



BACHELOR'S THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
BACHELOR OF ARTS IN COMPUTATIONAL LINGUISTICS

Comparing and Combining Rule Based and Machine Learning Techniques for Sentiment Analysis

Author:

Valentin DEYRINGER

Supervisor:

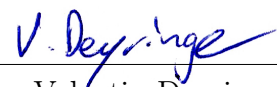
Ph.D. Christopher CULY

SEMINAR FÜR SPRACHWISSENSCHAFT
EBERHARD KARLS UNIVERSITÄT TÜBINGEN

August 2014

I hereby declare that this paper is the result of my own independent scholarly work. I have acknowledged all the other authors' ideas and referenced direct quotations from their work (in the form of books, articles, essays, dissertations, and on the Internet). No material other than that listed has been used.

Tübingen, August 29, 2014



Valentin Deyringer

Contents

List of Figures	iv
List of Tables	iv
1 Introduction	1
2 Sentiment Analysis	1
2.1 Motivation for Sentiment Analysis	2
2.2 Difficulties in Sentiment Analysis	3
2.3 Approaches to Sentiment Analysis	4
2.3.1 Support Vector Machines	5
2.3.2 Naïve Bayes Classification	6
2.3.3 Rule Based Approaches	7
3 The Experiment	8
3.1 Resources	8
3.2 Implemented Algorithms	9
3.3 Results	10
3.4 Conclusion	13
3.5 Future Work	14
4 Online Sentiment Analysis Tool	15
4.1 Future Work	17
5 Acknowledgments	17
References	17

Abstract

This paper compares a rule based system and two machine learning techniques as approaches to sentiment analysis. The machine learning techniques used are Naïve Bayes classification and Support Vector Machines, both trained and tested on a set of movie reviews, restaurant reviews and a combination of these two data sets. Afterwards, combinations of the used classification methods are tested. It is found that the machine learning techniques yield better results than the rule based method. By combining the algorithms the misclassification rate can be reduced considerably. Furthermore, a tool implementing the three algorithms to measure and visualize sentiment on the text level is presented.

List of Figures

2.1	Russell’s Model of Emotional Affect	2
2.2	Scatterplot Featuring a Linear Support Vector Machine’s Decision Boundary	6

List of Tables

3.1	Accuracies of Naïve Bayes Classifiers on different test sets	11
3.2	Accuracies of Support Vector Machines on different test sets . . .	12
3.3	Accuracy of rule based approach on different test sets	13
3.4	Accuracies and misclassification of combined classifications tested on the combined test data set	13

1 Introduction

Natural Language Processing (*NLP*) is a discipline concerned with the automation of linguistic processes by means of algorithmic solutions. The ultimate goal of *NLP* is to make machines understand human language. The knowledge gained by *NLP* is not only of interest for linguists or cognitive scientists but can also provide useful information in other areas.

There is a wide range of free and commercial toolkits which offer convenient ways for *NLP* tasks. One of these that is used for this work is *NLTK*¹ (Bird, Klein, & Loper, 2009), a platform providing several libraries for working with natural language in the Python programming language. Another library for Python used in this work is *scikit-learn* (Pedregosa et al., 2011), which allows for easy ways to perform machine learning tasks, the state of the art techniques used in *NLP* for text classification.

An important branch of *NLP* is text classification which aims to assign categories to a text. These categories can be of various types. For instance, one could classify a text for the gender of its author (Argamon, Koppel, Fine, & Shimoni, 2003).

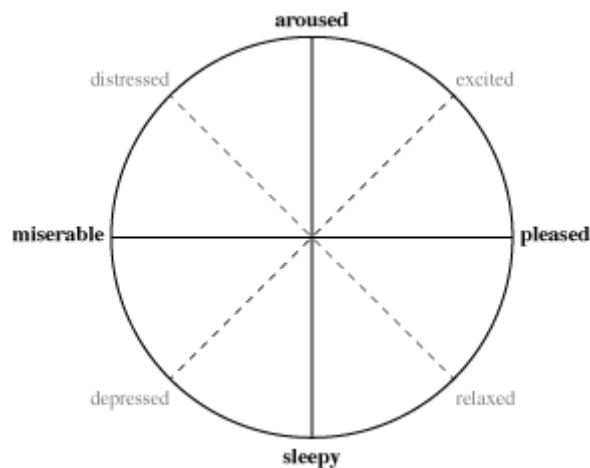
Another text classification task which has drawn a lot of interest over the last few years is sentiment analysis. The objective of sentiment analysis is to extract the subjective information a text conveys. The following section will give a more precise description of sentiment analysis, its motivation and the difficulties it faces.

This work will present different approaches to sentiment analysis, and evaluate results of implemented algorithms. Additionally combinations of the techniques are tested. The results of these evaluations helped towards developing a tool which is then presented. This tool offers an interface to perform sentiment analysis for texts based on the algorithms implemented for the experiment. A main feature of the tool is the visualization of the outputs.

