clauses, negation words, topic location and opinion weight.

They input data they use is from forums, which means there is a main topic and a main message on which people can react. This means that each forum consists of threads which all have a topic and a person who write the main message. Usually, each thread will have answers from different people with a timestamp, the name of the author and some text. They used conditional sentences from five different forums with different topics and manually annotate 1378 different sentences. With using 10-fold cross validation they retrieve the results which they compare using accuracy, precision, recall and the F-score. Experiments were conducted with different combinations of the feature set, but all the features together gained the best result for all metrics.

They also used three different classification strategies. Firstly, a clause-based strategy where one classifier is used for the condition and one for the consequent. Secondly, a consequent-based classifier which classifies only the consequent clauses, because the clause often contains no opinion. The third classification method used is whole-sentence-based which predicts the opinion of the whole sentence. This last classification method gained the best result for all the metrics, although the increase in precision is the lowest compared to the other classification strategies [34].

Concluding, when dealing with conditional sentences a different sentiment analysis approach is needed than for non-conditional sentences. However, this is not in the scope of the thesis which is why no further research will be done in this direction.

2.3.3. Sarcasm

Sarcasm is when the writer of a text means the opposite of what he or she says. In sentiment analysis this means that the author writes something positive and means it in a negative way and vice versa. This is something which is hard to deal with, especially since even humans can have a hard time to detect sarcasm [47]. Based on the experiences of Liu, sarcastic sentences do not occur that often in reviews of products or services [25].

Tsur et al. [47] used a semi-supervised learning method to identify sarcasm on a sentence level for product reviews. They use a small set of manually labelled sentences which are given a score between one and five, where one means clear absence of sarcasm and five means definitely sarcastic. To expand this training set they created search engine queries and they stated that sarcastic sentences often occur in texts with other sarcastic sentences. They made queries of the basic parts of the sarcastic sentences

and used this to extract sentences which contain this basic part as well or are accompanied with this query. They use two types of features: synthetic and high-frequency words (HFWs) and content words (CWs) pattern-based features. For the HFW and CW pattern-based features they classify words into high-frequency words (HFWs) and content words (CWs). They then define patterns as a sequence of alternately HFWs and CWs and it has to start and end with a HFW. After some cleaning the selecting of patterns is started, which are all given a feature value. Based on how each sentence is structured, a feature value is assigned to each sentence, this can be an exact match (meaning that the components of the pattern are appearing in the right order without any other words in between), a sparse match, incomplete match, or no match.

They also use punctuation-based features, which are normalised such that they have the same weight as the pattern-based features. For the punctuation-based features they look per sentence at the length, number of ?, number of !, number of quotes and number of capitalised words. For the classification they use a KNN-like strategy, they generate feature sets for every sentence in the training and test set and compute the Euclidean distance. For the evaluations the authors constructed two experiments. The first experiment is 5-fold cross validation, which scores best on precision and f-score when all the features are used, while recall is lower than when only using the HFW and CW pattern-based features. The second experiment is a gold standard where 15 different annotators were used to check the retrieved labels for some sentences to be either sarcastic or not sarcastic. For this test the precision and f-score are lower than for the first experiment, however the recall is higher.

2.4. Reviews

In this part of the literature survey, methods specific for reviews are studies through literature. *Since one of the aims of this research is to identify if feature sets specifically constructed for reviews perform better than a more general approach,* it is important to identify what methods are common amongst sentiment analysis methods for reviews. Since a sentiment analysis methods exists of different parts the following parts are studies in more detail for reviews:

- **Features**, even though features are part of the methodology they are identified apart from the full approach in the papers. The reason for this is the focus on feature sets within the research for this thesis.
- **Methodology**, in this section a more general explanation is given of the sentiment analysis method.
- **Evaluation**, this part focuses on the evaluation part of the sentiment analysis method for reviews.

By comparing methods used in the past it is possible to find out which methods are used most frequently and perform best to be able to construct a method specifically for sentiment analysis in reviews. All the terminology used in this section is previously introduced in Section 2.1.4.

Reviews are often given by customers to rate or give their opinion about a product. Most of the reviews in literature are about movies or products. The following items are part of a review:

- A review always consist of a **target or product** of which the rating and review is about.
- Furthermore, usually a **rating** is given by the consumer when writing and review.
- Finally, the most important part of the review is the **text** explaining the experiences of the consumer with the topic of the review.

2.4.1. Features

Since a lot of different feature methods are used a few of them for review are explained in this subsection. An overview of features used for review data can be found in table 2.3 and for preprocessing methods in table 2.2.

Dave et al. [8] propose a method that uses information retrieval methods for feature extraction and scoring. They use two tests to see how well certain feature strategies perform. The first test is

on seven different categories where six are used for training and the last one is a test to see how well the classifier deals with new domains. For the second test an even amount of positive and negative reviews are retrieved randomly from the four largest categories. For these different tests they compare the performance of different features on these test, but they also compare using different features.

The feature extraction starts with input data which which looks like part of a Web page, so the HTML tags are removed from this documents and the text is separated into sentences with a tokenizer. After that, the sentences are split into single-word tokens. To make sure the features aren't too specific some tokens are replaced with a more generic label, these are numerical tokens, all tokens with a product's name, low-frequency words and words that only occur in a certain product category. After this, they use n-grams where trigrams perform best for test one and bigrams perform best for test two. The authors settle for a bit more complicated approach than n-grams where the use a tree of substrings up to cutoff length, the threshold they settled upon is information gain relative to a node's parent. These features should also be given a score, this is why several techniques for this are compared and for both tests intensity times the document frequency gives the best result.

After this, they also compare smoothing methods of which Laplace smoothing gives the best result. After selecting the features and smoothing them they assign the features scores, for this they introduce their own scoring method. However, the scoring method does not look at the strength for a given feature, if a term occurs three times this is seen as the same as when a term occurs 23 times. This is why they look at reweighing measures, multiplying by document frequency and taking the log of this yields the best result. This part is especially interesting, since this paper is the only one which uses a smoothing an re-weighting method in their approach.

The feature extraction method proposed by Drake et al. [11] includes the occurrence of words in reviews. A word-score correlation metric is used for this that shows how much a word influences if a review is positive or negative. This means that if a word is positively correlated with the score of the review it would usually occur in reviews which score above the average and the same follows for the opposite. Words that occur too often or too little are removed since they are not useful for the learning.

The words that are not removed are stored in a vocabulary with words and their score of correlation. The features per review are extracted as follows, the review is converted into a Boolean vector, where the Boolean representing a word is true when the word occurs in the vocabulary.

Mejova and Srinivasan [31] compare different features and feature extraction methods to see which one perform better. Firstly, they compare stemming versus full words, where stemming changes the word to its morphological roots as explained in 2.1.3. Stemming has the benefit of resulting in smaller features sizes, however Mejova and Srinivasan find out that the accuracy goes down when using stemming.

Secondly, the compare binary versus term frequency weighting. With the binary option it is checked if a feature is present or not while term frequency computes the relative importance. However, there is no clear change in performance between the two options. Furthermore, exchanging a label with a negation has proven to be useful.

Also, for different sizes of n-grams they check what happens with the performance. Using a bigram or trigram along decreases the performance, while using unigrams and bigrams performs best for small dataset and unigrams, bigrams and trigrams together gives the best result for larger datasets. When looking at term frequency often the rare occurring words are removed from the vocabulary to reduce the amount of features. For different datasets removing the least occurring features had different influences on the performance.

The performance of the classifier can also be improved by removing less important features, these can be found by calculating their Mutual Information. For all the dataset the performance was lower when using all the features, for the large dataset using only a few percentage of the features worked best while for the small dataset a larger amount of the features where used.

When using POS-tagging it is possible to limit the feature space to only verbs, adjectives and nouns, when using all three of these instead of one this improves the performance.

Finally, lexicon-based features can be used to improve the performance. For this existing libraries can be used in which words are given a polarity already and the words with a strong polarity can be used as features. However, it differs per dictionary and dataset if the performance increases.

Table 2.2: Overview of preprocessing methods in papers about reviews

Preprocessing method	Present in papers
Stemming	[18], [8]
Stopword removal	[18]
Spelling checking	[18]

Table 2.3: Overview of features in papers about reviews

Feature	Present in papers
Tokenization (sentence level)	[18], [8]
Tokenization (word level)	[49], [8]
POS tagging	[33], [18] [1], [28], [31]
Negation detection	[38], [39], [8], [31]
Named Entity Recognition (NER) tagger	[33]
N-grams	[38] [2], [24] [3], [49] [3], [8] [3], [28] [4], [31] [4]
Binary count/presence	[38], [11], [28], [31]
Frequency measurement / weigth	[24], [18], [8], [31]
Lexicon	[33], [39] [5], [18] [5], [8] [5], [31] [6]
Word-document correlation	[11]
PMI	[33], [39]
MINIPAR parser (POS and relationships)	[39], [8]
Statistical substitution	[8], [31]
Information gain	[8]
Laplace smooting	[8]
Phrases	[31]
Mutual Information Based selection	[31]
Pattern extraction (high frequency and content words)	[31]
Punctuation occurrences (?,!," ", capitals, # words)	[31]

2.4.2. Methodology

The methodology of the papers' feature methods explained in the previous subsection 2.4.1 are discussed in this subsection. The reason for explaining the methodology of only three papers, while the full study contains more of them, is explaining all of them will not give any new insights. Since most of the methods combine feature extraction methods, use a machine learning algorithm and them evaluate the result the insights in explaining more methods will be limited, which is why different features and their final results (accuracy, recall, precision etc.) are compared in table 2.3 and 2.4. Also, the machine learning algorithm which will be used in this thesis is set on SVM as explained in 2.1.4.

Dave et al. [8] compare different feature extraction and scoring techniques and different testing situations. The data they use are reviews which contain a binary rating and use reviews from different categories. They tries to determine with these different techniques if the reviews are positive or negative and find that these information retrieval techniques perform as well as traditional machine learning methods. Since this paper compares a lot of different techniques and for different testing situation it is difficult to draw a conclusion which full technique works best. However, they do use a clear baseline for all the tests to compare the performance of the techniques. Also, the performance metrics chosen for the different tests are not always clear as well.

Drake et al. [11] explain a method to summarise the sentiment of a review with a real-valued

[1] NLProcessor linguistic parser
[2] unigrams
[3] unigrams, bigrams
[4] unigrams, bigrams, trigams
[5] WordNet
[6] SentiWordNet

score. They use multiple machine learning algorithms to compare the performance of two-class versus real-valued classification. While doing this, they also compare the performance of regression and classification-based algorithms for different sizes of sentiment categories. To compare these objectives, four learning methods were used in the experiments. From the learning methods used two of them are classification-based, Naive Bayes and SVM, and the other two are regression-based, linear regression and Support Vector Regression. As explained in section 2.4.1 a Boolean vector describing the occurrence of words in a correlation vocabulary is assigned as a feature to each review. For all the learning algorithms these vectors are used as input features. The Naive Bayes algorithm thinks of every possible score as a different class, based on the feature vector it predicts the class that is most likely. Whereas, linear regression tries to learn a function that maps the feature vector to a score. Finally, the SVM algorithms used are the variant of Joachim with the default setting used.

Mejova and Srinivasan [31] explore feature selection methods for sentiment analysis. They use three review datasets with different sizes to compare feature strategies give the best results. They found that training a classifier for a large dataset with a small amount of feature which are ranked high by mutual information (MI) performed better than using all features, however this does not count for a small dataset. They use SVM as a classifier since it is not their goal to determine what classifier would work best for the task and SVM generally outperforms other classifiers.

2.4.3. Evaluation

Since the evaluation methods for reviews and social data do not differ that much, only a few are discussed in detail here. For a general introduction of the terms or metrics used in this section look at subsection 2.4.3. Also, table 2.4 gives an overview of the metrics used and their results for every paper mentioned in this section.

To evaluate the different classification and regression methods Drake et al. [11] use the average squared error to compare the performance of the different algorithms. This metric is used because when the number of classes is more than two it becomes important to also take into account if the predicted sentiment is close to the real sentiment (given by a human annotator). They also introduce a baseline algorithm to see if the proposed algorithms are at least better than some minimum standard. This baseline algorithm takes the average score of the training data as a prediction on the future data. The data used is from GameSpot reviews which have a score 1.0 and 10.0 as a label, this label is given by users of the product and reflects how much they liked or disliked the product. On a 1.0 to 10.0 range the best performing algorithm is the regression-based SVM. Furthermore, on the two-class case the classification algorithms perform best and one the three and four-class cases the results are mixed (after the four-class case onward the SVM regression is the best performer).

Mejova and Srinivasan [31] use overall accuracy and the F-measure for positive and negative classes to illustrate the performance of the classifier. The datasets used are three review datasets which are publicly available, which differ in size and distribution of positive and negative reviews. Also, they use 10-fold cross validation to test the classifier, cross-validation folds are compared amongst papers hereafter.

2.5. Social data

This section tries to do the same as the previous section 2.4 but instead of review data as input, the input used in this section is social data. Since a sentiment analysis methods exists of different parts the following parts are studies in more detail for social data:

- **Features**, even though features are part of the methodology they are identified apart from the full approach in the papers. The reason for this is the focus on feature sets within the research for this thesis.
- **Methodology**, in this section a more general explanation is given of the sentiment analysis method.
- **Evaluation**, this part focuses on the evaluation part of the sentiment analysis method for social data.

Table 2.4: Evaluation comparison of methodologies

Authors	Recall	Precision	MSE	Accuracy	F-score	Number classes	Size data
Drake et al. [11]	-	-	2.59	-	-	3	1000
McDonald et al. [28]	-	-	-	0.53	-	3	3926
Popescu and Etzioni [39]	0.93	0.8	-	-	-	2	13841
Hu and Liu [18]	0.8	0.79	-	-	-	2	~378
Mejova and Srinivasan [31]	-	-	-	0.89	F_p:0.9 F_n:0.82	2	~20.000
Dave et al. [8]	-	-	-	0.87	-	2	5920
Mullen and Collier [33]	-	-	-	0.86	-	2	1380
Pang et al. [38]	-	-	-	0.83	-	2	1400
Wang and Manning [49]	-	-	-	0.82	-	2	~8800
Li et al. [24]	-	-	-	0.81	-	2	2000

In order to understand what the difference is with the previous section and to show what social data is the following characteristics of social data are listed:

- Data retrieved from microblogging platforms, e.g. Twitter, Facebook, Instagram etcetera.
- The because of form these messages have and the limitations to the length some platforms incorporated they are usually short messages.
- The messages often contain with emoticons, hashtags, urls and other social data related characters.
- Use of slang and spelling mistakes.

2.5.1. Features

There are multiple methods proposed in literature already which features to use when extracting sentiment analysis from social data. In table 2.6 features used in social data literature are listed such that the most common used features can be identified. Also, preprocessing methods are compared in table 2.5. Also, in this section a few of the feature method listed in the table are explained in detail.

Barbosa and Feng [4] use two sets of features: meta-information about the words used in the tweet and the structure of how tweets are written. For the meta-information features they use a POS tagger and for each word also its prior subjectivity and polarity are used which are obtained from a lexicon of 8000 words. When a negative word is in front of this word the polarity is switched. Because slang and words that occur often on the Web are not present in the vocabulary the most popular words are care collected and added to this list.

For the second type of features, the syntax features, the frequency of several characters are counted and divided by the number of words in the tweet. These characters are retweet, hashtag, reply, link, punctuation (!,?), emoticons and upper cases. They find that for the subjectivity detection different features in terms of information gain are important than for polarity detection. For the subjectivity detection the tweet syntax features (good emoticons and upper case) have a bigger impact, while or the polarity detection the meta-information (negative polarity, positive polarity and verbs) has more relevance.

Davidov et al. [9] propose a supervised sentiment classification framework for Twitter data. They use four different types of features: single word features, n-gram features, pattern features and punctuation features. For the single word features, every word in a sentence is given a weight which is the inverted count of the word in the Twitter corpus. However, if the word appears in less than 0.5% of the training set it is not counted as a feature. The pattern feature means that all the words are classified into high-frequency words (HFWs) and content words (CWs). A pattern is defined as a sequence of HFWs with CWs and is required to start and end with a HFW. If the pattern components in the sentence appear in the right order there is an exact match. For a sparse match there can be

a word in the sentence in between pattern components that does not match the pattern. For the incomplete match there are even more words that do not match the pattern and there is no match if no or a single pattern component appears in the sentence. The n-gram feature is used for words with a sufficient frequency, the words that are given a weight for the single word features. Finally they use punctuation-based features which count the length of the sentence, the number of "!", the number of "?", number of quotes and number of capitalised /all capital words. Their final conclusion states all of these features contribute to the sentiment classification framework.

The features used by Ortigosa et al. [35] include some preprocessing steps like converting to lower-case, detecting idioms and dividing the Facebook message into sentences. After that the sentences are tokenized based on the whitespaces in the sentence.

Then, they try to detect emoticons, based on a list of emoticons which can be found on Wikipedia. A second tokenization phase is performed based on punctuation marks and interjections are labelled. All the tokens are assigned a score based on their sentiment, for this the classifier checks if the token is a positive or negative emoticon, interjection or a word stored in the sentiment lexicon. If words are not found in one of the dictionaries repeated letters or non-alphabetic characters are removed and the dictionaries are checked again. Also, if the token does not match with a word yet a spelling checker is used since there might be misspellings in Facebook messages. After this, negations are checked to see of any score should be reversed (positive/negative) and POS tagging is applied. Finally, the polarity of the sentence is calculated based on the number of tokens with a sentiment and their POS tag. The final polarity score is the sum of score of the tokens divided by the sum of all the candidates to receive a score.

For the machine learning classifier the features used where slightly different. This process starts with POS tagging, followed by removing words that are not names, adjectives, interjections or verbs. Then, the remaining words are grouped based on their score previously assigned in the lexicon approach, meaning that interjections, emoticons and words are different groups.

Since there are a lot of papers trying to improve sentiment analysis methods for social data it is too much to compare them all. It is already quite difficult to see for these named method which one works best. However, this also depends on the data the method is applied to and what features appear in that data. However, for social data is seems that unigrams don't work that well in contrast to reviews. For social data it seems a better practice to use syntactical features or dictionaries.

Social data specific features

Table 2.5: Overview of preprocessing methods in papers about social data

Preprocessing method	**Present in papers**
Stemming	[7] [1]
Stopword removal	[35], [7]
Removing repetitive letters	[35]
Spelling checking	[35]
Convert all to lower-case	[35]

Social data has special characteristics because of the form restricted by the microblogging platform it is retrieved from. For example, Twitter messages have a maximum size of 140 characters which means users should be selective with the words they use. Also, in the messages on microblogging platforms use of slang, emoticons, spelling mistakes and hashtags or other microblogging related characters is common. However, these microblogging specific characteristics could be exploited as features because they can be really informative.

Punctuation features can be used to detect generic features of sentences [4, 9]. Of interest here are the number of exclamation or question marks which can indicate a strong opinion about an entity. For example, "I love this!!!!!!" has a stronger positive sentiment than "I love this". In addition, certain syntax features can be used such as hashtags, links, replies and retweets [4]. Some of these syntax features thus can be platform dependent. Furthermore, emoticons also occur often in text on social

[1] Porter

media. These can also indicate the polarity of the message since using an emoticon already shows what the emotion of the user is [4, 15, 35]. All of the features mentioned before can be counted in different ways. This can be either binary (is the character present or not), on term frequency (how many times does a character occur) or another function to represent the amount of occurrences of a certain character. However, the emoticons can both be counted or be marked by their polarity.

Interjection detection tries to identify and label interjections, the amount of letters used extra when representing this injection can intensify the meaning. For example, "Haha" indicates something is funny while "hahahahaha" has a stronger intensity. These interjections can then be linked to being positive or negative. Also, spelling checking can be used as a preprocessing method since a lot of spelling mistakes are made in social media texts. Especially when using a dictionary this improves finding these words in the dictionary. However, the spelling checking must be applied carefully, since some corrections can influence the performance in a bad way [35].

Table 2.6: Overview of features in papers about social data

Feature	**Present in papers**
Tokenization (sentence level)	[35]
Tokenization (word level)	[35]
POS tagging	[30] [2], [35] [3], [4] [4]
Negation detection	[35], [4], [7]
N-grams	[9], [32] [5], [35] [6], [4] [7]
Character n-grams	[32]
Binary count	[9]
Lexicon	[30], [35], [4], [7] [8], [15] [9]
Punctuation (!, ?)	[9], [4]
Tweet syntax features (retweet, hashtag, link, reply)	[4]
Pattern based (HFWs and CWs)	[9]
Word2Vec	[32]
Upper-case occurrences	[4]
Emoticon detection	[35], [4], [15]
Interjection detection	[35]

2.5.2. Methodology

To see what approaches are proposed in papers to find sentiment analysis for social data some of these papers are discussed in this subsection.

Barbosa and Feng [4] introduce a method that detects sentiment in tweets (Twitter messages) that uses the characteristics of how tweets are written and the meta-information of the words occurring in the tweet. They propose a sentiment method with two steps, which first makes the distinction between objective and subjective message and in the second step divides the subjective tweets into positive and negative. They use noisy labels for their training data instead of manually annotating the training data. These noisy labels mean that the training data used is collected from three different websites that provide sentiment analysis for tweets. The tweets are collected for three weeks from the sites with the assigned sentiment label. They authors tried different machine learning algorithms available on Weka and picked the SVM for both classifiers as it gave the best results.

A paper which presents a method for sentiment analysis in Facebook is proposed by Ortigosa et al. [35], where they try to detect information about users' sentiment polarity and model their polarity to

[2] Standford Lex-Parser
[3] CESS-ESP
[4] Wordlist
[5] n=1 to n=4
[6] n=1 and n=2
[7] unigrams
[8] SentiWordnet
[9] SentiStrength, SentiWordNet, SenticNet, SASA, Happines Index, PANAS-t

identify when a significant emotional change occurs. Since for this research only the sentiment analysis in Facebook is of use the rest won't be explained in this literature study. They use a hybrid classification method which combines machine-learning techniques with a lexical-based approach to identify if a message is either positive, negative or neutral in Spanish. Both methods have their own advantages and shortcomings which is why the combination was chosen. The first stage is a lexicon-based classifier was built and use to retrieve a large number of labelled messages. It uses a dictionary of words which are annotated with a semantic orientation and also recognises positive or negative interjections, POS tagging, negation, misspellings and emotions. The labelled messages are then used as training set for the machine-learning based classifier, for this method only status messages are used. They compared various machine learning algorithms with different parameters to see which approach gives the best results. The accuracy obtained using lexicon-based techniques for preprocessing and SVM for classifying was the highest.

Gonçalve et al. [15] compare eight different lexicon methods for sentiment analysis and develop a new method which combines these existing approaches to provide an even better result. The combined approach preforms better than all the approaches on their own, this combined approach includes Happiness Index, Senti-WordNet, SASA, PANAS-t, Emoticons, SenticNet, and SentiStrength.

