Vrije Universiteit Amsterdam

Master thesis

Sentiment analysis in the healthcare domain

Author
Abel Armand Schneider

Supervisors
dr. R. Bekker
dr. J. Tomczak

December 31, 2020

Preface

This thesis is written for the graduation project to obtain the MSc degree in Business Analytics at the Vrije Universiteit in Amsterdam. The Master's program Business Analytics is a two year multidisciplinary program with courses in the field of mathematics, computer science and statistics.

The research that has been conducted in this research was part of an internship at Attendi, a company which aims to make the healthcare more personal by developing speech-to-text technology. The focus of this research is on applying sentiment analysis on user generated reviews in the healthcare domain.

I would like to thank Berend Jutte and Diederik de Rave for giving me the opportunity to conduct this research at Attendi and for their support along the way. Moreover, I would like to thank Rene Bekker for his supervision over the last 6 months and Jacub Tomczak for being the second reader of my thesis.

Executive summary

Goal research - The goal of this research is to construct lexicon based models and machine learning models for performing sentiment analysis on user generated reviews in the healthcare domain. Moreover, the performance of both kind of models will be compared to determine which kind of model performs better.

Methodology - Most of the methodology in this research is concerned with the construction of the lexicon based models. The most important aspects of the methodology used in this research include the construction of domain specific lexicons by means of a term frequency-inverse document frequency technique and the construction of classifiers for modification and negation.

Results - The performance of the lexicon based models and the machine learning models are tested on 2 datasets which contain medical reviews. The results indicate that lexicon based models tend to perform better on relatively long reviews whereas machine learning models tend to perform better on relatively short reviews.

Recommendation - The lexicon based models constructed in this research can be used in the natural language processing engine of Attendi. However, it is recommended to improve the performance of the negation classifier by expanding the scope of it. Furthermore, it is recommended to construct health-related sentiment lexicons based on corpora which contain more reviews than the amount of reviews that were used for this research

Contents

1 Introduction

Sentiment analysis is a natural language processing task and is defined as the task of estimating the sentiment in a given text as positive or negative [32, 30, 40]. Sentiment analysis, also known as opinion mining, is a recent sub discipline and can be considered as a mix between information retrieval and computational linguistics. Sentiment analysis is considered to be a difficult task due to the subtle manner in which sentiment can be expressed. On top of that, the border between the expression of sentiment and objective statement can sometimes be vague, therefore, even humans frequently struggle in recognizing and classifying sentiment [47].

The beginning of this century seems to mark the beginning of international awareness and interest in the research problems and opportunities corresponding to sentiment analysis [47]. The rapid growth of the internet and social media, combined with advances in natural language processing techniques, offered the possibility to mine and analyze text data expressed in multiple domains. One of the possible applications of the abundance of text data was the possibility to analyse sentiment expressed by users on the internet and social media. At first, sentiment analysis gained in popularity in industries such as tourism, marketing and the airline industry [21, 4]. In these industries, user generated reviews provided companies the opportunity to have a better insight in the experiences of users. Therefore, companies were able to develop a better understanding of which aspects of their service were enjoyed by users and which aspects of their service users disfavored.

Some years later sentiment, analysis became a popular field of interest in the healthcare domain as well [4, 47]. Sentiment analysis in the healthcare domain can serve multiple purposes. First of all, the sentiment of patients expressed in hospitals may reveal a lot about the current health situation of a patient, the perceived quality of care and the certainty of a diagnosis [41]. Secondly, sentiment expressed by people with mental disorders can be monitored and analyzed [21]. By doing this over a period of time, patterns could be derived and care givers could learn from these insights and develop a better understanding of their patients. Finally, sentiment expressed by medicine users on medical review sites contain a lot of valuable information. Therefore, applying sentiment analysis on these medical reviews results in a better understanding of medicines and their adverse side effects. [6, 9].

However, performing sentiment analysis on user generated reviews regarding medicines poses a lot of challenges due to the specialised nature of these text. Reviews regularly lack coherence and contain spelling errors and slang. Furthermore, multiple studies prove that sentiment analysis is domain dependent [8, 5]. For example, well-known general purpose lexicons such as SentiWordNet and General Inquirer have a low coverage of domain specific words [6]. Moreover, domain specific words that are present in these general purpose lexicons regularly have a wrong polarity.

Another challenge of applying sentiment analysis in the healthcare domain is the desire for transparency. Over the last years, machine learning techniques

arose which scored very well on sentiment analysis tasks. However, in some occasions, the logic underlying these models that generates the output is simply too complex to understand for humans. This is undesirable in the healthcare domain and some even argue this could be illegal according to the General Data Protection Regulation (GDPR). Therefore, in the healthcare domain, simple rule-based algorithms, like for example a lexicon based model, are generally preferred over incomprehensible machine learning models due to transparency constraints.

To construct a lexicon based model, at first, we explore the possibilities for creating a health-related sentiment lexicon by proposing statistical techniques like term frequency-inverse document frequency and pointwise mutual information. These techniques are selected based on previous studies done on creating domain specific lexicons for sentiment analysis in the healthcare domain [22, 8, 4, 17]. Thereafter, classifiers for detecting negation, modification and emoticons are reviewed. These classifiers are mainly inspired by previous studies which performed sentiment analysis on medical reviews [34, 6, 48]. Finally, the goal of this research is to construct multiple lexicon based models, which have incorporated a health-related sentiment lexicon and classifiers for modification, negation, and emoticons, and to compare the performance of these lexicon based models to the performance of the machine learning models.

The paper is structured as follows: in section 2, we present literature regarding the topic of sentiment analysis. In section 3, we describe the methodology used for constructing a lexicon based model and the methodology for constructing baseline machine learning models. Section 4 provides an exploratory data analysis of the data used for this research. The results are described in section 5. Finally, section 6 provides a conclusion and section 7 provides a discussion of this paper.

2 Literature study

In general, two approaches for sentiment analysis are described thoroughly in the literature: the lexicon approach and the machine learning approach. The lexicon approach does not need experiences to learn. Instead, the lexicon approach is focused on the use of collection of terms, phrases, expressions and sentimental idioms known [6, 8]. On the other hand, The machine learning approach is aimed at providing the ability to automatically learn and improve from experience without being explicitly programmed. In the case of sentiment analysis, these experiences consist of providing text which are being labeled as containing positive or negative sentiment [43]. In this section, both the literature regarding the lexicon based approach and the machine learning approach are discussed.

2.1 Lexicon approach

As mentioned before, this study will examine the performance of lexicon based models on sentiment analysis. Generally, the definition of a lexicon is a list of all the words used in a particular language or subject [47]. However, the definition of a lexicon in sentiment analysis differs. In the field of sentiment analysis, and therefore in this research as well, with a lexicon is meant a subjectivity lexicon. A subjectivity lexicon contains only sentiment bearing words. Typically, a lexicon based model is built around a lexicon that contains sentiment words. In addition to the lexicon, classifiers can be added to the lexicon based model to handle more complex grammatical relations. This section will first review lexicons which are used for sentiment analysis purposes and review the structure of these lexicons. Thereafter, methods for adapting a lexicon to a specific domain will be examined. Finally, classifiers which are frequently added to lexicon based models will be reviewed.

2.1.1 Lexicons

Lexicons contain words which are assigned values for the polarity, and in some cases, intensity, and subjectivity [8]. The name for a word with the corresponding values is called an entry. In most lexicons, the values for the polarity range from -1 (very negative) to 1 (very positive) and the values for the subjectivity and intensity are in the range of 0 (very weak) to 2 (very strong). SentiWordNet (SWN) is one of the most well-known lexicons that exists [20, 4]. SWN is based on the database of WordNet. Each entry in SWN contains a positive, negative and objective score in the range of 0.0 to 1.0, with the overall sum of the 3 categories being equal to 1. Each term can consist of multiple synsets. Hereby, a synset indicates the possible senses of a term. For example, in SWN the synset *estimable (3)*, corresponding to the sense *may be computed or estimated* of the adjective *estimable* has an objectivity score of 1.0 (and thus automatically a positive and negative score of 0.0). However, the synset *estimable (1)*, corresponding to the sense *deserving of respect of high regard* has a positive score of 0.75 and an objective score of 0.25.

Another lexicon that is often used in studies regarding sentiment analysis is General Inquirer (GI). GI is a lexicon that lists terms as well as different senses

for these terms [29]. Each term can be assigned the label of being a positive or a negative term. On top of these polarity labels, each term can be labeled as a negation term, an intensifier or a diminisher. Because a term can consist of multiple senses, each sense is assigned a label. An example of a term with two different senses is the word *fun*. The first sense is a noun or adjective meaning *enjoyment* or *enjoyable*. Secondly, a sense of *fun* can be a verb meaning *to ridicule* or *to make fun of*. In this example, the first sense is classified as positive whereas the second sense is classified as negative.

Most lexicons contain adjectives, nouns, verbs and adverbs. These 4 part-of-speech (POS) terms have proven to be capable of expressing opinions and sentiment [3, 5]. Generally, adjectives appear the most in subjectivity lexicons as the presence of adjectives is most closely related to subjectivity and the expression of sentiment [50, 18]. However, not in all lexicons does the majority of words consists of adjectives. Goeuriot et al (2012) developed a lexicon which contained an equal amount of adjectives, nouns, verbs and adverbs. The lexicon based model which used this lexicon achieved promising results.

The majority of the lexicon based approaches are tailored to the English language [26, 36]. To make use of the existing tools for sentiment analysis in English, i.e. existing sentiment lexicons and sentiment analysis models, studies examined the possibility to apply machine translation techniques for converting non-English texts into English [36, 7]. Thereafter, the existing tools could be re-used instead of having to create new tools for each particular language. In general, sentiment analysis of machine-translated text yields worse results than sentiment analysis of the original text. The reason for this is that machine translation typically wrongly translates substantial amounts of text. Moreover, machine translation has the tendency to reduce well-structured texts into sentence fragments [26]. However, in some cases sentiment analysis of machine translated texts results in better performance compared to sentiment analysis of the original text [13]. This is particularly the case for languages which are not easy to interpret by natural language processing tools.

Another possibility to benefit from the already available sentiment lexicons for English is to map an English sentiment lexicon to a different language. This mapping could be done by means of traversing language-specific semantic lexical resources [26]. Moreover, a non-English lexicon can be constructed by bootstrapping from a list of initial seed examples [28]. This method is language independent and can be applied to each language for which WordNet exists. WordNet contains not only sentiment carrying words and can be seen as a thesaurus as it groups words together based on their meanings. This method has led to the first manually annotated lexicon for the Dutch language called Cornetto. Cornetto originates from a thesaurus and thus, is not tailored to sentiment analysis, therefore, the performance of Cornetto is regularly suboptimal.

A second Dutch Lexicon which is frequently used for sentiment analysis of Dutch texts is called Pattern [16]. Pattern was constructed by extracting 1100 Dutch adjectives from Dutch book reviews and, thereafter, these adjectives were annotated in terms of polarity, subjectivity and intensity strength. This set of adjectives was expanded by examining which words had the highest semantic

relatedness with the original set of 1100 adjectives. Eventually, this expansion led to a sentiment lexicon containing approximately 4000 sentiment carrying words. As opposed to Cornetto, Pattern was constructed for sentiment analysis purposes and contains only sentiment bearing words.

2.1.2 Different types of lexicons

In the lexicon approach, a distinction can be made between general purpose lexicons and domain specific lexicons. SWN and GI are usually defined as general purpose lexicons as they were constructed for the purpose of multi-domain sentiment analysis.

General purpose lexicons have some limitations. Multiple studies have proven that the sentiment a word contains is often dependent on the domain in which it is used [47, 7]. For example, consider the word *small* in SWN. The word *small* has a polarity of -0.25 in SWN which is the right polarity in most domains. Consider the use of the word *small* in the hotel domain in the following sentence: *The rooms are very small.* However, when a review was written about a digital camera containing the following sentence: *the camera is great because it has a small size.* It would be more appropriate if the polarity of the word *small* would be positive. Another example is the word *warm* which has a positive polarity in most general purpose lexicons. However, in the healthcare domain the polarity of the word *warm* tends to be negative because it is often associated with inflammation reactions and fever.

Next to sentiment words having the wrong polarity in certain domains, some words carry a subjective burden in a specific domain whereas it refers to objective information in another domain [9, 2]. Therefore, a limitation of general purpose lexicons can be that they have a low coverage of domain specific words. Multiple studies that applied sentiment analysis in the healthcare domain experienced this problem [6, 8]. For example, the word *headache* has an objective sentiment in SWN. However, in the health related domain the polarity of this word should be updated to ensure it has a negative polarity [8]. Another example is the word *heatstroke* which tends to have an objective sentiment in general purpose lexicons whereas it should have a negative sentiment in the healthcare domain.

These two limitations show that general purpose lexicons are suboptimal when applying sentiment analysis in particular domains. This gave rise to the emergence of domain specific lexicons. Domain specific lexicons are constructed by adapting a lexicon to the domain in which it is used. Gourieut et al (2016) and Zuibar et al (2018) adapted a general purpose lexicon into a domain lexicon and the accuracy of their sentiment analysis models on medical reviews increased significantly. In the following section, multiple approaches to construct a domain lexicon will be reviewed.

2.1.3 Construction domain specific lexicon

An option to construct a domain lexicon would be to manually tag all words and annotate them. However, this would be very time consuming in terms of

the annotator time and effort and is therefore not interesting for the scope of this research. Other options are less time consuming and frequently used for the creation of a domain specific lexicon. These options are the bootstrapping technique, pointwise mutual information (PMI) and term frequency-inverse document frequency (TF-IDF). In the next section, these techniques will be reviewed.

Bootstrapping technique

The bootstrapping technique, also named the dictionary approach, starts with manually selecting and annotating a small set of seed words for the domain of interest. Preferably, the selected words in the seed set should contain either strong positive or negative sentiment in the domain of interest [23, 12]. Subsequently, for each word in the seed set, all WordNet relations (hyponym, hypernym, and antonym) are then traversed to discover words that have a relation with the words in the seed set [26, 25]. Hypernyms are considered as the generic terms of hyponyms. So is for example *disease* a hypernym of *cancer*, the hyponym in this example. Antonyms are word with opposite meanings. *Good* is for example an antonym of the word *bad*. Based on the annotated values of the words in the seed set, and the type of relations between the word in the seed set and the words in WordNet, the new word will receive a polarity value. In general, hyponyms and hypernyms of a word in the seed set will receive the same polarity. On the contrary, antonyms of a word in the seed set will receive the opposite polarity.

Thereafter, for the newly discovered sentiment words, the process of traversing the WordNet relations is repeated to explore more sentiment words. For every iteration of the algorithm, a diminishing factor will be applied on the polarity score of the newly discovered words. This will ensure that words that have the shortest path to a word in set seed will be assigned a polarity score which is most similar to that word. Moreover, this will ensure that words that are related to a sentiment word by means of many steps will have a strongly diminished polarity score.

Pointwise mutual information

Another method to expand a set of seeds is by means of the pointwise mutual information (PMI). The PMI of pairs can be computed to add words to the original set of seeds. The PMI of two words can be computed as shown in equation 1:

$$PMI(term, term_i) = log_2 \frac{Pr(term, term_i)}{Pr(term)Pr(term_i)} \tag{1}$$

In this formula, term is the target term, term_i is the seed term and Pr stands for probability. More specifically, $Pr(term, term_i)$ denotes the probability of the joint distribution of the target term and the seed term. whereas $Pr(term)Pr(term_i)$ denotes the probability of the individual distributions. The statistical dependence between a word in the set seed and a target term is calculated based on their co-occurrence in a given corpus [21, 9, 29]. A positive value for the PMI of a pair of word implies that that the word out of the set

seed and the target word occur more often together than under the assumption of independence. On the other hand, a negative value for the PMI implies that the words occur less together than would be expected under the assumption of independence.

Target words with a PMI value higher than a user-defined threshold will be added to the original set of seeds [8]. On the contrary, target words with a PMI value lower than a user-defined threshold are not added. Instead of computing the PMI of a target word in combination with a single word in the set of seed. Turney and Littman (2003) constructed a bootstrapping algorithm that computed the measure of association of target words with the positive class and the negative class. Hereby, the positive class consisted of a set of positive words and the negative class consisted of a set of negative words.

Although PMI is considered a reliable approach for constructing a domain specific lexicon, it comes with some limitations: first of all, a big corpus size is required to obtain good results. It remains unclear what this size should be but most studies use corpora containing at least 10 million words [14, 11]. For certain domains it can be challenging to acquire a corpus of this magnitude. Furthermore, a second limitation can be that target words and their antonyms occur often in similar context [10]. Therefore, the use of PMI could result in words being added to a lexicon with the wrong polarity. A solution for this could be to add conjunction rules to the PMI method. Since words of different polarities are hardly ever conjoined by the word *and* but are generally conjoined by the word *but*, a PMI method that takes the conjunction of 2 words into consideration could be useful.

Term frequency-inverse document frequency

Finally, another way to construct a domain lexicon is by making use of a term frequency-inverse document frequency (TF-IDF) weighting mechanism [8, 17]. TF-IDF originates from information retrieval and essentially, calculates the relative frequency of the occurrence of words in a specific document compared to the inverse proportion of that word over the entire document corpus [33]. The formulas to derive the term frequency (TF) and the inverse document frequency (IDF) can be seen below in equation 2 and equation 3:

$$TF(t) = \frac{Number\ of\ times\ term\ t\ appears\ in\ a\ document}{Total\ number\ of\ terms\ in\ the\ document} \tag{2}$$

$$IDF(t) = \ln \frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ terms\ t\ in\ it} \tag{3}$$

After obtaining the TF and IDF, the TF-IDF of a word can be computed as can be seen in equation 4:

$$TF\text{-}IDF(t) = TF(t) * IDF(t) \tag{4}$$

In sentiment analysis, TF-IDF is applied to determine whether words are more inclined to occur more often in negative reviews or in positive reviews [8, 6]. Hereby, according to TF-IDF, words that occur relatively more often in positive reviews compared to negative reviews should have a positive value and words that occurs relatively more often in negative reviews should have a negative polarity.

This implies that if there is no alignment between the TF-IDF outcome of a word and the polarity of a word in the general purpose lexicon, the polarity of the word in the general purpose lexicon is changed. Moreover, words that are not included in a general purpose lexicon, because it seemed like they did not contain sentiment, can be added to the lexicon in case the TF-IDF outcome of the word indicates that the particular word tends to occur more often in positive or negative reviews.

Zuibar et al (2016) make use of TF-IDF for the construction of a domain specific lexicon. In their research, SWN is used as a general purpose lexicon and terms are adjusted or added based on TF-IDF. At first, they compute the count-based probability of each term in the testing set. Thereafter, the polarity class of the term (i.e. positive or negative) is predicted by computing the probabilities of a term occuring in positive and negative reviews. In case there is no alignment between the prediction and the original polarity value in the general purpose lexicon, the polarity value is changed accordingly.

Demiroz et al (2012) constructed a different algorithm which used TF-IDF for generating a domain specific lexicon. In their research, they take the natural logarithm of the term frequency and the inverse document frequency of both polarity classes. Next, they define a new measure called ΔTF(IDF). This measure determines whether the polarity of a word is correct. In their study, the polarity of a word is only changed if the ΔTF(IDF) score indicates that a word belongs to a different class than its polarity class in SWN.

After either a disagreement between the polarity of a word in the general purpose lexicon and the TF-IDF score has been observed, or finding a word that was not present in the general purpose lexicon but the TF-IDF score indicated that the word occurred relatively more often in the positive or negative class, the polarity of the words can be changed in multiples ways [17]. The following alternatives are for words that appeared already in a general purpose lexicon but with a polarity score that was not in accordance with the TF-IDF outcome of the word:

Flip method: the flip method simply flips the polarity of the word. If, for example, the polarity of the word was -0.3, the polarity after using the flip method will be 0.3.

Objective flip: the objective flip can only modify words with an objective polarity into either positive or negative. Similarly, words with a positive or negative polarity can only be modified into objective.

Shift method: shifting the polarity of the word towards the other pole with a fixed amount. The fixed amount can be similar for all words or can be dependent on the polarity value of the word in the general purpose lexicon.

Next to determining the method for modifying the polarity of words, the amount of words that will be modified should also be considered. Some methods for choosing which words to modify are reviewed below:

Top-k: modifying the polarity of the top $k\%$ of the words having a discrepancy between their TF-IDF score and their polarity score.