2 Sentiment Analysis

Sentiment analysis or opinion mining is the task of extracting the subjective information of a text. This task can be performed in a general fashion by extracting the overall sentiment of a document or more specifically by classifying several parts of a document. There is also a difference between general sentiment of a document and topic dependent sentiment, i.e. the authors attitude towards a specific topic.

¹Natural Language Toolkit

Figure 2.1: Russell's Model of Emotional Affect⁴

Sentiment values can be represented on different scales. Some work just focuses on the two levels of positive and negative sentiment, or on an extended scale with the additional levels of neutral and/or mixed sentiment. Other systems offer more fine grained scales. For example, a scale ranging from -10 to $+10$ with negative values depicting negative sentiment and positive values depicting positive sentiment could be used. It is also possible to represent sentiment on a multidimensional scale. *Tweet Viz*² for instance places documents on a two-dimensional emotional plane. Similar to the model presented by Russell (1980) shown in Figure 2.1, this plane is defined by the axis of activeness ranging from subdued to active and the axis of pleasure ranging from highly unpleasant to highly pleasant. The dimensions to measure emotions can be extended further. The *ANEW*³ dictionary presented in Bradley & Lang, 1999 lists words with their ratings for the three dimensions of valence, arousal and dominance.

The task of sentiment analysis has been approached with different methods which are machine learning based or rule based. The most common approaches are explained in Section 2.3.

2.1 Motivation for Sentiment Analysis

There are numerous free and commercial systems implementing sentiment analysis algorithms. These systems serve different purposes. The most popular field is investigating the reputation of companies and their products. There are several

²http://www.csc.ncsu.edu/faculty/healey/tweet_viz/

³Affective Norms for English Words

⁴Russell's Model of Emotional Affect. (2013). [image] Available at: http://www.csc.ncsu.edu/faculty/healey/tweet_viz/figs/circumplex.png [Accessed 29 Aug. 2014].

companies providing social media monitoring systems which also display sentiment measures.

Another application area is the financial market. Bollen, Mao, and Zeng (2011) examined the correlation between the sentiment conveyed by huge amounts of data from the microblogging platform twitter and the *DJIA*⁵. Bollen et al. states that predictions of the *DJIA* can be improved significantly by including public mood measurements performed on tweets.

Sentiment analysis is also applicable in research fields like sociology. In a recent study Kramer, Guillory, and Hancock (2014) examined the phenomenon of emotional contagion via text in the social network facebook.

2.2 Difficulties in Sentiment Analysis

Sentiment analysis is not a trivial task. Da Silva Cardoso (2013), based on S. Mukherjee & Bhattacharyya, 2013 lists the following difficulties:

- Sarcasm, irony and implicit sentiments

The words *only* and *lovely* in *My computer crashed only ten times this week. What a lovely machine!* are not meant literally but rather express an opposite negative sentiment.

- Domain dependency

In general, the sentence *Did he read the book?* does not convey sentiment. But when this statement is made about a director of a movie in a movie review it is negative. (Pang, Lee, & Vaithyanathan, 2002)

- "Thwarted expectations" (Pang et al., 2002)

A previous statement might be reversed afterwards like in the following statement: *Based on the book and the cast, this should be the best movie ever. But it isn't!*

- Pragmatics

Special characteristics of a text which the character representation does not resemble may play a role for the sentiment. For example, words may be underlined or written in italics for emphasis. This also has an effect on the sentiment these words convey.

- World knowledge

The Sentence *You must be Einstein!* implies that a person is very intelligent

⁵Dow Jones Industrial Average

and thus conveys positive sentiment, although the word *Einstein* would not generally be classified as positive. (Da Silva Cardoso, 2013)

- Subjectivity detection

The sentence *I hate love stories.* combines the negative word *hate* and the positive word *love*. A human reader resolves this consecutiveness without any problems. A machine in contrast may struggle to resolve such cases. (A. Mukherjee & Liu, 2012)

- Entity identification

The sentence *I really like my new Mercedes, it's so much more comfortable than my old rusty Ford.* is a positive statement with *Mercedes* as target word. The statement is negative when *Ford* is targeted. Grammatical analysis is needed to resolve the orientation of the sentiment bearing words.

- Anaphora

It is a very challenging task for machines to interpret which entity a pronoun refers to like *he* in *Lenoardo DiCaprio is a way better actor than Matt Damon. He definitely deserves an academy award..* This is especially important for sentiment analysis with a target word.

- Negation

A generally positive word might be negated and thus convey negative sentiment and vice versa: *There was not one good scene in the whole movie!* The algorithms implemented for this work contain a basic negation handling procedure.