2.5.3. Evaluation

This subsection highlights a few of the evaluation methods of the papers discussed regarding sentiment analysis in social data. The metrics used in evaluation methods are introduced in subsection 2.4.3. In order to compare the performance of the different papers an overview can be found in table 2.7.

To evaluate their approach Barbosa and Feng [4] use a test data set which is half the size of the training set with manually labelled tweets. Furthermore, they use approaches previously reported in literature to compare their approach to. The approaches they use for comparison are from Pang and Lee [37] which is used for classifying reviews on a sentence-level and a unigram-based classifier which is trained with an SVM. For their own approach they compare different settings of their classifier, they clean the training data and they compare a subjectivity version where the training data is cleaned and one with no-cleaning. Also, for the polarity classifier multiple versions are compared to each other of which the best uses majority voting to combine the trained sources. To compare their own approach and the ones earlier presented in literature the error rate is used as a metric. Also, the look at different training sizes to see if the error rate increases. They conclude that their approach has a lower error rate that the other approaches and keeps almost the same error rate when increasing or decreasing the training size.

Because Ortigosa et al. [35] use a lexicon-based classifier to label the training data for the machine learning classifier, the lexicon-based classifier should be evaluated. For this evaluation 1000 messages where randomly selected and given to a human to manually classify them. The manually annotated messages where after that compared to the labels of the lexicon-based method. The dataset which results from the lexicon-based method is used to evaluate the results of the machine learning classifiers. As an evaluation metric the accuracy is used, which is know not to be a good metric as it is often not clear what is meant by it. It does not necessary mean that when a higher accuracy is retrieved the classifier is better and more useful. This makes me question how well the method really performs since they also don't explain what is meant by accuracy and this is the only metric they use.

2.6. Research gaps and questions

This literature study gives an overview of what exists in literature to solve the problem statement. However, since the goal of the thesis is to do research about a problem of a small sub-problem which has not been discovered yet, it is important to see what is missing in literature as well. Below a list can be found with questions that are discovered while looking at the current methods.

- What feature selection methods work best for review data?
- What feature selection methods work best for social data?
- What features extraction methods should be chosen for new data sources?

Table 2.7: Evaluation comparison of methodologies

Authors	Accuracy	F-score	Error rate	Coverage	Number classes	Size data
Ortigosa et al. [35]	0.83	-	-	-	3	3000
Gonçalves et al. [15]	-	0.73	-	0.95	2	11.813
Barbosa and Feng [4]	-	-	0.187	-	2	~150.000
Davidov et al.[9]	0.64	-	-	-	2	1000
Choudhury and Breslin [7]	0.69	-	-	-	2	600
Meena and Prabhakar [30]	0.72	-	-	-	2	10.000
Mohammad and Brovo-Marquez [32]	0.66	-	-	-	2	7097

- When comparing features sets for different data sources do different sets of features based on the characteristics of a data source work better than a one-size-fits-all approach?
- Why are feature reduction and smoothing techniques used so little in literature?
- Do certain machine learning classifiers work better for either social data or reviews?

Because the research of my master thesis can not answer all the questions the research should be specified further. Based on this literature study the following research questions can be presented:

RQ1 Can we construct feature sets with reasonable features for review data for which we can identify differences in performance when using pair-wise comparison?

RQ2 Can we construct feature sets with reasonable features for social data for which we can identify differences in performance when using pair-wise comparison?

Main research questions Can Sentiment Analysis be improved when constructing feature sets specifically for the data type?

3

Project context

This chapter contains confidential information and therefore cannot be displayed online.

4

Data processing

As explained previously, the main objective of this thesis is to compare feature sets for sentiment analysis when using different data types as input. This means the data is extremely important to be able to analyse the influence of the different feature sets on the different data. Therefore, this chapter explains the data on which the sentiment analysis methods are performed.

The first step in this process is collecting the data which is explained in 4.1. Since the structure of the data within Unilever is different than the data defined in literature it was needed to make a clear overview of which data types belong to which groups within Unilever versus within this thesis as can be seen in Section 4.2. After that, the datasets needed to be expanded which is explained in Section **??**. Finally, in Section 4.3 the processing of the data, which is needed before the methods can be implemented.

4.1. Data collection

This part is left out of this online version because of confidentiality.

Table 4.1: Characteristics of manually annotated dataset provided by Unilever

Data group	number of sentences	# positive	# neutral	# negative
Social data	633	248	275	110
Reviews	625	367	134	124

4.2. Grouping of data

This part is left out of this version because of confidentiality.

Unilever uses a certain division of data that works very well from the business perspective. However, for conducting sentiment analysis these categories do not fit that well together. This is why a new division should be made which shows the categories the data falls into.

From this groups it also shows which parts of the data provided by Unilever could be used. Since the manually annotated dataset provided by Unilever consists of the groups according to Unilever for the social and carelines part only a smaller amount of the data can be used. From the social data only *Twitter, Instagram and Facebook* data can be used, the *reviews* data can be fully used.

Since the data is grouped from here on it is important to explain what is meant by the names and which data falls into which category.

- **Reviews data**: all reviews collected by Unilever, extracted mostly from sites like Amazon.
- **Social data**: only social data extracted from microblogging platforms. This means from Twitter, Facebook and Instagram.

4.2.1. Annotation procedure phone data
This part is left out of this version because of confidentiality.

4.3. Data processing
Parts of this section are left out of this version because of confidentiality.

The data processing starts with splitting the data into the data needed within this research and the data belonging to a different group such that it is not useful. To distinguish data that would be useful. For example, in the carelines data only the phone data is used. Because of this, the processing of the data started with connecting parts of the datasets because information needed for splitting the data was missing. This means connecting the manual annotated dataset received from Unilever with their data in the database system in order to find additional information about the data types.

4.3.1. Tools
Different Python packages contributing to the implementation were used. The most important ones are NLTK which is a package which provides several natural language processing functions, this is used for example for tokenization and POS tagging. Furthermore, Scikit-learn is used for the machine learning tasks in Python, the classifier used within this implementation is imported from the Scikit-learn package. The CountVectorizer which is used to translate words into term frequency features. Finally, a package called imbalanced-learn is used for the SMOTE algorithm which deals with imbalanced classes as explained in 5.6.

4.3.2. Preprocessing
Preprocessing is a crucial step to enable correct extraction of features from the sentences. The text needs to be cleaned in order for some of the feature extraction methods to work properly. For example, when using a dictionary to extract features, some words before preprocessing might not be found. The reason for this can be because they contain an upper case or because of a certain form of the word e.g., a verb ”walking” instead of ”walk”.

Due to these reasons multiple methods are used to preprocess the sentences. However, it is important to note that some of the feature extraction methods explained in Chapter 5 are applied before preprocessing the data.

First of all, to allow for comparison, all letters are transformed to lowercase. All the data sources contain upper case letters which makes it hard to compare words when some of them contain upper cases and some not as they might not be seen as the same.

Contractions are also handled, to help detect negations. Contractions are words or syllables in a shortened version. This shortened version is created by removing letters and using an apostrophe instead. Examples are ”is not” to ”isn’t”, ”were not” to ”weren’t”. When writing a formal text contractions are usually avoided. However, when writing informally, which is the case for all of the data, contractions are used a lot. When not handling contractions in the right way this can lead to missing negations which is an important feature in sentiment analysis since it can shift the polarity of a sentence. What makes it especially difficult to deal with contractions is the fact that they can have multiple forms. For example, ”you’ll” can indicate ”you will” or ”you shall” [43]. One way to deal with contractions is to use a dictionary which stores the mappings and can be used to replace the contraction with the full form. This is the most common and simple solution to solve this problem and since the negations are the most important to detect this is a sufficient solution.

After that, unicode normalisation is applied. Some words contain special characters which are explained by a unicode symbol, for example Café has the last character with a acute which in python is represented by a unicode character. These characters should be normalised to allow for comparisons or look ups.

Punctuation is also important to remove, since it says nothing important about the text on sentence level for sentiment. This is why they are simply removed and replaced with a white space. This punctuation include for example dots, exclamation marks, apostrophe etc. However, for the social data this step is when using punctuation as a feature applied after extracting punctuation features. After this, numbers are removed from the sentences as they are not indicating any sentiment.

After these steps, the sentences are tokenized on word level, which means each token represents a word in the sentence. Then, stopword removal is applied, since stopwords often have little or no significance to the text. However, this preprocessing step is not run at the beginning since the NLTK stopword list also contains the words "no" and "not" which are important sentiment shifters. Because of this reason the negator, which is explained in Section 4.3.3, is ran before the stopword remover.

4.3.3. Negation

Even though negation of sentiment classification are hard to handle, it is shown in literature that the identification of negations improves the accuracy [20, 31]. This is one of the reasons to take negations into account before creating the feature vector. Other reasons are the clear influence on the polarity in sentences and the common use of the feature [7, 8, 35, 38].

Negation words often change the sentiment in the opposite direction, for example from positive to negative. Some of the most common sentiment shifting words are no, not, never, none, nobody, nowhere, neither-nor, nothing, and cannot. Additionally, the problem of detection negations is more difficult than simply detecting negative words and shifting the sentiment of the word it influences. A negation word can affect the sentiment in a sentence in different ways, the most common manners a negative word influences a sentence are explained in the following list [26].

- The negation word directly negations a positive or negative sentiment expression. *For example: "A lot of people don't like the smell of this laundry detergent"* This type of negation simple shifts the sentiment of the sentiment word in the sentence. However, simply reversing the sentiment of these sentiment words should not be applied as a general rule. *For example: "This is not the best iced tea I tasted" does not mean "This iced tea tastes bad"*

- When a certain action or function cannot be performed, these sentences often do not contain any sentiment words. Where a sentiment word contains an opinion and thus is a word that is either a positive sentiment word or a negative sentiment word. *For example: "The fridge door does not open."* For these sentences it can be quite hard to determine what the desired outcome of these functions or actions is in relation to the domain. Also, there is no sentiment word in these sentences to shift.

- Finally, a desirable or undesirable state is negated which contains no sentiment word. *For example: "This soap has no smell."* The sentiment of these sentences is difficult to determine, because it is not clear what the desired outcome should be. The desired outcome is dependent on the product in this case which means knowledge about the product is required to determine the sentiment.

A common way of incorporating negation modelling into the feature set for sentiment analysis is to add artificial words. This means if a negation word is prior to a word x instead of considering the word as feature x it is represented as the feature NOT_x. For example, Pang et al. [38] negate every word until the next punctuation mark.

This method works as follows: "I do not like the new Ben and Jerry's ice cream flavour." becomes: I do not NOT_like NOT_the NOT_new NOT_Ben NOT_and NOT_Jerry's NOT_ice NOT_cream NOT_flavour. The advantage of this approach is the feature vector contain a plain and negated feature of each word. However, these negation words and the plain version of the word are treated as two completely different entities. Also, because the NOT_ label is added to all the word following the negation word the feature space is increased with sparse features. The impact of this negation implementation is limited, Pang et al. report a small improvement when using their implementation negations as feature for classification [50].

Another solution is using a polarity lexicon, which contain a list of polarity expressions and their polarity type such as SentiWordNet. This method is usually implemented in combination with a rule-based polarity classifier which calculates the number of positive and negative polarity expressions and decides on the polarity which occurs the most in the text. A polarity lexicon can also be used as features for a supervised classifier.

Rosenberg and Bergler [41] state that negation words are not sufficient to determine the semantics of negation, context and focus are also important. The linguistic notion of focus of the negations plays an important role which is why they propose some heuristic rules.

4.3.4. POS tagging

As explained in Section 2.1.3 with POS tagging the part of speech is assigned to each word. A reason for using a POS tagger is to detect the meaning of a word, for example in the following two sentences the word 'like' has a different meaning which can be captured when the POS of the word is known:

- "I really like this product"
- "I heard about products like smoothies and juices"

Another reason to use a POS tagger is because the grammatical structure in a sentence can have influence on the sentiment in a sentence. It has been shown that adjectives are important indicators of sentiment [26].

There are a lot of different implementations of POS taggers. The reason for this is implementing a POS tagger involves two challenges, firstly finding a possible tag for each words and secondly some words have different tags depending on their context. For example, the word 'set' can be a noun, an adjective, or a verb [5].

Since POS taggers can be implemented using different techniques, it is important to pick a suitable tagger for the data. Some of these techniques are rule-based, stochastic or transformation-based learning approaches [17].

Since POS tagging is a supervised learning problem, when choosing the tagger it is important to know the performance of the tagger and the data it is trained upon, because if the tagger is trained on a completely different domain than the tagger probably will not perform well on the data.

The tagger which is chosen in this thesis is the Perceptron tagger.

Using POS as features within sentiment analysis has different possibilities. First of all, the most common implementation of POS tags is counting the different type of POS tags per sentence. Another possibility is incorporating the POS tags into n-grams.

4.3.5. N-grams

The definition of n-grams is already explained in Section 2.1.3, however this subsection explains the implementation of n-grams in this thesis. This preprocessing method is often combined with a term feature weighting method which counts the occurrence of the term and the counts are used as features. Before the terms can be counted the sentences should be translated into single words for each term (when using unigrams) or pairs or triples of multiple consecutive words (depending on the value of n).

Since n-grams are constructed from words and their place in the text, all the implementations give the same result. When implementing n-grams their presence or frequency are counted per sentence. In this thesis the CountVectorizer, which is explained in 5.5, is used to represent the n-grams in the feature vector.

4.4. Data visualisation

In order to design features for the different data types it is important to understand the sentence and POS characteristics of the data types. This is why in this section the visualisation of the data is presented. Different techniques are used to identify how the data looks. The data visualised in this section contains the test and training data as explained in Sections 4.2 and **??**.

4.4.1. Sentence characteristics

First of all, the average length of the sentences is studied. These values can be found in Figure 4.1. As can be seen the social data contain the most words per sentences followed by phone data and reviews have the least amount of words per sentence. This result is intuitively an interesting result as reviews

or phone data are expected to contain the longest sentences. However, it can be that social messages are longer because they contain hashtags.

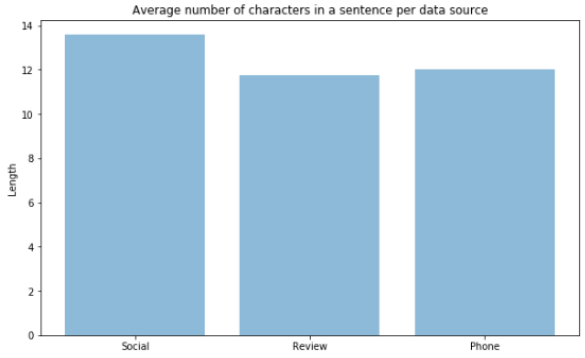

Figure 4.1: Average length of sentences per data source

It is also interesting to see how many letters are contained in a sentence. It might be that the review data has more longer words since this text is more formal than social data. However, when counting the average amount of letters per sentence for the different data types in Figure 4.2 is can be seen that the amount of letters per sentence for social data is the largest. This can be explained by the hashtags which sometimes even are words joined together, for example "#veganfoodporn". When using multiple of these hashtags, the number of letters per sentence increases a lot.

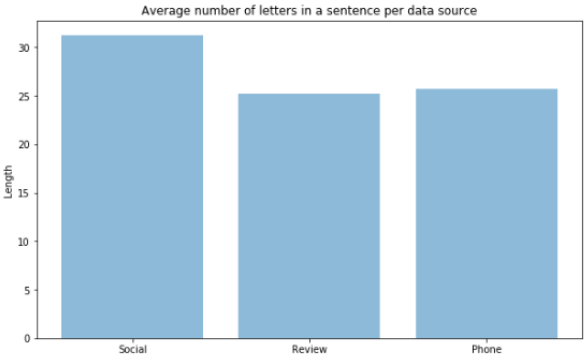

Figure 4.2: Average number of letters in a sentence per data source

Finally, the number of characters is counted to see if the assumption about more special characters occurring per sentence in social data. In Figure 4.3 it can be observed that the amount of characters in social data is a lot higher than for reviews or phone data. This is a significant difference which shows that the number of characters can be used to represent social data as a feature.

4.4.2. Part of Speech

Part of speech (POS) tags say something about the type of words in the sentences as explained in Section 2.1.3. It is interesting to see if POS tags such as nouns are used more in a specific type of data, this is why a POS tagger is used and the occurrences are counted per data type as shown in Figure 4.4. Results that are interesting within this figure are that reviews contain the least nouns compared to social and phone data. Also, reviews seem to contain a lot of adverbs and phone data contains more conjunctions.

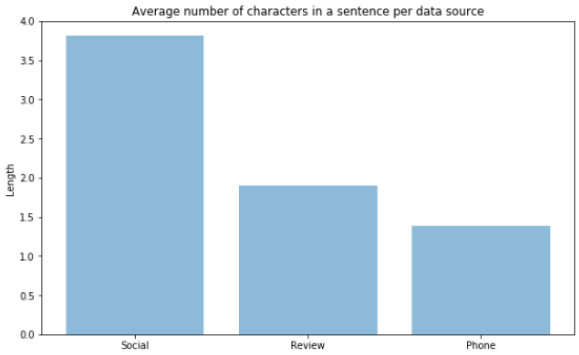

Figure 4.3: Average number of characters in a sentences per data source

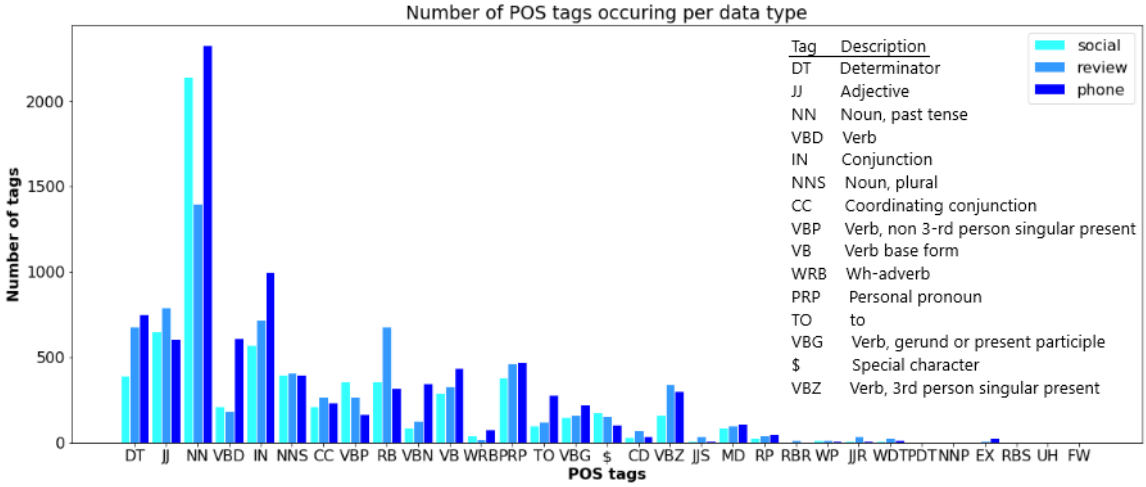

Figure 4.4: Number of POS tags per data type

5

Methodology

This chapter gives an overview of the methodology used to answer the research questions. These research questions are repeated in the following list:

- Can we construct feature sets with reasonable features for review data for which we can identify differences in performance when using pair-wise comparison?
- Can we construct feature sets with reasonable features for social data for which we can identify differences in performance when using pair-wise comparison?
- Can sentiment analysis be improved when constructing feature sets specifically for the data type?

The overall structure of the implementation is explained in detail. The first part of this chapter gives a general overview of the structure of the methods chosen, while the second part of the chapter goes into detail explaining the design choices made during the implementation of the methods.

First of all, in Section 5.1 a general overview of the sentiment analysis pipeline is given which is shown in figure 5.1. The details of this pipeline are explained in the other sections. In Section 5.2 the review feature sets are explained and in Section 5.3 the social feature sets are introduced and explained in detail. Section 5.4 explains the feature extraction methods which are used in the feature sets and the details of their implementation. For some of the features, a feature weighing method needs to be applied, which is explained in 5.5. After that, in Section 5.6 is explained how the phone data is incorporated into the methodology. After that, in Section 5.7 the classification method is described.

5.1. Sentiment analysis pipeline

In order to understand the implementation details of the sentiment analysis pipeline, first a general overview should be given. Sentiment analysis generally consist of a few general steps which are shown in Figure 5.1. The process starts with divided the data into a training and test dataset, on which each feature set is applied which results in a feature vector. The training set is used to train the model upon and the test set is used to see how the trained model performs. The feature set is where the focus is within this research, which is why multiple feature sets will be implemented and compared to gain insights about different feature extraction methods in combination with different data types. In the training phase, the feature set in combination with the labels of each sentence are given as input to the SVM classifier, which creates a classification model as output. Within the test phase, the feature vector can be given as input to the classification model which assigns a label to each sentence. As a result of classification a set with predicted labels is generated and a result of manual annotation a gold standard labelled set exists which can be compared to see how well the classification model performed.

In order to answer the research questions several feature sets are designed for review data and also several feature sets are designed for social data. There are two reasons why we use multiple feature sets and why they are designed specific for the data type, which are the following:

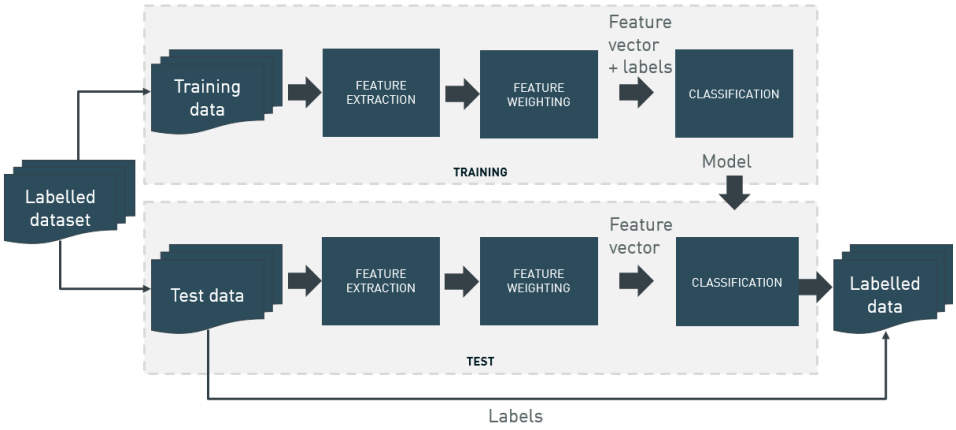

Figure 5.1: Sentiment analysis pipeline

- There is no consensus in the literature about features that should be chosen. Often the reason for choosing features are not explained in papers.
- Not clear if you should use different features for different data types and which features work well for which data type. To our knowledge there are no studies yet which try to identify the differences between features for different data types.