Threshold: modifying the polarity of the words of which the TF-IDF score exceeds a fixed threshold and that are not in accordance with their polarity score in the general purpose lexicon that is used.

Iterative: modifying the polarity of a word one at a time. Thereby, only accepting the modified polarity if it results in an improved accuracy on the validation set.

2.1.4 Classifiers lexicon based model

As mentioned at the beginning of this chapter, lexicon based models generally contain a sentiment lexicon and classifiers. The sentiment lexicon forms the foundation of a lexicon model and other classifiers are incorporated in the lexicon model to handle more complex grammatical relations. The most common classifiers which appear in the literature about lexicon based models are the modifier classifier, the negation classifier and the emoticon classifier [34, 6, 48]. For this reason, these classifiers will be reviewed in this section. Moreover, some individual rules that could be added to a lexicon model will be reviewed because they can improve the performance of lexicon based models as well [22].

Modifier Classifier

Modifiers are words that change the polarity strength of sentiment words [6, 44]. Modifiers can be classified into two categories: amplifiers and diminishers. Amplifiers increase the polarity of sentiment words, whereas diminishers reduce the polarity of sentiment words [37].

In the Dutch language, adjectives can be used as adverbs as opposed to the English language in which the ending -ly is usually required. For example, *verschrikkelijk mooi* means *really beautiful* instead of *terrible* + *beautiful* [16]. Despite the fact that *verschrikkelijk* is generally considered as a word that expresses a negative sentiment, it amplifies the positive polarity of the *mooi* in the example sentence. To quantify how much an amplifier or a diminisher changes the polarity of the adjacent sentiment word, some studies make use of lexicons in which each entry contains a value for the intensity [26, 37]. Generally, this value can range from 0 to 2. Modifiers with intensity values higher than 1 are considered amplifiers and modifiers with intensity values lower than 1 are considered diminishers. In case a sentiment words is being used as an modifier for an adjacent sentiment word, the value for the intensity of the modifier, in

combination with the polarity value of the adjacent sentiment word, determines the final polarity. The following example sentences will clarify how amplifiers and diminishers affect the polarity of adjecent sentiment words.

Hij doet dat verschrikkelijk goed.

In this sentence, *verschrikkelijk* is used as an modifier for the adjacent sentiment word *goed*. In case the intensity value of the word *verschrikkelijk* equals 1.5, making it an amplifier, and the polarity value of *goed* equals 0.6, the final polarity is $1.5 * 0.6 = 0.9$.

Hij doet dat redelijk slecht.

In this sentence, *redelijk* is used as an modifier for the adjacent sentiment word *slecht*. In case the intensity value of *redelijk* equals 0.7, making it an diminisher, and the polarity value of the word *slecht* equals -0.6, the final polarity is $0.7 * -0.6 = -0.42$.

Negation Classifier

Next to modifiers, negation terms have an impact on sentiment words as well. Often by reversing the polarity of sentiment words [6, 48, 49]. Negation terms that are used frequently are *niet, nooit* and *geen*. The following example sentences will show how the use of a negation term can flip the polarity of a sentence.

Het medicijn werkt goed.

This sentence contains positive sentiment because of the appearance of the word *goed*.

Het medicijn werkt niet goed.

However, in the second sentence, the negation term *niet* reverses the polarity of the sentiment word *goed* and, therefore, changes the polarity of the sentence from positive to negative.

In the literature, multiple techniques on how to quantify negation are described [48, 6, 27]. First of all, the polarity flip technique simply multiplies the term which has been negated with -1. This technique works well for the negation of sentiment words that are slightly positive or negative. For example, the word *fine* (polarity 0.3) would receive a polarity of -0.3 after it has been negated into *not fine*. However, for sentiment words that carry strong positive or strong negative sentiment, this technique does not lead to preferred outcomes [33]. Consider the term *excellent* which has a polarity of 1, after the negation of the term into *not excellent*, the polarity flip technique would assign a polarity of -1 to this term. Intuitively, this seems far too negative for this term and a more fitting polarity score would be around 0.

A different technique is the polarity shift technique [48]. This technique shifts the polarity of a sentiment word, which has been negated, in the direction of the opposite polarity with a fixed amount. In case this fixed amount is determined to be 0.8, which is regularly used in the literature, *not fine* will result

in a polarity of -0.5 and *not excellent* will result in a polarity of 0.2. As opposed to the polarity flip technique, this method tends to be more suitable for the modification of very positive and negative words. However, for words that contain less sentiment, the polarity shift technique may result in these words getting assigned a wrong polarity.

Connecting negation terms and sentiment words

Next to determining the polarity of sentiment words that have been negated, it can be challenging to derive the scope of a negation term [49]. Generally, negation terms are the preceding word of the sentiment word they negate. However, different sentence constructions do exist as well. Therefore, parsing of sentences enables a better understanding of the linguistic structure of a sentence, and hence, understanding which word is being negated by the negation term [7].

Parsing a sentence may reveal which words are linked to each other by dependency relations [42]. There are 2 types of dependency relations between words, namely: direct relations and indirect relations. A direct relation implies that one word depends directly on the other word or they both depend on a third word directly. On the contrary, an indirect relation implies that one word depends on the other word through other words or both words depend on a third word indirectly. Sentiment word can either be negated by negation word through a dependency relation which is direct or a dependency relation which is indirect.

Emoticon classifier

Another way to express sentiment is by using emoticons. Emoticons are a symbolic illustration of mind, mood, emotional state and feelings [6]. Compared to text, the meaning of emoticons are less dependent on the language and domain in which they are used. Over the years, emoticons have gained in popularity on social media and public reviews and, therefore, their evaluation and classification have become more important for sentiment analysis applications. In their research, Asghar et al (2017) used 230 emoticons of which 120 were labeled positive and 110 were labeled negative. Three human annotators manually assigned polarity scores (between -1 and 1) to all emoticons. Thereafter, the emoticons were incorporated to the sentiment analysis model by using if-then rules.

Customized rules

In addition to the modification, negation and the emoticon classifier, customized rules could be added to the lexicon based model to improve the performance [3, 24]. These customized ruled are aimed at specific linguistic structures and Appel et al (2017) incorporate many of these rules into their lexicon based model. For example, a rule is aimed at the meaning of the word *but*. This rule states that if a sentence contains *but*, all previous sentiment in that sentence should be disregarded and only the sentiment of the part after *but* should be considered. Another rule addresses the meaning of the word *unless*. In case a sentence contains the word *unless* and *unless* is followed by a negative clause, the clause after *unless* is disregarded. Finally, a rule that works in a similar fashion is the

following: when the word *despite* appears in a sentence, the clause after *despite* is disregarded and the part before despite is considered.

2.1.5 Classification by lexicon based models

In sentiment analysis, there are different levels on which sentiment analysis can be performed. The most well known are aspect level, sentence level and document level.

Aspect-based

The goal of aspect-based analysis, also called feature-based sentiment analysis, is to identify aspects of entities and assigning a sentiment to each aspect [6, 15]. Consider the following example sentence: *het medicijn werkt goed maar het is te duur.* In this sentence, two aspects of the medicine are described, namely: the efficacy of the medicine and the price of the medicine. Gwang et al (2018) conducted a study in which they applied sentiment analysis on aspects of the medical experience patients had with medicines. In this research, human annotators, based on the content of a sentence, assigned the sentence to one of the following categories: overall, effectiveness, side effects, condition, dosage, cost. Thereafter, sentiment analysis was performed on the sentences of all categories. The results showed significant discrepancies in the accuracy between the categories.

Sentence-based

Sentence-based analysis is aimed at finding the polarity of sentences [35, 27, 31]. Generally, sentences can be defined as positive, negative and neutral. Appel et al (2018) constructed an algorithm that counts the positive and negative words in a sentence. Hereby, positive words which are negated count as negative words and vice versa. In case a sentence contains more positive than negative words, it is labelled as a positive sentence. Sentences that contain more negative words than positive words are labelled as a negative sentence. In case a sentence contains an equal amount of positive and negative words, an alternative process is followed to determine the polarity of the sentence. The first step of the alternative process consists of labeling the sentence in accordance with the polarity of the word with the strongest polarity or intensity score. If the first step did not suffice, the second step is to make a hierarchy of importance around the part-of-speech (POS) particles. Hereby, the order of most influential to least influential is: adjectives, adverbs, verbs and nouns.

Asghar et al (2018) have a different approach for sentence-based analysis. In their algorithm, the sentiment score is computed by adding all the scores of sentiment words, modifiers and emoticons. Sentences with an aggregated score above 0 are labelled as a positive sentence, sentences with an aggregated score below 0 are labelled as negative and sentences with an aggregated score of 0 are labeled as neutral.

Document-based

Finally, sentiment analysis may also be applied on document-level. An example of sentiment analysis on document level was performed by Gueriot et al (2017). In their study, at first, sentences in reviews were assigned a label based on the frequency in which positive and negative sentiment words occurred in that sentence. Again, sentences containing more positive words than negative words were considered positive and vice versa. Thereafter, a review was considered positive if it contained more positive sentences than negative sentences and the other way around for reviews containing more negative sentiment words.

Another way to classify reviews on document level is by looking at the aggregated sentiment value of all words or phrases that express sentiment [9]. According to this method, an aggregated sentiment value above 0 results in a review being classified as positive and an aggregated sentiment value below 0 leads to the review being classified as negative. A variation on this classification technique is proposed by Kennedy et al (2005). In their study, they count the number of positive and negative phrases that occur in a review. Thereby, a positive word which is negated counts as a negative phrase according to their method. Reviews that contain more positive phrases than negative phrases are classified as positive and vice versa.

2.2 Machine learning approach

Next to the lexicon approach, the machine learning approach is frequently used for sentiment analysis of reviews [30, 41, 50]. In machine learning, 2 different types of algorithms can be distinguished: supervised and unsupervised learning methods [41]. Supervised machine learning methods require labeled data to train classifiers. On the contrary, unsupervised machine learning methods do not require labeled data. Instead, unsupervised learning methods cluster data according to their similarity. For the scope of this research, only supervised methods will be examined.

2.2.1 Features

Machine learning models use characteristics of the text, called features, as input to the model. Therefore, the performance of a machine learning model depends to a large extent on the feature choice. Generally, machine learning models for sentiment analysis use a bag-of-words vector representation as features [47, 19]. A bag-of-words is a representation of a document that describes the occurrence of words within that document. For this representation, the word order in the document is irrelevant and can therefore not be derived from a bag-of-words. From a bag-of-words vector representation, N-gram(s) can be chosen as a feature. An N-gram is a contiguous sequence of n items from a given sequence of text or speech. Generally, an N-gram of 1, also called unigram, is chosen as feature for machine learning models aimed at sentiment analysis [19, 24].

Next to using unigrams as feature for machine learning models, other studies use different features. Mukhtar et al (2019) use bigrams (n=2), part-of-speech information and the position of the of the terms in the text as features. Hereby, the position of the term could, for example, be the presence of the term in the first or last sentence of the review. The position of a term can be relevant because studies show that sentiment is often expressed in the beginning and end of a review. However, the performance of the model was lower compared to the model that used unigrams as features. Moreover, Kennedy et al (2005) use unigrams that occurred at least 3 times in the data set to remove spellings errors and terms that were very rare. This resulted in a better performance of the model.

2.2.2 Machine learning techniques

In the literature regarding sentiment analysis by machine learning models, three different techniques are frequently reviewed: naive Bayes, support-vector machine (SVM) and a decision tree. The following section will briefly discuss the working of these three models.

Naive Bayes

The naive Bayes classifier simplifies learning by assuming that features are independent of each other. In practice, this assumption tends to be wrong frequently. However, despite this wrong assumption, naive Bayes models perform well in text classification and sentiment analysis [46, 19]. Moreover, naive Bayes models require little training data for estimating the classification parameters.

The naive Bayes classifier is derived from Bayes' rule. Bayes' rule is as follows:

$$P(c|r) = \frac{P(c)P(r|c)}{P(r)} \tag{5}$$

Then, the naive Bayes classifier, which can be derived from Bayes' rule, assigns a review (r) to the class c* based on the following formula

$$c* = \underset{c}{\operatorname{argmax}}\, P(c|r) \tag{6}$$

Since $P(r)$ plays no role in selecting c^* and $P(c)$ is more or less equal for the positive and negative class, the term $P(r|c)$ strongly influences the outcome of the naive Bayes classifier. To determine the term $P(r|c)$, the naive Bayes classifier decomposes this term by making the assumption that all the features in the document are conditionally independent given the class of r. This leads to the following formula:

$$P(c|r) := \frac{P(c)(\prod_{i=1}^{m} P(f_i|c)^{n_i(r)})}{P(r)} \tag{7}$$

In this formula, m is the total number of features and n_i indicates how often feature f_i occurs in document r.

Support vector machine

SVM is a supervised kernel method for machine learning. A kernel method uses a mapping function to embed the data in suitable feature space [29]. Thereafter, a linear algorithm is used to discover nonlinear patterns. During the learning phase of the classifier, the classifier assigns weights to all labeled instances and constructs support vectors. Thereafter, the support-vector machine aims to find a hyperplane which separates the document vectors in one class from the other class with the largest separation, or margin, between the two classes. The nearest point from the hyperplane on both sides is called a support vector. Finally, new data points are classified according to their nearest support vector. Support vector machines have proven to be highly effective at text categorization and sentiment analysis [41, 1].

Decision tree

A decision tree is a supervised learning method used for classification and regression. Based on the training data, a tree structure is formed containing decision nodes and leaf nodes. Decision nodes have at least 2 branches and the branches represent the possible outcomes of a conditional statement. A leaf node represents a decision or a regression.

In case unigrams are used as features for the decision tree, each decision node contains a word which appeared in the training set. Moreover, the two branches of that particular decision node represent the presence or absence of the specific word [39]. The leaf nodes present either a positive review or a negative review. The position of a decision node in the three is dependent on the information

gain the corresponding word returns. For example, the presence of the word disaster will probably be more revealing than the presence of the word morning. Therefore, the decision node corresponding to the presence of the word disaster in a review will probably be higher in the tree compared to the decision node corresponding to the presence of the word morning.

2.3 Comparison lexicon based approach and machine learning approach

As mentioned before, eventually, the goal of this research is to develop lexicon based models and machine learning models and compare their performance with regard to sentiment analysis tasks. The evaluation of the performances will be reviewed in the section *results*. This subsection, will review the differences between lexicon based models and machine learning models which are derived from the literature.

The biggest advantage of the lexicon approach is the simplicity of adjusting the model [44]. Problems observed in the output can be targeted directly, making the model more refined over time. Furthermore, the lexicon model can easily be customized to handle various grammatical relations in a sentence. This customization occurs in the form of adding or changing rules to the model [38]. As opposed to the lexicon model, customization of the machine learning model is hardly possible. However, the performance of machine learning models can be improved by changing the features used as input for the model.

Moreover, as lexicon models are based on clearly defined rules, the interpretability of the classification of reviews tends to be higher. Most lexicon based models output a score which is derived by applying rules regarding emoticons, modifiers, negations and sentiment words [4, 6]. Lexicon based models are able to classify on document level by using the classification of individual sentences in a review. On the contrary, the only output created by machine learning models is usually the class in which the review is classified. Thereby, omitting which features resulted in the prediction of the review. Obviously, machine learning models are able to classify on sentence level as well. However, the performance of machine learning models on sentence levels tends to be significantly lower compared to their performance on document level [44].

On the other hand, the machine learning approach has an edge over the lexicon approach in terms of performance [41, 38]. This is especially the case for sentiment analysis on document level where a 10% difference in accuracy is not uncommon. A reason for this is that sentiment expressed without the use of sentiment words is difficult to grasp for the lexicon based model. Consider the sentence, *the next time I hear this song on the radio, I will throw my radio out of the window*. The absence of sentiment carrying words will result in the prediction of a neutral review by the lexicon based model. However, the machine learning model, after being fed with manually labeled training data containing similar expressions, could be able to detect a negative sentiment and classify accordingly.

Finally, a limitation of both the lexicon approach and the machine learning approach is the classification of sentences in which irony or sarcasm is expressed [49]. Irony is defined as the process of intentionally using words or expressions for uttering meaning that is different from the one they have when used literally. Therefore, a lexicon model can classify ironic sentences wrongly by taking the literal meaning of expressed sentiment words into consideration. Machine learning models tend to wrongly classify ironic sentences as well because these

models do not possess the capabilities to recognize irony. Sarcasm is closely related to irony and is generally defined as ironic or satirical wit that is intended to insult, mock or amuse [45]. Sarcasm can be expressed in many different ways, however, in tweets and reviews it is often expressed such that negative activities or states are described as a really positive event. For example the sentence, *absolutely adore it when my bus is late*, is a clear example of sarcasm as most people do not adore when busses are late.

3 Methodology

In this chapter, the necessary steps for constructing lexicon based models and machine learning models will be discussed. Section 3.1 describes the data acquisition and is relevant for both kind of models. Section 3.2 - 3.7 are aimed at the methodology involved in constructing the lexicon based models. Section 3.8 reviews the machine learning models that are used for this research. Finally, section 3.9 describes the evaluation techniques used for this research. The proposed framework for the methodology section aimed at the lexicon approach can be seen in figure 1.

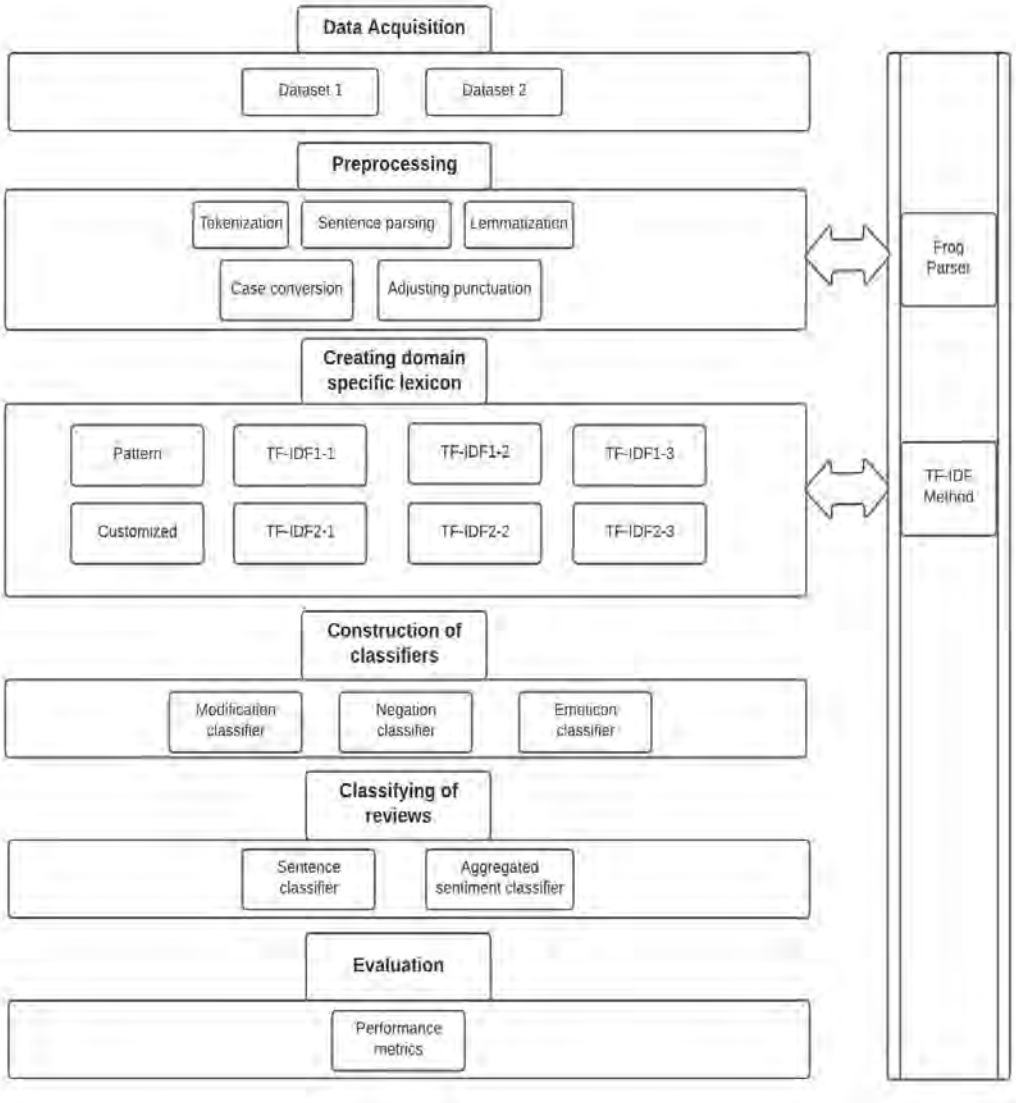


Figure 1: Framework methodology section

3.1 Data acquisition

For this research two different data sets will be used. Both datsets contain medical reviews in which medicine users express their feeling towards the medicine they used. The first data set consists of 8500 medical reviews and originates from the website www.meldpuntmedicijnen.nl. This dataset will from now on be named *dataset 1*. The second dataset consists of 5438 medical reviews and originates from the website www.mijnmedicijn.nl. This dataset will from now on be named *dataset 2*. The data will be more closely examined in chapter 4: *Exploratory data analysis*.