2.3 Approaches to Sentiment Analysis

For classification in general and sentiment analysis in particular, the state of the art techniques are based on machine learning algorithms. These algorithms attempt to detect patterns in a given training data set. The recognized structures are then used to process unseen data. In case of classification, this process aims to assign a class to a new data point.

There are supervised and unsupervised machine learning algorithms. Supervised algorithms are given labelled data whereupon the algorithms try to find the best features describing the different classes.

For unsupervised algorithms, the training data is not labelled. The algorithms cluster similar data points and separate the different classes this way.

There are also intermediate machine learning techniques. So called semi-supervised algorithms combine a set of labelled data and unlabelled data for learning. The performance achieved exceeds the performance that could be obtained by training only on one of these sets. Active learning techniques are basically supervised methods but require manual confirmation of output for difficult cases and improve their assumptions based on learning from the new input.

In the following, we will look at the machine learning techniques which find the most application for sentiment analysis in detail. These are the supervised methods of Naïve Bayes classification and Support Vector Machines. (Syamlal & Bruins, 2007; Pang et al., 2002; Narayanan, Arora, & Bhatia, 2013) Both techniques are examined more precisely in the experiment in Section 3 and implemented in the tool presented in Section 4. Other machine learning algorithms like Maximum Entropy Classification or Decision Trees are not taken into account for this work.

Additionally, rule based approaches are presented out of which one is also part of the experiment and the implemented tool.

2.3.1 Support Vector Machines

The machine learning approach of Support Vector Machines has been implemented in several systems and is found to be performing well for classifying texts for their sentiment. (Syamlal & Bruins, 2007; Pang et al., 2002)

Support Vector Machines represent the data points as feature vectors in a feature space. The algorithm fits a hyperplane separating the data such that the distance of the hyperplane and the data is maximal. Figure 2.2 shows an example of such a hyperplane separating data points linearly. We can see that the data is not linearly separable. To still separate the data points correctly, the hyperplane needs to be bent. This can be achieved by applying the so called *kernel trick*. This method is based on a transformation of the feature space into a higher dimension such that the data becomes linearly separable. The feature space is afterwards transformed back into the original dimension. The resulting hyperplane is not linear any more but correctly separates the data into the different classes. New data points are classified by computing their feature vectors and assigning the class according to which side of the hyperplane the data point is found on in the feature space.

There may be measuring errors in the data or the data points may overlap naturally. To account for this, the soft margin method introduced by Cortes and Vapnik (1995) allows for such errors. The slack variables ξ_i are introduced to

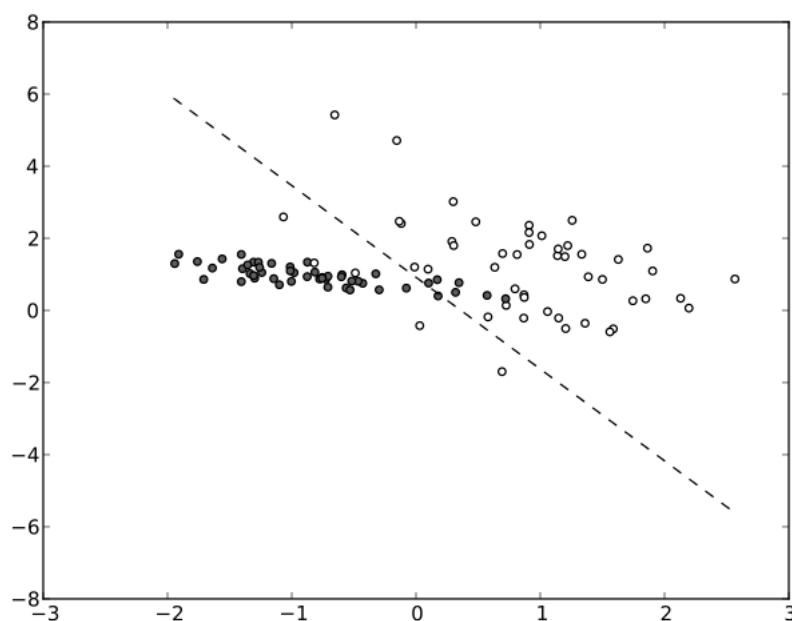


Figure 2.2: Scatterplot Featuring a Linear Support Vector Machine’s Decision Boundary (Dashed Line)⁷

measure the number of training errors and eventually minimize their effect on the training of the classifier.

Basically, Support Vector Machines allow only two classes. It is still possible to create Support Vector Machines which can handle more classes. This can be achieved by calculating hyperplanes hierarchically per class while treating all other classes as one opposite class.

2.3.2 Naïve Bayes Classification

The method of Naïve Bayes Classification has also been implemented successfully for sentiment analysis. (Narayanan et al., 2013)

A Naïve Bayes Classifier is a maximum likelihood classifier based on the Bayes Theorem. Narayanan et al. described the following equations. Given the features of an unseen data point, the probability of a document d belonging to class c_i is given by

$$P(c_i|d) = \frac{P(d|c_i) \times P(c_i)}{P(d)} \quad (1)$$

⁷Linear-svm-scatterplot. (n.d.). [image] Available at: <http://upload.wikimedia.org/wikipedia/commons/4/46/Linear-svm-scatterplot.svg> [Accessed 29 Aug. 2014].