Because of these reasons, **three features sets for reviews are chosen and three feature sets for social data**. In order to study on which data the feature sets perform best, all three data sources are tested on the feature sets. Since the phone data is a summary of a phone call the structure of the text will be most similar to review text. This is why there are no phone summary feature sets and the assumption is made that the feature sets for review data will perform better on the phone data than the social data feature sets.

There are three different reasons for choosing the three feature sets for social data and reviews. An overview of these features sets within the sentiment analysis pipeline is given in Figure 5.2.

- The first feature set is chosen because these features are used the most in the literature studied. The most used features are identified for review data in Table 2.3 and for social data in Table 2.6.
- Secondly, the sentiment analysis methods for the specific data sources are compared based on their performance which is compared in Tables 2.4 and 2.7. The best performing methods for both for review and social data are chosen based on metrics other than accuracy since it does not take class imbalance into account. Since these methods have been found to work well on this specific data it is interesting to see if the performance is also high for the data used within this thesis. For review data we follow the feature sets proposed in Mejova et al. [31], and for social data the features in Barbosa et al. [4].
- Finally, two additional feature sets, one for each of the data types, are proposed in this thesis which are self-constructed. The reason to add additional features is because the structure of the text of social en review data is very different, however the first two types of feature sets do not always deal with the specific characteristics of these data types. For these feature sets, based on the data type, features which are expected to improve the performance for a specific data type are selected.

5.2. Review data feature sets

In this section the feature sets of review data are explained. Some details of the preprocessing method are given in Section 4.3 and of the feature extraction methods are given in Section 5.4. The reasons for choosing the different feature sets were discussed in Section 5.1.

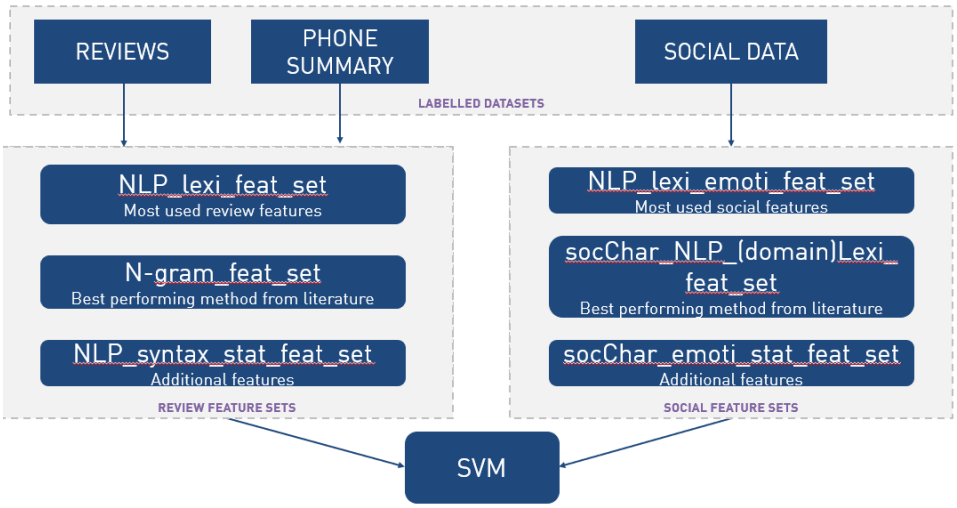

Figure 5.2: Overview feature sets

5.2.1. Review feature set 1: Most used features

This feature set is constructed by selecting the most used features in the literature studied in this thesis. It is chosen with the assumption that features which are used often in papers will perform well. It is seen by the results in the papers that the authors of previous literature have a reason for choosing certain features. By comparing the papers studied in Chapter 2 and their features, the most used features can be identified. The features used in literature can be found in Table 2.3. If the feature occurs three times or more in the table it is selected as a frequently used feature.

The review most used feature set is shown in Figure 5.3 and starts with tokenization. After that, the tokenized words are given to a lexicon, the lexicon which is used is WordNet [1], this lexicon is also chosen because it is used most in the literature studied. Details of implementing the WordNet lexicon can be found in Subsection 5.4.1. Another feature is created by using a negator on the tokenized words. The implementation details of the negator can be found in Section 4.3.3. After the negations are identified, both the POS tags and uni- and bigrams are extracted. Both the lexicon, POS tags and uni- and bigrams are added to the feature vector based on their frequency in each sentence. This term frequency is explained in Section 5.5.1. This feature set will be referred to as $NLP\_lexi\_feat\_set$ when mentioned after this section, this name explains that NLP features (negator, POS tagging, 1,2-grams) and a lexicon are used.

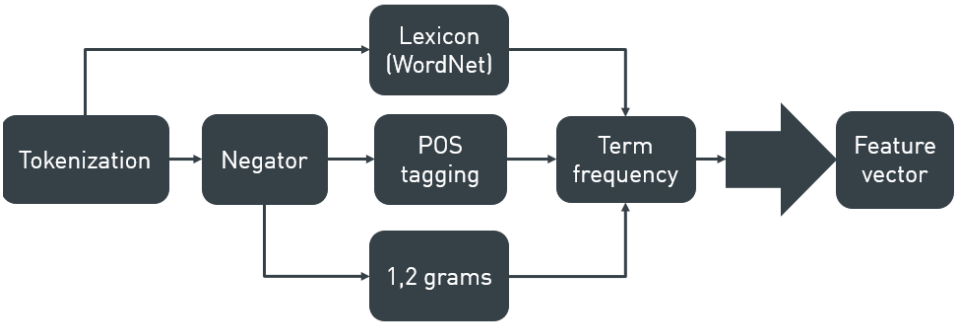

Figure 5.3: Review feature set 1: $NLP\_lexi\_feat\_set$ (Most used review features in literature)

[1] https://wordnet.princeton.edu/

5.2.2. Review feature set 2: Best performing method from literature

The second feature set is selected based on the performance of the methods for review data studied in literature. This means that the features from the best performing method in literature are replicated. Only the methods that report different metrics than solely accuracy are considered, since accuracy on its own can be misleading and does not take into account how many items are misclassified. For choosing this set, selection table 2.4.3 is used and it resulted in picking the method of Mejova et al. [31] to replicate. The reason for replicating the feature set of an existing method is knowing how it performs on different data. It is interesting to see how well an existing method performs on different data and if this method works well for this data as well.

Mejova et al. [31] compare different feature definition and selection strategies. First, they compare basic units extracted from text, such as: stemming, n-grams and phrases. Secondly, they compare feature extraction methods such as term frequency and binary weights. They also compare POS and different lexicons. The datasets they use are three different ones with various sizes, such that they know how the dataset affects the performance. The reviews have two classes in which they can occur, positive or negative. For the classification they use SVM, which is the same as used in this research. They features used within the feature set are the best performing of the paper which is n-grams (with n=1,2,3) and term frequency, explained in Section 5.5.1, as is shown in Figure 5.4. This feature set will be referred to as $n-gram\_feat\_set$ when mentioned after this section, this name explains only n-gram features are used in this feature set.

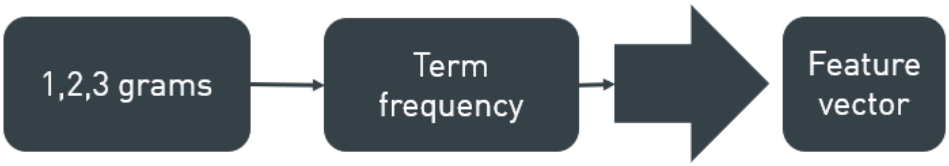

Figure 5.4: Review feature set 2: $n-gram\_feat\_set$ (features from best performing method in literature, which for review data is Mejova et al. [31])

5.2.3. Review feature set 3: Additional feature set

This feature set looks at additional features which are not used in the previous feature sets, but can be expected to improve the performance of sentiment analysis for review text. Intuitively, it is easy to say that review text looks different than text from microblogging platforms. Review text is more formal and contains more full sentences, whereas microblogging text is shorter, contains more special characters and spelling mistakes or slang. However, when looking into this deeper, it is harder to say what the exact differences are. This is why linguistic features are used, which look at the natural language characteristics of the text which are for example the syntax. On the contrary, non linguistic techniques look at the text as a series of words, characters, phrases, sentences etc. These non linguistic techniques look at the number of occurrences of these words or terms for example. However, these non-linguistic features might not make a big difference when comparing the differences between review and social data especially in the domain of customer feedback. This is why a linguistic feature set is designed especially for the review data.

Luigi and Matteo [10] look at syntactic features for online reviews and identify the use of syntactic patterns improves sentiment analysis. Gamon [13] looked into sentiment analysis for customer feedback data and shows that the addition of deep linguistic analysis features consistently improves accuracy to this domain. Gamon uses different linguistic features from which a few are selected, these are POS, semantic relations, and length measures.

An overview of the feature set is given in Figure 5.5. The process starts with tokenization, after which some length measures, as explained in Subsection 5.4.6, are calculated and represented in the feature vector. Furthermore, named entities are extracted and negations are detected. The detection of named entities is explained in Subsection 5.4.3 and of negations is explained in Subsection 4.3.3. After this both POS tags and syntactical relations (introduced in Section 5.4.4) are identified and by

their term frequency represented in the feature vector. Finally, the words are filtered based on their POS tags and only verbs, nouns, adjectives, adverbs and negations are taken into account in the PMI score. For each of these words the positive and negative PMI score is shown in the feature vector. How the PMI score is calculated can be found in Section 5.4.5. This feature set will be referred to as $NLP\_syntax\_stat\_feat\_set$ when mentioned after this section, the name NLP refers to the negation and NER features, the name syntax refers to the syntactic relations and POS tags and the statistical (stat) feature is point-wise mutual information (PMI).

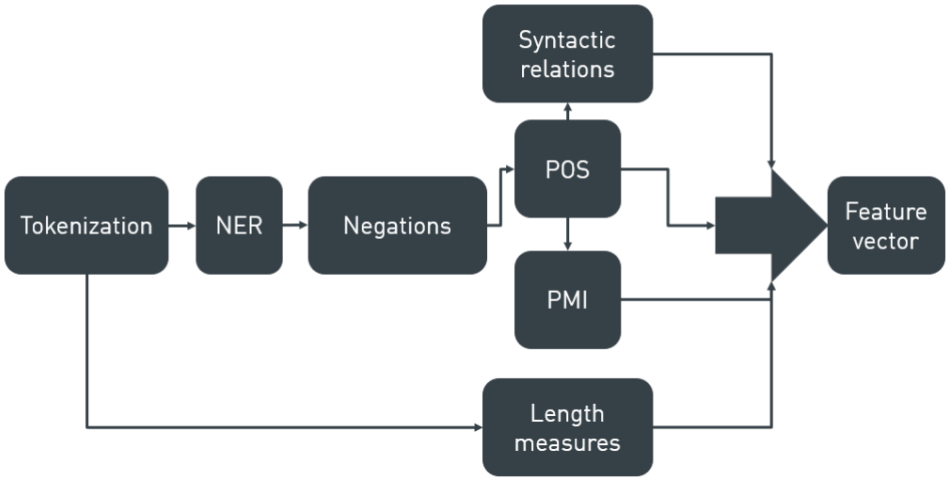

Figure 5.5: Review feature set 3: $NLP\_syntax\_stat\_feat\_set$ (Additional feature set)

Because this feature set is self-constructed within this thesis, it is important to give a clear explanation why the features are chosen, for each of the features this is explained in the list below.

- *Syntactic relations*: Each linguistic element has an intrinsic value of sentiment that is propagated through the structure of the sentence [10]. By using a dependency parser the influence each word has on the sentence with the grammatical relation is explained. Also, this feature is proposed by multiple papers for review text as a linguistic feature and was found to improve sentiment analysis [10, 13]

- *NER*: This feature is included on the one hand to generalise the text as kind of a preprocessing feature since all products and organisations are seen as the same. A reason to include NER is because sentences with a NER label are more likely to contain an explicit or implicit sentiment [47].

- *Pointwise mutual information (PMI)*: The reason for using PMI is that using only the other features represents only the characteristics of the sentence, but does not show the context or semantic orientation of the words in the feature vector. Luigi and Matteo [10] show the importance of both syntactic features and context-based features which shown the sentiment of the text in the review in combination with the context. However, the words and their meaning are an important part of representing the polarity of the text. With PMI the probability of a word occurring in a positive or negative sentence can statistically be represented. The corpus used for this is the training set in which the relation between words and the labels of the sentences in which they occur is used when computing this score. The PMI score shows the probability of a word in a sentence being positive or negative based on how many times it occurs in positive or negative sentences in the corpus. It takes into account both the context of the words and the semantic orientation.

- *Length measures*: The reason to also include length measures it to have a good combination of different features. Gamon [13] identified the importance of linguistic features and proposes length measures as the most interesting feature to add.

5.3. Social data feature sets

This section explains the feature sets chosen especially for social data. Just as the review feature sets, the social sets are also designed by using the most used feature methods, the best performing method in literature and a method designed especially for the additional feature set of social data.

5.3.1. Social feature set 1: Most used features

For the most used features of the social data the feature of the methods in literature using social data are compared. The features used in this literature are listed in Table 2.6. The features occurring three times or more are used in the most used features set.

The structure of this feature set is shown in Figure 5.6. It starts with tokenization and the tokenized sentences are then given to a lexicon as explained in Subsection 5.4.1, which is SentiWordNet. SentiWordNet is one of the most used polarity lexicons in literature for social data and assigns to each synset (which is a group of words which are synonyms) of WordNet three sentiment scores [7, 15]. For the other features, first emoticon detection, which is explained in 5.4.2, is applied and after that both negations are detected and n-gram (with n=1,2) are constructed. For both of the emoticons and the n-grams the term frequency of the terms per sentence are counted. Finally, the sentiment scores of the words and the term frequencies of emoticons, uni- and bigrams together forms the feature vector. This feature set will be referred to as $NLP\_lexi\_emoti\_feat\_set$ when mentioned after this section, the name NLP refers to the negation and 1,2-grams features, the name lexi refers to the lexicon and emoti to the emoticon detection feature.

The biggest differences with the most used features set for reviews are that instead of POS tags, emoticon detection is used and instead of WordNet, SentiWordNet is used. The reason that this is the same is probably the fact that these frequent used features in literature are because of this reason also used by a lot researchers. However, this might not always be the best reason to pick a method. This is why it is interesting to see if these most used features methods perform better than the other features sets since this is closer to a general approach.

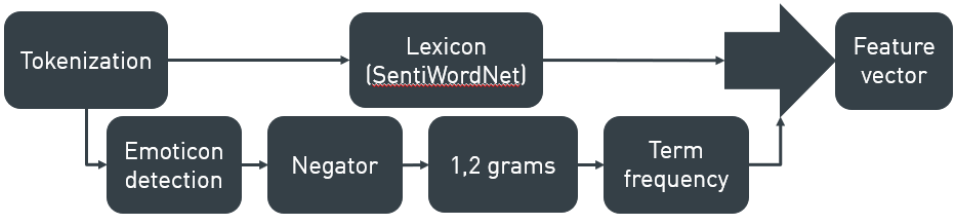

Figure 5.6: Social feature set 1: $NLP\_lexi\_emoti\_feat\_set$ (Most used features)

5.3.2. Social feature set 2: Best performing method from literature

The second feature set for social data is based on the performance of the methods compared in the literature study, Section 2.5. These existing methods are compared based on their performance in Table 2.5.3. In this comparison, methods that only report accuracy are not considered since this is not a reliable metric as it does not show all the important information about the classification performance. Based on this comparison the method of Barbosa and Feng [4] is chosen to implement as a feature set.

Barbosa and Feng propose a two-step sentiment analysis method, which first classifies the data as subjective and objective and then further divides the subjective data into positive and negative. They do not use manually annotated data but use sentiment detection websites which generate noisy labels. For the subjective/objective and the positive/negative classifications almost the same feature extraction methods are used, which is why the feature extraction methods used for the positive/negative classification are used. The differences between the original method of Barbosa and Feng and the implementation used in this research, our method does not use a two-step classification, should be taken into account when comparing the performance of the implementation in this research and the performance of the method of Barbosa and Feng.

Although Barbasa and Feng propose a two-step classification method, the features used for both classification steps are almost the same. The feature used in the positive versus negative classification step will be used in this feature set. There are two different feature types which they use: the tweet syntax features (frequency of retweet, hashtag, punctuation etc.) and meta-features (POS tags). The tweet syntax features count the amount of certain characters and divide it by the amount of words per sentence. These number are then represented for each character in the feature vector. The meta-features start with a negator, of which the details are explained in Subsection 4.3.3. After, that both the POS tags are counted (see Subsection 4.3.4) and the words are given a score by using a polarity lexicon (SentiWordNet), as explained in Subsection 5.4.1. Finally, some popular words occurring in Unilever data are added to this lexicon as explained in the following paragraph. The details of representing a lexicon in the feature vector is explained in Section 5.4.1. This feature set will be referred to as $socChar\_NLP\_(domain)Lexi\_feat\_set$ when mentioned after this section, socChar refers to the social characters of which the occurrence is counted, the name NLP refers to the negation and POS tag features, the name (domain)Lexi refers to the lexicon and domain words which are added to this lexicon.

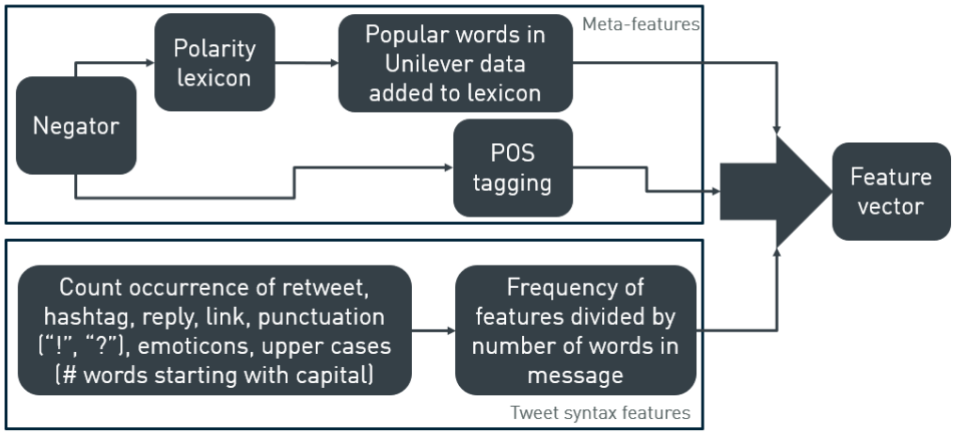

Figure 5.7: Social feature set 2 : $socChar\_NLP\_(domain)Lexi\_feat\_set$ (Best performing method from literature)

5.3.3. Social feature set 3: Additional feature set

This feature set tries to include features which are not used before in the social feature sets but for a given reason are interesting to look at to possible improve the performance of sentiment analysis for social data. In order to identify which features are interesting or useful to add for microblogging data Taboada [46] shows that in the research about sentiment analysis about linguistics of online text, such as tweets and Facebook posts, most researchers clean the data, correct spelling mistakes, remove hashtags and URLs which makes the text look more like formal written text. However, the very nature of microblogging data is the presence of capitalisation and punctuation, which is why both punctuation and the percentage of capitalisation per sentence is taken as a feature. Both of these characteristics can indicate a strong opinion in a text. For this reason the special characters that occur a lot in social data are taken as features especially for this data source.

An overview of these additional features set is given in Figure 5.8. The features which are used in this feature set are an emoticon dictionary, which is explained in Subsection 5.4.2. Furthermore, the sum of all polarity scores for all words, which are calculated using PMI, which is explained in Subsection 5.4.5. Finally, the occurrence of retweets, hashtags reply, punctuation, negation words and upper cases are counted for each sentence. This feature set will be referred to as $socChar\_emoti\_stat\_feat\_set$ when mentioned after this section, socChar refers to the social characters of which the occurrence is counted, the name emotic refers to the emoticon detection feature and the statistical (stat) feature is point-wise mutual information (PMI).

Another characteristic which could be used to improve the performance of sentiment analysis for social data is the use of social relations [19]. However, the dataset used in this research does not include these social relations thus within this research it is not possible.

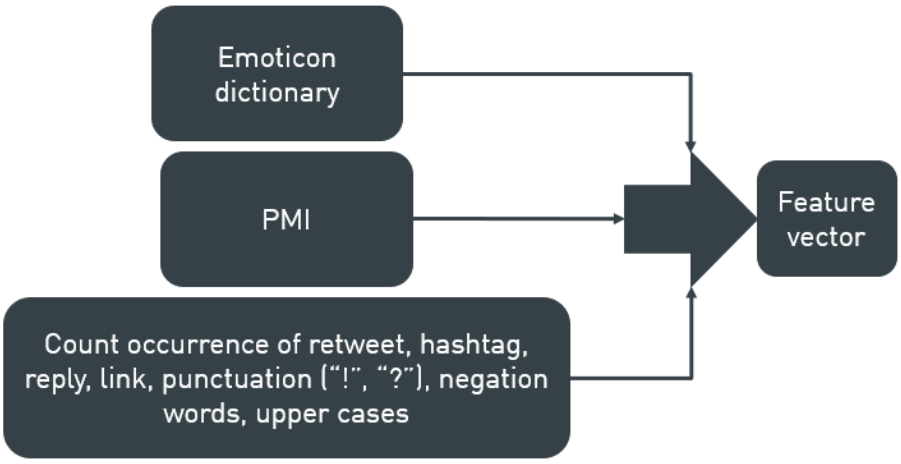

Figure 5.8: Social feature set 3: $socChar\_emoti\_stat\_feat\_set$ (Additional feature set)

Since the assumption that social features which describe the features and characteristics of social data would benefit sentiment analysis for social data is stated by previous literature, it is interesting to investigate if sentiment analysis for social data can be improved when using only social features. Also, Agarwal et al. [1] identify that using 100 abstract linguistic features performs as well as a unigram baseline (with 10000 features) for social data. Most of the linguistic features they use include the occurrence of certain characters. For example, number of exclamation marks, number of question marks, number of negation words, number of extremely-positive, extremely-negative, positive, and negative emoticons. The social features which are mentioned to improve performance in previous research are the ones chosen to represent the additional social feature set. Thus these features are both chosen for the reason that they are specific to social data and according to the research of Agarwal et al. [1] are better than only using unigram features.

Given that this feature set is selected based on the reasoning of Agarwal et al. [1] it is important to give a clear description why the features are chosen, for each of the features this is explained in the list below.

- *Emoticon dictionary*: this feature is chosen because it is more specific to microblogging data. Furthermore, emoticons can improve sentiment analysis because they often reflect the emotion used by the writer of the text, they are strong indicators of sentiment.
- *Pointwise mutual information*: Since individual words can all say something about the sentiment of a full sentence it is important to take all of them into account. By using the scores of all these words in a sentence a general idea is given of the polarity of the text. How the scores for each of these words is computed is explained in Section 5.4.5.
- *Count of microblogging characters*: this feature is chosen because it is specific to microblogging data as it is used in previous work. Also, a lot of methods in literature for social data use a count (or binary presence) of microblogging features [1, 4, 22]. Even though this feature is already included in the best performing method in literature in Section 5.3.2, this feature extraction method does characterise the elements of social data extremely well and for that reason it is included again.