3.2 Preprocessing

This section describes how the reviews, that are used for this research, are preprocessed for the lexicon based models. The preprocessing of the reviews for the machine learning models is different and will be explained in section 3.8.

The preprocessing of the reviews for the lexicon based models consisted of the following steps:

Tokenization

Tokenization is the process of chunking the text of the reviews into small pieces, called tokens. Tokenization can be done on multiple levels, some examples include: word level, sentence level and paragraph level. Tokenization is generally the first step in the preprocessing process because other preprocessing steps are dependent on the tokens retreived from the text. In this research, tokenization will be done on word level because sentiment will also be derived on word level. Consider the following example sentence: *mijn voet doet pijn en is gezwollen.* The outcome of the tokenization can be seen in in table 1.

Token 1	Token 2	Token 3	Token 4	Token 5	Token 6	Token 7	Token 8
mijn	voet	doet	pijn	en	is	gezwollen	.

Table 1: Tokenization of example sentence

Sentence Parsing

As revieved in section 2.1, part-of-speech (POS) information of the words in the reviews can be useful since only 4 POS labels are related to the expression of sentiment. These 4 POS labels are adjectives, adverbs, nouns and verbs. For this reason, a parser is used for assigning POS labels to each word in the reviews. The following example sentences will clarify the importance of assigning POS labels to the words in the reviews:

Ik houd er wel een naar gevoel aan over.

In this example sentence, *naar* is used as an adjective and it expresses negative sentiment.

We gaan naar Friesland op vakantie.

On the contrary, in this example sentence , the word *naar* is used as an preposition and it does not express sentiment.

These example sentences show that the same word may have a different meaning dependent on the POS label and the usage of the particular word in a sentence. Therefore, the lexicon based models constructed in this research will only derive sentiment from words which are assigned the POS label adjectives, adverbs, nouns or verbs.

Next to determining the part-of-speech labels, parsing of sentences can also reveal sentence structures and dependency relations. This information can be used to determine which words are negated by negation terms. The parser used in this research is called FROG and will be reviewed in section 3.3. Moreover, section 3.5 will review negation of sentiment words and dependency relations.

Case conversion

All uppercase letters have been converted into lowercase letters. This has been done because all entries, in the lexicons used for this research, are in lowercase.

Lemmatization

Lemmatization is a technique aimed at the removal of inflectional endings of a word. After the removal, the base of a word remains which is known as the lemma. An example of this is the word *ziek* which is a lemma for the inflected forms *ziekte*, *ziektes* and *ziekig*. The lemmatization step is applied because our general purpose lexicon contains mostly lemmatized forms of sentiment carrying words. Therefore, by using the lemmatized form of words, more sentiment bearing words are recognized. Lemmatization of the reviews will be done by FROG.

Adjusting punctuation

For the lexicon based model, all reviews have to end with a point, an exclamation mark or a question mark. These are the only three options for punctuating the end of a sentence. In case a reviews does not end with one of these characters, a point is added to the end of the review.

Moreover, the performance of the POS parser improves when the punctuation of the reviews is more in accordance with official grammatical rules. The parser used for this research does for example not recognize the end of a sentence in case two dots are written instead of one. Therefore, multiple combinations of punctuation symbols are replaced by a single punctuation sign. An overview of this can be seen in appendix A.

3.3 Part-of-speech tagger

The part-of-speech (POS) parser that is used in this research is called FROG. FROG is included in a unified software distribution called LaMachine. FROG is a POS tagger developed for the Dutch language, and more specifically, for Dutch tweets [11]. Due to the maximum length of tweets of 140 characters, tweets frequently lack coherence and structure. Moreover, they often contain grammatical- and spelling errors. Due to the similarities between tweets and the medical reviews used for this research, FROG has been chosen as parser for this research.

The output of the sentence, *Marie vroeg zich af of hij nog zou komen*, generated by the FROG parser can be seen in table 2.

Index	Text	Lemma	Morph	POS	Po	Depindex	Dep
1	Marie	Marie	[Marie]	SPEC(deeleigen)	1.00000	2	Su
2	vroeg	Vragen	[vroeg]	WW(pv,verl,ev)	0.53254	0	ROOT
3	zich	zich	[zich]	VNW(refl, pron,obl,red,3,getal)	0.99997	2	se
4	af	af	[af]	VZ(fin)	0.99685	2	Svp
5	of	of	[of]	VG(onder)	0.73333	2	Vc
6	hij	hij	[hij]	VNW(pers,pron,nomin, vol,3,ev,masc)	0.99965	8	Su
7	nog	nog	[nog]	BW()	0.99993	8	None
8	zou	zullen	[zal]	WW(pv,verl, ev)	0.99994	5	Body
9	komen	komen	[kom][en]	WW(inf, vrij,zonder)	0.86154	8	Vc
10	.	.	[.]	LET()	0.99995	9	punct

Table 2: FROG output example sentence

The columns in table 2 have the following meaning:

Index: the position of the word in a sentence. The first word in a sentence has index 1.

Text: the word itself.

Lemma: the lemmatized form of the word.

Morph: the morphological segmentation of the word.

POS: the part-of-speech tag that corresponds to the word.

Posprob: the confidence of the Parser of the assigned POS to the word.

Depindex: index number of the head word on which the current word is dependent.

Dep: type of dependency relation between current word and its head word.

3.4 Lexicons

In this section, the general purpose lexicon that is used for this research and the construction of the domain specific lexicons will be reviewed.

3.4.1 General purpose lexicon

For the construction of the lexicon based model, Pattern is used as the general purpose lexicon. Pattern consist of 3917 entries and table 3 shows how the entry of the word *razend* looks in Pattern.

Word form	Sense	Cornetto$_ID$	Word$_Net_ID$	Polarity	Subjectivity	Intensity	Reliability
razend	geweldig	r$_a$ − 14522	a-01387319	0.8	1.0	1.9	1.0

Table 3: Example entry in Pattern lexicon

For the lexicon based model, only the following columns are relevant and will therefore be explained briefly:

Polarity: Indicates the polarity value of the word. Polarity values range from -1 (very negative) to 1 (very positive).

Intensity: Represents a value which can be used as a multiplier if word in entry is used as a multiplier for successive adjective.

3.4.2 Creating domain specific lexicon

As described in the literature section, the bootstrapping technique, pointwise mutual information and term frequency-inverse document frequency are methods for creating a domain specific lexicon. Due to the limited amount of reviews, pointwise mutual information is not a feasible method for this research. Out of the remaining options, term frequency-inverse document frequency (TF-IDF) has been chosen for this research due to the good performance of this method in related studies.

To be more specific, for creating a domain specific lexicon, the TF-IDF method constructed in the article of Demiroz et al (2012) is chosen. In that research, and therefore in this research as well, $TF(w, c)$ expresses the occurrence of the word w in the class c $\{+, -\}$ (either positive or negative). Moreover, $IDF(w)$ is the proportion of documents in which the word w occurs and is computed by dividing the total number of documents (N) by the number of documents which contain the word w. Obviously, many variations on how to compute TF-IDF do exist. The TF-IDF variant which is chosen in this research can be seen in the formulas below:

$$TF\text{-}IDF(w_i, +) = \ln\left(TF(w_i, +) + 1\right) * \ln\left(N/DF(w_i)\right) \tag{8}$$

$$TF\text{-}IDF(w_i, -) = \ln\left(TF(w_i, -) + 1\right) * \ln\left(N/DF(w_i)\right) \tag{9}$$

In these equations, equation 8 depicts how to compute the TF-IDF score of words in the positive class and equation 9 depicts how to compute the TF-IDF score of words in the negative class.

Thereafter, a new measure is defined for polarity adaptation of words. This measure is called $\Delta TF\text{-}IDF$ and it estimates what the polarity of a word should be based on its occurence in positive reviews and negative reviews. The formula can be seen below in equations 10.

$$\begin{aligned}(\Delta TF)IDF(w_i) &= TF\text{-}IDF(w_i,+) - TF\text{-}IDF(w_i,-) \\ &= [TF(w_i,+) - TF(w_i,-)] * IDF(w_i)\end{aligned} \tag{10}$$

The TF-IDF computations are applied on both datasets individually. For dataset 1, the TF-IDF computations are applied on 3200 reviews: 1600 positive reviews and 1600 negative reviews. In table 4, the outcome of some TF-IDF computations from words appearing in dataset 1 can be seen. For dataset 2, the TF-IDF computations are applied on 2174 reviews: 992 positive reviews and 1082 negative reviews. More negative reviews are selected because the positive review have an average length of 120 words and the negative review have an average length of 110 words. In table 5 the outcome of some TF-IDF computations from words appearing in dataset 2 can be seen.

In table 4 and 5, $TF(w_i,+)$ indicates how often the word occurs in positive reviews and $TF(w_i,-)$ indicates how often the word occurs in negative reviews. Moreover, $IDF(w_i)$ is computed by taking the natural logarithm of the outcome of dividing the number of reviews used for these TF-IDF computations (3200 in the first dataset and and 2174 in the second dataset) by the document frequency of the word. Finally, $(\Delta TF)IDF(w_i)$ is the $\Delta TF\text{-}IDF$ score which corresponds to word w_i.

w_i	$TF(w_i,+)$	$TF(w_i,-)$	$IDF(w_i)$	$(\Delta TF)IDF(w_i)$
waard	11	5	5.51	3.81
super	71	11	3.97	7.10
geweldig	44	5	4.38	8.82
troep	6	50	4.19	-8.24
misselijkheid	12	33	4.49	-4.05
hel	1	11	5.77	-11.34

Table 4: Partial list of $(\Delta TF)IDF$ computations on dataset 1

w_i	$TF(w_i, +)$	$TF(w_i, -)$	$IDF(w_i)$	$(\Delta TF)IDF(w_i)$
overleven	22	1	4.73	11.57
zaadlozing	15	2	4.91	8.21
positiever	41	15	3.99	3.84
gif	1	15	5.04	-10.47
extreme	3	12	5.77	-6.22
jeuk	20	50	4.49	-3.25

Table 5: Partial list of $\Delta TF)IDF$ computations on dataset 2

Adding words to sentiment lexicon

After TF-IDF has been calculated on the words in the datasets, words that do not appear in the original sentiment lexicon of Pattern but have a $\Delta TF\text{-}IDF$ score which implies a tendency to a certain polarity class, are added to the sentiment lexicon. For both datasets, 3 different lexicons are constructed. TF-IDF1-1, TF-IDF1-2 and TF-IDF1-3 are constructed based on TF-IDF computations based on dataset 1. TF-IDF2-1, TF-IDF2-2 and TF-IDF2-3 are constructed based on TF-IDF computations based on dataset 2. Basically, the first number in the name of the lexicon denotes on which dataset the lexicon has been contructed and the second number denotes according to which conditions words are added to the lexicon.

The conditions which apply for assigning polarities based on $\Delta TF\text{-}IDF$ values are as follows:

- For lexicons TF-IDF1-1 and TF-IDF2-1, words with a $\Delta TF\text{-}IDF$ score in the range [10,inf] are added with polarity of 1 and words with with a $\Delta TF\text{-}IDF$ score in the range [-inf,-10] are added with a polarity of -1.
- For lexicons TF-IDF1-2 and TF-IDF2-2, the words with a $\Delta TF\text{-}IDF$ score in the range [10,inf] are added with polarity of 1 and words with a $\Delta TF\text{-}IDF$ score in the range [-inf,-10] are added with polarity of -1. Moreover, words with a $\Delta TF\text{-}IDF$ score in the range [5,10] are added with polarity of 0.7 and words with with a $\Delta TF\text{-}IDF$ score in the range [-10,5] are added with polarity of -0.7.
- For lexicons TF-IDF1-3 and TF-IDF2-3, the words with a $\Delta TF\text{-}IDF$ score in the range [10,inf] are added with polarity of 1, words with with a $\Delta TF\text{-}IDF$ score in the range [-inf,-10] are added with polarity of -1, the words with a $\Delta TF\text{-}IDF$ score in the range [5,10] are added with polarity of 0.7 and words with with a $\Delta TF\text{-}IDF$ score in the range [-10,5] are added with a polarity of -0.7. Moreover, words with a $\Delta TF\text{-}IDF$ score in the range [2,5] are added with polarity of 0.5 and words with with a $\Delta TF\text{-}IDF$ score in the range [-5,2] are added with a polarity of -0.5.

An overview of all these conditions can be seen in table 6.

Lexicon	$\Delta TF\text{-}IDF$ **value**	**Assigned polarity**
TF-IDF1-1	$[10, \infty]$	1
TF-IDF2-1	$[-\infty, -10]$	-1
TF-IDF1-2	[10,∞]	1
TF-iDF2-2	$[-\infty, -10]$	-1
	[5,10]	0.7
	$[-10, -5]$	-0.7
TF-IDF1-3	[10,∞]	1
TF-IDF2-3	$[-\infty, -10]$	-1
	[5,10]	0.7
	$[-10, -5]$	-0.7
	[2,5]	0.5
	$[-5, -2]$	-0.5

Table 6: Conditions for adding words to lexicons

Some examples will clarify the condition stated in table 6. Consider the words and corresponding $\Delta TF\text{-}IDF$ values in table 7. Note that these values were derived from dataset 1.

Word	$\Delta TF\text{-}IDF$ value	Word	$\Delta TF\text{-}IDF$ value
waard	3.81	geweldig	8.82
troep	-8.24	hel	-11.34

Table 7: Words with corresponding $\Delta TF\text{-}IDF$ value

As can be seen in table 7, the $\Delta TF\text{-}IDF$ value of the word *waard* is 3.81 which is in the range [2,5]. Therefore, the word *waard* is added to lexicon TF-IDF1-1 with polarity 0.5 The word *geweldig* has a $\Delta TF\text{-}IDF$ value of 8.82 which is in the range [5,10]. Therfore, the word *geweldig* is only added to lexicons TF-IDF1-2 and TF-IDF1-3 with a polarity of 0.7. The word *troep* has a $\Delta TF\text{-}IDF$ value of -8.24 which is in the range $[-10. - 5]$. Therefore, the word *troep* is added to lexicons TF-IDF1-2 and TF-IDF1-3 with a polarity of -0.7. Finally, the word *hel* has a $\Delta TF\text{-}IDF$ value of -11.34 which is in the range $[-\infty, -10]$. Therefore, the word *hel* is added to lexicons TF-IDF1-1, TF-IDF1-2 and TF-IDF1-3 with a polarity of -1.

Adjusting polarity of sentiment words

Next to adding words to the sentiment lexicon that were not yet present in the original lexicon of Pattern, words that had a positive or negative polarity in Pattern, but the TF-IDF outcome indicated the opposite polarity, are changed according to the conditions specified in table 6 as well. The following table shows some examples of words of which the polarity is changed.

Word	Original polarity	New polarity	Word	Original polarity	New polarity
warm	0.6	-0.5	effect	0.4	-0.5
nadeel	-0.5	0.5	nood	-0.5	0.5

Table 8: Words of which polarity is changed

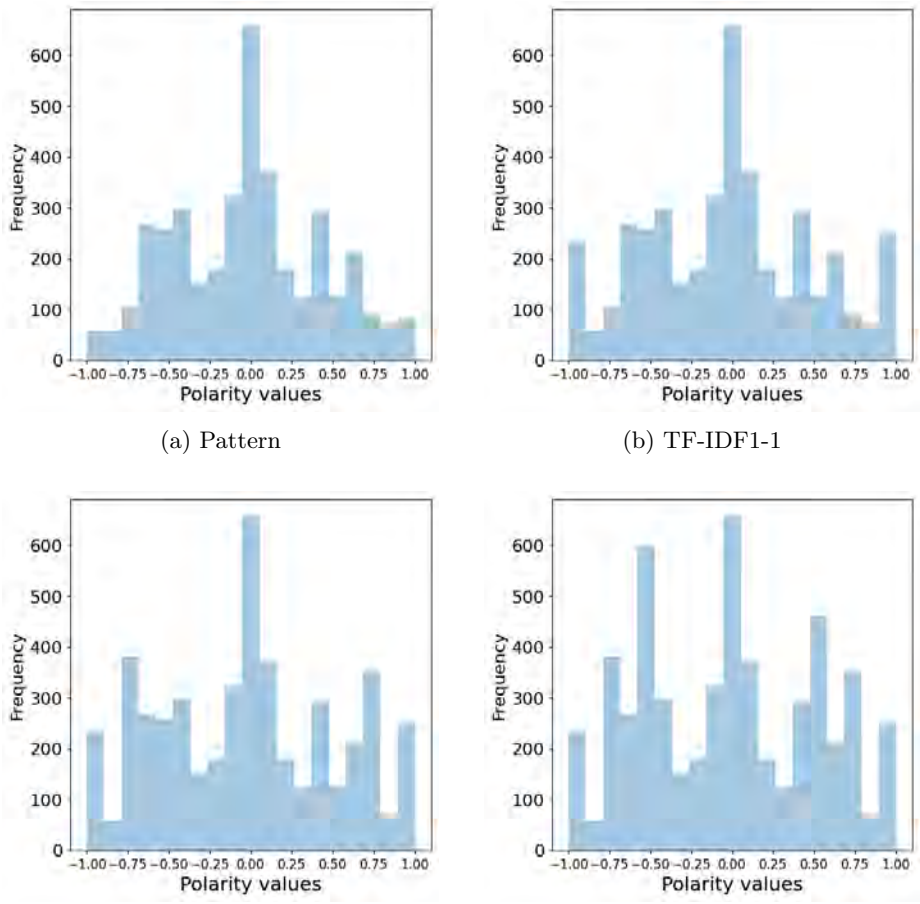
Resulting Lexicons

An overview of the original lexicon of Pattern and the lexicons that are constructed based on TF-IDF computations on dataset 1 and dataset 2 can be seen in table 9. Hereby, the first number of the lexicon indicates whether the lexicon is based on dataset 1 or dataset 2. The second number of the lexicon indicates according to which conditions words are added to the lexicon.

Lexicon	Number of words	Lexicon	Number of Words
Pattern	3918	Pattern	3918
TF-IDF1-1	4264	TF-IDF2-1	4087
TF-IDF1-2	4802	TF-IDF2-2	4472
TF-IDF1-3	5478	TF-IDF2-3	5132

Table 9: Overview lexicons

The polarity distributions of the lexicons that are based on dataset 1 can be seen in figure 2.