With the probability of the feature f_i belonging to class c_i as

$$P(f_i|c_i) = \frac{\text{count}(f_i) + k}{(k + 1) \times m} \quad (2)$$

$\text{count}(f_i)$ is the count of the feature f_i appearing in texts of class c_i in the training data set and m being the number of all features observed in the training data set. By adding the constant k , usually 1, to the feature count, features of the new data point not observed in the training data are accounted for. This step is called laplacian smoothing.

With the simplifying assumption of features in a text being conditionally independent of each other we get

$$P(c_i|d) = \frac{(\prod P(f_i|c_j)) \times P(c_j)}{P(d)} \quad (3)$$

with f_i as the individual features of a text. The output of the classifier is the class that the highest posterior probability was computed for.

2.3.3 Rule Based Approaches

In contrast to machine learning techniques, it is also possible to classify a text with predefined rules. These rules consist of an antecedent and a sentiment value as consequent. The antecedent can be a word or a more complex construct like a combination of words that need to appear in a certain distance.

A set of rules can be defined as a lexicon of positive words and a lexicon of negative words. An example for such a sentiment lexicon is one that is used for the general inquirer (Stone, Dunphy, & Smith, 1966) consisting of 1637 words classified as positive and 2007 words classified as negative. Another lexicon is used for the commercial system *LIWC*⁸ (Pennebaker, Chung, Ireland, Gonzales, & Booth, 2007).

Rules can be extracted from pre-classified texts or by estimating a resulting sentiment value for the antecedents in another way. Prabowo and Thelwall (2009) for example, implemented an algorithm which computed sentiment values of words based on hit counts of searches performed on Google. The used queries basically consisted of the word to compute a value for and a word associated with strong sentiment.

It is also possible to construct a lexicon with help of lexical semantic nets like *WordNet* (Miller, Beckwith, Fellbaum, Gross, & Miller, 1990). Hu and Liu (2004) and Kim and Hovy (2004) implemented algorithms which started with a seed set

⁸Linguistic Inquiry and Word Count

of a few sentiment bearing words and collecting all words that could be associated with these by traversing *WordNet*.

For classification, a text is scanned for the rules' antecedents. The simplest way of computing the sentiment is comparing the counts applied rules and assigning the class which was yielded to by the most applied rules. This kind of computing the sentiment was applied in the rule based approach presented in Section 3.2. If the consequences of the rules are numerical the sentiment value of the text can be computed with mathematical functions processing these values.

3 The Experiment

For this work an experiment was carried out which compared Support Vector Machines, Naïve Bayes Classification and a rule based approach. Subsequently, the different algorithms were combined to minimize the misclassification rate.

All computations were performed on a Laptop running an Intel Core i5 at 1.7GHz.

3.1 Resources

For setting up supervised machine learning algorithms like Naïve Bayes Classification and Support Vector Machines, a training data set is needed.

A further test data set is required for feature selection and to estimate the quality of the classification methods. Test data sets are also meaningful for non machine learning techniques.

One of the used data sets is a set of movie reviews from *IMDb*⁹ as introduced in Maas et al., 2011. This dataset contains 50000 reviews evenly split into a training set of 25000 reviews and a test set of 25000 reviews. The overall distribution of labels is balanced such that there are 25000 positive reviews and 25000 negative reviews, 12500 for training and 12500 for testing each.

The second data set used is the *Yelp academic data set*¹⁰. This set contains 330071 restaurant reviews with ratings from one to five stars. Reviews with five stars were assumed to be positive, reviews with one star were assumed to be negative. It is important to note that the correlation between star ratings and sentiment is not a perfect agreement. Five star ratings may contain negative features as well as one star ratings may contain positive features. However, by manually checking a small set of ten randomly selected reviews for each polarity, the assumptions made could be confirmed. One negative review which was an

⁹<http://www.imdb.com>

¹⁰The *Yelp academic data set* is available at: http://www.yelp.com/academic_dataset

edited review constituted an exception. The author cited his previous positive text and then added his new negative text. This phenomenon is an instance of thwarted expressions as presented in Section 2.2 and is not considered further in this work. The original data set contained 95807 reviews rated with five stars and 26383 reviews rated with one star. For comparison reasons, this data set was scaled down to a maximum number of 50000 texts. The first 25000 reviews rated with five stars and the first 25000 reviews rated with one star were taken according to their ordering in the original data set. Both sets of negative and positive texts then were split into 12500 texts for training and 12500 texts for testing. The rest of 72190 reviews with one or five stars was discarded. The ratio between positive and negative reviews in the original data was not kept. A classifier trained on an unbalanced data set would greatly weight features of the smaller class which is not desirable in the case of sentiment analysis. The leftover reviews could have been added to the test set. To keep the size of the different test data sets comparable, this was not done. Reviews rated with 2, 3 or 4 stars were not taken into account for this work.