5.4. Feature extraction

In the process of extracting features from the sentences the text is first preprocessed, which is explained in Section 4.3, and afterwards some methods can be applied to transform the text into features.

These methods are called feature extraction methods.

In order to classify the data, features are used in this classification which, together with the labels (positive, negative, neutral), will be given as input to the machine learning algorithm. With this input the classifier will then construct a model, which for new input data can predict the labels. For the input to the classifier the text is translated into numerical values which are put into a feature vector and given to the classifier. Thus, the rows of this feature vector are the different sentences and the columns represent the features of the sentence. It is important to state that some features which are extracted from the sentences are still in text form, for these features, feature weighting is applied, which is explained in Subsection 5.5.1, to create numerical values.

The key for sentiment analysis is the engineering of effective features, like for most supervised learning applications [26]. When choosing features there are a few things to take into consideration. First of all, the size of the dataset affects the performance for some of the techniques. It has been proven that using a small set of features works better for smaller datasets, while using a lot of features is too selective for smaller datasets. Furthermore, the domain of the text in the dataset is also of significant influence. There are certain features that work better either for a specific domain or in a non-domain dependent dataset [31]. Finally, the classification technique can also perform better or worse in combination with certain features.

5.4.1. Lexicon

Even though lexicons are used often within sentiment analysis as rules, they can also be used for feature selection [31]. Lexicon-based feature methods usually add a set of words which they contain to the features space [29]. The words that are added to the feature space are then checked to be present in the sentences. The biggest problem which should be taken into account when using a lexicon is words and phrases can have different orientations depending on the domain they occur in.

There are two types of lexicons which are used within this research to extract features: namely a) the dictionary and b) a polarity lexicon. The dictionary contains a list of seed words, for each of the words the occurrence in a sentence is checked. This means this creates a large feature set which is extremely sparse. This can also lead to the feature set growing to an unmanageable size [8]. The dictionary used in this research is WordNet [2], which contains synsets of words and their semantic relations.

The polarity lexicon used within this research is SentiWordNet [3], which is based on WordNet and associates scores to part of the synsets from WordNet. The scores which are used are positive, negative, and neutral which matches the sentiment scores within this research perfectly. SentiWordNet will assign based on the POS label of the word, a score between [0, 1] to the word for positive, negative and neutral [15]. The SentiWordNet lexicon is used for two social feature sets, which can be found in Section 5.3.1 and Section 5.3.2.

There are a few reasons for choosing these two lexicons. First of all, because they are used the most in other research studied [7, 8, 15, 18, 39]. Secondly, they are open to use and easy to incorporate into the project. However, a domain dependent lexicon would probably perform better. However, to my knowledge it is difficult to find such a lexicon for the product domain which can be used for free and within a company.

Domain dictionary

In addition to a polarity lexicon, Barbosa and Feng [4] use a specific Web vocabulary with slang words. For example, words as ”yummy” or ”ftw” are added to this lexicon which is created from popular words used in online discussions. However, this extra popular online words’ lexicon from Barbosa and Feng is not available which is why a similar lexicon needs to be created.

In order to create a similar lexicon specific to the data of this feature set, which is social data, the most popular words of the Unilever social data, which are not linked to the SentiWordNet lexicon, are identified. The reason for using popular words specific in the Unilever data is because this data is

[2] https://wordnet.princeton.edu/ retrieved on 19-6-2018
[3] http://sentiwordnet.isti.cnr.it/ retrieved on 4-7-2018

related to a domain which might benefit from using a list of words which are common within the social media text. It would be interesting to see if this added word lexicon has a beneficial influence on the performance.

In order to see if this lexicon has effect, a simple version with fifty words is created which are manually given a sentiment label between -1 and 1. If this lexicon does have effect it can be a design choice for Unilever to create such a lexicon and maintain this for sentiment analysis and possible update this every 2 years since part of the words are slang and thus sensitive to change.

Table 5.1: Ten most occurring words of Unilever dictionary

Word	Sentiment
like	1.0
without	-0.5
workout	0
na	-1.0
love	1.0
oh	0
like4like	0
plus	0.5
palms	0
skin	0

5.4.2. Emoticon detection

Emoticon detection is used often as a feature since emoticons can give a summary of the sentiment implied by the user. There are different options to implement emoticon detection. One of the implementations is simple counting the number of emoticons occurring in the text which is proposed by Barbosa and Feng [4] and thus also implemented in social best performing method from literature as explained in Section 5.3.2.

However, this is not the only implementation option. Other possibilities are using a emoticon lexicon, which is used in social additional feature set 5.3.3, or change the emoticon to a label which can then be counted as a term when using n-grams. An emoticon lexicon is the most used method in the literature studied in which they use a lexicon with a polarity assigned to the emoticons [15, 35]. Since the emoticon lexicon is the most popular option, this approach is also used within social most used feature set 5.3.1. The different emoticons are added to a dictionary in combination with a score. The emoticon labels are identified using Wikipedia [4] which provides a list with all the emoticon used and their meaning.

5.4.3. Named Entity Recognition

Named entity extraction in sentiment analysis can be defined as follows:

1. Within a corpus C, find all mentions M or entity expression.
2. All the entity expressions M should be clustered into groups, where each group represents a real-world entity [26].

This generally is a very complex problem, for one reason because it depends on the context which groups of entities are of interest. For example, when Unilever is interested in sentences about ice-creams the retrieved text about ice-cream might be a combination of ice-creams of Unilever and the competitors.

There are a lot of different implementations of NER which can be selected to incorporate this into the implementation. There exists some good entity extraction APIs which can be used for entity extraction.

[4] https://en.wikipedia.org/wiki/List_of_emoticons retrieved on 2-8-2018

However, since part of the data is confidential it was chosen not to use an API for this since the data then would need to uploaded. Instead spaCy [5] is used, which provides a trained model which can identify the entities in text. Examples of the labels used in the spaCy model are person, location, product, date etc. The NER is used in the review additional feature set in Section 5.2.3 and the purpose of this feature is to generalise the text.

5.4.4. Syntactic relations

Relations between sentiment expressions and their target can be identified by a syntactic parser or dependency parser [26]. To find the dependencies between words a dependency graph is created for each sentence. The dependency structure tries to find the syntactic structure that consists of relations between lexical items, these relations are called dependencies. Syntactic parsing can also be quite complex, since a sentence can be ambiguous and therefore have multiple parse trees.

Dependencies are created by drawing arrows between lexical items. This process starts at the HEAD or ROOT. When parsing a sentence for each word (including ROOT) is chosen which other word it is dependent of. From the dependencies a tree can be constructed, an example is given in Figure 5.9.

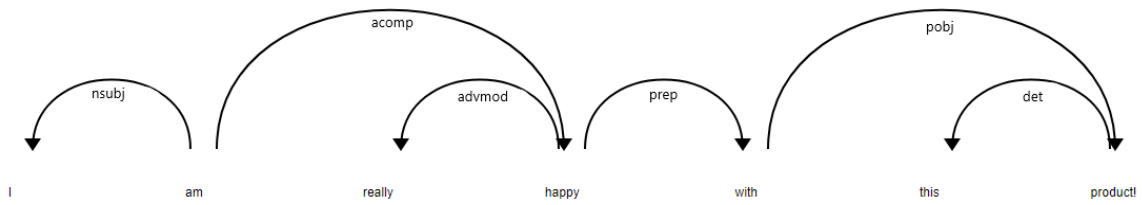

Figure 5.9: Example dependency parsing

In order to use this type of feature, again the Python package spaCy is used which provides a trained model which can be used to predict the dependencies. This means every word is assigned a label which explains the dependency within the sentence. These labels are represented as term frequencies within the feature vector. This feature method is used in review feature set 3, which can be found in Subsection 5.2.3.

5.4.5. Pointwise mutual information (PMI)

PMI measures the statistical dependency between two terms occurring in a document [26]. This can be used to represent syntactic patterns of the text by connecting the terms in the sentences with it appearing mostly in positive or negative sentences. For each word the PMI is calculated with respect to the labels in the training data, thus the training data is used as the corpus. For instance, the word ”excellent” will occur more often in positive sentences while the word ”poor” will occur mostly in negative sentences [33]. This means that the words based on their natural language are determined to be more positive or more negative. Also, when comparing PMI with other association measures, such as Jaccard, Dice, and Normalised Google Distance, it performs better [12].

When representing this score for every selected word in the sentences (only verbs, nouns, adjectives, adverbs and negations) the following formulas are used, to find the PMI scores for each word correlated with positive sentences and negative sentences:

$$PMI(\text{word, positive sentences}) = \log\left(\frac{\left(\frac{\text{number of sentences labelled positive and contain the word}}{\text{total number of sentences in dataset}}\right)}{\left(\frac{\text{number of positive sentences}}{\text{total number of sentences in dataset}}\right)\left(\frac{\text{number of sentences containing the word}}{\text{total number of sentences in dataset}}\right)}\right) \tag{5.1}$$

[5] https://spacy.io/usage/linguistic-features retrieved on 15-7-2018

$$PMI(\text{word, negative sentences}) = \log\left(\frac{\left(\frac{\text{number of sentences labelled negative and contain the word}}{\text{total number of sentences in dataset}}\right)}{\left(\frac{\text{number of negative sentences}}{\text{total number of sentences in dataset}}\right)\left(\frac{\text{number of sentences containing the word}}{\text{total number of sentences in dataset}}\right)}\right) \tag{5.2}$$

The range of PMI can differ from $-\infty \leq PMI(x,y) \leq min[-\log p(x), -\log p(y)]$ which is why it is necessary to normalise the PMI ranks. For this the normalised pointwise mutual information (NPMI) which has a range between $[-1,1]$, where -1 means it's never occurring together, 0 means independence and 1 means complete co-occurrence.

$$NPMI(word,y) = \frac{PMI(word,y)}{\log(p(word,y))} \tag{5.3}$$

This feature extraction method is used in both the review additional feature set, explained in Section 5.2.3 and the social additional feature set, explained in Section 5.3.3.

5.4.6. Length Measures

Within the set of lexical features Gamon [13] also uses length measures. Since review text has more structure in sentences it makes sense to take these measures into account. The lengths that are added to the feature vector are length of the sentence, clauses, adverbial/adjectival phrases, and noun phrases. It makes sense that these measures appear more in review text since the sentences of this text probably is more structured. This feature extraction method is used in the review additional feature set, explained in Section 5.2.3.

5.5. Feature weighting

Sometimes when features are extracted from text they still need to be translated to a numerical value. For example, when extracting n-grams the words are extracted in a certain manner from the original text. However, the n-grams still represent (combinations of) words instead of numbers. For this reason feature weighting methods are used. In this section only term frequency is explained since this is the main method used for frequency weighting in this research. This method is chosen because of the research of Mejova et al. [31] in which they compare binary occurrence and term frequency and identify that term frequency works a little better. Furthermore, the focus of this research is on the feature extraction methods and not on the feature weighting methods.

5.5.1. Term frequency

Term frequency is generally used to give weights to features representing documents [31]. There are different formulas to assign a score to different terms, however the most simple version is for each occurrence of a list to check how many times it appears in each sentence. Term frequency is in this research generally used to translate text into numerical values which appear in the feature vector. Unless otherwise specified, this method is used.

5.6. Imbalanced data

When looking at the amount of positive, negative and neutral labels in Table **??**. We see that the amount of labels in each class are not equal. Especially for the phone and review data types the classes are highly imbalanced. For the phone data only 22 sentences are labelled positive, while 437 sentences are labelled neutral. Also, for the review data the amount of positive labels is higher than the neutral and negative labels combined.

Imbalanced data means when one or two of the different classes (positive, neutral, negative) contain more samples than the others. In this case the neutral and negative class contain more samples than the positive class. Imbalanced data classification problems result often in minority class samples to be misclassified. The reason for this is the design of most machine learning algorithms, which often tries to optimise the accuracy of the overall classification [27].

There are multiple solutions which can be considered to solve the problem of imbalanced data of which the most commonly used ones are resampling, instance weighting and thresholding. Resampling under-samples or over-samples the data to rebalance the data. When using instance weighting different error-classification costs are assigned to different classes. Finally, thresholding uses a decision threshold of the classifier to balance the precision and recall [45].

5.6.1. SMOTE

Since Synthetic Minority Over-sampling Technique (SMOTE) is one of the most popular methods for oversampling and also because it performs better in previous literature for text classification problems when using SVM this technique is chosen to deal with the imbalanced data in thus research [45]. SMOTE uses over-sampling of the minority class by creating synthetic minority class examples [6]. Another reason to choose SMOTE is because it uses oversampling to rebalance the data classes. Because the amount of data used within this project is not big it is important not to under-sample the data since that means removing instances from the majority class. Also, the majority class is a different class feature each of the data types.

Even though, SMOTE is for simplicity often explained as a solution to a two-class problem it is also applied to multi-class problems [6]. When addressing a multi-class problem as in this research it can be specified to the algorithm which class to SMOTE for. Since the different data types have different distributions, the settings for the SMOTE algorithm can be different. Therefore, the settings for each of the data types is defined as shown in Table 5.2. For social the the positive and neutral class are not that different in size while the negative class is clearly smaller which is why the resample setting for the SMOTE algorithm is set on 'minority'. The review data has two classes which are almost equal in size (neutral and negative) and one class which is bigger than the two combined and thus the resample setting is 'all'. Finally, for the phone data the positive class is really small, even though the two other classes are not equal as well, the minority class should be resampled and the setting is thus 'minority'.

Table 5.2: Characteristics of imbalanced data and SMOTE settings. Minority means resample minority class, all means resample all classes.

Data group	Minority class	# positive	# neutral	# negative	resample settings
Social data	Negative	186	221	89	Minority
Reviews	Neutral & Negative	367	134	124	All
Phone	Positive	22	437	279	Minority

5.7. Classification

We described features, feature extraction, and feature weighting. Now we can describe the classifier which is trained on these features. Since sentiment analysis is formulated as a supervised learning problem this also requires a supervised learning method. This is why the support vector machine (SVM) is used to build a model for classification.

There are multiple reasons for choosing the SVM, first it is important to note that the focus of this thesis is on the identification, extraction, and evaluation of the feature sets within the sentiment analysis pipeline. This is why one classifier is picked which is reliable and most used in the literature of sentiment analysis. Furthermore, text is an ideal type of data to use for SVM classification due to the sparse nature of text. Text is sparse because when representing for example words in a feature vector most elements are zero. Very few features are irrelevant and they tend to be correlated with each other while being linearly separable into categories [29].

A support vector machine can be used for both classification and regression, however since we are solving a three class problem a classifier is better suited as identified by Dave et al. [8]. The idea is to find a hyperplane that best divides the data into different classes, which is called classification. The data points closest to the hyperplane are called support vectors. The hyperplane should be chosen such that the distance between the hyperplane and the support vectors is as big as possible while the support vectors are still on the correct side of the hyperplane.

Although the SVM used in this report is a multi-class SVM, the SVM will be explained in this Section

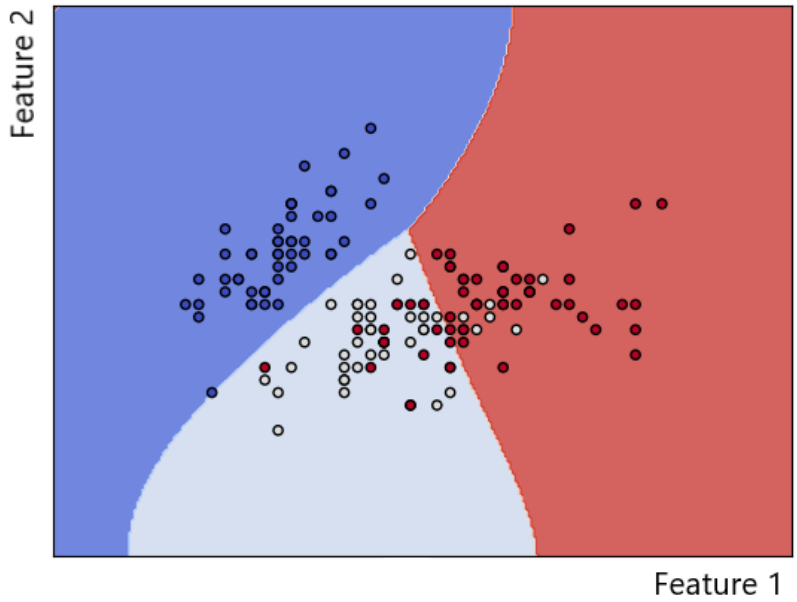

Figure 5.10: SVM hyperplanes, hyperplanes are the lines used to divide the data into different classes. The distance between the hyperplane and the support vectors should be as big as possible while the support vectors are still on the correct side of the hyperplane.

with two classes because this is easier to explain. The input data is composed from n vectors x_i. Each x_i has a value y_i associated with it which explains if the instance belongs to the positive (+1) or negative (-1) class. Also, the vector x_i usually ends up having a lot of dimensions, thus x_i is defined as a p-dimensional vector. The formal definition of the initial dataset is as follows.

$$D = \{(x_i, y_i)|x_i \in R^p, y_i \in \{-1.1\}\}_{i=1}^{2} \tag{5.4}$$

In order to find the optimal hyperplane equation 5.5 is used.

$$w^T x + b = 0 \tag{5.5}$$

$$\begin{aligned} Positive &: w^T x + b \geq +1 \\ Negative &: w^T x + b \leq -1 \end{aligned} \tag{5.6}$$

The distance from the hyperplane to the points is:

$$\frac{|w^T x| + b}{||w||} \tag{5.7}$$

From these equations the optimisation problem can be defined with the following objective function:

$$\max_{w,b} \frac{2}{||w||} \text{ subjective to } \begin{cases} w^T x_i + b \geq +1 \\ w^T x_i + b \leq -1 \end{cases} \tag{5.8}$$

The quadratic optimisation problem:

$$\max_{w,b} \frac{1}{2} w^T w \text{ subjective to } \{y_i(w^T x_i + b) \geq +1) \tag{5.9}$$

Therefore the solution is:

$$\begin{aligned} w &= \textstyle\sum_i \alpha_i y_i x_i \\ b &= y_i - w^T x_i \end{aligned} \tag{5.10}$$

Thus, the classifier is defined by:

$$w^T x + b = \sum_i \alpha_i y_i x_i^T x + b \tag{5.11}$$

5.8. Summary

- This chapter starts with explaining the sentiment analysis pipeline, where the data is divided into a training and test set, this data is preprocessed, features are extracted and feature weighting is applied, a classifier is trained on these features and therefor able to predict the class of new sentences. The output of this classifier together with the ground truth labels which are generated with manual annotation are used to evaluate the model.
- Different feature sets are proposed because there is no consensus in literature which features work well and because its not clear how different features perform for different data types. Because of this three feature sets are proposed for review data and three feature sets are proposed for social data.
- These three feature sets are the most used features in literature, the best performing method for the data type in literature and additional feature sets which are expected to perform well for the data type.
- The feature extraction methods which are used in the feature sets are explained in Section 5.4, which are lexicons, emoticon detection, named entity recognition, syntactic relations, pointwise mutual information and length measures.
- The feature weighting method used in this thesis is explained to translate text to numerical values.
- The data used within this thesis is for each data type imbalanced, the technique used to deal with this is SMOTE and is explained in Section 5.6.
- Finally, the classifier is explained in Section 5.7. We use a support vector machine because it is reliable, most used in literature of sentiment analysis and text is the ideal data type to use for SVM classification.

6

Evaluation: comparing different feature sets

The goal of this chapter is to study which of the feature sets perform best for the data type they are constructed for. In the previous chapters the different feature sets have been explained which are analysed in this chapter.

First, in Section 6.1 the benchmarks are introduced, in order to compare the proposed feature sets with state of the art sentiment analysis methods. The gold standard is explained is explained in Section 6.2. Then, in Section 6.3 the metrics used for the evaluation are explained and what they represent. In Section 6.4 the setup of the experiment is explained. In order to evaluate the results effectively some hypotheses have been drawn in Section 6.5, with which we are able to answer the research questions. After that, the results are presented in Section 6.6 and with these results the hypotheses are answered. Then, the results are discussed in Section 6.7. After that, in Section 6.8 we dive deeper into the results of the review data and propose some improvements to this data and the feature sets for review data. Finally, a summary is given of this chapter.

6.1. Benchmarks

Two benchmarks are used to compare the to proposed feature sets with state of the art methods, three feature sets for review data and three feature sets for social data. This comparison is used to see how our proposed feature sets perform compared to some state of the art methods for sentiment analysis.

Even though the benchmarks are trained on different data which means The Unilever method is used to compare the feature sets against the current implementation of Unilever. The implementation details of the Unilever method are explained in Chapter 3. The Unilever method is trained on a bigger dataset than the dataset used in this thesis, the data used to train the Unilever model is also about products from Unilever. However, the dataset used to train the Unilever method is not annotated by humans which is why this is not used to train the different feature sets.

The Vader method is used as benchmark to see how our proposed feature sets are performing compared to a state of the art sentiment analysis method. The Vader benchmark is used since this is a sentiment analysis method which is provided with the NLTK package in Python and thus is a standard sentiment analysis model. Also, Vader is method has shown to perform quite good generally, the method has a F1-score of 0.96 for social data and 0.63 for Amazon review data [14]. It is a rule based model which uses a sentiment lexicon to determine the sentiment of the words. This method is also trained on different data than used within this thesis, the data used within this thesis is given as test data to the method to analyse the performance of the Vader method. The performance of the benchmarks are reported in Table 6.1. Although these methods are not trained on the same data as in this thesis they are state of the art methods and therefor it is used as a comparison to our proposed feature sets. The metrics which we focus on in the comparison are F1-score and Kappa, the reason for this will be explained in Section 6.4.

Method	Data type	Accuracy	Precision	Recall	F1-score	Kappa
Unilever	Review	57.44%	56.72%	54.99%	53.24%	33.03%
	Social	52.62%	52.76 %	49.83%	49.65%	25.69%
	Phone	63.28%	50.06%	47.88%	45.48%	22.11%
Vader	Review	64.16%	56.65%	54.66%	55.14%	37.15%
	Social	55.65 %	53.21%	54.18%	53.24%	30.69%
	Phone	55.15%	51.41 %	60.60%	45.33%	24.62%

Table 6.1: Performance of baselines on the data used in this research

6.2. Gold standard

A gold standard is a labelled dataset which is used for training and testing the model which contains entities of which the labels are known to be true. Using a gold standard dataset is a standard aspect in sentiment analysis. The reason for using a gold standard dataset is being able to conclude if the trained model performs well. Especially when testing the performance of the model, the test set needs to consist data of which it is sure these labels are correct.