(a) Pattern

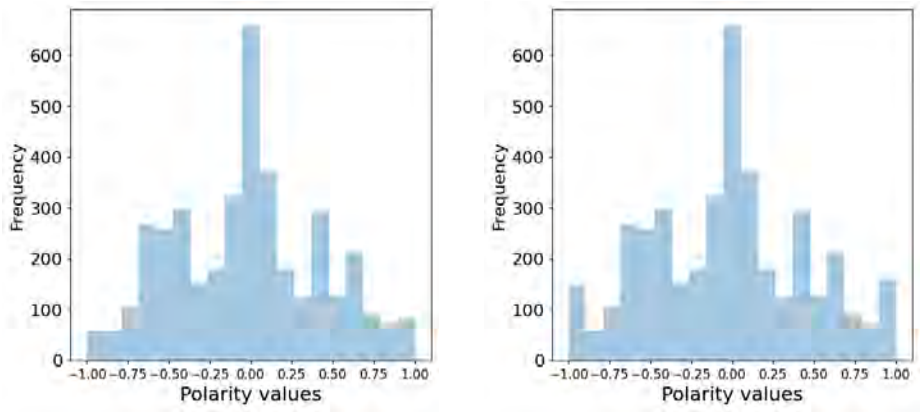
(b) TF-IDF1-1

(c) TF-IDF1-2

(d) TF-IDF1-3

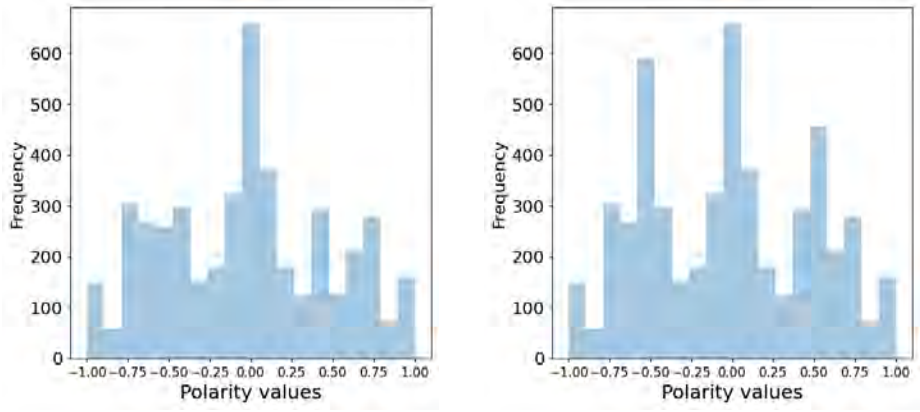
Figure 2: Polarity distributions lexicon based on dataset 1

The polarity distributions of the lexicons based on dataset 2 can be seen in figure 3.



(a) Pattern

(b) TF-IDF2-1



(c) TF-IDF2-2

(d) TF-IDF2-3

Figure 3: Polarity distributions lexicons based on dataset 2

Customized lexicon

Finally, an additional lexicon has been constructed which is called the *customized lexicon*. As mentioned in the introduction, in the healthcare domain there is a desire for transparent and intuitive models. The lexicons which are constructed by means of TF-IDF methods contain words which derive their polarity value from their occurrences in positive and negative reviews. Therefore, many words may receive a polarity which are not intuitive to doctors. Some examples of words which received a polarity which may not be intuitive for humans can be seen in table 10.

Word	Polarity	Word	Polarity
penis	1	wist	-0.5
aangezien	0.7	beginnen	-0.7
injecties	0.5	doctor:	-1

Table 10: Words with inexplicable polarity

Due to the desire for a transparent and intuitive sentiment analysis model, a customized lexicon has been constructed by adding only a small selection of words to the original lexicon of Pattern. This customized lexicon is made by adding 108 sentiment bearing words to the original Pattern lexicon. The added words are selected, and given a polarity value, by a human annotator. The majority of the words that are added to the customized lexicon are medical terms. The list of words that were added to the original lexicon of Pattern to construct the customized lexicon can be seen in appendix B.

3.5 Classifiers

As mentioned in the literature study, to handle more complex grammatical relations in text data, classifier will be incorporated into the lexicon based models. The following section will describe the use of the modifier classifier, the negation classifier and the emoticon classifier.

Modifier classifier

The literature study revealed that modifiers can adjust the polarity of sentiment words. Modifiers can be distinguished into diminishers and amplifiers. Diminishers are modifiers with an intensity value below 1 while amplifiers are modifiers with an intensity value higher than 1. In this research, modifiers have to be the preceding word of a sentiment word to be detected by the lexicon based model. If a modifier is detected, then the polarity of the neighboring sentiment word is computed as follows:

$$modified\ pol_{\text{score}}(s) = intensity_{\text{score}}(m) * original\ pol_{\text{score}}(s)\ if\ (m\ \epsilon\ M\ and\ s\ \epsilon\ S) \tag{11}$$

In this formula, the modified polarity of the sentiment word (s) is calculated by multiplying the intensity score of the modifier (m) with the original polarity of the sentiment word. Moreover, S is a set containing all sentiment words and M is a set containing all modifiers.

As mentioned before, every entry in the lexicon of Pattern contains a value for the intensity. These values will be used in this research for the modification classifier. Table 11 shows the intensity values of some amplifiers which are present in the sentiment lexicon of Pattern and 12 shows the intensity values of some diminishers.

Amplifier	Intensity	Amplifier	Intensity
extreem	2	ontzettend	1.6
enorm	1.9	heel	1.5
super	1.7	veel	1.3

Table 11: Partial list of amplifiers

Diminisher	Intensity	Diminisher	Intensity
redelijk	0.9	minder	0.5
middelmatig	0.9	beetje	0.4
tamelijk	0.6	amper	0.3

Table 12: Partial list of diminishers

The following example sentence shows how the modifier classifier is applied:

Het medicijn werkt heel goed.

In this example sentence, the word *heel* serves as an amplifier for the word *goed*. The intensity value of the word *heel* is 1.6 and the polarity value of the word *goed* is 0.55. Therefore, the modified polarity is $1.6 * 0.5 = 0.88$

Finally, some modifiers have been added to the the lexicons after human inspection. These modifiers, with corresponding intensity values can be seen in table 13.

Modifier	Intensity	Modifer	Intensity
meer	1.3	minder	0.3
veel	1.3	weinig	0.4
tegen	0	iets	0.7
beetje	0.4	enige	0.4
zo	1.5		

Table 13: Modifiers added to lexicons

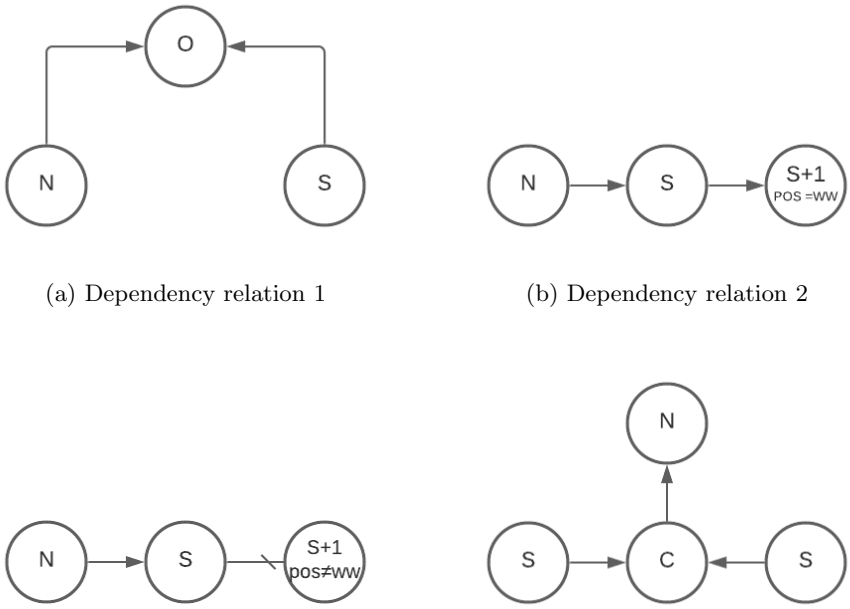
Note in table 13 that the intensity value of the word *tegen* is set at 0. The reason for this is that *tegen* is frequently used in sentences in which users describe for which (*tegen*) complaint they received medication. Therefore, the word that follows *tegen* is often not a description of their current state but a state which users wanted to change. The following example sentence will clarify this:

Ik kreeg dit middel tegen buikpijn.

Negation classifier

The second classifier which will be incorporated into the lexicons based models is the negation classifier. At first, based on manually inspecting which words were used for negating sentiment words, the following list of negation terms was constructed: *niet, geen, nooit, nergens, niets, niks, weg, over, verdwijnen, verhelpen, opgelost, voorbij* en *afnemen.*

Thereafter, 50 sentences containing negation of a sentiment word were parsed to determine the dependency relations between the sentiment word which was negated and the negation word itself. These dependency relations were incorporated into the model and can be seen below in figure 4.



(a) Dependency relation 1

(b) Dependency relation 2

(c) Dependency relation 3

(d) Dependency relation 4

Figure 4: Possible dependency relations between negation word and sentiment word

An example sentence for dependency relation 1 is: *de buikpijn is verdwenen.* In this sentence the negation word (N) is *verdwenen* and the sentiment word (S) is *buikpijn*. Both the sentiment word and the negation word depend on another word (O) which equals *is* in the example sentence.

An example sentence for dependency relation 2 is: *ik heb geen buikpijn gehad.* In this sentence the negation word (N) is *geen* and the sentiment word (S) is *buikpijn*. In this example sentence, the negation word depends on the sentiment word and the sentiment word depends on the word after the sentiment word (S+1), which equals *gehad*. Moreover, the word after the sentiment word has to be a verb.

An example sentence for dependency relation 3 is: *ik heb geen buikpijn op het moment.* In this sentence the negation word (N) is *geen* and the sentiment word (S) is *buikpijn*. In the example sentence, the negation word depends on the sentiment word. Moreover, in case the word after the sentiment word is a verb, the sentiment word may not depend on the word directly after the sentiment word. Without this addition, some phrases were unjustified recognized as negations.

Dependency relation 4 is slightly different than the three previous dependency relations. The three previous dependency relations negated only one sentiment word. However, in dependency relation 4, two sentiment words are negated. An example sentence for dependency relation 4 is: *ik heb geen jeuk of pijn* In this sentence, both the sentiment words (S) *jeuk* and *pijn* depend on the conjunctor

(C) *of*. Moreover, the conjuctor (C) *of* depend on the negation word(N) *geen*.

Modification of negated words

The polarity of negated sentiment words has to be modified. This is done by applying the polarity shift method, which was reviewed in the literature study, with some adjustments. First of all, the polarity of sentiment words with an original polarity between -0.3 and 0.3, is set to 0. These words contained already little sentiment and this sentiment vanishes even more after negation. Consider the following the example sentence:

Ik heb niet ademloos naar de uitleg van de arts geluisterd, het was een warrig verhaal.

In the previous sentence, the word *ademloos* has a polarity of 0.2. However, after negation by means of the word *niet*, the polarity of the phrase *niet ademloos* is set to 0.

Secondly, the polarity of sentiment words with an original polarity of more than 0.3 are reduced with 0.9 after negation. Consider the following example sentence:

Ik ben niet blij met de zorg die ik heb ontvangen.

In this example sentence, the sentiment word *blij* has a polarity of 0.6. However, after negation by means of the sentiment word *niet*, the polarity of the phrase *niet blij* is set to $0.6 - 0.9 = -0.3$.

Finally, the polarity of sentiment words with an original polarity of less than -0.3 are increased with 1.2. Consider the following example sentence:

De buikpijn is verdwenen.

In this example sentence, the sentiment word *buikpijn* has a polarity of -0.5. However, after negation by means of the sentiment word *verdwenen*, the polarity of the phrase *klachten verdwenen* is set to $-0.5 + 1.2 = 0.7$

The values 0.9 and 1.2 were chosen because they resulted in the best performance on the validation set. An overview of the negation rules can be seen in table 14:

Original polarity	Polarity after negation
$(-0.3, 0.3)$	0
$[-1, -0.3]$	Original polarity+ 1.2
$[0.3, 1]$	Original polarity -0.9

Table 14: Polarity modification of negated words

Emoticon classifier

6 manually annotated emoticons are added to the domain specific lexicons. The polarity of the emoticons range from -1 to 1 and mainly emoticons which contain strong sentiment (either positive or negative) are added. The emoticons that were included in the lexicons can be seen in table 15. Because the emoticons are added to all domain specific lexicons, and will therefore be used by all lexicon based models, the term emoticon classifier will not be used from here on.

Emoticon	Polarity	Emoticon	Polarity
:-))	0.9	:-(	-0.7
:)	0.7	:(	-0.7
:-]	0.7	D-':	-0.9

Table 15: Emoticons that are added to lexicons

Customized rules

Next to all classifiers, some semantic rules, aimed at specific linguistic structures, are added to the lexicon based model. These rules were inspired by the study of Appel et al (2016) and can be seen in table 13.

Rule number	**Description:**	**Example sentence**
R1	If sentence contains *maar*, disregard all previous sentiment expressed before the word *maar*	*Ik voelde mij slecht maar nu veel beter.*
R2	If sentence contains *behalve*, disregard all sentiment expressed after the word *behalve*	*Iedereen is tevreden over dit medicijn behalve als je gek bent.*
R3	If sentence contains *ondanks*, disregard all sentiment expressed after the word *ondanks*	*Dit medicijn is verschrikkelijk ondanks dat het in eerste instantie goed werkte.*

Table 16: Customized rules

3.6 Classification of reviews

For the lexicon based model, two different classifiers for determining the class (positive or negative) of a review are used: the sentence classifier and the aggregated sentiment classifier. The sentence classifier and the aggregated sentiment classifier are inspired by other studies which used these classifiers [23, 5]. The two classification approaches will be explained and an example will clarify how they work.

Sentence classifier

The sentence classifier classifies reviews based on the number of positive and negative sentences. A sentence is considered positive when its polarity is above the threshold value of 0.3. For negative sentences, this threshold value is set at -0.3. A review containing more positive than negative sentences is considered positive. A review containing more negative than positive sentences is considered negative. In case a reviews contains an equal number of positive and negative sentences, the polarity class of the sentence with the strongest sentiment value determines the polarity.

Aggregated sentiment classifier

The aggregated sentiment classifier classifies reviews based on the aggregated sentiment value. The aggregated sentiment value is computed by adding up all the polarity values of sentiment words and phrases in a review. An aggregated sentiment value above 0 results in the the review being classified as positive. An aggregated sentiment value below 0 results in the review being classified as negative. The following example review will show how both classifiers work:

pff, de medicijnen zijn verschrikkelijk! mijn plezier in het leven is weg. Mijn nek voelt wel weer goed.

The sentence classifier would split up this review in 3 sentences based on the punctuation symbols. Each sentence would receive a polarity value. The results of these steps can be seen in table 17.

Thereafter, the sentence classifier counts the number of negative sentences (2 in this example) and positive sentences (1 in this example). Because there are more negative sentences than positive sentences in this review, the sentence classifier classifies the review as negative.

Sentence number	Sentence	Words/phrases that carry sentiment	Polarity sentence
Sentence 1	*pff, de medicijnen zijn verschrikkelijk!*	verschrikkelijk	-0.7
Sentence 2	*mijn plezier in het leven is weg.*	plezier weg	-0.5
Sentence 3	*mijn nek voelt wel weer goed.*	goed	0.7

Table 17: Example sentence classifier

On the other hand, the aggregated sentiment classifier will add all sentiment expressed in the review. Thereby, not taking into consideration in which sentence the sentiment is expressed. The aggregated sentiment value of the example review would be $-0.7 - 0.5 + 0.7 = -0.5$. Because the aggregated sentiment value is below 0, the aggregated sentiment value would classify this example review negative as well.

3.7 Algorithm lexicon based model

The following section will describe the algorithm which supports the lexicon based sentiment analysis models. Obviously, the algorithm differs based on whether the classifiers (modification and negation) are incorporated into the model and based on which classification method is used (sentence classifier or aggregated sentiment classifier). In the step-by-step procedure described below, all classifiers are incorporated into the model and the sentence classifier is used.

Therefore, the lexicon based model combines the domain specific lexicon, the classifiers for negation and modification and the customized rules for assigning sentiment to reviews. The following steps will show how the sentiment of a review is determined:

Step 1

Set value *sentiment_polarity* to 0 and create list *polarities_sentences*.

Step 2

Look for sentiment word in sentence. In case a sentiment word is found, derive the polarity from the domain specific lexicon. Thereafter:

- Check whether sentiment word is negated. In case this is true, compute the polarity of the sentiment word after negation and add to *sentiment_polarity*.

- Check whether sentiment word is modified. In case this is true, compute the polarity of the sentiment word after modification and add to *sentiment_polarity*.

- Check whether one of the customized rules apply to this sentence. in case this this true, compute the polarity after applying the customized rule.

Step 3

Look for end of sentence symbols. As mentioned before, a point, exclamation mark and a question mark indicate the end of sentence. If one of these characters is found, the value of sentiment_*polarity* is stored in *polarities_sentences* and *sentiment_polarity* is set to 0 again.

Step 4

Repeat steps 2-3 until the end of the review is reached.

Step 5

Classify the review based on the values in the list *polarities_sentences*.

For clarity, step 1-4 of the algorithm can be seen in figure 5.

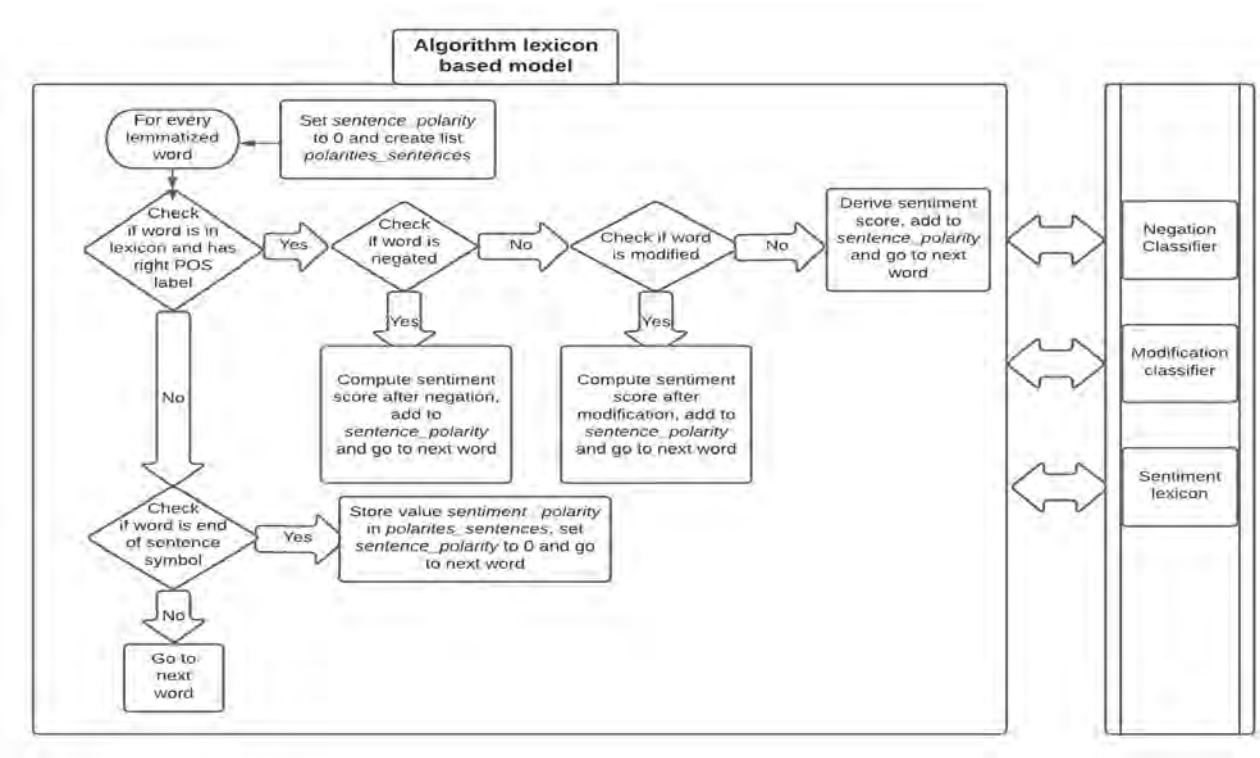


Figure 5: Framework algorithm

3.8 Machine learning models

Based on the literature study, three kind of machine learning models will be used for sentiment analysis in this research: support vector machine models, naive Bayes models and decision tree models. These three machine learning techniques perform well on sentiment analysis tasks and do not require a lot of the training data. The models will be trained on training sets of different magnitudes to determine how this impacts the performance. For dataset 1, the machine learning models will be trained on 250, 500, 1000 and 3200 reviews. For dataset 2, the machine learning models will be trained on 250, 500, 1000 and 2000 reviews. All training sets will contain the same amount of positive and negative reviews.

The machine learning models will use different kind of features. At first, models will be constructed that use the unigrams in the reviews as features. As mentioned in the literature study, unigrams are a N-gram of 1 and perform generally well on sentiment analysis tasks. Secondly, models will be constructed that use only the lemmatized form of words in the reviews that occur at least 3 times. By using lemmas that occur at least 3 times, the aim is to reduce the sparsity in the data and, therefore, improve the performance of the machine learning model.