A third data set of 50000 texts was composed from 25000 texts of the *IMDb* data set and 25000 texts of the *Yelp* data set, 12500 positive and 12500 negative texts each. This third data set also contains 25000 texts for training and 25000 texts for testing, evenly split in positive and negative texts.

3.2 Implemented Algorithms

The rule based algorithm implemented for the experiment is based on the sentiment lexicon presented by Stone et al. (1966), as explained in Section 2.3.3. The input text needs to be split into tokens whereupon these tokens need to be lemmatized to match the dictionaries' entries. Additionally, a basic negation handling tries to detect inverting words. When the strings *not*, *n't* or *no* were found, the following words were given a prefix marking them as negated until a delimiter was found which revoked negation. Based on this, all features found for one class were added to the opposed class with the same added prefix marking the feature as negated. The sentiment value is then computed on the counts of sentiment bearing words found in the text. If no such word is found in the text the sentiment value *neutral* is assigned. When the counts for negative and positive words are equal, the sentiment value *mixed* is assigned. The two sets of words have an overlap of 15 words¹¹ which are considered as conveying mixed

¹¹Words appearing in both word lists: *arrest, board, deal, even, fine, fun, hand, help, hit, laugh, make, matter, order, particular, pass*.

sentiment and do not have an influence on the counts.

Based on the implementation of Naïve Bayes Classification presented in Narayanan et al., 2013 the features to use for both of the machine learning approaches were examined. A slight change was made by changing the tokenization step to the built-in tokenization method of *NLTK* and filtering stop words for unigrams based on *NLTK*'s stop word list for english. This way the basic implementation achieved an accuracy of 87.5% when trained on tested on the *IMDb* data set. The computation time of training was about 3.5 minutes. Lemmatizing the tokens decreased that accuracy to 87.2% while training time was more than threefold the initial training time. The time consumption is attributable to the process of Part of Speech tagging which is needed for correct lemmatization. Extending the feature sets by token fourgrams also resulted in less accurate classification with 87.4% accuracy while training time increased by about 50 seconds.

Based on these results, the features taken into account for the experiment were token uni-, bi- and trigrams with basic negation handling. As the findings in Narayanan et al., 2013 propose, the features extracted from the training sets were restricted to the top 32000 features based on mutual information. This restriction also keeps training time of the classifiers reasonable.

Since the implementations differ, other features and feature selection methods could achieve better results. However, the features were kept the same for comparison reasons.

The used Naïve Bayes Classifier is the built-in algorithm provided by *NLTK*. The utilized Support Vector Machine is an implementation offered by the *sci-kit learn* library. All default values of slack variables were retained.

After reading in the data and performing all preprocessing steps, solely training took 26.67 seconds for the Naïve Bayes Classifiers and 16.92 seconds for the Support Vector Machines, both in average.

3.3 Results

The different classifiers were all tested on the three test data sets.

Table 3.1 shows the accuracy of the different Naïve Bayes Classifiers. For the Naïve Bayes Classifiers we can see that the classifier trained on the *IMDb* training data set yielded the best results when averaged over the test data sets. The lowest accuracy is obtained when applying the classifier trained on the *Yelp* data set to the *IMDb* test set.

When examining the results closer, the source for this low accuracy is found. The balance of correct classifications for the two classes made by the Naïve Bayes

	<i>IMDb</i> Data	<i>Yelp</i> Data	Combined Data	Average
trained on <i>IMDb</i>	87.676 %	84.136 %	85.976 %	85.929 %
correct positive	85.688 %	74.400 %	80.072 %	80.053 %
correct negative	89.664 %	93.872 %	91.880 %	91.805 %
trained on <i>Yelp</i>	61.104 %	85.384 %	73.272 %	73.252 %
correct positive	23.144 %	72.144 %	47.536 %	71.412 %
correct negative	99.064 %	98.624 %	99.008 %	98.899 %
trained on Comb.	81.432 %	87.608 %	84.424 %	84.488 %
correct positive	67.096 %	77.096 %	71.896 %	72.029 %
correct negative	95.768 %	98.120 %	96.952 %	96.947 %

Table 3.1: Accuracies of Naïve Bayes Classifiers on different test sets

Classifiers is also shown in Table 3.1. The Naïve Bayes Classifier trained on the *Yelp* training data set has a strong tendency towards assigning the texts to the negative class. Correct assignments for the positive class are rare and the accuracy for assigning the class positive to a post drops as low as 23% for the *IMDb* test data set. That is, this classifier has a high recall for the negative class, while its precision in general is very low. The recall of a classifier is measured depending on how many data points it can assign correctly to a class without missing to label data points of this class. Precision in contrast shows how many wrong classifications are made.