Most commonly human annotators are used for this process. However, this is a time-expensive process since for supervised learning algorithms as much data as possible is required in order to train and test the model. For this reason crowdsourcing platforms are often used for this task. However, the problem with crowdsourcing platforms is some workers might not take the effort to answer correctly.

Also, when using experts to label the data manually this is done by humans and thus can contain mistakes. Furthermore, there might be some cases where different annotators disagree about the label since this is not always obvious. However, considering all the limitations human annotation is still the most accurate way of collecting labelled data for sentiment analysis and thus seen as the gold standard. In order to assure the quality of the labelled dataset and confidentiality issues experts are used to annotate the data which results in the gold standard dataset.

6.3. Evaluation metrics

The metrics used to evaluate the results of the different feature sets are introduced in Section 2.4.3. This section will explain the meaning of each of the metrics.

- Accuracy: with accuracy you can measure what percentage of the output of the model is the same as the input. Accuracy is a standard metric to report for sentiment analysis and is reported in almost all literature studied in Chapter 2. Using this metric alone is unreliable since it does not take all the important factors into account. It only looks at the amount of instances which are labelled correct and incorrect but it does not look at class imbalance.

- Precision: looks at the amount of predictions of a certain class are actually correctly predicted. Thus, the true positives are taken and divided by the total sum of true positives and false positives. It is important to look at this to see how many times an instance is wrongly added to a certain class. Thus, precision takes the amount of misclassified instances to a class into account.

- Recall: takes into account the how many times an instance of a certain class is predicted to be in that class. It takes the true positive and divides it by the true positives and false negatives. Thus, recall takes the instances into account from a certain class which are not detected.

- F1-score: this is a combination of precision and recall, which is looked at if both precision and recall are equally important. The F1-score (with $\beta = 1$) is the harmonic average between precision and recall, for example when either recall or precision would be more important a different value could be chosen for β. If recall would be more important a larger β could be chosen. Since both precision and recall are important for the F1-score is used as an overall comparison that captures both precision and recall equally.

- Kappa: this score is useful to report when class imbalance occurs, as is the case in this research. It uses the probability of random chance, which means it can be a reliable metric even for class imbalance. Also, multi-class classification is better handled by the Kappa score compared to

Table 6.2: Interpretation of Kappa score [23]

Kappa statistic	Strength of Agreement
≤ 0.00	Poor
0.00-0.20	Slight
0.21-0.40	Fair
0.41-0.60	Moderate
0.61-0.80	Substantial
0.81-1.00	Almost perfect

precision and recall. Thus, for both multi-class and imbalanced problems it is a good evaluation metric, which makes it a good metric for this research. Landis and Koch [23] introduced a scale which explains how to interpret the Kappa score as shown in Table 6.2. However, there is no standardised way to interpret the Kappa scores.

6.4. Procedure

In order to understand the results of this experiment, first the setup of the experiment should be explained. There are four main components which are used for this experiment; the feature sets, the datasets, validation of the model and comparison metrics.

Feature sets The feature sets where explained in Chapter 5, all of these feature sets are evaluated in this experiment. These feature sets are used as a full sentiment analysis pipeline which is also explained in Chapter 5. Within this pipeline only the feature set is changed to be able to answer the hypotheses which are drawn in Section 6.5. There are three review feature sets;

- The $NLP\_lexi\_feat\_set$, where the features are selected because they are used the most in review literature.
- The $n-gram\_feat\_set$, which is chosen because it is the best performing in literature review feature set.
- The $NLP\_syntax\_stat\_feat\_set$, which consists of additional review features which are assumed to represent the important characteristics of review data.

And there are also three social feature sets;

- The $NLP\_lexi\_emoti\_feat\_set$, where the features are selected because they are used the most in social literature.
- The $socChar\_NLP\_(domain)Lexi\_feat\_set$, which is chosen because it is the best performing in literature social feature set.
- The $socChar\_emoti\_stat\_feat\_set$, which consists of additional review features which are assumed to represent the important characteristics of social data.

Dataset The construction of the datasets is introduced in Chapter 4. There are a review, social, and phone dataset, which contain of sentences and labels indicating if the sentences are positive, neutral, or negative. Within this experiment only the review and social data are used on the corresponding feature sets.

Model validation In order to be able to see if the method performs well, each dataset needs to be divided into a training and a test set. The reason for this is to be able to see if the method performs as you would want it to, which means the test data can not be used to train the model because otherwise the model would be trained on sentences it would have seen already. In order to split the dataset into a training and test set a trade off should be made between being able to keep enough training

instances and enough test data to carry out the performance assessment.

However, to ensure that the method does not only work well for that specific training and test case, which is called overfitting, k-fold cross validation is used as explained in Section 2.1.5. In order to ensure each fold is representative stratified k-fold is used, which aims to ensure that each class is equally represented in each fold. The k, which determines in how many different sets the data should be divided of which one is the test set and the others are the training sets, is set on 10. The choice for k is a trade off between a big enough test set and enough sentences to train the model on, because the dataset is really small the choice has been made to make the training set as large as possible.

Comparison metrics In order to compare the results the F1-score and Kappa are chosen as metrics. The reason for choosing these two metrics is that the F1-score includes both the misclassified and not detected instances, which are taken into account by precision and recall respectively. Since F1-score takes both precision and recall into account it represents a trade-off of both. Furthermore, Kappa takes class imbalance and the possibility of predicting the class of an instance by chance into account. It compares how much better the classifier performs compared with guessing the target distribution. The other metrics which are explained in Section 6.3 are still reported to show the full information and to be able to discuss the results afterwards.

6.5. Hypotheses

In this section based on the research questions, some hypotheses are drafted. With these hypotheses the research questions related to the feature sets can be answered and insights in the results can be explained in a more clear manner.

RQ1: Can we construct feature sets with reasonable features for review data for which we can identify differences in performance when using pair-wise comparison?

The three review features sets which are compared are the $NLP\_lexi\_feat\_set$, $n{-}gram\_feat\_set$ and $NLP\_syntax\_stat\_feat\_set$. We use a pair-wise comparison in the hypotheses to identify which feature set results in the highest performance.

Hypothesis 1 The $NLP\_lexi\_feat\_set$ (most used review feature set) performs better than the $n-gram\_feat\_set$ (best performing from literature review feature set) for review data.

This first hypothesis addresses the comparison between the $NLP\_lexi\_feat\_set$ and the $n-gram\_feat\_set$ by using F1 and Kappa scores.

1a The $NLP\_lexi\_feat\_set$ has a higher *F1-score* than the $n-gram\_feat\_set$ for *review data*.

1b The $NLP\_lexi\_feat\_set$ has a higher *Kappa score* than the $n-gram\_feat\_set$ for *review data*.

Hypothesis 2 The $NLP\_lexi\_feat\_set$ (most used review feature set) performs better than the $NLP\_syntax\_stat\_feat\_set$ (additional review feature set) for review data.

This second hypothesis addresses the comparison between the $NLP\_lexi\_feat\_set$ and the $NLP\_syntax\_stat\_feat\_set$ by using F1 and Kappa scores.

2a The $NLP\_lexi\_feat\_set$ has a higher *F1-score* than the $NLP\_syntax\_stat\_feat\_set$ for *review data*.

2b The $NLP\_lexi\_feat\_set$ has a higher *Kappa score* than the $NLP\_syntax\_stat\_feat\_set$ for *review data*.

Hypothesis 3 The $n-gram\_feat\_set$ (best performing from literature review feature set) performs better than the $NLP\_syntax\_stat\_feat\_set$ (additional feature set) for review data.

This third hypothesis addresses the comparison between the $n-gram\_feat\_set$ and the $NLP\_syntax\_stat\_feat\_set$ by using F1 and Kappa scores.

3a The $n-gram\_feat\_set$ has a higher *F1-score* than the $NLP\_syntax\_stat\_feat\_set$ for *review data*.

3b The $n-gram\_feat\_set$ has a higher *Kappa score* than the $NLP\_syntax\_stat\_feat\_set$ for *review data*.

RQ2: Can we construct feature sets with reasonable features for social data for which we can identify differences in performance when using pair-wise comparison?

Hypothesis 4 The $NLP\_lexi\_emoti\_feat\_set$ (most used social feature set) performs better than the $socChar\_NLP\_(domain)Lexi\_feat\_set$ (best performing from literature social feature set) for social data.

This fourth hypothesis addresses the comparison between the $NLP\_lexi\_emoti\_feat\_set$ and the $socChar\_NLP\_(domain)Lexi\_feat\_set$ by using F1 and Kappa scores.

4a The $NLP\_lexi\_emoti\_feat\_set$ has a higher *F1-score* than the $socChar\_NLP\_(domain)Lexi\_feat\_set$ for *social data*.

4b The $NLP\_lexi\_emoti\_feat\_set$ has a higher *Kappa score* than the $socChar\_NLP\_(domain)Lexi\_feat\_set$ for *social data*.

Hypothesis 5 The $NLP\_lexi\_emoti\_feat\_set$ (most used social feature set) performs better than the $socChar\_emoti\_stat\_feat\_set$ (additional feature set) for social data.

This fifth hypothesis addresses the comparison between the $NLP\_lexi\_emoti\_feat\_set$ and the $socChar\_emoti\_stat\_feat\_set$ by using F1 and Kappa scores.

5a The $NLP\_lexi\_emoti\_feat\_set$ has a higher *F1-score* than the $socChar\_emoti\_stat\_feat\_set$ for *social data*.

5b The $NLP\_lexi\_emoti\_feat\_set$ has a higher *Kappa score* than the $socChar\_emoti\_stat\_feat\_set$ for *social data*.

Hypothesis 6 The $socChar\_NLP\_(domain)Lexi\_feat\_set$ (best performing from literature social feature set) performs better than the $socChar\_emoti\_stat\_feat\_set$ (additional feature set) for social data.

This sixth hypothesis addresses the comparison between the $socChar\_NLP\_(domain)Lexi\_feat\_set$ and the $socChar\_emoti\_stat\_feat\_set$ by using F1 and Kappa scores.

6a The $socChar\_NLP\_(domain)Lexi\_feat\_set$ has a higher *F1-score* than the $socChar\_emoti\_stat\_feat\_set$ for *social data*.

6b The $socChar\_NLP\_(domain)Lexi\_feat\_set$ has a higher *Kappa score* than the $socChar\_emoti\_stat\_feat\_set$ for *social data*.

6.6. Results

This section presents the results, finds support or no support for the hypotheses, and answers the research questions based on these results. The results relevant to this section are presented in separate tables for the review and social dataset and feature set, they can be found in Table 6.3 and 6.4 respectively.

6.6.1. Review feature sets comparison

This section describes the results of the comparison of the review feature sets on review data. For each of the hypotheses we will evaluate if the results provide enough support.

Table 6.3: Performance of review features sets with review data as input.

Review feature set	Accuracy	Precision	Recall	F1-score	Kappa
$NLP\_lexi\_feat\_set$	64.18% (+/- 3.89%)	55.25% (+/- 6.85%)	51.87% (+/- 4.94%)	**52.49**% (+/- 5.54%)	32.66% (+/- 7.36%)
$n-gram\_feat\_set$	60.85% (+/- 5.90%)	52.01% (+/- 7.74%)	52.45% (+/- 7.07%)	51.82% (+/- 7.28%)	31.46% (+/- 10.15%)
$NLP\_syntax\_stat\_feat\_set$	61.92% (+/- 4.14%)	54.63% (+/- 9.32%)	53.39% (+/- 5.62%)	50.76% (+/- 6.46%)	31.51% (+/- 7.70%)

Hypothesis 1: The $NLP\_lexi\_feat\_set$ performs better than the $n-gram\_feat\_set$ for review data.

1a The $NLP\_lexi\_feat\_set$ has a higher *F1-score* than the $n-gram\_feat\_set$ for *review data*.
The F1-score for the $NLP\_lexi\_feat\_set$ is 52.49% and for the $n-gram\_feat\_set$ 51.82%, as can be found in Table 6.3. This means the F1-score for the $NLP\_lexi\_feat\_set$ is higher. We use Welch's t-test to see if the difference between the F1 scores of the $NLP\_lexi\_feat\_set$ and the $n-gram\_feat\_set$ is statistically significant. Our null hypothesis states that two independent samples have identical average (expected) values. We cannot reject the null hypothesis, t-value = -1.876, p-value = 0.077 (we reject the null hypothesis at p<0.05), degrees of freedom = 18. Thus, the difference is not statistically significant.

1b The $NLP\_lexi\_feat\_set$ has a higher *Kappa score* than the $n-gram\_feat\_set$ for *review data*.
The Kappa score for the $NLP\_lexi\_feat\_set$ is 32.66% and for the $n-gram\_feat\_set$ 31.46%, as can be found in Table 6.3. The Kappa score of the $NLP\_lexi\_feat\_set$ is higher than the Kappa score of the $n-gram\_feat\_set$. We use Welch's t-test to see if the difference between the Kappa scores of the $NLP\_lexi\_feat\_set$ and the $n-gram\_feat\_set$ is statistically significant. Our null hypothesis states that two independent samples have identical average (expected) values. We cannot reject the null hypothesis, t-value = -2.048, p-value = 0.056 (we reject the null hypothesis at p<0.05), degrees of freedom = 18. Thus, the difference is not statistically significant.

Since for both sub-hypothesis 1a and 1b the difference is not statistically significant, we have no support for Hypothesis 1.

Hypothesis 2: The $NLP\_lexi\_feat\_set$ performs better than the $NLP\_syntax\_stat\_feat\_set$ for review data.

2a The $NLP\_lexi\_feat\_set$ has a higher *F1-score* than the $NLP\_syntax\_stat\_feat\_set$ for *review data*.
The F1-score for the $NLP\_lexi\_feat\_set$ is 52.49% and for the $NLP\_syntax\_stat\_feat\_set$ 50.76%, as can be found in Table 6.3. Thus, the F1-score of the $NLP\_lexi\_feat\_set$ is higher than the F1-score of the $NLP\_syntax\_stat\_feat\_set$. We use Welch's t-test to see if the difference between the F1 scores of the $NLP\_lexi\_feat\_set$ and the $NLP\_syntax\_stat\_feat\_set$ is statistically significant. Our null hypothesis states that two independent samples have identical average (expected) values. We cannot reject the null hypothesis, t-value = 0.852, p-value = 0.405 (we reject the null hypothesis at p<0.05), degrees of freedom = 18. Thus, the difference is not statistically significant.

2b The $NLP\_lexi\_feat\_set$ has a higher *Kappa score* than the $NLP\_syntax\_stat\_feat\_set$ for *review data*.

The Kappa score for the $NLP\_lexi\_feat\_set$ is 32.66% and for the $NLP\_syntax\_stat\_feat\_set$ 31.51%, as can be found in Table 6.3. Thus, the Kappa score of the $NLP\_lexi\_feat\_set$ is higher than the Kappa score of the $NLP\_syntax\_stat\_feat\_set$. We use Welch's t-test to see if the difference between the Kappa scores of the $NLP\_lexi\_feat\_set$ and the $NLP\_syntax\_stat\_feat\_set$ is statistically significant. Our null hypothesis states that two independent samples have identical average (expected) values. We cannot reject the null hypothesis, t-value = 0.119, p-value = 0.907 (we reject the null hypothesis at p<0.05), degrees of freedom = 18. Thus, the difference is not statistically significant.

Even though both the F1-score and the Kappa score are higher for the $NLP\_lexi\_feat\_set$ the statistical significance test fails for this observation too. Hence, we found no support for Hypothesis 2.

Hypothesis 3: the $n-gram\_feat\_set$ performs better than the $NLP\_syntax\_stat\_feat\_set$ for review data.

3a The $n-gram\_feat\_set$ has a higher *F1-score* than the $NLP\_syntax\_stat\_feat\_set$ for *review data*.
The F1-score for the $n-gram\_feat\_set$ is 51.82% and for the $NLP\_syntax\_stat\_feat\_set$ 50.76%, as can be found in Table 6.3. Thus, the F1-score of the $n-gram\_feat\_set$ is higher than the F1-score of the $NLP\_syntax\_stat\_feat\_set$. We use Welch's t-test to see if the difference between the F1 scores of the $n-gram\_feat\_set$ and the $NLP\_syntax\_stat\_feat\_set$ is statistically significant. Our null hypothesis states that two independent samples have identical average (expected) values. We cannot reject the null hypothesis, t-value = 0.600, p-value = 0.557 (we reject the null hypothesis at p<0.05), degrees of freedom = 18. Thus, the difference is not statistically significant.

3b The $n-gram\_feat\_set$ has a higher *Kappa score* than the $NLP\_syntax\_stat\_feat\_set$ for *review data*.
The Kappa score for the $NLP\_lexi\_feat\_set$ is 31.46% and for the $NLP\_syntax\_stat\_feat\_set$ 31.51%, as can be found in Table 6.3. Thus, the Kappa score of the $n-gram\_feat\_set$ is lower than the Kappa score of the $NLP\_syntax\_stat\_feat\_set$. We use Welch's t-test to see if the difference between the Kappa scores of the $n-gram\_feat\_set$ and the $NLP\_syntax\_stat\_feat\_set$ is statistically significant. Our null hypothesis states that two independent samples have identical average (expected) values. We cannot reject the null hypothesis, t-value = 2.095, p-value = 0.051 (we reject the null hypothesis at p<0.05), degrees of freedom = 18. Thus, the difference is not statistically significant.

Again, there is no statistical significant difference and therefore we found no support for Hypothesis 3.

6.6.2. Social feature set comparison

Table 6.4: Performance of social features sets with social data as input.

Social feature set	Accuracy	Precision	Recall	F1-score	Kappa
$NLP\_lexi\_emoti\_feat\_set$	61.46% (+/- 5.37%)	57.61% (+/- 7.38%)	56.84% (+/- 6.29%)	56.78% (+/- 6.70%)	38.00% (+/- 8.75%)
$socChar\_NLP\_(domain)Lexi\_feat\_set$	55.89% (+/- 6.91%)	54.85% (+/- 6.51%)	54.85% (+/- 5.76%)	53.54% (+/- 6.16%)	32.23% (+/- 9.67%)
$socChar\_emoti\_stat\_feat\_set$	57.69% (+/- 7.15%)	57.26% (+/- 8.96%)	57.52% (+/- 7.89%)	55.51% (+/- 7.89%)	34.26% (+/- 10.87%)

Hypothesis 4: The $NLP\_lexi\_emoti\_feat\_set$ performs better than the $socChar\_NLP\_(domain)Lexi\_feat\_set$ for social data.

4a The $NLP\_lexi\_emoti\_feat\_set$ has a higher *F1-score* than the $socChar\_NLP\_(domain)Lexi\_feat\_set$ for *social data*.
The F1-score for the $NLP\_lexi\_emoti\_feat\_set$ is 56.78% and for the $socChar\_NLP\_(domain)Lexi\_feat\_set$ 53.54%, as can be found in Table 6.3. The F1-score for the $NLP\_lexi\_emoti\_feat\_set$ is higher. We use Welch's t-test to see if the difference between the F1 scores of the $NLP\_lexi\_emoti\_feat\_set$ and the $socChar\_NLP\_(domain)Lexi\_feat\_set$

is statistically significant. Our null hypothesis states that two independent samples have identical average (expected) values. We cannot reject the null hypothesis, t-value = 0.959, p-value = 0.350 (we reject the null hypothesis at p<0.05), degrees of freedom = 18. Thus, the difference is not statistically significant.

4b The $NLP\_lexi\_emoti\_feat\_set$ has a higher *Kappa score* than the $socChar\_NLP\_(domain)Lexi\_feat\_set$ for *social data*.
The Kappa score for the $NLP\_lexi\_emoti\_feat\_set$ is 38.00% and for the $socChar\_NLP\_(domain)Lexi\_feat\_set$ 32.23%, as can be found in Table 6.3. A similar trend as in Hypothesis 4a is observed, the Kappa score of the $NLP\_lexi\_emoti\_feat\_set$ is higher. The F1-score for the $NLP\_lexi\_emoti\_feat\_set$ is higher. We use Welch's t-test to see if the difference between the Kappa scores of the $NLP\_lexi\_emoti\_feat\_set$ and the $socChar\_NLP\_(domain)Lexi\_feat\_set$ is statistically significant. Our null hypothesis states that two independent samples have identical average (expected) values. We cannot reject the null hypothesis, t-value = 1.221, p-value = 0.238 (we reject the null hypothesis at p<0.05), degrees of freedom = 18. Thus, the difference is not statistically significant.

We found no statistical significant difference and therefore we found no support for Hypothesis 4.

Hypothesis 5: The $NLP\_lexi\_emoti\_feat\_set$ performs better than the $socChar\_emoti\_stat\_feat\_set$ for social data.

5a The $NLP\_lexi\_emoti\_feat\_set$ has a higher *F1-score* than the $socChar\_emoti\_stat\_feat\_set$ for *social data*.
The F1-score for the $NLP\_lexi\_emoti\_feat\_set$ is 56.78% and for the $socChar\_emoti\_stat\_feat\_set$ 55.51%, as can be found in Table 6.3. The F1-score for the $NLP\_lexi\_emoti\_feat\_set$ is higher. We use Welch's t-test to see if the difference between the F1 scores of the $NLP\_lexi\_emoti\_feat\_set$ and the $socChar\_emoti\_stat\_feat\_set$ is statistically significant. Our null hypothesis states that two independent samples have identical average (expected) values. We cannot reject the null hypothesis, t-value = 0.959, p-value = 0.350 (we reject the null hypothesis at p<0.05), degrees of freedom = 18. Thus, the difference is not statistically significant.

5b The $NLP\_lexi\_emoti\_feat\_set$ has a higher *Kappa score* than the $socChar\_emoti\_stat\_feat\_set$ for *social data*.
The Kappa score for the $NLP\_lexi\_emoti\_feat\_set$ is 38.00% and for the $socChar\_emoti\_stat\_feat\_set$ 34.26%, as can be found in Table 6.3. The Kappa score for the $NLP\_lexi\_emoti\_feat\_set$ is higher. We use Welch's t-test to see if the difference between the Kappa scores of the $NLP\_lexi\_emoti\_feat\_set$ and the $socChar\_emoti\_stat\_feat\_set$ is statistically significant. Our null hypothesis states that two independent samples have identical average (expected) values. We cannot reject the null hypothesis, t-value = 0.125, p-value = 0.902 (we reject the null hypothesis at p<0.05), degrees of freedom = 18. Thus, the difference is not statistically significant.

We found no support for Hypothesis 5 as there was no statistical significant difference.

Hypothesis 6: the $socChar\_NLP\_(domain)Lexi\_feat\_set$ performs better than the $socChar\_emoti\_stat\_feat\_set$ for social data.