3.9 Evaluation

In this section, the evaluation metrics that are used in this research will be reviewed. The following evaluation metrics are used: precision, recall, and F-score For all these metrics, the performance on the positive class and the performance on the negative class will be reviewed. The formulas can be seen below:

$$Precision\ positive\ class = \frac{tp}{tp + fp} \tag{12}$$

$$Precision\ negative\ class\ = \frac{tn}{tn + fn} \tag{13}$$

$$Recall\ positive\ class\ = \frac{tp}{tp + fn} \tag{14}$$

$$Recall\ negative\ class\ = \frac{tn}{tn + fp} \tag{15}$$

$$Total\ recall = \frac{tp}{tp + fn}\ +\ \frac{tn}{tn + fp} \tag{16}$$

$$F\text{-}score\ positive\ class = \frac{2(precision\ positive\ class)(recall\ positive\ class)}{precision\ positive\ class + recall\ positive\ class} \tag{17}$$

$$F\text{-}score\ negative\ class = \frac{2(precision\ negative\ class)(recall\ negative\ class)}{precision\ negative\ class + recall\ negative\ class} \tag{18}$$

In these formulas, *tp* stands for *true positive*, *tn* stands for *true negative*, *fp* stands for *false positive* and *fn* stands for false negative.

In the *results* section, the only evaluation metrics which will be shown are the recall for the positive class, the recall for the negative class and the total recall. These evaluation metrics are chosen based on similar studies regarding the topic of sentiment analysis. Appendix C will display the results including all evaluation metrics:

4 Exploratory data analysis

In this section, the two datasets that are used for this research will be explored. As mentioned before, the dataset that originates from the website www.meldpuntmedicijnen is called dataset 1 and the dataset that originates from www.mijnmedicijn.nl is called dataset 2. Most of the reviews in dataset 1 describe the experience a user had with a medicine which aimed to cure *physical* complaints. Some examples include medicines which aim to relieve the pain, medicines which aim to help with regard to respiratory problems and medicines which aim to help with regard to digestion issues. On the other hand, the reviews in dataset 2 describe the experience a user had with a medicine which aimed to cure *psychological* complaints. Some example include medicines which aim to help with regard to depression and panic attacks.

4.1 Description datasets

Dataset 1 consists of 8500 medical reviews. Every review contains 2 columns, namely: *review* and *mood*. Every user that writes a review has the possibility to select 4 moods. Moods 1 and 2 are displayed by a sad looking emoticon, mood 3 is displayed by an emoticon which looks neutral and mood 4 is displayed by a happy looking emoticon. The emoticons can be seen in figure 6.

Figure 6: Emoticons to display mood in dataset 1. The moods are displayed in ascending order

Dataset 2 contains 5438 reviews. This dataset contains multiple columns which quantify the experience the user had with regard to the medicine. For the scope of this research, only the columns *reviews* and *mood* will be taken into account. Every user has the possibility to give a number from 1 to 5 in the column *mood*. Hereby, the numbers 1 and 2 indicate a negative experience, 3 a neutral experience and 4 and 5 indicate a positive experience

The distribution of the moods for both datasets can be seen in figures 7 and 8.

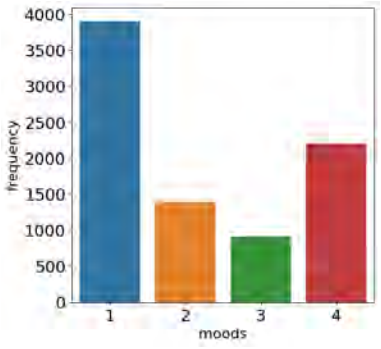


Figure 7: Moods dataset 1

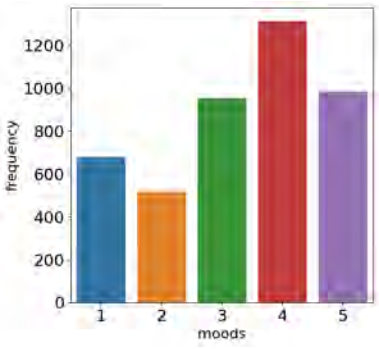


Figure 8: Moods dataset 2

To develop more intuition of the reviews, 2 reviews per dataset will be shown below. Per dataset, 1 positive review will be shown and 1 negative review will be shown.

Dataset 1, positive: *"'ik gebruik de tramadol al enkele jaren (soms stop ik tussendoor). het helpt goed tegen de pijn waardoor mijn lichaam even kan ontspannen. ik heb geen last van bijwerkingen."*

Dataset 1, negative: *'waardeloos! het helpt totaal niet, terwijl rhinocort bij mij altijd heeft gewerkt. sinds ik budesonide van sandoz gebruik is het alsof ik oud bloemenwater opsnuif en heb continu een loopneus. slecht medicijn. met rhinocort nooit meer last van een waterige neus. nu weer continu last."*

Dataset 2, positive: *'ik heb dit middel 3 jaar lang gebruikt op 20 mg na herhaaldelijke depressies en een genetische gevoeligheid voor depressie waardoor het steeds terug zal komen. onlangs 4 maanden gestopt omdat mijn lichaam niet meer reageerde op het medicijn en een verhoging tot 40 mg tot enorme impotentieproblemen leidde, ook was ik erg aangekomen. nu na 4 maanden weer begonnen omdat ik merk dat ik de stabiliteit nodig heb, dan wel in deze periode van mijn leven kan gebruiken. wat ik nu merk is de hoofdpijn die ik altijd heb bij het opstarten, een vrij heftige schelle pijn die als het goed is na 1 a 2 weken weer weggaat. het rare is dat ik nooit andere bijwerkingen heb dus voor mij is dit middel ideaal, ik word ook niet raar, de situatie verergert niet, niks, behalve dan die vervelende hoofdpijn maar dat overleef ik wel. bij de eerste pil had ik een wat droge mond.'*

Dataset 2, negative: *"leek in eerste instantie goed te gaan, ik genoot weer van het leven maar het maakte een ommezwaai in volledig out of control geld uitgeven, de drugs opzoeken en living on thé edge. mijn relatie stond er niet bijster goed voor maar zo'n persoon als dat ik nu zag had ik niet eerder ontdekt. niet onprettig maar wel een slag in mijn bankrekening. was hier eerder al vatbaar voor maar het nam me volledig in mijn bezit. gek genoeg: ik vond het wel prima na zoveel gedonder. 'ben er nu maar mee gestopt om erger te voorkomen. je kunt jezelf wel geweldig voelen maar is het wel echt? "*

4.2 Data cleaning

In this research, sentiment analysis is applied on 2 different classes: positive and negative. Therefore, neutral reviews are removed from the datasets. For both datasets, reviews with a value *3* for the variable *mood* are considered to be neutral. Furthermore, for dataset 1, reviews with the values *1* or *2* for the variable *mood* are merged and form the negative class. The reviews containing a value *4* for the variable *mood* form the positive class. For dataset 2, the reviews containing the values *1* or *2* for the variable *mood* are merged and form the negative class. The reviews containing the values *4* and *5* for the variable *mood* are merged and form the positive class. The results of merging the data and the removal of the neutral reviews can be seen in figures 9 and 10.

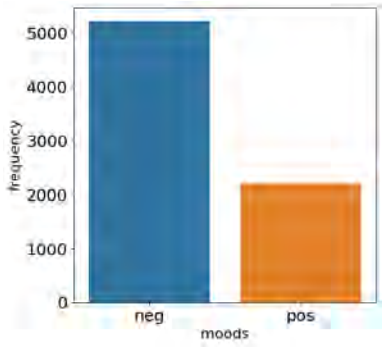


Figure 9: Moods after merging dataset 1

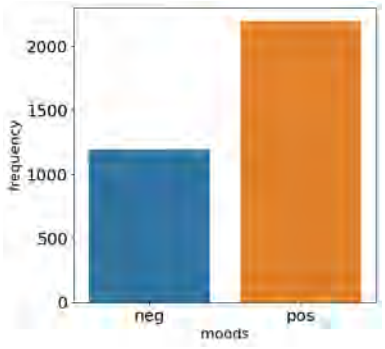


Figure 10: Moods after merging dataset 2

After merging the reviews, dataset 1 contains 5169 negative reviews and 2311 positive reviews. Dataset 2 contains 1282 negative reviews and 2304 positive reviews.

4.3 Length reviews

In this section, some plots will be made to have a better understanding of the data. At first, the distribution of the length of the reviews per dataset is illustrated by two boxplots which can be seen in figure 11.

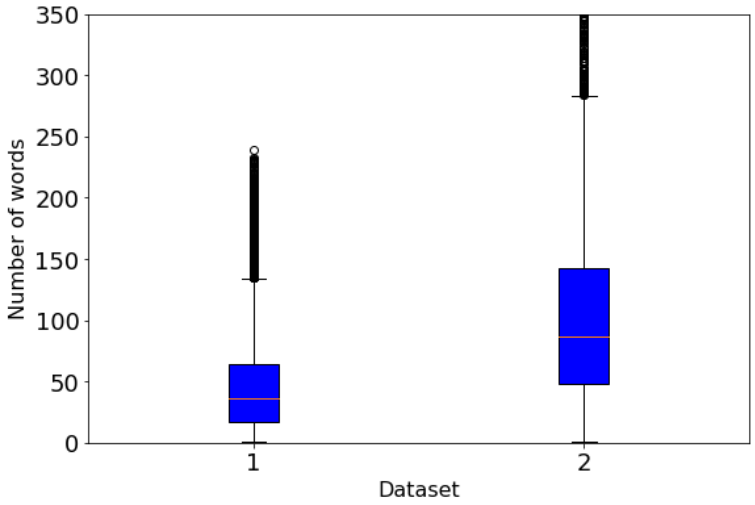


Figure 11: Length reviews per dataset

As can be seen in figure 11, reviews in dataset 2 (median 87) tend to be longer than the reviews in dataset 1 (median 36). Moreover, there is more variation in the length of the reviews in dataset 2 in comparison with the reviews in dataset 1.

Moreover, after having concluded that there are differences in the distribution of the length of the reviews between the two datasets, the distribution of the length of the reviews of the different classes (positive vs negative) will be shown. At first, figure 12 shows the distribution of the length of the reviews in dataset 1 for both classes. Secondly, figure 13 shows the distribution of the length of the reviews in dataset 2 for both classes.

As can be seen in figures 12 and 13, the differences between the distribution of the length of the review per class does not differ very much for both datasets. For dataset 1, the median of length of the reviews for the negative class is 35 and for the positive class it is 37. For dataset 2, the median of length of the reviews for the negative class is 80 and for the positive class it is 91. As mentioned before in section 3.4, the differences between the *average* length of the reviews per class is taken into consideration for the TF-IDF computations.

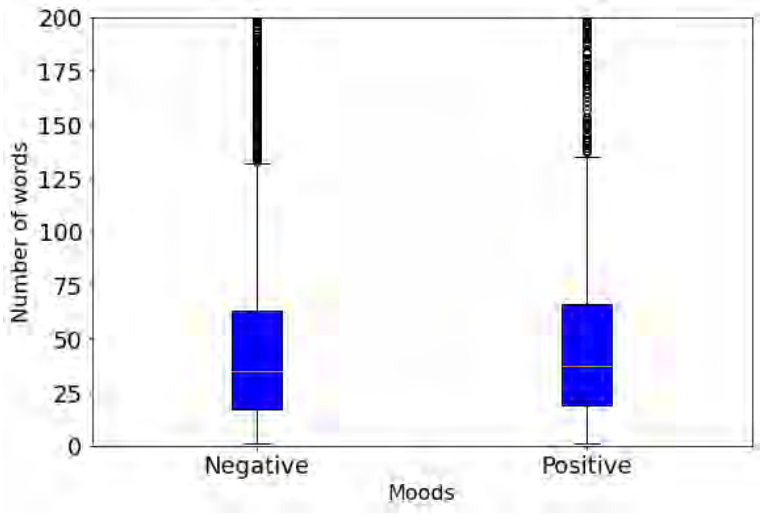


Figure 12: Length reviews dataset 1 per class

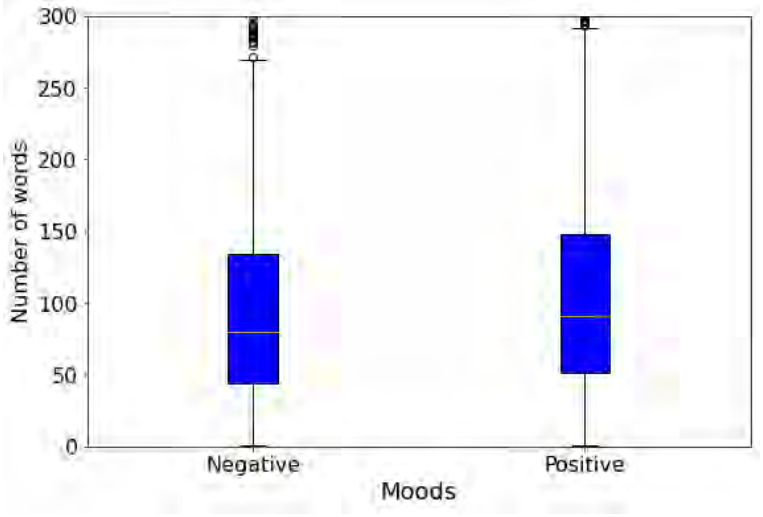


Figure 13: Length reviews dataset 2 per class

4.4 Training data and test data

This section will review how many reviews are used for training the models and how many reviews are used for testing the models for both datasets.

Dataset 1

After merging, dataset 1 contains 5169 negative reviews and 2311 positive reviews. 1600 positive reviews and 1600 negative reviews are used for the TF-IDF computations for constructing the domain specific lexicons. These 3200 reviews

can be seen as the training data for dataset 1. In the results section, the models are tested on 400 positive reviews and 400 negative reviews which do not appear in the training data. These 800 reviews can be seen as the test data for the lexicon based models for dataset 1.

The machine learning models for dataset 1 are trained on the training data from dataset 1. The models are trained on 250 reviews, 500 reviews, 1000 reviews and 3200 reviews to determine what the influence of the training size is on the performance of the models. Moreover, the machine learning models are tested on the same test data as on which the lexicon based models are tested.

Dataset 2

After merging, dataset 2 contains 1282 negative reviews and 2314 positive reviews. 1000 positive reviews and 1102 negative reviews are used for the TF-IDF computations for constructing the domain specific lexicons. These 2102 reviews can be seen as the training data for dataset 2. In the results section, the models are tested on 200 positive reviews and 200 negative reviews which do not appear in the training data. These 400 reviews can be seen as the test data for dataset 2.

The machine learning models for dataset 2 are trained on the training data from dataset 2. The models are trained on 250 reviews, 500 reviews, 1000 reviews and 2000 reviews to determine what the influence of the training size is on the performance of the models. Moreover, the machine learning models are tested on the same test data as on which the lexicon based models are tested.

5 Results

In this section, the results of the lexicon based models and the machine learning models are displayed. As discussed in section 3.9, this section will only show the performance of the models according to the *total recall*, *recall positive class* and the *recall negative class* evaluation metrics. The performance of the models according to the other evaluation metrics can be seen in appendix C. First, the results of the lexicon based models will be shown in section 5.1. Thereafter, the results obtained with the machine learning models are shown in section 5.2. Finally, an overview of all results and a comparison between the lexicon based models and the machine learning models will be presented in section 5.3.

5.1 Lexicon based model

In this section, the results of the lexicon based model are shown. This section is divided in 2 subsections. Subection 5.1.1 will review the performance of the lexicon based models on the reviews in dataset 1. Subsection 5.1.2 will review the performance of the lexicon based models on the reviews in dataset 2. For both datasets, the performance of all lexicon based sentiment analysis models will be evaluated. This implies that lexicon based models which use lexicons that are constructed based on dataset 2 will be tested on the reviews in dataset 1. Moreover, lexicon based models which use lexicons that are constructed based on dataset 1 will be tested on the reviews in dataset 2. This is done to evaluate how well lexicons perform on a different domain than the domain on which they have been constructed. As described in the methodology section, TF-IDF1-1, TF-IDF1-2 and TF-IDF1-3 are based on TF-IDF computations on dataset 1 and TF-IDF2-1, TF-IDF2-2 and TF-IDF2-3 are based on TF-IDF computations on dataset 2. Moreover, the customized lexicon is constructed by means of a human annotator which added 108 sentiment words to the original lexicon of Pattern.

Section 3.5 described the construction of the modifier classifier and the negation classifier. In this research, all lexicon based models are tested which have *not* incorporated the classifiers for modification and negation and all lexicon based models are tested which have incorporated the classifiers for modification and negation. This is done to evaluate the contribution these classifiers have with regard to the performance of the lexicon based models. Moreover, as discussed in section 3.6, two different methods for classifying reviews are used: the sentence classifier and the aggregated sentiment classifier. For all lexicon based models the performance will be tested by using both classifiers.

5.1.1 Results on dataset 1

As mentioned before, in subsection 5.1.1, only the results on the reviews of dataset 1 are shown. For the first dataset, the performance of the models is tested on 800 reviews. These 800 reviews were excluded from the reviews that were used for the construction of the TF-IDF based lexicons. The 800 reviews consist of 400 positive reviews and 400 negative reviews. At first, the performance of the lexicon based models which use lexicons that are constructed based on dataset 1 will be shown. Thereafter, the results of the lexicon based models

which use lexicons that are constructed based on dataset 2 will be presented. Finally, the results of the models using the customized lexicon will be shown.

Performance lexicons based on dataset 1

At first, the performance of the sentiment analysis models which use lexicons that are constructed based on dataset 1 will be shown. As mentioned before, this will both be done by lexicons based models which have incorporated the classifiers for modification and negation and by lexicon based models which have not incorporated the classifiers for modification and negation. First, the lexicon based models which have not incorporated the classifiers for modification and negation will be shown. In figure 14 all models classify reviews by means of the sentence classifier. In figure 15, all models classify reviews by means of the aggregated sentiment classifier.

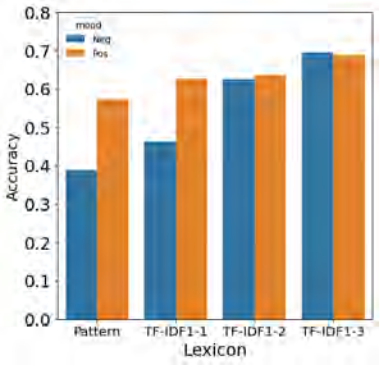


Figure 14: Recall lexicon based models using sentence classifier

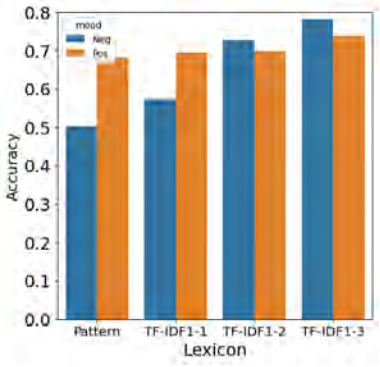


Figure 15: Recall lexicon based models using aggregated sentiment classifier

As can be seen in the figures above, the models using lexicon TF-IDF1-3 perform best, followed by models using lexicon TF-IDF1-2, TF-IDF1-1 and the original lexicon of Pattern. Furthermore, it could be noted that the models which use the aggregated sentiment classifier tend to perform better than the models which use the sentence classifier.

Next, the performance of the sentiment analysis models including the classifiers for modification and negation can be seen in the following figures. Figure 16 depicts the results when using the sentence classifier whereas figure 17 depicts the results when using the aggregated sentiment classifier.

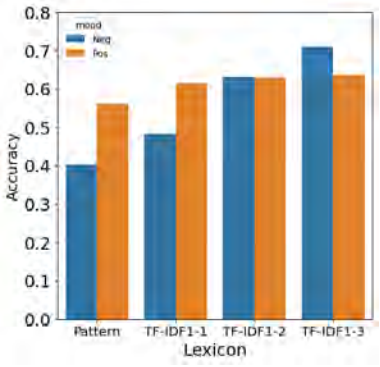


Figure 16: Recall based on sentence classifier

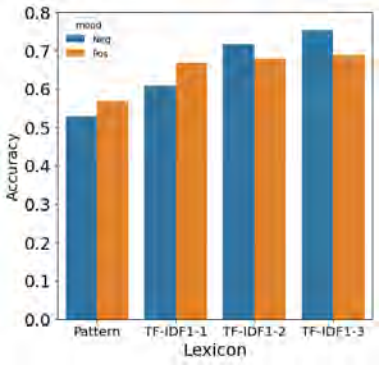


Figure 17: Recall based on aggregated sentiment classifier

As can be seen in the figures above, the models using lexicon TF-IDF1-3, again, perform best followed by the models using lexicon TF-IDF1-2, TF-IDF1-1 and the original lexicon of Pattern. Furthermore, the models using the classifier based on aggregated sentiment value perform better than the models which use the sentence classifier. Another interesting observation, which can be made after comparing figures 14 and 15 with figures 16 and 17, is that the sentiment analysis models containing classifiers for modification and negation perform more or less similar to the sentiment analysis models which do not contain these classifiers.