The Naïve Bayes Classifier trained on the combined training data set showed similar but less noticeable behaviour. This is due to the combined training data set including a subset of the *Yelp* training data set.

Furthermore, all of the Naïve Bayes Classifiers are struggling to assign the positive class correctly when tested on the *Yelp* test data set.

In Table 3.2, the accuracies of the different Support Vector Machines are listed. The Support Vector Machine trained on the combined data set has the highest average accuracy. Only the accuracy of the classifier trained on the *Yelp* data set applied to the *IMDb* test set is below 80%. Even that accuracy is almost 77% which is fairly good. Some of the accuracies even exceed 90%. Compared to the Naïve Bayes Classifiers, only the Support Vector Machine trained on the *IMDb* data set is slightly inferior to its counterpart.

The tendency towards assigning one of the classes observed for the Naïve Bayes Classifiers is not present for the Support Vector Machines. Only the classifiers trained on the *Yelp* training data set and the combined training data set seem have slight difficulties of assigning the class positive correctly, when tested on the

	<i>IMDb</i> Data	<i>Yelp</i> Data	Combined Data	Average
trained on <i>IMDb</i>	85.812 %	80.420 %	83.132 %	83.121 %
correct positive	85.728 %	80.784 %	83.184 %	83.232 %
correct negative	85.896 %	80.056 %	83.080 %	83.011 %
trained on <i>Yelp</i>	76.976 %	92.152 %	84.344 %	84.491 %
correct positive	73.064 %	92.592 %	82.496 %	82.717 %
correct negative	80.888 %	91.712 %	86.192 %	89.597 %
trained on Comb.	84.296 %	91.276 %	87.696 %	87.756 %
correct positive	67.096 %	91.720 %	87.984 %	82.267 %
correct negative	95.768 %	90.832 %	87.408 %	91.336 %

Table 3.2: Accuracies of Support Vector Machines on different test sets

IMDb test data set. The Support Vector Machines have a higher precision with slightly higher recall.

We can also see that the Support Vector Machines performing best on a test data set are the ones trained on the corresponding training data set. This is an indicator of the Support Vector Machines overfitting for the domain they are trained on. Since all slack variables were retained in this experiment, an adjustment of these may prevent this in further studies.

The results of the rule based approach are shown in Table 3.3.

Compared to the results of the machine learning techniques, the results of the rule based algorithm are relatively poor. In contrast to a baseline classification assigning the same class to all texts in the training set (50% accuracy), these results are still good when taking into account the very easy implementation. When comparing the balance of the algorithm correctly assigning the positive class and correctly assigning the negative class, we can observe that barely half of the negative texts were assigned for all three test sets. This may indicate slighter coverage of the negative scope than the positive.

Note that the rule based approach does also assign the classes neutral to the texts, if the counts of positive and negative words are zero or the class mixed if the counts are greater than zero and equal. The machine learning classifiers do always assign one of the classes positive or negative to the data points, since they were only trained on these two classes. This way, a comparison of the techniques has only limited validity.

The results achieved on the combined training data set are close to the mean of the results on the other two data sets for all the measures shown above. This is

	<i>IMDb</i> Data	<i>Yelp</i> Data	Combined Data	Average
Overall accuracy	63.676 %	64.740 %	63.864 %	64.093 %
correct positive	73.408 %	78.344 %	75.568 %	75.773 %
correct negative	53.944 %	51.136 %	52.160 %	52.410 %

Table 3.3: Accuracy of rule based approach on different test sets

	Correct	Misclassifications	No Assignment
NB + Rules	56.148 %	4.832 %	39.020 %
SVM + Rules	58.080 %	5.248 %	36.672 %
NB + SVM	78.776 %	5.104 %	16.120 %
NB + SVM + Rules	52.596 %	2.308 %	45.096 %

Table 3.4: Accuracies and misclassification of combined classifications tested on the combined test data set

not surprising since the combined data set consists of evenly sized subsets of the other two data sets.

An additional calculation was carried out by combining the different classifiers, such that a sentiment value was only assigned when the outputs of the classifiers combined coincided. Table 3.4 shows the accuracies and misclassification rate of these combined measurements performed on the combined test data set. For this computation, the Naïve Bayes Classifier trained on the *IMDb* training data set and the Support Vector Machine trained on the combined training data set were chosen, since they yielded the best averaged results in previous computations. By combining the different approaches, we can achieve a great improvement in precision. We can see that the misclassification rate for the combined data set dropped considerably from a maximum of about 36% for rule based classification and a minimum of about 12% for Support Vector Machines to maximally 5.22% with a combination of exactly these two techniques. However, the recall is also decreased to a minimal value of 52.6% for the combination of all three algorithms.