6a The $socChar\_NLP\_(domain)Lexi\_feat\_set$ has a higher *F1-score* than the $socChar\_emoti\_stat\_feat\_set$ for *social data*.
The F1-score for the $socChar\_NLP\_(domain)Lexi\_feat\_set$ is 53.54% and for the $socChar\_emoti\_stat\_feat\_set$ 55.51%, as can be found in Table 6.3. The F1-score for the $socChar\_emoti\_stat\_feat\_set$ is higher. We use Welch's t-test to see if the difference between the F1 scores of the $socChar\_NLP\_(domain)Lexi\_feat\_set$ and the $socChar\_emoti\_stat\_feat\_set$ is statistically significant. Our null hypothesis states that two independent samples have identical

average (expected) values. We cannot reject the null hypothesis, t-value = -0.774, p-value = 0.449 (we reject the null hypothesis at p<0.05), degrees of freedom = 18. Thus, the difference is not statistically significant.

6b The $socChar\_NLP\_(domain)Lexi\_feat\_set$ has a higher *Kappa score* than the $socChar\_emoti\_stat\_feat\_set$ for *social data*.
The Kappa score for the $socChar\_NLP\_(domain)Lexi\_feat\_set$ is 32.23% and for the $socChar\_emoti\_stat\_feat\_set$ 34.26%, as can be found in Table 6.3. The Kappa score for the $socChar\_emoti\_stat\_feat\_set$ is higher. We use Welch's t-test to see if the difference between the Kappa scores of the $socChar\_NLP\_(domain)Lexi\_feat\_set$ and the $socChar\_emoti\_stat\_feat\_set$ is statistically significant. Our null hypothesis states that two independent samples have identical average (expected) values. We cannot reject the null hypothesis, t-value = -0.627, p-value = 0.539 (we reject the null hypothesis at p<0.05), degrees of freedom = 18. Thus, the difference is not statistically significant.

Since in this hypothesis both the F1-score and Kappa were lower for the $socChar\_NLP\_(domain)Lexi\_feat\_set$ than for the $socChar\_emoti\_stat\_feat\_set$ we would reject the hypothesis if the difference is significant. However, the differences are not significant thus we cannot find support to reject this hypothesis.

6.7. Discussion

In this section we will discuss what the results from Section 6.6 mean when answering the research questions. Also the implications of our results in relation to the research questions will be discussed. In order to discuss the differences between the feature sets we also take the features used in each feature set into account. An overview of the features used in each of the feature sets is given in Table 6.5

When discussing the results for each research questions we will also take the results of the benchmarks into account, introduced in Section 6.1. Even though these benchmarks are trained on different data and thus we can not compare the methods, we use them to see how our feature sets perform compared to some state of the art methods.

Table 6.5: Overview of features used in each feature set

Most used features		Best performing	method from literature	Additional	features
Review	**Social**	**Review**	**Social**	**Review**	**Social**
N-grams (n=1,2)	N-grams (n=1,2)	N-grams (n=1,2,3)	Number of social chars	PMI	PMI
POS tagging	Emoticon detection	Term frequency	POS tagging	POS tagging	Emoticon lexicon
Lexicon (WordNet)	Lexicon (SentiWordNet)		Lexicon (SentiWordNet)	Syntactic relations	Number of social chars
Negator	Negator		Negator	Lenght measures	
Term frequency	Term frequency		Term frequency		
			Domain lexicon		

6.7.1. RQ1: Can we construct feature sets with reasonable features for review data for which we can identify differences in performance when using pair-wise comparison?

To answer the question if we can construct feature sets with reasonable features for review data for which we can identify differences in performance when using pair-wise comparison, we try to identify with the hypotheses which feature sets performs the best using F1 and Kappa on review data. Thus, with the hypotheses we would be interested to see significant differences when comparing F1 and Kappa scores. For Hypothesis 1 until 3 we did not find statistical significant differences in the results, which is why we could not find support for these hypotheses.

Because we only compare different feature sets in our hypotheses it is also interesting to see how our feature sets perform compared against some state of the art benchmarks. The performance of the benchmarks are reported in Table 6.1. Since the benchmarks are trained on different data we know

this is not a fair comparison, but we only use this comparison as a general idea of how the feature sets perform against state of the art methods. When answering the research question we are only looking at the performance using F1 and Kappa of the benchmarks on review data. A summary of the F1 and Kappa scores of the review feature sets and the benchmarks are given in Table 6.6. From this table we can see that both the F1-scores and the Kappa scores of both benchmarks are higher than the F1-scores and the Kappa scores of the feature sets. Since the F1 and Kappa scores of all the review features sets are lower than the F1-scores of the social feature sets, while the benchmarks for review data have a higher F1 and Kappa score than for social data or phone data this result is not as we expected. Furthermore, we saw while generating the results that the neutral class of review data had a lower F1 score than the positive and negative class. We look deeper into this in Section 6.8.

Table 6.6: Summary performance review feature sets against benchmarks

	F1-score	Kappa
Unilever benchmark	53.24%	33.03%
Vader benchmark	54.66%	37.15%
$NLP\_lexi\_feat\_set$	52.49%	32.66%
$n-gram\_feat\_set$	51.28%	31.46%
$NLP\_syntax\_stat\_feat\_set$	50.76%	31.52%

The variance shows the sensitivity of the learning algorithm to the specifics of the training data. This is why it is also important to take into account the variances for each of the feature sets. The higher the variance the more sensitive the feature set is to noise of the data. We do not want the prediction model to be different for each of the different folds resulting from the cross-validation. We can see in Table 6.3 that the $NLP\_lexi\_feat\_set$ has the lowest variance for each of the metrics and $n-gram\_feat\_set$ has the highest variance for each of the metrics. Since the $n-gram\_feat\_set$ uses the least feature extraction methods, as we can see in Table 6.5, this is probably the reason for the highest variance, since the other two feature sets contain more different features.

Thus, we can conclude for this research question that we can not find any significant difference in performance when constructing reasonable features sets for review data. The most used feature set has the highest performance in terms of both F1-score and Kappa.

6.7.2. RQ2: Can we construct feature sets with reasonable features for social data for which we can identify differences in performance when using pair-wise comparison?

To answer if we can construct feature sets with reasonable features for social data for which we can identify differences in performance when using pair-wise comparison, we looked at Hypothesis 4 until 6. Again, we did not find support for these hypotheses because there was no significant difference between the F1 and Kappa score between the social feature sets.

Different than for the review feature sets, for the social feature sets F1-scores and Kappa are all higher than the benchmarks tested on social data. A summary of the F1 and Kappa scores of the social feature sets and the benchmarks are given in Table 6.7. Even though the benchmarks are trained of different data the results do show that our feature sets are comparable against state of the art methods.

Table 6.7: Summary performance social feature sets against benchmarks

	F1-score	Kappa
Unilever benchmark	49.65%	25.69%
Vader benchmark	53.24%	30.69%
$NLP\_lexi\_emoti\_feat\_set$	56.78%	38.00%
$socChar\_NLP\_(domain)Lexi\_feat\_set$	53.54%	32.23%
$socChar\_emoti\_stat\_feat\_set$	55.51%	34.26%

In section 6.7.1 we explained why it is important to look at the variance of the feature sets. We

can see that for the social feature sets the $socChar\_emoti\_stat\_feat\_set$ has the highest variance and thus reacts the most to noise. Furthermore, the $socChar\_NLP\_(domain)Lexi\_feat\_set$ has the lowest variance for the F1-score which means the sentences are less sensitive to sentences which are misclassified or not detected. The $NLP\_lexi\_emoti\_feat\_set$ has the lowest variance for Kappa which means that this feature set is less sensitive to class imbalance than the other two.

We did not perform an extra analysis since there were no unexpected occurrences for the social data, we saw for review data that the neutral class performed using F1 worse than the other classes. However, for social data the three classes (i.e. positive, negative, and neutral) have comparable results. Also, the class imbalance in the social gold standard dataset is smaller compared to review data which suggests that class imbalance has an impact on the performance for sentiment analysis.

Again, we can conclude for this research question that we can not find any significant difference in performance when constructing reasonable features sets for social data. Compared to the other social feature sets the $NLP\_lexi\_emoti\_feat\_set$ has the highest performance for both F1-score and Kappa. Also, all the social feature sets have a higher performance than the benchmarks from which we assume they are comparable to state of the art methods.

6.8. Further investigation review data

During the discussion of the results we identified none of the review feature sets perform better than the social feature sets, while the benchmarks for review data perform better looking at the F1 and Kappa score than social data. The benchmarks in Table 6.1 have a higher performance for review data compared to the social and phone data. Also, the F1-score of the neutral class is for all the features sets lower than the positive and negative class. Since the combination of these occurrences are not expected it is worth to investigate why this happens.

The first intuition was to look at the confusion matrix to make sense of what the F1-score and the Kappa score mean. The confusion matrices for the three review feature sets are shown in Table 6.8, Table 6.9, and Table 6.10.

Table 6.8: Confusion matrix for review *most used feature set*. The true labels are shown at the top column names and the predicted labels are shown at the vertical rows. The diagonal which starts at the top left corner shows the correctly predicted labels.

True/Predicted	Negative	Neutral	Positive
Negative	54	25	45
Neutral	23	37	74
Positive	26	31	310

Table 6.9: Confusion matrix for review *best performing method in literature*. The true labels are shown at the top column names and the predicted labels are shown at the vertical rows. The diagonal which starts at the top left corner shows the correctly predicted labels.

True/Predicted	Negative	Neutral	Positive
Negative	63	30	31
Neutral	32	42	60
Positive	42	50	275

When analysing the confusion matrices in more depth a few reasons for unexpected occurrences in performance for review data where identified. Below a list with supporting arguments for these unexpected performances are given.

- Imbalanced nature of the data: this follows from the confusion matrices in Table 6.8, Table 6.9 and Table 6.10. The number of positive sentences are predicted correctly more often than for negative and neutral sentences; also the F1-score per class shows this. In the review dataset, the number of positive labelled sentences is 367, while negative and neutral are 124 and 134

Table 6.10: Confusion matrix for review *additional feature set*. The true labels are shown at the top column names and the predicted labels are shown at the vertical rows. The diagonal which starts at the top left corner shows the correctly predicted labels.

True/Predicted	Negative	Neutral	Positive
Negative	71	11	42
Neutral	36	36	62
Positive	62	31	274

respectively as shown in Table **??**. This is a big class imbalanced which is clearly seen in Table 6.8, Table 6.9 and Table 6.10. Also we can identify that for the negative and neutral class the number of sentences which are incorrectly labelled most often have the label positive which is caused by positive being the majority class.

- Nature of review data: when looking at how many wrongly predicted labels each class has for negative and neutral sentences, it can be observed that the neutral class is most often, taking the average of the three feature sets gives 69% of the sentences, wrongly classified. The features and the wrongly classified sentences are studied for all three feature sets. This led to the hypothesis that due to the nature of review data the neutral sentences are most often wrongly predicted, which can be observed in Table 6.8, Table 6.9 and Table 6.10. This hypothesis is studied further in Section 6.8.1 using manual inspection. These neutral sentences sometimes contain negative and positive words even though they do not contain either positive or negative sentiment over the full sentence. For example, "Love but dries my skin!-Clears my face up and makes it smooth, but it also dries my skin out too" the label given by a human annotator is neutral, the label given by the model is positive. This sentence contains both positive and negative sentiment, which makes it neutral. In reviews, people often tend to mention both a positive and a negative fact about a product.

- Only sentences of a review: a review normally consists of multiple sentences, the meaning of a sentence by itself can be limited for certain sentences. We studied sentences and in our dataset only one sentence per review was include. All the sentences of a review together provide the message of the consumer, which is why a sentence by itself can have no meaning or might not be understood because the context is missing. This makes it on the on hand for the human annotator hard to label the sentences for reviews and also for the classification model hard to determine what is meant by the sentence. Because of this the quality of the annotated data will be lower, since the human annotator probably does not know which labels to give for certain sentences and thus is forced to label a sentence which does not reflect the sentiment of the given label well.

- Domain dependent: since the review data is extremely domain dependent it needs enough data to learn which words are positive or negative in this specific domain. For example, when using n-grams the word good probably occurs most in the positive sentences and thus is a useful feature for the positive sentences. The features do not occur enough for the model to properly learn the distinction between the positive, negative and neutral sentences.

6.8.1. Data improvements

Because the neutral sentences have the lowest performance and thus will effect the performance the most when trying to improve the results these are studied in more detail. When looking only at the data we suggest there are three reasons why neutral sentences result into the lowest performance. First of all there are the neutral sentences which are both positive and negative. This is a due to the fact that consumers which write a review about a product often reflect on a good and bad aspect of the product, which can result in a sentence with both positive and negative sentiment. To balance this positive and negative sentiment these sentences are often labelled as neutral by a human annotator. The second reason is there exist sentences for which the human annotator might doubt about the sentiment, because the human annotator is forced to label the sentences positive, negative, or neutral different humans might label these sentences with a different label. These doubtful sentences thus make it harder for the model to classify because they might not clearly fall in one of the three classes.

Furthermore, there are also the sentences which are clearly neutral and thus reflect the neutral class well.

To identify how big these categories are to see the effect of each of these types of sentences, we annotated the 134 neutral sentences again with four human annotators. Instead of giving the human annotators only the options positive, negative, or neutral we also added the labels for 'postive+negative' (pos+neg), 'missing context' and 'I don't know' (when the annotator does not know which label to choose). The results of this are given in Table 6.11. When three or more annotators agreed about the label this label was assigned, but when there was no agreement among the annotators the label 'mixed' was given. The table shows that the biggest group of the neutral sentences are mixed, which means the annotators do not agree.

Table 6.11: Labels of neutral sentences for review data analysed by humans

Pos + neg	Missing context	Mixed	Neutral	Positive	Negative
17.9%	9.0 %	38.8%	17.2%	15.7%	1.5%

The results from Table 6.11 suggest that the training data might influence the performance of the feature sets. In the next chapter we compare the data types by selecting the best performing review and social feature set. Because of the unexpected results in performance for the review data we are not sure which feature set works the best for review data. This is why we propose a best solution for improving the quality of our gold standard for review data. By improving our gold standard we suspect the performance of the review feature sets will be higher. Thus, the following changes are made to the review data, the performance resulting from these changes is reported in Table 6.14:

- The positive class is reduced randomly in size such that the difference between the classes is less extreme. In order to keep enough data to train the model on and to compare it to the social data the positive class is reduced such that the size of the review dataset becomes around 500 sentences. The part of the positive sentences which are removed are selected randomly multiple times to make sure the change in performance is not because of specific sentences which are left out. The results of this change on the performance of the review feature sets is presented in Table 6.12.

Table 6.12: Performance of review feature sets on review data after reducing the size of the positive class to create less imbalance

Review feature set	Accuracy	Precision	Recall	F1-score	Kappa
$NLP\_lexi\_feat\_set$	60.36% (+/- 4.87%)	55.50% (+/- 7.26%)	52.75% (+/- 5.79%)	**52.50**% (+/- 6.11%)	**33.01**% (+/- 8.04%)
$n-gram\_feat\_set$	59.77% (+/- 5.93%)	55.38% (+/- 11.55%)	50.33% (+/- 6.89%)	50.03% (+/- 7.76%)	29.97% (+/- 9.92%)
$NLP\_syntax\_stat\_feat\_set$	56.00% (+/- 8.96%)	53.66% (+/- 10.95%)	50.70% (+/- 7.47%)	49.80% (+/- 7.91%)	28.06% (+/- 11.74%)

- The original neutral sentences which are labelled positive or negative by the extra annotation are removed from the neutral class of the review data. These are instances which are probably not neutral and thus can influence the learned patterns in this dataset a lot or cause for noise in the performance. By removing these sentences the quality of the data should be slightly improved. The changes in performance after removing these sentences are shown in Table 6.13.

Table 6.13: Performance review feature sets on review data after removing the neutral sentences which after the extra annotation appeared to be positive or negative.

Review feature set	Accuracy	Precision	Recall	F1-score	Kappa
$NLP\_lexi\_feat\_set$	68.45% (+/- 4.23%)	58.79% (+/- 6.83%)	55.50% (+/- 6.70%)	**56.14**% (+/- 6.62%)	**39.28**% (+/- 8.64%)
$n-gram\_feat\_set$	66.45% (+/- 2.60%)	54.85% (+/- 8.22%)	48.85% (+/- 4.25%)	49.26% (+/- 5.29%)	30.23% (+/- 6.77%)
$NLP\_syntax\_stat\_feat\_set$	60.84% (+/- 4.26%)	55.05% (+/- 6.24%)	52.58% (+/- 5.40%)	51.51% (+/- 5.33%)	30.29% (+/- 6.88%)

The changes to the data together are shown in Table 6.14, the feature sets which are proposed in Chapter 5 are still the same. By looking at the changes applied on the data apart from each other in Table 6.12 for the reducing of the imbalanced data and in Table 6.13 by removing the neutral sentences which appear to be positive or negative after extra annotation we can compare the effect of each of these changes.

Table 6.14: Performance of review features sets with review data as input after reducing the size of the majority class and removing bad annotations.

Review feature set	Accuracy	Precision	Recall	F1-score	Kappa
$NLP\_lexi\_feat\_set$	64.88% (+/- 6.18%)	59.48% (+/- 7.65%)	55.84% (+/- 5.87%)	**56.05**% (+/- 6.13%)	**39.41**% (+/- 9.59%)
$n-gram\_feat\_set$	63.11% (+/- 6.68%)	57.74% (+/- 13.29%)	51.59% (+/- 6.48%)	51.51% (+/- 7.24%)	33.55% (+/- 10.78%)
$NLP\_syntax\_stat\_feat\_set$	59.78% (+/- 7.49%)	54.37% (+/- 8.39%)	52.52% (+/- 6.27%)	52.19% (+/- 7.15%)	32.82% (+/- 9.92%)

When comparing the combined changes results as in Table 6.14 with Table 6.3 we see that the F1 and kappa score of almost all review feature sets has increased. Only the F1-score for the $n-gram\_feat\_set$ has decreased slightly. The F1 and Kappa score of the $NLP\_lexi\_feat\_set$ has increased the most when combining both changes. We also see that the variances when only removing neutral sentences which appear to be positive or negative after using extra annotators has lower variances than when reducing the imbalance of the classes. We assume the reason for this is that reducing the class imbalance means that we have less sentences to train and test the method on which results in more noise and therefore higher variance. While removing the neutral sentences which after using more annotators removes some noise because when keeping sentences with a label which might be incorrect this can result in learning from features which actually belong to a different class.

Thus, the results in Table 6.3 support our suggestions regarding the label quality of the neutral sentences in review data. Using only one of the two changes proposed does not lead to an improvement in performance using F1 and Kappa score for all the feature sets, while using both of the changes proposed improves almost all of the F1 and Kappa scores of the three review feature sets. We proposed a best solution to improve the quality of the gold standard for review data which resulted to an improvement in performance or almost stayed the same for all the review feature sets.

6.8.2. Feature set improvements

Because the $n-gram\_feat\_set$ and the $NLP\_syntax\_stat\_feat\_set$ still were not improved that much compared to the performance reported in Table 6.3 we assume there might be features in these feature sets which do not support the characteristics of review data. This means we might need to add one or two feature extraction method or remove a feature which does not support this data type. We did see that the performance looking at F1 and Kappa improved a few percentage for the $NLP\_lexi\_feat\_set$. Therefore, we tested for all the features from the $NLP\_lexi\_feat\_set$ one by one if adding them to the $n-gram\_feat\_set$ and the $NLP\_syntax\_stat\_feat\_set$ would result in a higher performance in F1 and Kappa than the results after the data improvement changes which are reported in Table 6.14. The ones that are improving the performance for both F1 and Kappa are reported in Table 6.15.

Table 6.15: Performance of review features sets with review data as input after reducing the size of the majority class and removing bad annotations and adding some features. When the F1-score or Kappa score is higher than both baselines they are shown in bold.

Review feature set	Accuracy	Precision	Recall	F1-score	Kappa
$NLP\_lexi\_feat\_set$	64.88% (+/- 6.18%)	59.48% (+/- 7.65%)	55.84% (+/- 5.87%)	56.05% (+/- 6.13%)	39.41% (+/- 9.59%)
$n-gram\_feat\_set$	63.11% (+/- 6.68%)	57.74% (+/- 13.29%)	51.59% (+/- 6.48%)	51.51% (+/- 7.24%)	33.55% (+/- 10.78%)
$n-gram\_feat\_set + POS$	66.08% (+/- 5.73%)	61.55% (+/- 7.34%)	57.15% (+/- 5.43%)	**57.58%** (+/- 5.96%)	**40.75%** (+/- 9.29%)
$NLP\_syntax\_stat\_feat\_set$	59.78% (+/- 7.49%)	54.37% (+/- 8.39%)	52.52% (+/- 6.27%)	52.19% (+/- 7.15%)	32.82% (+/- 9.92%)
$NLP\_syntax\_stat\_feat\_set + n-grams(n = 1,2)$	63.71% (+/- 6.95%)	59.36% (+/- 8.90%)	55.87% (+/- 6.35%)	56.20% (+/- 6.71%)	38.52% (+/- 10.20%)

In Table 6.15 we can see that after adding POS tags to the $n-gram\_feat\_set$ and adding n-grams to the $NLP\_syntax\_stat\_feat\_set$ those are also comparable to the state of the art methods. Also, we see that after adding these features that the $n-gram\_feat\_set$ has the highest F1 and Kappa score compared to the other features sets for review data.

6.9. Summary

- Two benchmarks, Unilever and Vader, are introduced and used to compare the feature sets to state of the art methods. By using these methods we suggest that if the performance of our feature sets in F1 and Kappa is higher than the benchmarks our feature sets are comparable to these state of the art methods.

- The metrics which are reported are explained: accuracy, precision, recall, F1-score, Kappa. In the evaluation we focus on the F1-score because this takes both precision and recall into account which represent misclassified and undetected sentences respectively and Kappa which takes class imbalanced into account.
- The procedure of the experiment is explained. There are three review and three social feature sets used, the experiment is conducted on review and social data for the respective feature sets. The model is validated with 10-fold cross validation.
- We draw 3 hypotheses for the review feature sets which is a pair-wise comparison of the performance using F1 and Kappa of the feature sets for review data. We also introduce 3 hypotheses for social data which is a pair-wise comparison of the social features sets on social data.
- We identify unexpected occurrences in performance for review data. While the performance for F1 and Kappa are high compared to social data for the benchmarks, the performance of the review feature sets is lower than those benchmarks. Since this result is unexpected we researched why these results might occur.
- We did not find significant differences when comparing the feature sets for both review and social data. Compared to the other social feature sets the $NLP\_lexi\_emoti\_feat\_set$ has the highest performance for both F1-score and Kappa.
- We propose there are four reasons why the review data has high performance for the benchmarks compared to the social data and phone data and lower performance for the review feature sets: imbalanced data, nature of review data, size of the training set, quality of the annotated data.
- To improve the performance of the review feature sets on review data the gold standard review dataset is improved after an extra annotation procedure of human annotators and small feature changes are made to the feature sets that we not performing better than the benchmarks after the data changes.
- The $n - gram_feat_set + POS$performs best for the review data.