Performance lexicons based on dataset 2

All previous models contained lexicons which are constructed based on dataset 1. However, it is also interesting to see the performance of the lexicon based models which use lexicons that are constructed based on TF-IDF computations on dataset 2. Again, at first the results are shown of the models that do not contain the classifiers for modification and negation. These results can be seen in figures 18 and 19.

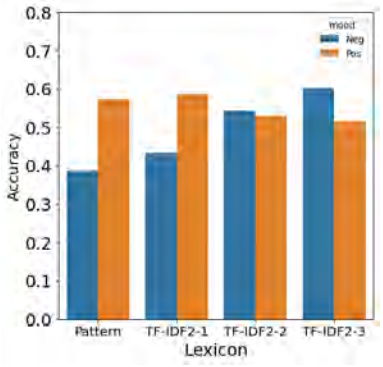


Figure 18: Recall based on sentence classifier

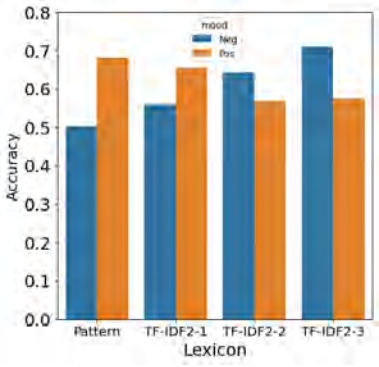


Figure 19: Recall based on aggregated sentiment classifier

As can be seen in figures 18 and 19, the lexicon based models using lexicons that are constructed based on dataset 2, perform worse compared to the lexicon based models which use lexicons that are constructed based on dataset 1. Moreover, it seems that the models which use lexicon TF-IDF2-3, TF-IDF2-2, TF-IDF2-1 and the original lexicon of Pattern perform relatively similar. Finally, the models which use the aggregated sentiment classifier perform better than the models which use the sentence classifier.

Finally, the results of the lexicon based models, using lexicons that are constructed based on dataset 2 and using classifiers for modification and negation can be seen in figures 20 and 21.

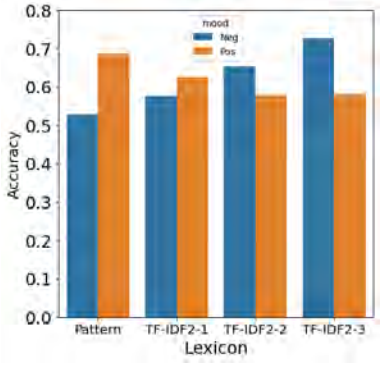


Figure 20: Recall based on sentence classifier

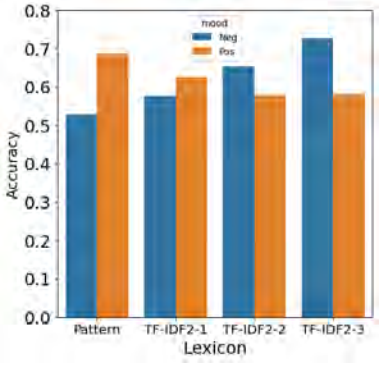


Figure 21: Recall based on aggregated sentiment classifier

After comparing figures 20 and 21 with figures 18 and 19, it seems like the performances of these models is quite similar. This implies that the impact of the modification and negation classifier is small for these models. Moreover, for the reviews in dataset 1, it can be concluded that the performance of the lexicon based models which use lexicons that are constructed based on dataset 1 perform significantly better than the lexicon based models which use lexicons that are based on dataset 2.

Performance customized lexicon

Finally, the performance of the lexicon based models using the customized lexicon can be seen in figure 22 and 23. In these figures, the abbreviation *NC* stand for *no classifiers* and this implies that the classifier for modification and negation are not incorporated into the model. Moreover, the abbreviation *WC* stand for *with classifiers* and this implies that the classifier for modification and negation are incorporated into the model. Again, in the figure on the left the sentence classifier is applied and in the figure on the right the aggregated sentiment classifier is applied.

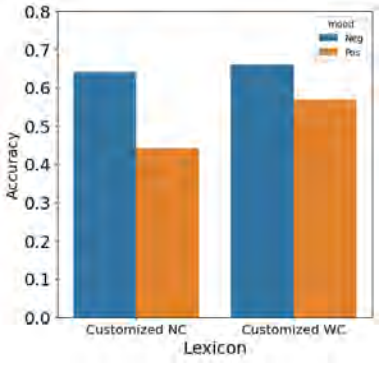


Figure 22: Recall based on sentence classifier

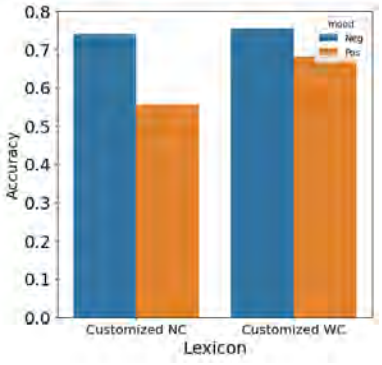


Figure 23: Recall based on aggregated sentiment classifier

As can be seen in figures 22 and 23, the models using classifiers perform significantly better than the models which do not use these classifiers. After the incorporation of the classifiers, the performance of the models on the positive reviews improves in particular. Moreover, the models which apply the aggregated sentiment classifier perform better than the models which use the sentence classifier. Finally, after comparing the performance of the lexicon based models which use the customized lexicon with models which use the lexicons that are constructed by means of TF-IDF computations, it can be concluded that the models which use the lexicons that are constructed based on TF-IDF computations perform better.

5.1.2 Results on dataset 2

For the second dataset, the performance of the sentiment analysis models is tested on 400 reviews. These 400 reviews were excluded from the reviews that were used for the construction of the TF-IDF based lexicons. The 400 reviews consist of 200 positive reviews and 200 negative reviews. The result of the sentiment analysis models on dataset 2 will be shown in the same manner as the results on dataset 1 were shown. This implies that at first, the results of the lexicon based models using lexicons which are based on dataset 2 (same lexicons as on which the models are tested) are used. Thereafter, the results of the lexicon based models using lexicons which are based on dataset 1 are shown. Finally, the results of the lexicon based models using the customized lexicon are

presented.

Performance lexicons based on dataset 2

At first, the results of the models using lexicons which are based on dataset 2 are shown. Figures 24 and 25 show the results of the lexicons based models models without the classifiers for modification and negation. Again, the figure on the left applies the sentence classifier and the figure on the right applies the aggregated sentiment classifier.

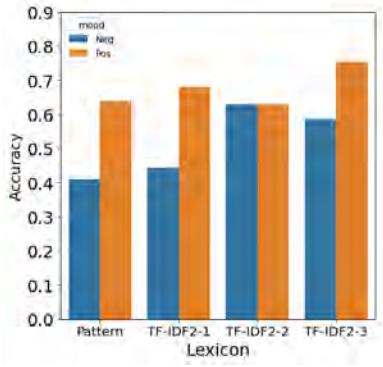


Figure 24: Recall based on sentence classifier

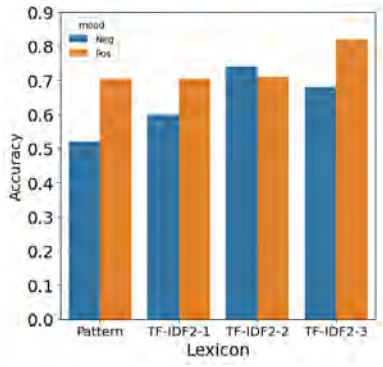


Figure 25: Recall based on aggregated sentiment classifier

As can be seen in figure 24 and 25, the models using lexicon TF-IDF2-3 perform best, followed by models using lexicon TF-IDF2-2, TF-IDF2-1 and the original lexicon of Pattern. Morever, the lexicon based models using the classifier based on the aggregated sentiment value performs better than the classifier based on the sentiment of sentences. Finally, it is worth noting that the sentiment analysis models, using every lexicon except of lexicon TF-IDF2-2, achieve a significant higher recall on positive reviews compared to negative reviews.

Next, the performance of the sentiment analysis models containing lexicons based on dataset 2 and having the classifier for modification and negation incorporated, can be seen in figures 26 and 27.

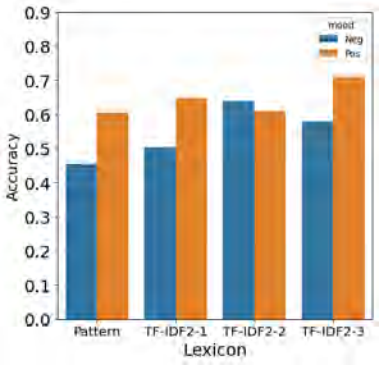


Figure 26: Recall based on sentence classifier

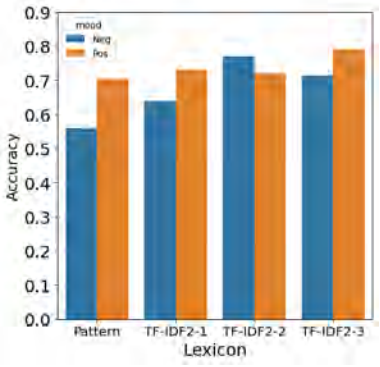


Figure 27: Recall based on aggregated sentiment classifier

As can be seen in figure 26 and 27, the models using lexicon TF-IDF2-3 and TF-IDF2-2 perform best, followed by the models using lexicon TF-IDF2-1 and the original lexicon of Pattern. Moreover, again, the lexicon based models using the classifier based on the aggregated sentiment value perform better than the models which use the sentence classifier. Finally, after comparing figures 26 and 27 with figures 24 and 25, it seems like the models containing classifiers for modification and negation perfrom more or less similar to the sentiment analysis models which do contain these classifiers.

Performance lexicons based on dataset 1

Next, the results of the sentiment analysis models containing lexicons that are based on TF-IDF computations on dataset 1 are shown. At first the results of the sentiment analysis models, which do not contain the classifiers for modification and negation, are shown and can be seen in figures 28 and 29.

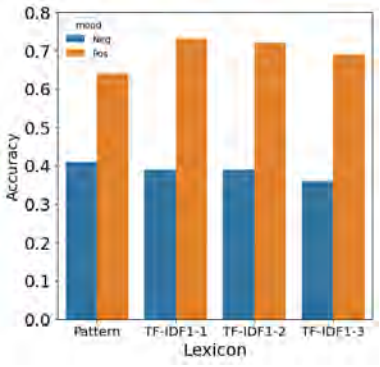


Figure 28: Recall based on sentence classifier

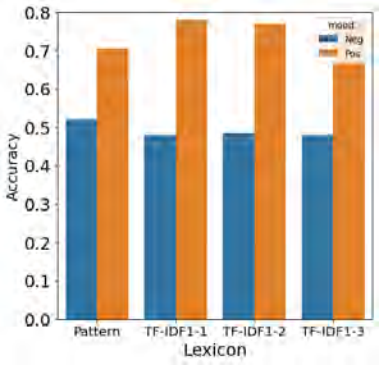


Figure 29: Recall based on aggregated sentiment classifier

As can be seen in figures 28 and 29, the performance of the sentiment analysis

models does not strongly depend on the lexicon that is used. This is derived from the fact that the performance of all models is quite similar. All models have a stronger tendency to classify reviews as positive. Therefore, the recall on positive reviews is significantly higher than the recall on negative reviews.

Finally, the results of the sentiment analysis models using lexicons based on dataset 1 and using all classifiers can be seen in figures 30 and 31.

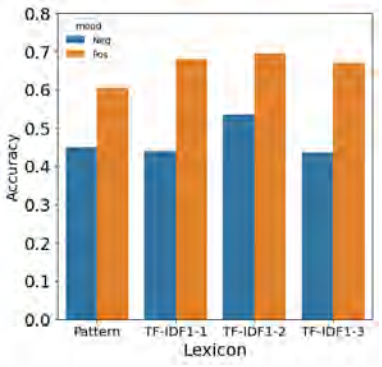


Figure 30: Recall based on sentence classifier

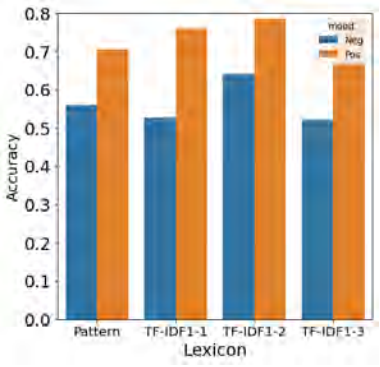


Figure 31: Recall based on aggregated sentiment classifier

As can be seen in figures 30 and 31, the performance of the sentiment analysis models is again relatively independent of the lexicons used. On top of that, All models have a stronger tendency to classify reviews as positive. Therefore, the recall on positive reviews is significantly higher than the recall on negative reviews. Moreover, after comparing figures 30 and 31 with figures 28 and 29, it can be derived that the influence of the classifiers for modification and negation is negligible for these models.

Performance customized lexicon

Finally, the performance of the lexicon based models using the customized lexicon can be seen in figures 32 and 33. In these figures, the abbreviation *NC* stand for *no classifiers* and this implies that the classifier for modification and negation are not incorporated into the model. Moreover, the abbreviation *WC* stand for *with classifiers* and this implies that the classifier for modification and negation are incorporated into the model. Again, in the figure on the left the sentence classifier is applied and in the figure on the right the aggregated sentiment classifier is applied.

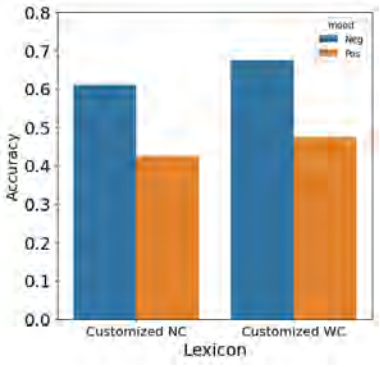


Figure 32: Recall based on sentence classifier classifier

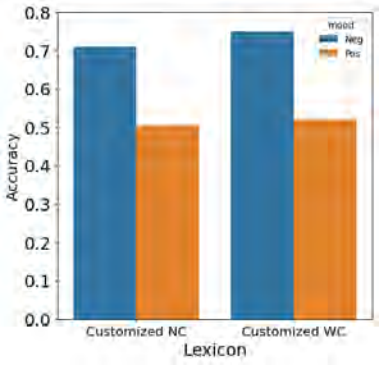


Figure 33: Recall based on aggregated sentiment classifier

As can be seen in figures 32 and 33, the classifiers for modification and negation slightly improve the performance of the models. Moreover, the models have the tendency to classify reviews as negative. Which results in a significant higher recall on the negative reviews compared to the positive reviews. Finally, again, the models using the aggregated sentiment classifier perform better than the models which use the sentence classifier.

5.2 Machine learning models

In the following section, the results of the machine learning models will be shown. As discussed in section 3.8, the machine learning techniques that are used are the following: decision tree models, naive Bayes models and support vector machine models.

Due to the different number of reviews that were available in both datasets, there are slight variations in the size of the training data. Dataset 1 has been trained on 250, 500, 1000 and 3200 reviews. Dataset 2 has been trained on 250, 500, 1000 and 2000 reviews. For both datasets, all training data consisted of an equal amount of positive and negative reviews. For all machine learning models, two different kind of features are used: unigrams and lemmas that appear at least 3 times in the training data. From now on called *unigrams* and *lemmas*. subsection 5.2.1 will show the results of the machine learning models on dataset 1. Subsection 5.2.2 will show the results of the machine learning models on dataset 2.

5.2.1 Results on dataset 1

At first, the results on dataset 1 will be shown. Figure 34 shows the performance of the decision tree models which use unigrams as features and figure 35 shows the performance of decision tree models which use lemmas as features.

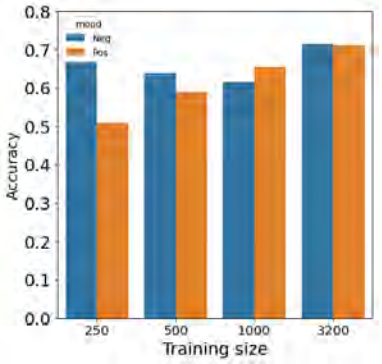


Figure 34: Recall of decision tree models using unigrams

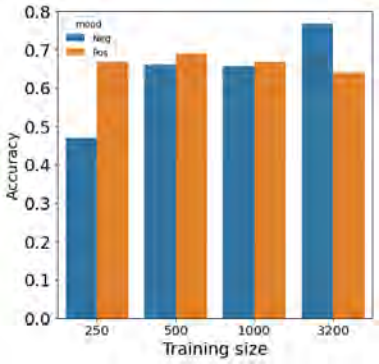


Figure 35: Recall of decision tree models using lemmas

As can be seen in figures 34 and 35, more training data results in a better performance on the positive reviews for the models which use unigrams as features. Moreover, more training data results in a better performance on the negative reviews for the models which lemmas as features.

Next, figures 36 and 37 show the performance of the naive Bayes models. Again, the figure on the left shows the performance of the models which use unigrams as features and the figure on the right shows the performance of the models which use lemmas as features.

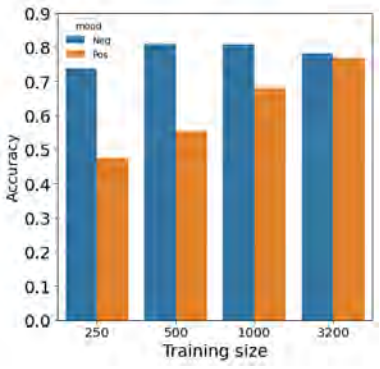


Figure 36: Recall of naive Bayes models using unigrams

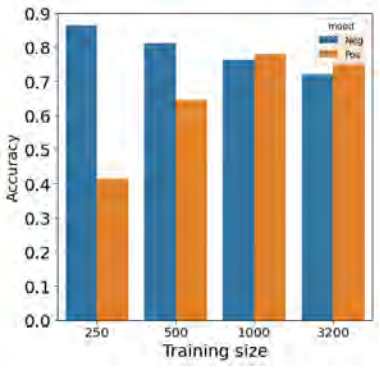


Figure 37: Recall of naive Bayes models using lemmas

As can be seen in figure 36 and 37, the models have a strong tendency to classify reviews as negative. Moreover, the total performance of the models, which can be derived by taking the recall for the positive class and the negative class into account, increases in case the models have been trained on more reviews.

Finally, figures 38 and figure 39 show the performance of the support vector machine models.

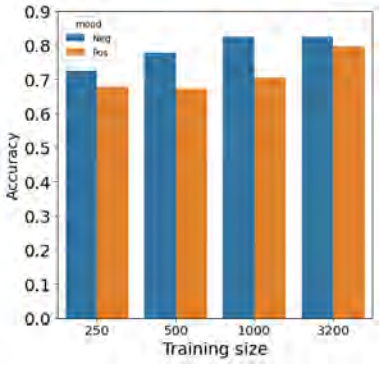


Figure 38: Recall of SVM models using unigrams

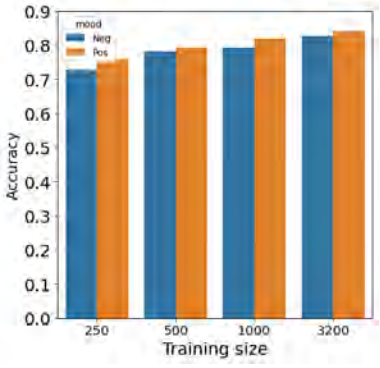


Figure 39: Recall of SVM models using lemmas

As can be seen in figure 38 and 39, the performance of the models is relatively good for all training sizes and more training results in a better performance of the models for both the positive class and the negative class.

5.2.2 Results on dataset 2

For dataset 2, the results of the machine learning models will be shown in a similar manner as the results of dataset 1 were shown. At first, the results of the decision tree models can be seen in figures 40 and 41.