3.4 Conclusion

Based on the achieved results, the machine learning techniques perform better than the rule based approach. Furthermore, the Support Vector Machines are slightly better than Naïve Bayes classification in most cases. The implementations of the used algorithms are very basic. Some simple adjustments may already yield better results but are not covered in this work.

Combining different algorithms can help increasing the precision greatly. But this

improvement comes at the cost of a decreased recall. On the one hand, we can see that only a small number of data points is assigned with a wrong category, which is an indicator for high precision. On the other hand the combined approaches end up with up to 45% of the posts left unclassified. This shows that the recall has dropped tremendously. The measure that may be compared with this rate of non-classified data points is the percentage of misclassified data of the previous computations. Checking this number for all the classifiers tested on the same test data set (combined data set) we find the Naïve Bayes Classifier trained on the *Yelp* training data set performing worst by misclassifying about 27% of the data. This is a decline of 18 percentage points.

Assigning weights to the outputs of the combined classifiers may counteract this effect. By introducing a threshold of confidence for all classifiers, the overall result could be computed based on these confidence levels. As a very basic example, if one of the classifiers could assign a class with a high level of confidence, the other results would not need to be considered. We can also see, that the best results of the combined approaches is achieved when merging the results of the two machine learning classifiers. This combination has the highest amount of correctly assigned classes and the least amount of unclassified data points.

3.5 Future Work

To improve the obtained results, the sentiment lexicon used for the rule based approach could be extended and adjusted. The rules could also be extended to taking into account more words which lead to a sentiment value when appearing in a specified proximity or even more complex structures. A way of adding single words to the lexicon is implemented in the tool presented in Section 4.

While the effect of negation is taken into account, this work does not consider the effect of valence shifters like intensifiers and diminishers. Kennedy and Inkpen (2006) show that these can also have an impact on the obtained results. The implemented negation handling is very basic and could additionally be improved further.

Moreover, the data sets used for training could be improved by manually checking their grading based on the star ratings. Because of the large size of these data sets this might be very time and cost intensive and therefore not feasible. Especially the *Yelp* data seems to pose difficulties which are beyond the scope of this work. These difficulties could come from an incorrect mapping of star ratings and sentiment values as described in Section 3.1.

Since the used data of online reviews usually also contains emoticons (:-), :-

(, ;), ...) and the built-in tokenization of *NLTK* which is used fails for these, the tokenization method presented in O'Connor, Krieger, & Ahn, 2010 called *ttokenize*¹² could be used. However, when taking into account token bigrams and trigrams as done for the machine learning techniques, emoticons split into separate tokens are considered as one again and this improvement is unnecessary. Then again, this does not hold for the rule based approach.

Wang and Manning (2012) report that Naïve Bayes Classifiers perform better on short texts and Support Vector Machines perform better on long texts. This findings could be measured more precisely and improve the results. Other aspects of a text could enhance the results of the different approaches further.

The feature selection performed before training of the machine learning classifiers is a baseline based on Narayanan et al., 2013. The features could also be selected by the TF-IDF measure or with the methods presented by O'Keefe and Koprinska (2009).

For an extension of the sentiment scale covered by the machine learning techniques, the ratings with three stars in the *Yelp* data set could be used for building a corpus of reviews conveying neutral or mixed sentiment. This would again require for manual work.

4 Online Sentiment Analysis Tool

Based on the experiment presented in Section 3 a web application was implemented. This application is hosted at *valentind.pythonanywhere.com* and the source code can be found at *www.github.com/valentindey/sentiment_webapp*.

The presented tool might be useful for users who need to investigate sentiment and want to grasp the functionality of the underlying model better. The users could come from different fields, including researchers interested in the function and performance of the machine learning algorithms used, or even psychologists studying the writings of an author.

The purpose of this tool is to provide a way of classifying a single text for its sentiment based on all three presented classification methods and to visualize corresponding output texts. The user may enter a text and adjust several settings to their needs. Capitalization of the input words is ignored.

When the text is analyzed, the overall sentiment is shown, which is the outputted sentiment value of all three classifiers if these values match, mixed otherwise. Additionally the result and an output text of each individual classifier is shown.

¹²*Ttokenize* is available at: <https://github.com/brendano/tweetmotif>

The user may select the machine learning classifiers used for classification of the text by selecting the appropriate training data set. For adjusting the output of the rule based approach, positive and negative words can be specified. By adding a word to both categories, it is treated as conveying mixed sentiment. Adding emoticons to the word lists will not affect the output since punctuation in the input text are separated by the tokenizer (see Section 3.5). Moreover, the words entered should be the lemmas of the words. For example entering a plural form of a word will not match the exact same plural form in the input text. Rather the singular form needs to be entered, because the lemmas of words in the input text are compared. If words treated as negative polarity items are entered, they will not affect the output. Their information content as negating word is rated higher. Negative polarity items mean words that change the orientation of the following words as described in Section 3.2.