7

Evaluation: comparing across data types

This chapter includes the second experiment which looks at the differences between the data types when conducting sentiment analysis. The reason for comparing the differences between data types is to see if a specific feature set for each data type can improve sentiment analysis. This experiment will be used to answer the main research question: *Can sentiment analysis be improved when constructing feature sets specifically for the data type?* In the previous chapter the performance of the proposed feature sets are evaluated, but this chapter looks at how the different data types react on the best performing feature sets.

First, in Section 7.1 the setup of the experiment is explained to understand how the experiment is conducted. Then, in Section 7.2 several hypotheses are proposed to be able to test the research question. Section 7.3 shows the results which are used to answer the hypothesis. Then, in Section 7.4 we discuss these results and connect them to the main research question. Finally, we present some additional results which are identified during the evaluation and worth mentioning.

7.1. Procedure

In order to understand the results of this experiment, first the setup of the experiment should be explained. The goal of this experiment is to identify the differences between using different feature sets on different data types. There are four main components which are used for this experiment; the feature sets, the datasets, validation of the model and comparison metrics. The validation of the model and comparison metrics are the same as in the previous experiment in Section 6.4 and therefore will not be explained again.

Feature sets In this chapter we are comparing the best performing feature sets for the data types to be able to answer the main research question. The main research question is: *Can sentiment analysis be improved when constructing feature sets specifically for the data type?* This is why for each data type the best performing feature set constructed for that data type is selected and compared with the other data type variant. For review data this is the $n-gram\_feat\_set + POS$ which will be compared to the $socChar\_NLP\_(domain)Lexi\_feat\_set$. For social data this will be $NLP\_lexi\_emoti\_feat\_set$ and the $NLP\_lexi\_feat\_set$. Finally, for phone data which we assume it will perform similarly to review data which is why we compare the $n-gram\_feat\_set + POS$ and $socChar\_NLP\_(domain)Lexi\_feat\_set$.

Dataset For this experiment we use a review, social and phone dataset, which consist of sentences and their labels indicating for each sentence if it is positive, neutral, or negative. The construction of the datasets is introduced in Chapter 4. Within this experiment for the review, social and phone data the performance differences per data type are analysed for the best performing feature sets.

For the review data set some changes were made based on Section 6.8 this is done because unexpected results in performance for review data occurred and therefore this was further analysed.

In these analysis the neutral sentences appeared to need extra labelling and some data improvements were made as explained in Section 6.8.1.

7.2. Hypotheses

Some hypothesis are drawn because it is not clear how different features perform on different data types. Since some hypotheses are also drafted in Chapter 6, the numbers of these hypotheses start at 7 to prevent confusion when talking about the hypotheses. Following, a list of all the hypotheses drafted for this chapter can be found.

The feature set which was found to perform the best for F1 and Kappa score on review data in Chapter 6, which is the $n-gram\_feat\_set+POS$ and the social variant of this feature set, the $socChar\_NLP\_(domain)Lexi\_feat\_set$, are selected to compare upon the review data. This hypothesis compares these two feature sets for review data to see whether a feature set constructed for review data performs better for F1 and Kappa on review data than a feature set constructed for social data.

Hypothesis 7 For review data the $n-gram\_feat\_set+POS$ is performing better than the $socChar\_NLP\_(domain)Lexi\_feat\_set$.

7a For review data the $n-gram\_feat\_set+POS$ has a higher F1-score than the $socChar\_NLP\_(domain)Lexi\_feat\_set$.

7b For review data the $n-gram\_feat\_set+POS$ has a higher Kappa score than the $socChar\_NLP\_(domain)Lexi\_feat\_set$.

The feature set which was found to perform the best for F1 and Kappa score on social data in Chapter 6, which is the $NLP\_lexi\_emoti\_feat\_set$ and the review variant of this feature set, the $NLP\_lexi\_feat\_set$, are selected to compare upon the social data. This hypothesis compares these two feature sets for social data to see whether a feature set constructed for social data performs better for F1 and Kappa on social data than a feature set constructed for review data.

Hypothesis 8 For social data the $NLP\_lexi\_emoti\_feat\_set$ is performing better than the $NLP\_lexi\_feat\_set$.

8a For social data the $NLP\_lexi\_emoti\_feat\_set$ has a higher F1-score than the $NLP\_lexi\_feat\_set$.

8b For review data the $NLP\_lexi\_emoti\_feat\_set$ has a higher Kappa score than the $NLP\_lexi\_feat\_set$.

This hypothesis compares these two feature sets for phone data, we suspect phone data to be comparable to review data more than to social data which is why the best performing method on the review data in the previous work was selected. The feature set which was found to perform the best for F1 and Kappa score on review data in Chapter 6, which is the $n-gram\_feat\_set+POS$ and the social variant of this feature set, the $socChar\_NLP\_(domain)Lexi\_feat\_set$, are selected to compare upon the phone data. This hypothesis compares these two feature sets for phone data to see whether a feature set constructed for review data performs better for F1 and Kappa on phone data than a feature set constructed for social data.

Hypothesis 9 For phone data the $n-gram\_feat\_set+POS$ is performing better than the $socChar\_NLP\_(domain)Lexi\_feat\_set$.

9a For phone data the $n-gram\_feat\_set+POS$ has a higher F1-score than the $socChar\_NLP\_(domain)Lexi\_feat\_set$.

9b For phone data the $n-gram\_feat\_set+POS$ has a higher Kappa score than the $socChar\_NLP\_(domain)Lexi\_feat\_set$.

7.3. Results

This section will answer the hypotheses which are drafted in Section 7.2. In each subsection one of the hypothesis will be answered by finding support or reject the sub-hypotheses which are needed to answer the main hypotheses.

7.3.1. Hypothesis 7: Review data

To answer the sub hypotheses related to hypothesis 7 Table 7.1 will be used. Hypothesis 7 will answer if the $n-gram\_feat\_set + POS$ or the $socChar\_NLP\_(domain)Lexi\_feat\_set$ will perform better on review data.

Table 7.1: Comparison performance of feature sets on **review data** with F1-score and Kappa score

Feature set	F1-score	Kappa score method
$n-gram\_feat\_set + POS$	57.58% (+/- 5.96%)	40.75% (+/- 9.29%)
$socChar\_NLP\_(domain)Lexi\_feat\_set$	51.04% (+/- 6.91%)	29.15% (+/- 8.96%)

Hypothesis 7: For review data the $n-gram\_feat\_set + POS$ is performing better than the $socChar\_NLP\_(domain)Lexi\_feat\_set$.

7a For review data the $n-gram\_feat\_set + POS$ has a higher F1-score than the $socChar\_NLP\_(domain)Lexi\_feat\_set$.
The F1-score for the $n-gram\_feat\_set + POS$ is 57.58% and for the $socChar\_NLP\_(domain)Lexi\_feat\_set$ 51.04%, as can be found in Table 7.1. We can clearly see that the score of the $n-gram\_feat\_set + POS$ is higher. We use Welch's t-test to see if the difference between the F1 scores of the $n-gram\_feat\_set + POS$ and the $socChar\_NLP\_(domain)Lexi\_feat\_set$ is statistically significant. Our null hypothesis states that two independent samples have identical average (expected) values. We can reject the null hypothesis, t-value = 2.466, p-value = 0.024 (we reject the null hypothesis at p<0.05), degrees of freedom = 18. Thus we found support for this hypothesis.

7b For review data the $n-gram\_feat\_set + POS$ has a higher Kappa score than the $socChar\_NLP\_(domain)Lexi\_feat\_set$.
The Kappa score for the $n-gram\_feat\_set + POS$ is 40.75% and for the $socChar\_NLP\_(domain)Lexi\_feat\_set$ 29.15%, as can be found in Table 7.1. We can clearly see that the Kappa score of the $n-gram\_feat\_set + POS$ is higher. We use Welch's t-test to see if the difference between the Kappa scores of the $n-gram\_feat\_set + POS$ and the $socChar\_NLP\_(domain)Lexi\_feat\_set$ is statistically significant. Our null hypothesis states that two independent samples have identical average (expected) values. We can reject the null hypothesis, t-value = 2.357, p-value = 0.031 (we reject the null hypothesis at p<0.05), degrees of freedom = 18. Thus we found support for this hypothesis.

We found support for of the sub-hypotheses, which means we can support Hypothesis 7.

7.3.2. Hypothesis 8: Social data

For this hypothesis the performance of best performing feature set for social data is compared against the review variant of this feature set. The results of the performance of social data for the feature sets are shown in Table 7.2.

Table 7.2: Comparison performance of feature sets on social data with F1-score and Kappa score

Feature set	F1-score	Kappa score method
$NLP\_lexi\_emoti\_feat\_set$	56.78% (+/- 6.70%)	38.00% (+/- 8.75%)
$NLP\_lexi\_feat\_set$	57.21% (+/- 5.75%)	37.47% (+/- 7.18%)

Hypothesis 8: For social data the $NLP\_lexi\_emoti\_feat\_set$ is performing better than the $NLP\_lexi\_feat\_set$.

8a For social data the $NLP\_lexi\_emoti\_feat\_set$ has a higher F1-score than the $NLP\_lexi\_feat\_set$.
The F1-score for the $NLP\_lexi\_emoti\_feat\_set$ is 56.78% and for the $NLP\_lexi\_feat\_set$ 57.21%, as can be found in Table 7.2. The F1-score for $NLP\_lexi\_emoti\_feat\_set$ is lower. We use Welch's t-test to see if the difference between the F1 scores of the $NLP\_lexi\_emoti\_feat\_set$ and the $NLP\_lexi\_feat\_set$ is statistically significant. Our null hypothesis states that two independent samples have identical average (expected) values. We cannot reject the null hypothesis, t-value = -0.149, p-value = 0.883 (we reject the null hypothesis at p<0.05), degrees of freedom = 18. Thus, we cannot find support for this hypothesis.

8b For social data the $NLP\_lexi\_emoti\_feat\_set$ has a higher Kappa score than the $NLP\_lexi\_feat\_set$.
The Kappa score for the $NLP\_lexi\_emoti\_feat\_set$ is 38.00% and for the $NLP\_lexi\_feat\_set$ 37.47%, as can be found in Table 7.2. The Kappa score for $NLP\_lexi\_emoti\_feat\_set$ is higher. We use Welch's t-test to see if the difference between the F1 scores of the $NLP\_lexi\_emoti\_feat\_set$ and the $NLP\_lexi\_feat\_set$ is statistically significant. Our null hypothesis states that two independent samples have identical average (expected) values. We cannot reject the null hypothesis, t-value = 0.141, p-value = 0.900 (we reject the null hypothesis at p<0.05), degrees of freedom = 18. Thus, we can not find support for this hypothesis.

We can not find support for this hypothesis, the differences in performance between the $NLP\_lexi\_feat\_set$ and the $NLP\_lexi\_emoti\_feat\_set$ are not statistically significant.

7.3.3. Hypothesis 9: Phone data

For the last dataset, the phone data the assumption has been made that it is most similar to review data.

Table 7.3: Comparison performance of feature sets on phone data with F1-score and Kappa score

Feature set	F1-score	Kappa score method
$n-gram\_feat\_set + POS$	54.66% (+/- 7.92%)	56.51% (+/- 10.30%)
$socChar\_NLP\_(domain)Lexi\_feat\_set$	47.61% (+/- 3.88%)	33.10% (+/- 9.06%)

Hypothesis 9: For phone data the review best performing feature set is performing better than the social best performing feature set.

9a For phone data the $n-gram\_feat\_set + POS$ has a higher F1-score than the $socChar\_NLP\_(domain)Lexi\_feat\_set$.
The F1-score for the $n-gram\_feat\_set + POS$ is 54.66% and for the $socChar\_NLP\_(domain)Lexi\_feat\_set$ 47.61%, as can be found in Table 7.9. We can clearly see that the F1-score of the $n-gram\_feat\_set + POS$ is higher. We use Welch's t-test to see if the difference between the F1 scores of the $n-gram\_feat\_set + POS$ and the $socChar\_NLP\_(domain)Lexi\_feat\_set$ is statistically significant. Our null hypothesis states that two independent samples have identical average (expected) values. We can reject the null hypothesis, t-value = 2.398, p-value = 0.032 (we reject the null hypothesis at p<0.05), degrees of freedom = 18. Thus we found support for this hypothesis.

9b For phone data the $n-gram\_feat\_set + POS$ has a higher Kappa score than the $socChar\_NLP\_(domain)Lexi\_feat\_set$.
The Kappa score for the $n-gram\_feat\_set + POS$ is 56.51% and for the $socChar\_NLP\_(domain)Lexi\_feat\_set$ 33.10%, as can be found in Table 7.9. We can clearly see that the Kappa score of the $n-gram\_feat\_set + POS$ is higher. We use Welch's t-test to see if the difference between the Kappa scores of the $n-gram\_feat\_set + POS$ and the $socChar\_NLP\_(domain)Lexi\_feat\_set$ is statistically significant. Our null hypothesis states that two independent samples have identical average (expected) values. We can reject the null hypothesis, t-value = 5.118, p-value = 0.000 (we reject the null hypothesis at p<0.05), degrees of freedom = 18. Thus, we found support for this hypothesis.

We found support for both of the sub-hypotheses, which means we can support Hypothesis 9.

7.4. Discussion

By answering the sub-hypotheses we were able to answer the main hypothesis in Section 7.3. This section will discuss what these results mean for the main research question: *Can sentiment analysis be improved when constructing feature sets specifically for the data type?* With the results gained by answering the hypotheses we will answer the main research question.

7.4.1. Main research question: Can sentiment analysis be improved when constructing feature sets specifically for the data type?

To answer this research question we compared for each of the data types (i.e. reviews, social data, and phone data) the performance using F1 and Kappa of the best performing feature set for the data type. For review data these are the best performing feature sets from literature, the $n-gram\_feat\_set+POS$ which is compared to the $socChar\_NLP\_(domain)Lexi\_feat\_set$. For social data these are the most used features in literature feature sets, the $NLP\_lexi\_feat\_set$ and the $NLP\_lexi\_emoti\_feat\_set$. Finally, for phone data, which we assume will perform similarly to review data, we compare again the $n-gram\_feat\_set+POS$ and $socChar\_NLP\_(domain)Lexi\_feat\_set$. Looking back at Chapter 6 we found no support for any of the hypotheses when comparing the performance of the three feature sets, for reivew data, and social data separately. These hypotheses in Chapter 6 tried to identify if there is a significant difference comparing the performance of three feature sets, for review data, and social data separately.

To answer the main research question we used Hypothesis 7, 8 and 9, weather these hypotheses are supported by the results or not is used to answer the research question. Both the Hypothesis 7 and Hypothesis 9 are supported and it was found that feature sets constructed for a specific data type might not perform well for other data types. In this case the $socChar\_NLP\_(domain)Lexi\_feat\_set$ does not perform that well for review and phone data because of the significant difference in F1 and Kappa, which means that not all feature sets which work well for one data type will also perform well for other data types. We do know that the performance of the $socChar\_NLP\_(domain)Lexi\_feat\_set$ is comparable to the benchmarks for social data from which we assumed that for social data this feature set performs good. For both the review and phone data the $n-gram\_feat\_set+POS$ performs better than the $socChar\_NLP\_(domain)Lexi\_feat\_set$. In order to answer the research question with more confidence we would want to find support for all hypotheses. Hypothesis 7 and 9 show that there is a difference in performance between feature sets for different data types. However, for Hypothesis 8 we can not find support. Since both the $NLP\_lexi\_feat\_set$ and the $NLP\_lexi\_emoti\_feat\_set$ are used in Hypothesis 8 which do not have a lot of different features when comparing the features used in these feature sets it is not surprising we do not find support.

With the support for hypothesis 7 and 9 we can conclude for review and phone data that there is a difference between feature sets performance for different data types. It shows that feature sets which work well for one data type do not always work well for other data types. Even though the $socChar\_NLP\_(domain)Lexi\_feat\_set$ for social data is comparable to the benchmarks, for review data both the F1 and Kappa score are below the performance of the benchmarks. This means that feature sets which perform well for one data type do not always perform well for another data type.

Because we used the best performing feature set for each of the data types we can conclude that a specific feature set for a data type can give a higher performance for F1 and Kappa while this feature set might not perform well on other data types.

Thus, from the above discussion we conclude that a feature set designed for a specific data type might not work well for other data types. However, there is no significant difference between feature sets designed for a specific data type. Therefore, we can conclude that there might be a feature set which can improve sentiment analysis specifically for the data type, but a general feature set with standard features can be comparable to that result.

7.5. Additional results

Except for answering the research questions, our experiment also gave some interesting results which are worth mentioning.

Table 7.4: Overview of features per feature set

Most used features		Best performing	method from literature	Additional	features
Review	Social	Review	Social	Review	Social
N-grams (n=1,2)	N-grams (n=1,2)	N-grams (n=1,2,3)	Number of social chars	PMI	PMI
POS tagging	Emoticon detection	Term frequency	POS tagging	POS tagging	Emoticon lexicon
Lexicon (WordNet)	Lexicon (SentiWordNet)		Lexicon (SentiWordNet)	Syntactic relations	Number of social chars
Negator	Negator		Negator	Lenght measures	
Term frequency	Term frequency		Term frequency		
			Domain lexicon		

7.5.1. Review data

We identified that the most used features in literature for review and social data are comparable in performance for review data. When comparing the $NLP\_lexi\_feat\_set$ and the $NLP\_lexi\_emoti\_feat\_set$ as shown in Table 7.5 we identified the F1-score of the $NLP\_lexi\_feat\_set$ is higher while the Kappa score of the $NLP\_lexi\_emoti\_feat\_set$ is higher. Even though these differences are not significant this is an interesting result. The main difference between these feature sets is the $NLP\_lexi\_feat\_set$ uses POS tags as a feature, while the $NLP\_lexi\_emoti\_feat\_set$ does not as can be seen in Table 7.4. We suspect that because of the POS tags the $NLP\_lexi\_feat\_set$ has less misclassified sentences (higher precision) but these sentences which are classified correct using POS tags might fall into the majority class because of which the Kappa score is lower because the chance of guessing this label is higher.

Table 7.5: Comparison performance of most used feature sets on review data with F1-score and Kappa score

Feature set	F1-score	Kappa score method
$NLP\_lexi\_feat\_set$	56.05% (+/- 6.13%)	39.41% (+/- 9.59%)
$NLP\_lexi\_emoti\_feat\_set$	55.75% (+/- 5.94%)	39.53% (+/- 9.08%)

7.5.2. Social data

In this section we want to identify what the interesting results in our experiment are for social data. For social we also identified that the most used features in literature for review and social data are comparable in performance for review data as can be seen in Table 7.6. The same pattern which occurred for the review data is the F1-score of the $NLP\_lexi\_feat\_set$ is a little higher than the F1-score of the $NLP\_lexi\_emoti\_feat\_set$, while the Kappa score of the $NLP\_lexi\_emoti\_feat\_set$ is a little higher than the Kappa score of the $NLP\_lexi\_feat\_set$. Even though the difference are insignificant, it is interesting to see that general features work well for both data types. We suggest from this result that there exist general features which work well for multiple data types. Also taking into account that $NLP\_lexi\_emoti\_feat\_set$ gave the highest performance using F1 and Kappa for social data, this might point out that most used features work the best for social data because of the characteristics of social data. Or the $socChar\_NLP\_(domain)Lexi\_feat\_set$ and the $socChar\_emoti\_stat\_feat\_set$ either most used features do not contain features which are increasing performance of social data, which means there might be other features which are specific to social data. This result can also be caused by the small amount of data because of which there are not enough social characters (i.e. smileys, exclamation marks etc.) in the social data used.

Table 7.6: Comparison performance of the most used feature sets on social data with F1-score and Kappa score

Feature set	F1-score	Kappa score method
$NLP\_lexi\_feat\_set$	57.19% (+/- 5.79%)	37.48% (+/- 7.17%)
$NLP\_lexi\_emoti\_feat\_set$	56.78% (+/- 6.70%)	38.00% (+/- 8.75%)

The $n-gram\_feat\_set + POS$ performs better on social data than $socChar\_NLP\_(domain)Lexi\_feat\_set$. This is surprising given the fact that a lot of social features are included in the social best performing method from literature and that the social feature set contains more features than the review feature set. The review best performing method from literature only uses n-grams with n=1,2,3 and POS tags as features.

Also, we identified when using the review best performing method from literature without POS tags as features it performed even better as can be seen in Table 7.7. Because of these reasons we wanted to investigate if n-grams with n=1,2,3 compared to n-grams with n=1,2 generally works better for social data. To confirm this we checked if the review and social most used feature set also gained a higher F1-score and Kappa value when using n-grams with n=1,2,3 which can be seen in Table 7.7. However, this table does not show any evidence for this which might be caused by the small amount of data or the n-grams with n=1,2,3 might perform well for specific sentences in this dataset.

Table 7.7: Social data using trigrams in most used features sets versus uni and bigram

Feature set	n	F1-score	Kappa score
$n-gram\_feat\_set$	1,2,3	58.73 %(+/- 7.47%)	39.59% (+/- 10.44%)
$NLP\_lexi\_feat\_set$	1,2	57.19% (+/- 5.79%)	37.48% (+/- 7.17%)
$NLP\_lexi\_feat\_set$	1,2,3	57.27% (+/- 6.06%)	37.98% (+/- 8.19%)
$NLP\_lexi\_emoti\_feat\_set$	1,2	56.78% (+/- 6.70%)	38.00% (+/- 8.75%)
$NLP\_lexi\_emoti\_feat\_set$	1,2,3	55.33% (+/- 7.40%)	35.61% (+/- 9.75%)

The $n-gram\_feat\_set + POS$ performs better than the $socChar\_emoti\_stat\_feat\_set$. The $socChar\_emoti\_stat\_feat\_set$ has only features which are specific to social data. Because we suspected this might be caused by the n-grams we added n-grams as well to the social additional feature set, which can be seen in Table 7.8. Since both the social and review additional feature set + n-grams score the highest we can at least assume that n-grams work well for social data.

Table 7.8: Social data adding n-grams to the social additional feature set

Feature set	F1-score	Kappa score
$socChar\_emoti\_stat\_feat\_set$	55.51% (+/- 7.89%)	34.26% (+/- 10.87%)
$add\_soc\_feat\_set + n-grams(n = 1,2)$	58.70% (+/- 8.59%)	40.55% (+/- 12.61%)
$NLP\_syntax\_stat\_feat\_set$	53.38% (+/- 8.38%)	31.59% (+/- 11.44%)
$NLP\_syntax\_stat\_feat\_set + n-grams(n = 1,2)$	59.06% (+/- 6.85%)	39.26% (+/- 11.51%)

From these results we can draw some general conclusions about social data. Using n-grams as features works best for social data, whereas specific social features do not seem to have a higher performance than using only n-grams. We identified that not a lot of emoticons are occurring in the social dataset we use in this thesis. Thus, the specific social features do not make a big difference in performance because there are not occurring many emoticons in the social data. One reason for this can be that the dataset is small and therefore they are not an important feature in this dataset. We do see a big difference with review data that shows that POS features do not work well for social data whereas they do work well for review data.