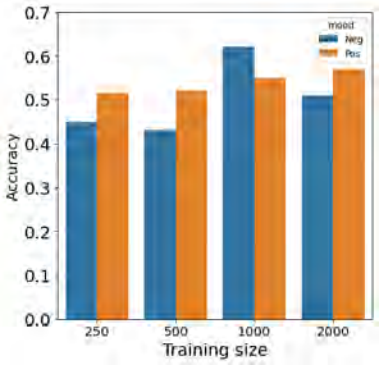


Figure 40: Recall of decision tree models using unigrams

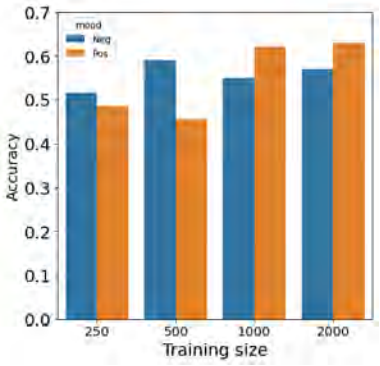


Figure 41: Recall of decision tree models using lemmas

As can be seen in figures 40 and 41, the performance of the models improves slightly in case they have been trained on a bigger training set. Moreover, the performance of the decision tree models is significantly lower than the performance of the decision tree models on dataset 1.

Next, in figures 42 and 43, the results of the naive Bayes models can be seen.

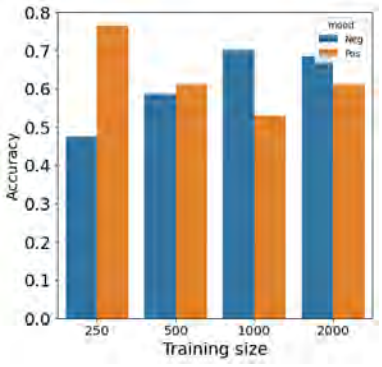


Figure 42: Recall of naive Bayes models using unigrams

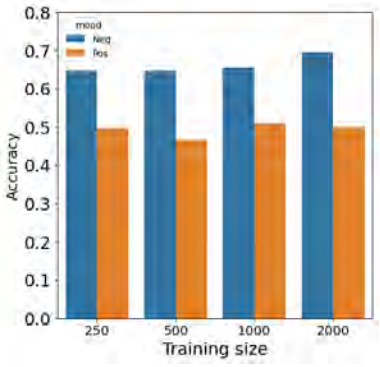


Figure 43: Recall of naive Bayes models using lemmas

As can be seen in figures 42 and 43, the naive Bayes models which use unigrams tend to perform better on the negative class in case they have been trained on more reviews. However, the performance on the positive reviews fluctuates in case the models have been trained on more reviews. Moreover, the performance of the naive Bayes which use lemmas as features is more or less similar for the different training sizes and the performance is significantly better on the nega-

tive reviews.

Finally, in figures 44 and 45 the results of the support vector machines can be seen.

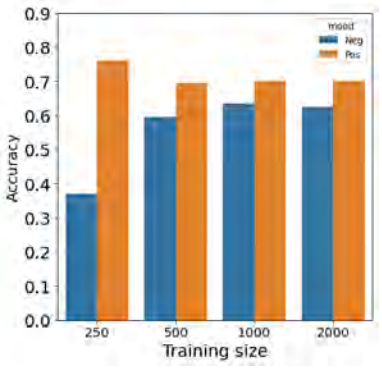


Figure 44: Recall of SVM models using unigrams

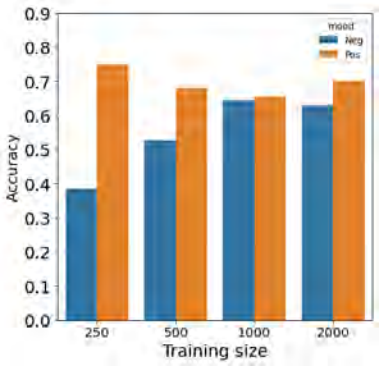


Figure 45: Recall of SVM models using lemmas

As can be seen in figures 44 and 45, increasing the training size results in a better recall for both kind of models on the negative class and results in a worse recall on the positive class. However, the total recall increases in case the models have been trained on more reviews.

5.3 Overview results

Finally, this section will provide an overview of the *total Recall* of all the models. The total recall can be computed by taking the average of the recall on the positive class and the negative class and was shown before in section 3.9 in the form of equation 16. Eventually, the performance of the lexicon based models and the machine learning models will be compared by means of the total recall evaluation metric.

At first, the performance of the lexicon based models and the machine learning models on dataset 1 will be compared by means of tables 18 and 19. Thereafter, the performance of the lexicon based models and the machine learning models on dataset 2 will be compared by means of tables 20 and 21. In these tables the following abbreviations have been used:

SC = *Sentence classifier*
ASC = *Aggregated sentiment classifier*
WC = *With classifiers*
NC = *No classifiers*
TR = *Total recall*

Model	TR dataset 1 SC	TR datset 1 ASC
Pattern NC	0.481	0.592
TF-IDF1-1 NC	0.544	0.635
TF-IDF1-2 NC	0.630	0.713
TF-IDF1-3 NC	**0.692**	**0.759**
Pattern WC	0.482	0.607
TF-IDF1-1 WC	0.549	0.637
TF-IDF1-2 WC	0.630	0.698
TF-IDF1-3 WC	0.673	0.721
TF-IDF2-1 NC	0.509	0.607
TF-IDF2-2 NC	0.536	0.606
TF-IDF2-3 NC	0.558	0.643
TF-IDF2-1 WC	0.507	0.601
TF-IDF2-2 WC	0.548	0.616
TF-IDF2-3 WC	0.562	0.654
Customized NC	0.541	0.641
Customized WC	0.614	0.718

Table 18: Results of lexicon based models on dataset 1

Model	TR dataset 1 using unigrams	TR dataset 1 using lemmas
Decision tree 250	0.589	0.568
Decision tree 500	0.613	0.665
Decision tree 1000	0.634	0.662
Decision tree 3200	0.712	0.703
Naive Bayes 250	0.608	0.639
Naive Bayes 500	0.682	0.729
Naive Bayes 1000	0.744	0.771
Naive Bayes 3200	0.775	0.802
SVM 250	0.701	0.742
SVM 500	0.726	0.787
SVM 1000	0.763	0.806
SVM 3200	**0.809**	**0.833**

Table 19: Results of machine learning models on dataset 1

At first, as can be seen in table 18, the models which use the aggregated sentiment classifier perform better than the models which use the sentiment classifier. Moreover, adding the classifiers for modification and negation to the models which use lexicons that are constructed based on TF-IDF computations does not result in an improvement on the total accuracy. However, adding these classifiers to the customized lexicon does result in a significant improvement of the performance. For the models which use the aggregated sentiment classifier and the customized lexicon, the total recall increases from 0.641 to 0.718 after the classifiers for modification and negation are added. Finally, the performance of the models which use lexicon that are constructed on dataset 1 perform signifanctly better the models which use lexicons that are constructed based on dataset 2.

As can be seen in table 19, more training data results for all kind of models in a higher total recall. Moreover, for the majority of the models, the performance of the models is better in case lemmas are used as features instead of unigrams. Furthermore, the best performing models seem to be the support vector machine models followed by the naive Bayes models and the decision tree models.

Finally, after comparing table 18 and 19, we conclude that the best performing lexicon based model on dataset 1 is the model which uses lexicon TF-IDF1-3 without the classifiers for modification and negation and which uses the aggregated sentiment classifier. This models achieves a total recall of 0.759. For the machine learning models, the best performing model is the support vector machine model which is trained on 3200 reviews and which uses lemmas as features. This models achieves a total recall of 0.833. Therefore, we conclude that the machine learning models perform better than the lexicon based models on dataset 1.

Next, table 20 will show the results of the lexicon based models on dataset 2 and table 21 will show the results of the machine learning models on dataset 2.

Model	TR dataset 2 SC	TR 2 datset ASC
Pattern NC	0.525	0.612
TF-IDF1-1 NC	0.555	0.627
TF-IDF1-2 NC	0.555	0.627
TF-IDF1-3 NC	0.525	0.627
Pattern WC	0.530	0.632
TF-IDF1-1 WC	0.560	0.662
TF-IDF1-2 WC	0.565	0.662
TF-IDF1-3 WC	0.552	0.650
TF-IDF2-1 NC	0.562	0.652
TF-IDF2-2 NC	0.630	0.725
TF-IDF2-3 NC	**0.670**	0.750
TF-IDF2-1 WC	0.577	0.685
TF-IDF2-2 WC	0.625	0.745
TF-IDF2-3 WC	0.645	**0.752**
Customized NC	0.517	0.607
Customized WC	0.575	0.635

Table 20: Results of lexicon based models on dataset 2

The results of the lexicon based models on dataset 2 are quite similar to the results of the lexicon based models on dataset 1. As can be seen in table 20, the models which use the aggregated sentiment classifier perform better than the models which use the sentiment classifier. In addition, adding the classifiers for modification and negation to the models which use lexicons that are constructed based on TF-IDF computations does not result in an improvement of the total recall. However, similar to dataset 1, adding the classifiers for modification and negation to the customized lexicon improves the total recall from 0.607 to 0.635. Finally, the models which use lexicons that are constructed based on dataset 2 perform significantly better than the models which use lexicons that are constructed based on dataset 1. This implies that even within the healthcare the performance of a lexicon is strongly dependent on the domain in which it is used.

As can be seen in table 21, more training data results in an improvement of the performance of the models. However, the increase in performance is smaller than the increase in performance that was observed in dataset 1. This could be caused by the fact that the reviews in dataset 2 tend to be longer and, therefore, it may be more difficult to find patterns in the data. Moreover, as opposed to dataset 1, the majority of the models perform better in case unigrams are used as features instead of lemmas. Finally, the best performing models are the support vector machine models followed by the naive Bayes models and the decision tree models.
Finally, after comparing table 20 and 21, we conclude that the best performing

Model	TR dataset 2 using unigrams	TR dataset 2 using lemmas
Decision tree 250	0.482	0.500
Decision tree 500	0.475	0.525
Decision tree 1000	0.585	0.585
Decision tree 2000	0.590	0.600
Naive Bayes 250	0.620	0.570
Naive Bayes 500	0.597	0.555
Naive Bayes 1000	0.615	0.582
Naive Bayes 2000	0.647	0.597
SVM 250	0.565	0.567
SVM 500	0.645	0.602
SVM 1000	**0.667**	0.650
SVM 2000	0.662	**0.665**

Table 21: Results of lexicon based models on dataset 2

lexicon based model on dataset 2 is the model which uses lexicon TF-IDF2-3 with the classifiers for modification and negation. This model achieves a total recall of 0.752. For the machine learning models, the best performing model is the support vector machine which is trained on 1000 reviews and uses unigrams as features. This model achieves a total recall of 0.667. Therefore, we conclude that the lexicon based models perform better on dataset 2.

6 Conclusion

This section will summarize our findings on the effectiveness of health-related sentiment lexicons and the effectiveness of the classifiers for modification and negation. Moreover, a conclusion will be drawn about the performance of lexicon based models on sentiment analysis tasks in comparison with the performance of machine learning models.

Multiple health-related lexicons were constructed in this research. The majority of the lexicons was constructed according to TF-IDF computations and a single lexicon was constructed by means of a human annotator which added 108 words to the original lexicon of Pattern. Generally, the health-related sentiment lexicons perform better than the original lexicon of Pattern. Moreover, after comparing the results on dataset 1, which mainly contained medical reviews regarding physical complaints, and the results on dataset 2, which mainly contained medical reviews regarding psychological complaints, the health-related lexicons perform better in particular on reviews which describe physical complaints. Finally, it should be noted that lexicon based models using lexicons which were constructed based on dataset 2 and tested on dataset 1, and the other way around, did perform only slightly better than the lexicon of Pattern. This implies that is important to construct lexicons based on the same domain as in which the lexicons will be used.

Next to the creation of health-related lexicons, classifiers for modification and negation were incorporated into the lexicon based models. As can be seen in the *results* section, adding these classifiers to the lexicons which were constructed according to the TD-IDF method does not results in a significant improvement of the performance. However, adding these classifiers to the customized lexicon does result in a significant improvement of the performance on both datasets. More research is necessary to understand in which way the creation of a domain specific lexicon and classifiers for modification and negation can be combined to improve the performance.

Finally, the goal of this research was to compare the performance of lexicon based models and machine learning models on sentiment analysis tasks. As mentioned before, the machine learning models performed better on the reviews in dataset 1 whereas the lexicon based models performed better on dataset 2. Therefore, based on these findings, it is difficult to draw a conclusion whether lexicon based models or machine learning models perform better on sentiment analysis tasks.

7 Discussion

After the conclusion of this research has been drawn in the previous section, this section will review the limitations and potential improvements for further research in the field of sentiment analysis.

Probably, the biggest limitation of this research was the availability of user generated medical reviews. In general, stakeholders which possess medical data were quite reluctant to share these reviews due to privacy concerns. Eventually, approximately 13.000 reviews were collected for this research divided over 2 datasets. This amount of reviews was enough for developing baseline lexicon based models and machine learning models. As mentioned before, the reviews in both datasets were quite different. Therefore, merging the two datasets did not results in desirable outcomes.

The performance of the lexicon based models and the machine learning models would probably improve in case more reviews could have been used for this research. In case more reviews would have been used, the TF-IDF method would be more accurate and depend less on randomness. In this research, for example, some words received a positive polarity value because they appeared 3 times in a positive review and 1 time in a negative review. Due to these these low frequencies, there is a relatively big chance that the word received a polarity value which is not in accordance with the real sentiment they express. For the machine learning models, more reviews would have resulted in more training data. As can be seen in the *results* section, generally, the machine learning models which were trained on the biggest training set (3200 reviews for dataset 1 and 2000 reviews for dataset 2) scored significantly better than the models that were trained on less reviews. Therefore, we assume that more training data would have resulted in a better performance of the machine learning models.

In addition to the limitations of this research, this section contains some recommendation to keep in mind during further research regarding sentiment analysis in the healthcare domain. At first, lexicon based models in the healthcare domain could incorporate more medical wisdom into their models. Many reviews that were wrongly classified in this research contained sentences which expressed medical statements such as: *Mijn bloeddruk is zo veel lager* or *mijn zelfvertrouwen is zo veel lager*. These two phrases do not contain (clear) sentiment words but both express sentiment in a medical context. Due to the desired level of *bloeddruk* and *zelfvertrouwen*, the word *lager* expresses a completely different sentiment in both phrases. Lexicon based models which incorporate knowledge about the desired level of certain medical conditions, and classifiers which are able to recognize when such information is expressed, would perform significantly better.

Another recommendation for future research would be to improve the performance of the negation classifier. In this research, the scope of the negation classifier was only one sentence. This implies that a sentiment word could only be negated in case that the negation word appeared in the same sentence as the sentiment word. However, in practice, many sentiment words were negated by means of negation words in different sentences and, therefore, this was not de-

tected by the negation classifier used for this research. An example to clarify this would be: *ik had altijd veel last van pijn en vermoeidheid. Dat is nu allemaal voorbij.* In this example, the negation word *voorbij* negates the two sentiment words *pijn* en *vermoeidheid*. However, because the negation word does not appear in the same sentence as the sentiment words, the negation is not recognized.

Finally, we recommend further research to consider incorporating sarcasm and irony into lexicon based models. As mentioned briefly in the literature study, sarcasm and irony are difficult to grasp for lexicon based models and, therefore, most studies do not even try to detect sarcasm and irony. However, many wrongly classified reviews in this research contained sarcasm or irony which was not recognized by the lexicon based models. For this reason, many reviews containing sarcasm or irony were wrongly classified.

Appendix A

Original string	New string
..	.
...	.
....	.
.....	.
.....	.
!!	!
!!!	!
!!!!	!
!!!!!	!
!!!!!!	!
??	?
???	?
????	?
?????	?
??????	?
!?	!
!!?	!
!!??	!
?!	?
??!	?
??!!	?
.!	.
.?	.
..?	.
..!	.
..??	.
..!!	.

Appendix B

Word	Polarity	Word	Polarity	Word	Polarity
aanvallen	-0.5	achteruit	-0.5	alert	0.5
angststoornis	-0.4	baal	-0.4	balen	-0.4
benauwdheid	-0.5	bijwerking	-0.4	bijwerking	-0.4
blaasontsteking	-0.5	bloeding	-0.5	bloedverlies	-0.5
buikpijn	-0.5	daadkrachtig	0.4	darmklachten	-0.4
depressie	-0.5	diarree	-0.5	doorbraak	0.5
down	-0.4	draaierig	-0.4	droom	0.4
duizeligheid	-0.5	eetlust	0.4	ellende	-0.5
energie	0.5	erger	-0.4	genieten	0.5
gezwollen	-0.5	haaruitval	-0.6	hartaanval	-0.5
hartklachten	-0.4	hartkloppingen	-0.4	helaas	-0.5
helpen	0.5	hoesten	-0.4	hoofdpijn	-0.5
hoofdpijn	-0.5	hooikoorts	-0.5	huiduitslag	-0.4
heerlijk	0.5	incontinentie	-0.4	jeuk	-0.5
keelpijn	-0.5	klacht	-0.4	klachten	-0.4
knallen	-0.4	knallend	-0.5	koort	-0.4
koppijn	-0.4	kortademig	-0.4	kortademigheid	-0.4
krampen	-0.4	maagklachten	-0.4	maagkrampen	-0.4
maagpijn	-0.4	malaise	-0.5	migraine	-0.4
misselijkheid	-0.6	moeheid	-0.5	nachtmerrie	-0.5
obstipatie	-0.5	onrust	-0.4	ontsteken	-0.4
ontsteking	-0.4	oorsuizen	-0.5	opgejaagd	-0.5
opgewekt	0.4	opgezette	-0.5	opvlieger	-0.4
overgeven	-0.6	paniekaanvallen	-0.4	pijn	-0.4
positief	0.4	pret	0.4	probleem	-0.4
reactie	-0.4	reuma	-0.5	rillingen	-0.5
rommel	-0.5	rotzooi	-0.5	rust	0.5
rustiger	0.5	somberheid	-0.5	spierpijn	-0.5
stabieler	0.5	steken	-0.4	stemmingswisselingen	-0.5
stoornis	-0.5	succes	0.6	sufheid	-0.3
tevredenheid	0.5	tintelingen	-0.4	topper	0.6
troep	-0.6	uitgerust	0.5	verbeterd	0.6
verbetering	0.4	verdrietig	-0.5	verkoudheid	-0.3
verlichting	0.4	vermoeidheid	-0.4	vertrouwen	0.4
vieze	-0.3	vooruitgang	0.3	werken	0.5
wondermiddel	0.7	zekerheid	0.4	zelfvertrouwen	0.4

Appendix C

In the following tables, the precision(P), recall(R) and F-score(F) of the models constructed in this research will be shown. Table 2 and .., show the results of the lexicon based models on dataset 1. Table .. and .., show the results of the lexicon based models on dataset 2 Finally, table ... shows the results of the machine learning models on dataset 1 and table shows the results of the machine learning models on dataset 1. The following abbreviates are used:

WC = *With classifiers*
NC = *No classifiers*

	Positive			Negative		
Model	P	R	F	P	R	F
Pattern NC	0.596	0.478	0.531	0.373	0.490	0.423
TF-IDF1-1 NC	0.627	0.529	0.571	0.461	0.562	0.505
TF-IDF1-2 NC	0.647	0.624	0.635	0.624	0.647	0.635
TF-IDF1-3 NC	0.721	0.695	0.708	0.695	0.721	0.708
Pattern WC	0.561	0.475	0.514	0.402	0.488	0.441
TF-IDF1-1 WC	0.614	0.534	0.571	0.484	0.566	0.521
TF-IDF1-2 WC	0.630	0.622	0.626	0.631	0.639	0.635
TF-IDF1-3 WC	0.637	0.769	0.657	0.710	0.670	0.689
TF-IDF2-1 NC	0.586	0.498	0.539	0.432	0.520	0.472
TF-IDF2-2 NC	0.530	0.527	0.529	0.542	0.545	0.544
TF-IDF2-3 NC	0.515	0.554	0.534	0.601	0.564	0.581
TF-IDF2-1 WC	0.556	0.497	0.525	0.459	0.518	0.486
TF-IDF2-2 WC	0.548	0.538	0.543	0.547	0.557	0.552
TF-IDF2-3 WC	0.515	0.559	0.536	0.609	0.566	0.587
Customized NC	0.441	0.542	0.486	0.641	0.543	0.588
Customized WC	0.568	0.617	0.592	0.660	0.614	0.636

Table 22: Results of lexicon based models which use sentence classifier on dataset 1