In contrast to most implemented tools, which often display a summary of results for a set of analysed documents, this application visualizes the results on text level. There is an option to set the color that positive words, negative words, words with mixed sentiment and negative polarity items are displayed in. Neutral words are always colored black. For the rule based approach the colorization applies to the words as specified in the initial dictionary (see Section 3.2) and the words additionally defined by the user.

Due to the classifiers' training on the same feature set, the output text of both machine learning techniques share the same colorization. This colorization is based on how many times a word in the input text appeared in features observed during training time for each of the classes.

The colors for the different words can be specified by the user¹³

The visualization aims to provide a way to see which parts of a text were computed as relevant features for training the classifiers. By playing around with the tool, we can find that an exclamation point was computed as a positive feature by training on the *Yelp* training data set and the combined training data set. Based on such findings further improvement suggestions for the implemented algorithms can be made. However, the colorization may not coincide with the result of classification.

¹³The color picker used is available at: <http://www.menucool.com/color-picker>

4.1 Future Work

To save the words specified by a user, a feature to store defined word lists could be implemented.

A feature allowing to rate the outputs could be implemented and based on these ratings and the corresponding input texts, the classifiers could be retrained.

The output texts are created by concatenating the tokens as they are split by the tokenizer with intermediate white spaces. Correctly reassembling the input text would make the output look nicer, while it would not change anything in the underlying algorithms.

5 Acknowledgments

I would like to thank Ph.D. Christopher Culy for his advice and support and all my fellow students and friends for helping me with this work by discussions, proofreading and their general support.

References

- Argamon, S., Koppel, M., Fine, J., & Shimoni, A. R. (2003). Gender, genre, and writing style in formal written texts. *To appear in Text*, 23, 3.
- Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python*. " O'Reilly Media, Inc."
- Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1), 1–8.
- Bradley, M. M., & Lang, P. J. (1999). *Affective norms for english words (anew): Instruction manual and affective ratings*.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273–297.
- Da Silva Cardoso, H. K. (2013). Sentiment analysis of correspondence corpora.
- Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al. (2003). *A practical guide to support vector classification*.
- Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining* (pp. 168–177).

- Kennedy, A., & Inkpen, D. (2006). Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22(2), 110–125.
- Kim, S.-M., & Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the 20th international conference on computational linguistics* (p. 1367).
- Kramer, A. D. I., Guillory, J. E., & Hancock, J. T. (2014). Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences*, 201320040.
- Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011, June). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (pp. 142–150). Portland, Oregon, USA: Association for Computational Linguistics. Available from <http://www.aclweb.org/anthology/P11-1015>
- Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database*. *International journal of lexicography*, 3(4), 235–244.
- Mukherjee, A., & Liu, B. (2012). Modeling review comments. In *Proceedings of the 50th annual meeting of the association for computational linguistics: Long papers-volume 1* (pp. 320–329).
- Mukherjee, S., & Bhattacharyya, P. (2013). Sentiment analysis: A literature survey. *arXiv preprint arXiv:1304.4520*.
- Narayanan, V., Arora, I., & Bhatia, A. (2013). Fast and accurate sentiment classification using an enhanced naive bayes model. In *Intelligent data engineering and automated learning-ideal 2013* (pp. 194–201). Springer.
- O'Connor, B., Krieger, M., & Ahn, D. (2010). Tweetmotif: Exploratory search and topic summarization for twitter. In W. W. Cohen & S. Gosling (Eds.), *Icwsn*. The AAAI Press.
- O'Keefe, T., & Koprinska, I. (2009). Feature selection and weighting methods in sentiment analysis. *ADCS 2009*, 67.
- Pang, B., & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on association for computational linguistics* (p. 271).

- Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the acl-02 conference on empirical methods in natural language processing-volume 10* (pp. 79–86).
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Pennebaker, J. W., Chung, C. K., Ireland, M., Gonzales, A., & Booth, R. J. (2007). The development and psychometric properties of liwc2007. *Austin, TX, LIWC. Net*.
- Prabowo, R., & Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics*, 3(2), 143–157.
- Russell, J. A. (1980). A circumplex model of affect. *Journal of personality and social psychology*, 39(6), 1161.
- Sidorov, G., Miranda-Jiménez, S., Viveros-Jiménez, F., Gelbukh, A., Castro-Sánchez, N., Velásquez, F., et al. (2013). Empirical study of machine learning based approach for opinion mining in tweets. In *Advances in artificial intelligence* (pp. 1–14). Springer.
- Stone, P. J., Dunphy, D. C., & Smith, M. S. (1966). The general inquirer: A computer approach to content analysis.
- Syamlal, R., & Bruins, J. (2007). Sentiment classification using a svm.
- Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 417–424).
- Wang, S., & Manning, C. D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2* (pp. 90–94).