7.5.3. Phone data

To understand the results for phone data we looked at the results in Table 7.9. We will explain the interesting results.

The best_perf_lit_rev_feat_set has a higher performance than the $socChar\_NLP\_(domain)Lexi\_feat\_set$. The same goes for the $NLP\_syntax\_stat\_feat\_set$ compared to the $socChar\_emoti\_stat\_feat\_set$. Because for the social feature sets more social data specific features are added, knowing this we suggest that phone data performs better on review feature sets.

Table 7.9: Comparison performance of feature sets on phone data with F1-score and Kappa score

Feature set	F1-score	Kappa score method
$NLP\_lexi\_feat\_set$	57.30% (+/- 10.28%)	**57.65%** (+/- 10.06%)
$n-gram\_feat\_set + POS$	54.66% (+/- 7.92%)	56.51% (+/- 10.30%)
$NLP\_syntax\_stat\_feat\_set + n-grams(n = 1, 2)$	57.03% (+/- 9.95%)	46.72% (+/- 6.55%)
$NLP\_lexi\_emoti\_feat\_set$	**58.42%** (+/- 10.11%)	57.07% (+/- 11.29%)
$socChar\_NLP\_(domain)Lexi\_feat\_set$	47.88% (+/- 4.12%)	33.67% (+/- 9.36%)
$socChar\_emoti\_stat\_feat\_set$	48.96% (+/- 9.11%)	35.36% (+/- 7.72%)

Again when we compare the $NLP\_lexi\_feat\_set$ and the $NLP\_lexi\_emoti\_feat\_set$ we identify the F1-score of the $NLP\_lexi\_emoti\_feat\_set$ is higher while the Kappa score of the $NLP\_lexi\_feat\_set$ is higher. This is in line with our assumption that most used features perform well for multiple data types.

Another interesting result is that phone data has for all the results a high kappa score, as high as 57.65%, which is significantly higher than kappa scores occurring for social and review data, while the F1-score is similar to the other two types. The reason why the F1 scores are not much higher is the dis-balance leads to large variation in F1-scores for the different classes. For some of the folds two of the two positive classes are misclassified which results in a F1-score of zero. This lowers the overall F1-score a lot, because F1 is calculated as the mean of the F1 of the positive, negative, and neutral class this lowers the value a lot.

The conclusions which can be drawn from this for phone data is that more features did improve the performance for phone data. There is no single feature which works really well for phone, only social features did decrease the performance of sentiment analysis for phone data. This shows that phone data is different from review and social data when using feature sets specific for a data type. However, when using most used features which are assumed to be more general this works well for all three data types.

7.6. Summary

- The setup of the experiment is explained, for each of the data type the best performing feature set is selected. For review data this is the $n-gram\_feat\_set + POS$ which will be compared to the $socChar\_NLP\_(domain)Lexi\_feat\_set$. For social data this will be $NLP\_lexi\_emoti\_feat\_set$ and the $NLP\_lexi\_feat\_set$. Finally, for phone data which we assume it will perform similarly to review data which is why we compare the $n-gram\_feat\_set + POS$ and $socChar\_NLP\_(domain)Lexi\_feat\_set$.
- For each of the data types we draw a hypothesis that uses the best performing feature set for each data type.
- We found support for Hypothesis 7: For review data the $n-gram\_feat\_set + POS$ is performing better than the best performing $socChar\_NLP\_(domain)Lexi\_feat\_set$.
- We also found support for Hypothesis 9: For phone data the review best performing feature set is performing better than the social best performing feature set.
- We can conclude that there might be a method which can improve sentiment analysis specifically for the data type, but a general method can be comparable to that result.
- We also found that both the $NLP\_lexi\_feat\_set$ and the $NLP\_lexi\_emoti\_feat\_set$ perform comparable for each of the data types which suggests that there are general features which work well for multiple data types.

8

Conclusion

In this thesis we tried to investigate if feature sets specifically designed for the data type can improve sentiment analysis. We use three data types to compare the changes in performance using different feature sets, namely reviews, social data and phone data, which is a summary of a phone call about a product from Unilever. To answer the research question we propose three feature sets for review data and three feature sets for social data. We focus on two aspects when answering this research question, comparing the different feature sets and comparing the data types. We tested our feature sets on a Unilever dataset which consists of sentences with one of the possible labels; positive, negative, or neutral. In this Chapter we look back at our research questions which are proposed in Chapter 1, we address each of the research questions by explaining our approach and the conducted evaluations.

8.1. RQ1: Can we construct feature sets with reasonable features for review data which we can use for a pair-wise comparison?

To answer research question 1, we conducted an experiment in Chapter 6 where we performed a pair-wise comparison using F1 and Kappa scores of three review feature sets. Each of the feature sets is chosen with a reason, the $NLP\_lexi\_feat\_set$ which contains NLP features and a lexicon is created from features which are most used in literature about review data. The $n-gram\_feat\_set$, where n-grams are the only features used, is chosen by comparing the performance using F1 of method from literature and using the same feature set and the $NLP\_syntax\_stat\_feat\_set$, where NLP, syntax, and statistical features are used, contains features which studying the review data we assumed might perform well for review data.

Our results showed that there was no significant difference in performance using F1 and Kappa when comparing the review feature sets. We also used two benchmarks on review data to see if the performance of our features sets using F1 and Kappa was comparable to state of the art methods. We only use the benchmarks for a general comparison since they are trained on different data we can not actually compare them with our feature sets. We identified that the F1 and Kappa scores of all the review features sets are lower than the F1-scores of the social feature sets, while the benchmarks for review data have a higher F1 and Kappa score than for social data or phone data this result is not as we expected. Therefore, we looked deeper into the results of review data and suggested a few reasons for these results, the data is imbalanced, review data contains sentences which are both positive and negative and thus result in a neutral label by a human whereas the model would see this differently, only one sentence of a review was used which means the context is missing, and the sentences are domain dependent.

Because of these reasons we proposed some best solutions to improve our review data to be able to retrieve more informative results. One of these solutions involved improving the gold standard dataset used for review data. Since the F1-score of the neutral class was the lowest we conducted an extra human labelling step with four annotators to see the effect of sentences with both positive and nega-

tive sentiment, missing context and being forced to assign a label when using only three classes (i.e. positive, negative, or neutral). From this we found that 38.8% of the neutral sentences received mixed labels, which means that the annotators do not agree with another. This is a problem which more often occurs in sentiment analysis since a label does not always fall into a specific class (i.e. positive, negative or neutral) and different humans have different interpretation about which sentences fall into which class. However, the changes made to the gold standard dataset for review data were the reduction of the majority class and removing the neutral sentences which the human annotators agreed upon where positive or negative. From this study we saw the effect imbalanced data and quality of the ground truth dataset has on the performance of sentiment analysis.

We also looked at improving the review feature sets by adding or removing one feature extraction method from the $NLP\_lexi\_feat\_set$, since the performance of this feature set using F1 and Kappa was improved the most by the changes to the gold standard dataset. This showed us that n-grams and POS features resulted in the highest performance using F1 and Kappa for review data. However, even after the improvements of the data and the features there still was no significant difference in performance for the review feature sets. Although, the performance of the feature sets were comparable to the benchmarks from which we assume that our feature sets are comparable to state of the art methods.

From this we can conclude that we constructed feature sets with reasonable features for review data, but we could not identify a feature set which has significantly higher performance using F1 and Kappa compared to other feature sets designed for review data. Moreover, class imbalance probably influences the performance and using classes with an equal balance would possible further improve the performance. Knowing what we know now, we suggest looking at syntactical features for review data might give a significant difference compared to other feature sets designed for review data.

8.2. RQ2: Can we construct feature sets with reasonable features for social data which we can use for a pair-wise comparison?

This research question is also answered in Chapter 6. We performed a pair-wise comparison using F1 and Kappa scores of three social feature sets. The reasoning behind choosing the feature sets for social data is comparable to the feature sets chosen for review data. The $NLP\_lexi\_emoti\_feat\_set$, which consist of NLP features and a lexicon, is created by selecting the most used features in sentiment analysis literature about social data. The $socChar\_NLP\_(domain)Lexi\_feat\_set$, where we use social character, NLP, and a (domain) lexicon as features, is selected by the best performing method from literature using F1 for social data and the $socChar\_emoti\_stat\_feat\_set$, which includes social characters, emoticon, and statistical features, contains features which studying the social data are proposed to perform well for social data.

The results again showed that there is no significant difference between the performance using F1 and Kappa for the social feature sets on social data. We also used the two benchmarks which are tested on the social data but trained on different data to check if our feature sets are comparable to state of the art methods. We assumed our feature sets are comparable to state of the art methods if they at least have a higher performance using F1 and Kappa than the benchmarks which for social data is true. We did not perform an extra analysis since there were no unexpected occurrences for the social data, we saw for review data that the neutral class performed using F1 worse than the other classes. However, for social data the three classes (i.e. positive, negative, and neutral) have comparable results. Also, the class imbalance in the social gold standard dataset is smaller compared to review data which suggests that class imbalance has an impact on the performance for sentiment analysis.

From this we can conclude that we constructed reasonable feature sets for social data, but we could not identify a feature set with a significant improvement compared to other reasonable feature sets for social data. Since our social dataset is quite small and there we not many specific social characters (i.e. smileys, exclamation marks etc.) which could support the specific social characters this experiment could be repeated on a bigger dataset. Furthermore, we did identify that syntactical

features do not work that well for social data, which logically follows from the fact that they are short messages. Besides, in social data the class imbalance was smaller than for review data which was also visible in the performance when comparing F1 and Kappa scores. However, using equal classes could possible improve the performance for social data.

8.3. Main Research question: Can sentiment analysis be improved when constructing feature sets specifically for the data type?

The main research question is answered in a second experiment which is explained in Chapter 7. To answer this research question we compared for each of the data types (i.e. reviews, social data, and phone data) the performance using F1 and Kappa of the best performing feature set for the data type. For review data these are the best performing feature sets from literature, the $n{-}gram\_feat\_set{+}POS$ which is compared to the $socChar\_NLP\_(domain)Lexi\_feat\_set$. For social data these are the most used features in literature feature sets, the $NLP\_lexi\_emoti\_feat\_set$ and the $NLP\_lexi\_feat\_set$. Finally, for phone data, which we assume will perform similarly to review data, we compare again the $n-gram\_feat\_set+POS$ and $socChar\_NLP\_(domain)Lexi\_feat\_set$. For each data type we created a hypothesis to evaluate the differences in performance of the selected feature sets.

The hypotheses showed that there is a significant difference in performance using F1 and Kappa for review data and phone data when comparing the feature sets. However, for social data we did not find a significant difference in performance using F1 and Kappa. This is why we conclude that a specific feature set for a data type can give a higher performance for F1 and Kappa while this feature set might not performance well on other data types.

We also saw when comparing the most used features from literature using the $NLP\_lexi\_feat\_set$ and $NLP\_lexi\_emoti\_feat\_set$ it gave comparable results for all the data types and since there is no significant difference between the feature sets as supported by our evaluation in 6 this shows that more general features can work well for multiple data types. Therefore, our final conclusion states there might be feature sets which can improve sentiment analysis specifically for the data type, but a general feature set with standard features can be comparable to that result.

8.4. Limitations

During this thesis certain circumstances apposed restrictions to our work, thereby limiting the generalizability of the experiments or influencing our results. This section will identify these limitations and the influence this might have on the work presented in this report.

This biggest threats to this work are the size of the data and the feature distribution. Since this research uses a sentiment analysis method where features are extracted from the input sentences and these features are the input to a machine learning algorithm this requires enough data to train the model constructed by the machine learning algorithm on. When there is not enough data to train the model on some features might not occur frequently and thus might not be important for the model while they might be important features to distinguish between the classes (i.e. positive, neutral, negative) when using more data.

Furthermore, since this research focuses on sentiment analysis at sentence level we often miss context which also influences the data quality of the labelled dataset which is then reflected in our results. Especially for review data which is a story of multiple sentences this can influence the performance. Moreover, sentence level is often too general for the product domain, since people often write about different aspects of a product within one or multiple sentences. Aspect level sentiment analysis would be more suitable in the product domain and captures the sentiment intended by the writer better. However, in this thesis the labelled dataset only contained sentences with their sentiment label (i.e. positive, negative, or neutral) which is why we were not able to construct a sentiment analysis method at aspect level.

For the three data types within this thesis (i.e. review, social and phone) only for the phone data the labelled data is created within this thesis. Therefore, we can question the quality of the labels assigned

social and review data. As explained in Section 6.8 the quality of the review data is not that good and even sentences which are found to be positive or negative after annotating the neutral sentences of review data with four annotators were removed. This questions the quality of human annotation of the social data as well.

Another limitation is the fact that the data is imbalanced, which we assume in our results influences the performance of the implementation within this thesis. For each of the data types there is a majority or minority class which is at least twice as big or small respectively as the other classes. Especially for the phone data the amount of positive sentences is really small (22 positive, 437 neutral, and 279 negative). This can influence the performance significantly as was seen with the review data. After reducing the majority class of the review data the performance best performing class for review data .

Furthermore, in the current implementation of the feature sets a general approach for implementing each feature which was chosen. For example the POS tags where extracted using Perceptron tagger from the NLTK package, however there are a lot more POS taggers available of which there might exist one that works better with the specific conditions of the implementation within this thesis.

On the other hand the features which are chosen for each of the feature sets can also be changed. We selected reasonable features for each of the feature sets, but this does not necessarily mean these are the best features. There are numerous amount of feature extraction techniques and combinations which can be formed of the features within feature sets.

For this research the classifier was set at the start of this thesis on SVM, however there are a lot more machine learning algorithms available which might be able to improve the performance of the classification even more.

To deal with the disbalance of the data SMOTE was used within the methodology of this thesis. However, there are other techniques possible to correct for the disbalance of the data. There might even be techniques which work better for this specific type of data.

8.5. Future work

Our work is to the best of our knowledge the first work that tries to detect if feature sets constructed specifically for the data type can improve sentiment analysis. Our results show an indication that there might be a feature set specifically designed for a data type which performs better than a more general feature set. Therefore, we present suggestions for future work that could overcome the limitations of this work and provide more insight into the differences between data types when conducting sentiment analysis.

Data: the data used within this thesis is a small annotated dataset which contains Unilever specific data. Future research could focus on using a larger dataset to see whether this changes the features which provide a higher performance.

In addition, the dataset used in this research is domain specific because it is collected by Unilever and therefore only contains data related to products. Future research could look at more general data or data from a different domain to improve the generalizability of the research.

Another research direction which is interesting and we did not have time to focus on in this research is the influence of imbalanced data on the performance of sentiment analysis. Although imbalanced data is a research area where already a lot of research has been done, the combination of imbalanced data in sentiment analysis has not been studied much. When generating a training and test set with different classes it is inevitable that the size of the classes are not equal. Since manual labelling is an expensive task it would be beneficial if there would a way to deal with class imbalance without extra work of generating extra instances of a minority class or removing a lot of labelled instances of the majority class, which is called undersampling. Doing research on how to deal with the issue of imbalanced data for sentiment analysis might reduce these costs and thus is worth looking at. Besides, there might be features which a less influenced by imbalanced data than others which could be a future research direction.

Since this study only looks at a small amount of data types, the next step could be to look at more different data types. In order to compare all the data types there could be more understand into their characteristics and the differences in performance when using different feature sets. Other data types

that can be used are for example news articles, forums, emails, and chat messages.

Features: when comparing the feature sets there was no significant difference for any of the feature sets. The reason for this might be the use of common features. We suspect that for review data syntactical features might improve the performance. Future research could look at the influence of these syntactical features which we suspect work well for reviews or feature sets which contain other different features.

Since most of the literature studied in the related work section does not look into why certain features are chosen. Future research could look into the reasons for choosing certain features.

Finally, we do not consider different machine learning algorithms in this research due to time limitations. Even though, in literature different algorithms are compared for sentiment analysis, we selected a machine learning algorithm for which we could find support it would work well for this research. However, extra research might identify different machine learning algorithms performing better than support vector machine.

Bibliography

[1] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media*. Association for Computational Linguistics, 30–38.

[2] Muhammad Zubair Asghar, Aurangzeb Khan, Shakeel Ahmad, and Fazal Masud Kundi. 2014. A review of feature extraction in sentiment analysis. *Journal of Basic and Applied Scientific Research* 4, 3 (2014), 181–186.

[3] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2009. Evaluation measures for ordinal regression. In *Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on*. IEEE, 283–287.

[4] Luciano Barbosa and Junlan Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd international conference on computational linguistics: posters*. Association for Computational Linguistics, 36–44.

[5] Johan Carlberger and Viggo Kann. 1999. Implementing an efficient part-of-speech tagger. *Software: Practice and Experience* 29, 9 (1999), 815–832.

[6] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[7] Smitashree Choudhury and John G Breslin. 2010. User sentiment detection: a YouTube use case. (2010).

[8] Kushal Dave, Steve Lawrence, and David M Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*. ACM, 519–528.

[9] Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd international conference on computational linguistics: posters*. Association for Computational Linguistics, 241–249.

[10] Luigi Di Caro and Matteo Grella. 2013. Sentiment analysis via dependency parsing. *Computer Standards & Interfaces* 35, 5 (2013), 442–453.

[11] Adam Drake, Eric Ringger, and Dan Ventura. 2008. Sentiment regression: Using real-valued scores to summarize overall document sentiment. In *Semantic Computing, 2008 IEEE International Conference on*. IEEE, 152–157.

[12] Shi Feng, Le Zhang, Binyang Li, Daling Wang, Ge Yu, and Kam-Fai Wong. 2013. Is Twitter a better corpus for measuring sentiment similarity?. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 897–902.

[13] Michael Gamon. 2004. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, 841.

[14] CJ Hutto Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf*.

[15] Pollyanna Gonçalves, Matheus Araújo, Fabrício Benevenuto, and Meeyoung Cha. 2013. Comparing and combining sentiment analysis methods. In *Proceedings of the first ACM conference on Online social networks*. ACM, 27–38.

[16] Hussam Hamdan, Patrice Bellot, and Frederic Bechet. 2015. Sentiment Lexicon-Based Features for Sentiment Analysis in Short Text. *Research in Computing Science* 90 (2015), 217–226.

[17] Fahim Muhammad Hasan, Naushad UzZaman, and Mumit Khan. 2007. Comparison of different POS Tagging Techniques (N-Gram, HMM and Brill's tagger) for Bangla. In *Advances and innovations in systems, computing sciences and software engineering*. Springer, 121–126.

[18] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 168–177.

[19] Xia Hu, Lei Tang, Jiliang Tang, and Huan Liu. 2013. Exploiting social relations for sentiment analysis in microblogging. In *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 537–546.

[20] Lifeng Jia, Clement Yu, and Weiyi Meng. 2009. The effect of negation on sentiment analysis and retrieval effectiveness. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 1827–1830.

[21] Ron Kohavi et al. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, Vol. 14. Montreal, Canada, 1137–1145.

[22] Efthymios Kouloumpis, Theresa Wilson, and Johanna D Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! *Icwsm* 11, 538-541 (2011), 164.

[23] J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics* (1977), 159–174.

[24] Shoushan Li, Sophia Yat Mei Lee, Ying Chen, Chu-Ren Huang, and Guodong Zhou. 2010. Sentiment classification and polarity shifting. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 635–643.

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

[26] Bing Liu. 2015. *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press.

[27] Rushi Longadge and Snehalata Dongre. 2013. Class imbalance problem in data mining review. *arXiv preprint arXiv:1305.1707* (2013).

[28] Ryan McDonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. 2007. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the 45th annual meeting of the association of computational linguistics*. 432–439.

[29] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal* 5, 4 (2014), 1093–1113.

[30] Arun Meena and TV Prabhakar. 2007. Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis. In *European Conference on Information Retrieval*. Springer, 573–580.

[31] Yelena Mejova and Padmini Srinivasan. 2011. Exploring Feature Definition and Selection for Sentiment Classifiers.. In *ICWSM*.

[32] Saif M Mohammad and Felipe Bravo-Marquez. 2017. Emotion intensities in tweets. *arXiv preprint arXiv:1708.03696* (2017).

[33] Tony Mullen and Nigel Collier. 2004. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

[34] Ramanathan Narayanan, Bing Liu, and Alok Choudhary. 2009. Sentiment analysis of conditional sentences. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 180–189.

[35] Alvaro Ortigosa, José M Martín, and Rosa M Carro. 2014. Sentiment analysis in Facebook and its application to e-learning. *Computers in Human Behavior* 31 (2014), 527–541.

[36] Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining.. In *LREc*, Vol. 10.

[37] Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 271.

[38] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 79–86.

[39] Ana-Maria Popescu and Orena Etzioni. 2007. Extracting product features and opinions from reviews. In *Natural language processing and text mining*. Springer, 9–28.

[40] Rudy Prabowo and Mike Thelwall. 2009. Sentiment analysis: A combined approach. *Journal of Informetrics* 3, 2 (2009), 143–157.

[41] Sabine Rosenberg and Sabine Bergler. 2012. Uconcordia: Clac negation focus detection at* sem 2012. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 294–300.

[42] Hassan Saif, Yulan He, and Harith Alani. 2012. Semantic sentiment analysis of twitter. In *International semantic web conference*. Springer, 508–524.

[43] Dipanjan Sarkar. 2016. *Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from Your Data*. Apress.

[44] Marina Sokolova, Nathalie Japkowicz, and Stan Szpakowicz. 2006. Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. In *Australasian joint conference on artificial intelligence*. Springer, 1015–1021.

[45] Aixin Sun, Ee-Peng Lim, and Ying Liu. 2009. On strategies for imbalanced text classification using SVM: A comparative study. *Decision Support Systems* 48, 1 (2009), 191–201.

[46] Maite Taboada. 2016. Sentiment analysis: an overview from linguistics. (2016).

[47] Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. ICWSM-A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews.. In *ICWSM*. 162–169.

[48] Bin Wang, Bruce Spencer, Charles X Ling, and Harry Zhang. 2008. Semi-supervised self-training for sentence subjectivity classification. In *Conference of the Canadian Society for Computational Studies of Intelligence*. Springer, 344–355.

[49] Sida Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. Association for Computational Linguistics, 90–94.

[50] Michael Wiegand, Alexandra Balahur, Benjamin Roth, Dietrich Klakow, and Andrés Montoyo. 2010. A survey on the role of negation in sentiment analysis. In *Proceedings of the workshop on negation and speculation in natural language processing*. Association for Computational Linguistics, 60–68.