	Positive			Negative		
Model	P	R	F	P	R	F
Pattern NC	0.681	0.569	0.620	0.503	0.621	0.666
TF-IDF1-1 NC	0.694	0.612	0.651	0.574	0.662	0.615
TF-IDF1-2 NC	0.698	0.711	0.705	0.727	0.714	0.721
TF-IDF1-3 NC	0.737	0.764	0.750	0.781	0.755	0.768
Pattern WC	0.686	0.573	0.624	0.508	0.627	0.561
TF-IDF1-1 WC	0.668	0.620	0.643	0.606	0.655	0.630
TF-IDF1-2 WC	0.678	0.698	0.688	0.717	0.698	0.707
TF-IDF1-3 WC	0.704	0.734	0.718	0.754	0.727	0.739
TF-IDF2-1 NC	0.655	0.589	0.620	0.560	0.628	0.592
TF-IDF2-2 NC	0.568	0.605	0.586	0.643	0.607	0.625
TF-IDF2-3 NC	0.576	0.656	0.614	0.710	0.635	0.670
TF-IDF2-1 WC	0.625	0.587	0.605	0.577	0.615	0.595
TF-IDF2-2 WC	0.579	0.616	0.597	0.653	0.617	0.634
TF-IDF2-3 WC	0.581	0.672	0.623	0.727	0.643	0.682
Customized NC	0.556	0.672	0.608	0.739	0.633	0.682
Customized WC	0.681	0.729	0.704	0.756	0.711	0.733

Table 23: Results of lexicon based models which use aggregated sentiment classifier on dataset 1

	Positive			Negative		
Model	P	R	F	P	R	F
Pattern NC	0.640	0.520	0.573	0.410	0.532	0.463
TF-IDF1-1 NC	0.730	0.544	0.623	0.390	0.590	0.469
TF-IDF1-2 NC	0.720	0.541	0.618	0.390	0.582	0.467
TF-IDF1-3 NC	0.690	0.518	0.592	0.360	0.537	0.431
Pattern WC	0.605	0.526	0.562	0.455	0.535	0.491
TF-IDF1-1 WC	0.680	0.548	0.607	0.440	0.578	0.500
TF-IDF1-2 WC	0.695	0.551	0.615	0.435	0.587	0.500
TF-IDF1-3 WC	0.695	0.551	0.615	0.435	0.587	0.500
TF-IDF2-1 NC	0.680	0.550	0.608	0.445	0.581	0.504
TF-IDF2-2 NC	0.630	0.630	0.630	0.630	0.630	0.630
TF-IDF2-3 NC	0.755	0.645	0.695	0.585	0.704	0.639
TF-IDF2-1 WC	0.680	0.578	0.625	0.505	0.612	0.553
TF-IDF2-2 WC	0.695	0.658	0.676	0.640	0.677	0.658
TF-IDF2-3 WC	0.670	0.614	0.641	0.580	0.637	0.607
Customized NC	0.425	0.521	0.468	0.610	0.514	0.558
Customized WC	0.475	0.593	0.527	0.675	0.562	0.613

Table 24: Results of lexicon based models which use sentence classifier on dataset 2

	Positive			Negative		
Model	P	R	F	P	R	F
Pattern NC	0.705	0.594	0.645	0.520	0.638	0.573
TF-IDF1-1 NC	0.780	0.600	0.678	0.480	0.685	0.564
TF-IDF1-2 NC	0.770	0.599	0.673	0.485	0.678	0.565
TF-IDF1-3 NC	0.775	0.598	0.675	0.480	0.680	0.563
Pattern WC	0.705	0.615	0.657	0.560	0.654	0.603
TF-IDF1-1 WC	0.760	0.615	0.680	0.525	0.686	0.594
TF-IDF1-2 WC	0.785	0.630	0.699	0.540	0.715	0.615
TF-IDF1-3 WC	0.780	0.619	0.692	0.520	0.702	0.597
TF-IDF2-1 NC	0.730	0.669	0.698	0.640	0.703	0.670
TF-IDF2-2 NC	0.710	0.731	0.720	0.740	0.718	0.729
TF-IDF2-3 NC	0.820	0.718	0.766	0.680	0.790	0.731
TF-IDF2-1 WC	0.730	0.669	0.698	0.640	0.703	0.670
TF-IDF2-2 WC	0.720	0.757	0.738	0.770	0.733	0.751
TF-IDF2-3 WC	0.790	0.734	0.761	0.715	0.772	0.742
Customized NC	0.505	0.635	0.562	0.710	0.589	0.643
Customized WC	0.520	0.675	0.587	0.750	0.609	0.672

Table 25: Results of lexicon based models which use aggregated sentiment classifier on dataset 2

	Positive			Negative		
Model	P	R	F	P	R	F
Decision Tree 250 unigrams	0.510	0.597	0.550	0.668	0.586	0.624
Decision Tree 500 unigrams	0.589	0.611	0.600	0.638	0.617	0.628
Decision Tree 1000 unigrams	0.655	0.620	0.637	0.614	0.649	0.631
Decicion Tree 3200 unigrams	0.711	0.706	0.709	0.714	0.720	0.717
Naive Bayes 250 unigrams	0.477	0.638	0.545	0.739	0.594	0.659
Naive Bayes 500 unigrams	0.553	0.738	0.632	0.810	0.653	0.723
Naive Bayes 1000 unigrams	0.681	0.773	0.724	0.808	0.724	0.764
Naive Bayes 3200 unigrams	0.767	0.773	0.770	0.783	0.778	0.780
SVM 250 unigrams	0.678	0.703	0.690	0.724	0.700	0.712
SVM 500 unigrams	0.673	0.745	0.707	0.778	0.712	0.744
SVM 1000 unigrams	0.704	0.793	0.745	0.823	0.742	0.780
SVM 3200 unigrams	0.795	0.8125	0.804	0.823	0.807	0.815
Decision Tree 250 lemmas	0.668	0.548	0.602	0.469	0.595	0.524
Decision Tree 500 lemmas	0.670	0.655	0.663	0.660	0.675	0.668
Decision Tree 1000 lemmas	0.668	0.651	0.659	0.656	0.672	0.664
Decicion Tree 3200 lemmas	0.640	0.725	0.680	0.766	0.688	0.725
Naive Bayes 250 lemmas	0.864	0.587	0.699	0.415	0.761	0.537
Naive Bayes 500 lemmas	0.813	0.688	0.746	0.646	0.782	0.707
Naive Bayes 1000 lemmas	0.762	0.770	0.766	0.781	0.773	0.777
Naive Bayes 3200 lemmas	0.762	0.823	0.792	0.842	0.786	0.813
SVM 250 lemmas	0.757	0.727	0.742	0.727	0.757	0.741
SVM 500 lemmas	0.790	0.778	0.784	0.783	0.795	0.789
SVM 1000 lemmas	0.818	0.790	0.804	0.791	0.819	0.805
SVM 3200 lemmas	0.841	0.822	0.832	0.825	0.844	0.834

Table 26: Results of machine learning models on dataset

	Positive			Negative		
Model	P	R	F	P	R	F
Decision Tree 250 unigrams	0.515	0.474	0.493	0.449	0.490	0.469
Decision Tree 500 unigrams	0.520	0.467	0.492	0.429	0.482	0.454
Decision Tree 1000 unigrams	0.551	0.582	0.566	0.619	0.588	0.603
Decicion Tree 3200 unigrams	0.568	0.583	0.576	0.609	0.594	0.601
Naive Bayes 250 unigrams	0.765	0.583	0.662	0.474	0.677	0.557
Naive Bayes 500 unigrams	0.609	0.585	0.597	0.584	0.608	0.596
Naive Bayes 1000 unigrams	0.530	0.630	0.576	0.700	0.607	0.650
Naive Bayes 2000 unigrams	0.609	0.651	0.629	0.685	0.645	0.665
SVM 250 unigrams	0.760	0.537	0.630	0.371	0.616	0.463
SVM 500 unigrams	0.693	0.622	0.656	0.594	0.668	0.629
SVM 1000 unigrams	0.698	0.647	0.672	0.633	0.686	0.659
SVM 2000 unigrams	0.698	0.641	0.669	0.624	0.682	0.652
Decision Tree 250 lemmas	0.484	0.490	0.487	0.515	0.509	0.512
Decision Tree 500 lemmas	0.454	0.515	0.483	0.589	0.528	0.557
Decision Tree 1000 lemmas	0.619	0.570	0.594	0.550	0.600	0.574
Decicion Tree 2000 lemmas	0.630	0.585	0.606	0.570	0.615	0.591
Naive Bayes 250 lemmas	0.494	0.573	0.531	0.646	0.570	0.605
Naive Bayes 500 lemmas	0.464	0.558	0.506	0.646	0.556	0.597
Naive Bayes 1000 lemmas	0.510	0.588	0.546	0.656	0.581	0.616
Naive Bayes 2000 lemmas	0.500	0.612	0.550	0.695	0.590	0.638
SVM 250 lemmas	0.750	0.540	0.628	0.385	0.615	0.474
SVM 500 lemmas	0.681	0.580	0.626	0.525	0.631	0.573
SVM 1000 lemmas	0.655	0.640	0.648	0.646	0.660	0.653
SVM 2000 2000	0.698	0.644	0.670	0.628	0.684	0.655

Table 27: Results of machine learning models on dataset 2

References

[1] Tahsin Ali et al. “Can i hear you? Sentiment analysis on medical forums”. In: Jan. 2013, pp. 667–673.

[2] Sattam Almatarneh and Pablo Gamallo. “Automatic Construction of Domain-Specific Sentiment Lexicons for Polarity Classification”. In: June 2018, pp. 175–182. ISBN: 978-3-319-61577-6. DOI: 10.1007/978-3-319-61578-3_17.

[3] Orestes Appel et al. “A Hybrid Approach to the Sentiment Analysis Problem at the Sentence Level”. In: *Knowledge-Based Systems* 108 (May 2016). DOI: 10.1016/j.knosys.2016.05.040.

[4] Dr. Muhammad Asghar. “A Unified Framework for Creating Domain Dependent Polarity Lexicons from User Generated Reviews”. In: (Jan. 2015). DOI: 10.6084/M9.FIGSHARE.1609618.

[5] Dr. Muhammad Asghar et al. “Lexicon based approach for sentiment classification of user reviews”. In: *Life Science Journal* 11 (Jan. 2014), pp. 468–473.

[6] Dr. Muhammad Asghar et al. “Lexicon-enhanced sentiment analysis framework using rule-based classification scheme”. In: *PLoS ONE* e0171649 (Feb. 2017), pp. 1–22. DOI: 10.1371/journal.pone.0171649.

[7] Dr. Muhammad Asghar et al. “Medical opinion lexicon: An incremental model for mining health reviews”. In: *International Journal of Academic Research* 6 (Jan. 2014), pp. 295–302. DOI: 10.7813/2075-4124.2014/6-1/A.39.

[8] Dr. Muhammad Asghar et al. “SentiHealth: creating health-related sentiment lexicon using hybrid approach”. In: *SpringerPlus* 5 (Dec. 2016). DOI: 10.1186/s40064-016-2809-x.

[9] Dr. Muhammad Asghar et al. “Subjectivity lexicon construction for mining drug reviews”. In: *Science International* 26 (Nov. 2013), pp. 145–149.

[10] Wouter Atteveldt et al. “Good News or Bad News? Conducting Sentiment Analysis on Dutch Text to Distinguish Between Positive and Negative Relations”. In: *Journal of Information Technology Politics* 5 (July 2008), pp. 73–94. DOI: 10.1080/19331680802154145.

[11] T. Avontuur et al. “Developing a part-of-speech tagger for Dutch tweets”. In: *CLIN 2012*. 2012.

[12] Carmen Banea, Rada Mihalcea, and Janyce Wiebe. “A Bootstrapping Method for Building Subjectivity Lexicons for Languages with Scarce Resources.” In: Jan. 2008.

[13] Carmen Banea et al. “Multilingual Subjectivity Analysis Using Machine Translation.” In: Jan. 2008, pp. 127–135. DOI: 10.3115/1613715.1613734.

[14] Mohammad Darwich et al. “Corpus-Based Techniques for Sentiment Lexicon Generation: A Review”. In: *Journal of Digital Information Management* 17 (Oct. 2019), p. 296. DOI: 10.6025/jdim/2019/17/5/296-305.

[15] Orphée De Clercq and Véronique Hoste. “Rude waiter but mouthwatering pastries! An exploratory study into Dutch Aspect-Based Sentiment Analysis”. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), May 2016, pp. 2910–2917. URL: https://www.aclweb.org/anthology/L16-1465.

[16] Tom De Smedt and Walter Daelemans. “” Vreselijk mooi!” (terribly beautiful): A Subjectivity Lexicon for Dutch Adjectives.” In: Jan. 2012, pp. 3568–3572.

[17] Gulsen Demiroz et al. “Learning Domain-Specific Polarity Lexicons”. In: Sept. 2012. DOI: 10.1109/ICDMW.2012.120.

[18] Yihan Deng, Matthaeus Stoehr, and Kerstin Denecke. “Retrieving attitudes: Sentiment analysis from clinical narratives”. In: *CEUR Workshop Proceedings* 1276 (Jan. 2014), pp. 12–15.

[19] Lopamudra Dey et al. “Sentiment Analysis of Review Datasets Using Naïve Bayes‘ and K-NN Classifier”. In: *International Journal of Information Engineering and Electronic Business* 8 (July 2016), pp. 54–62. DOI: 10.5815/ijieeb.2016.04.07.

[20] Andrea Esuli and Fabrizio Sebastiani. “SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining”. In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. Genoa, Italy: European Language Resources Association (ELRA), May 2006. URL: http://www.lrec-conf.org/proceedings/lrec2006/pdf/384_pdf.pdf.

[21] Ronen Feldman. “Techniques and Applications for Sentiment Analysis”. In: *Commun. ACM* 56 (Apr. 2013), 82–89. DOI: 10.1145/2436256.2436274.

[22] Lorraine Goeuriot et al. “Sentiment lexicons for health-related opinion mining”. In: *IHI'12 - Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium* (Jan. 2012). DOI: 10.1145/2110363.2110390.

[23] Lorraine Goeuriot et al. “Sentiment lexicons for health-related opinion mining”. In: *IHI'12 - Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium* (Jan. 2012). DOI: 10.1145/2110363.2110390.

[24] Emma Haddi, Xiaohui Liu, and Yong Shi. “The Role of Text Pre-processing in Sentiment Analysis”. In: *Procedia Computer Science* 17 (Dec. 2013), 26–32. DOI: 10.1016/j.procs.2013.05.005.

[25] Bas Heerschop, Alexander Hogenboom, and Flavius Frasincar. “Sentiment Lexicon Creation from Lexical Resources”. In: vol. 87. June 2011, pp. 185–196. DOI: 10.1007/978-3-642-21863-7_16.

[26] Alexander Hogenboom et al. “Multi-lingual Support for Lexicon-Based Sentiment Analysis Guided by Semantics”. In: *Decision Support Systems* 62 (June 2014). DOI: 10.1016/j.dss.2014.03.004.

[27] Md Islam and Diana Inkpen. "Second order co-occurrence PMI for determining the semantic similarity of words". In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2006)* (Jan. 2006).

[28] Valentin Jijkoun and Katja Hofmann. "Generating a Non-English Subjectivity Lexicon: Relations That Matter." In: Jan. 2009, pp. 398–405.

[29] Alistair Kennedy and Diana Inkpen. "Sentiment Classication of Movie and Product Reviews Using Contextual Valence Shifters". In: (Jan. 2005).

[30] Yihan Deng Kerstin Denecke. "Sentiment analysis in medical settings: New opportunities and challenges". In: *Elsevier* (2015).

[31] Aurangzeb Khan, Baharum Baharudin, and Khairullah Khan. "Sentiment Classification Using Sentence-level Lexical Based Semantic Orientation of Online Reviews". In: *Trends in Applied Sciences Research* 6 (Oct. 2011), pp. 1141–1157. DOI: 10.3923/tasr.2011.1141.1157.

[32] Kevin Lerman, Sasha Blair-Goldensohn, and Ryan McDonald. "Sentiment Summarization: Evaluating and Learning User Preferences." In: Jan. 2009, pp. 514–522.

[33] Jingjing Liu and Stephanie Seneff. "Review Sentiment Scoring via a Parse-and-Paraphrase Paradigm." In: Jan. 2009, pp. 161–169. DOI: 10.3115/1699510.1699532.

[34] Yue Lu et al. "Automatic construction of a context-aware sentiment lexicon: An optimization approach". In: Jan. 2011, pp. 347–356. DOI: 10.1145/1963405.1963456.

[35] Paweł Mandera, Emmanuel Keuleers, and Marc Brysbaert. "Explaining human performance in psycholinguistic tasks with models of semantic similarity based on prediction and counting: A review and empirical validation". In: *Journal of Memory and Language* 92 (2017), pp. 57 –78. ISSN: 0749-596X. DOI: https://doi.org/10.1016/j.jml.2016.04.001. URL: http://www.sciencedirect.com/science/article/pii/S0749596X16300079.

[36] Rada Mihalcea, Carmen Banea, and Janyce Wiebe. "Learning Multilingual Subjective Language via Cross-Lingual Projections". In: Jan. 2007.

[37] Antonio Moreno-Ortiz, Chantal Pérez-Hernández, and Maria Del-Olmo. "Managing Multiword Expressions in a Lexicon-Based Sentiment Analysis System for Spanish". In: *Proceedings of the 9th Workshop on Multiword Expressions*. Atlanta, Georgia, USA: Association for Computational Linguistics, June 2013, pp. 1–10. URL: https://www.aclweb.org/anthology/W13-1001.

[38] Jin-Cheon Na et al. "Sentiment Classification of Drug Reviews Using a Rule-Based Linguistic Approach". In: Nov. 2012, pp. 189–198. ISBN: 978-3-642-34751-1. DOI: 10.1007/978-3-642-34752-8_25.

[39] Stephen Oppong et al. "Business Decision Support System based on Sentiment Analysis". In: *International Journal of Information Engineering and Electronic Business* 11 (Jan. 2019), pp. 36–49. DOI: 10.5815/ijieeb.2019.01.05.

[40] Bo Pang and Lillian Lee. “A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts”. In: *Computing Research Repository - CORR* 271-278 (July 2004), pp. 271–278. DOI: 10.3115/1218955.1218990.

[41] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. “Thumbs up? Sentiment Classification using Machine Learning Techniques”. In: *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*. Association for Computational Linguistics, July 2002, pp. 79–86. DOI: 10.3115/1118693.1118704. URL: https://www.aclweb.org/anthology/W02-1011.

[42] Guang Qiu et al. “Expanding Domain Sentiment Lexicon through Double Propagation”. In: Jan. 2009, pp. 1199–1204.

[43] Francisco Ramírez-Tinoco et al. “Use of Sentiment Analysis Techniques in Healthcare Domain”. In: Jan. 2019, pp. 189–212. ISBN: 978-3-030-06148-7. DOI: 10.1007/978-3-030-06149-4_8.

[44] H. Reckman et al. “teragram: Rule-based detection of sentiment phrases using SAS Sentiment Analysis”. In: *SemEval@NAACL-HLT*. 2013.

[45] Ellen Riloff et al. “Sarcasm as contrast between a positive sentiment and negative situation”. In: *Proceedings of EMNLP* (Jan. 2013), pp. 704–714.

[46] Irina Rish. “An Empirical Study of the Naïve Bayes Classifier”. In: *IJCAI 2001 Work Empir Methods Artif Intell* 3 (Jan. 2001).

[47] Padmaja Savaram and Sameen S. “Opinion Mining and Sentiment Analysis - An Assessment of Peoples’ Belief: A Survey”. In: *International Journal of Ad hoc, Sensor Ubiquitous Computing* 4 (Feb. 2013), pp. 21–33. DOI: 10.5121/ijasuc.2013.4102.

[48] Maite Taboada et al. “Lexicon-Based Methods for Sentiment Analysis”. In: *Computational Linguistics* 37 (June 2011), pp. 267–307. DOI: 10.1162/COLI_a_00049.

[49] Michael Wiegand et al. “A Survey on the Role of Negation in Sentiment Analysis”. In: (July 2010).

[50] Salud María Zafra et al. “How do we talk about doctors and drugs? Sentiment analysis in forums expressing opinions for medical domain”. In: *Artificial Intelligence in Medicine* 93 (Apr. 2018). DOI: 10.1016/j.artmed.2018.03.007.