Sentiment Classification in Social Media

An Analysis of Methods and the Impact of Emoticon Removal

ANDREAS PÅLSSON

DANIEL SZERSZEN
Sentiment Classification in Social Media

An Analysis of Methods and the Impact of Emoticon Removal

ANDREAS PÅLSSON
DANIEL SZERSZEN

Degree Project in Computer Science, DD143X
Supervisor: Richard Glassey
Examiner: Örjan Ekeberg

CSC, KTH May 2016
Abstract

Sentiment classification is the process of analyzing data and classifying it based on its sentiment conveying properties and the process has a multitude of applications in different industries. However, the different application areas also introduce diverse challenges in implementing the methods successfully. This report examines two of the main approaches commonly used for sentiment classification which entail the use of machine learning and a glossary of weighted words respectively. In addition, preprocessing is explored as an enhancement to the previously mentioned approaches. The approaches are tested on data collected from Twitter to examine their performance in social media. The results indicate that lexicon-based classifiers are the most performant, and that removal of emoticons increases the correctness of classification.
Referat

Contents

1 Introduction ................................................. 1
   1.1 Problem Statement .................................... 1
   1.2 Research Question .................................... 1
   1.3 Hypothesis ............................................ 2
   1.4 Scope .................................................. 2
   1.5 Structure .............................................. 3

2 Background .................................................. 5
   2.1 Sentiment Classification .............................. 5
   2.2 Methods of Sentiment Classification ............... 6
      2.2.1 Learning-based Classifiers ...................... 6
      2.2.2 Lexicon-based Classifiers ...................... 9
   2.3 Measuring Performance .............................. 10
      2.3.1 Recall ............................................. 10
      2.3.2 Precision ......................................... 10
      2.3.3 F-measure ......................................... 10
   2.4 Related Work ......................................... 11

3 Method ...................................................... 13
   3.1 Programming Frameworks ............................. 13
   3.2 The Data .............................................. 13
   3.3 Preprocessing ......................................... 14
   3.4 Lexicon-based classifiers ............................ 14
      3.4.1 Neutrality Thresholds ......................... 15
   3.5 Learning-based classifiers ........................... 15
      3.5.1 Naive Bayes .................................... 15
      3.5.2 Support Vector Machine ....................... 15
      3.5.3 Maximum Entropy ................................ 15

4 Results ..................................................... 17
   4.1 VADER ................................................. 17
   4.2 VADER preprocessed .................................. 18
   4.3 Naive Bayes .......................................... 19
Chapter 1

Introduction

Social media is a growing source of data and information spread. However, the information is convoluted with varying interests, opinions and emotions. Moreover, the form of communication lacks standardized grammar, spelling, use of slang, sarcasm and abbreviations, and more. These variables can make extracting critical points, facts, and the sentiment of the message difficult in situations where a number of these aspects are present. Through natural language processing (NLP) it is possible to study and analyze these messages and objectively classify sentiments presented in social media.

Sentiment classification is the task of labeling data with a polarity through analysis of the properties contained within the data. Classification can be binary, meaning either positive or negative, or describe a detailed range of polarity at the expense of increased implementation complexity. Social media increases the complexity of the problem, necessitating analysis of informal communication which does not necessarily adhere to any grammatical or contextual rules. An interesting aspect of this topic is the difference between spoken and written language and evaluating which variables are the most important in conveying sentiment in written form.

1.1 Problem Statement

Sentiment classification in social media is difficult due to the informal nature of the communication. The informal nature introduces additional variables and properties that have to be evaluated compared to formal texts, necessitating additional resources spent on annotating the data and training the classifiers.

1.2 Research Question

The aim of this report is to investigate the performance of a number of sentiment classifiers on data from social media, namely the lexicon-based classifier VADER and the learning-based approaches of naive Bayes (NB), Maximum Entropy (Max-
CHAPTER 1. INTRODUCTION

Ent) and Support Vector Machine (SVM). In addition, the classifiers will evaluate
differently processed data sets to examine the effects of emoticons on their perfor-
mance. Therefore, the main research question is:

- Which classification approach performs best when evaluating social media
texts?

Moreover, the presence or absence of emoticons is of interest when classifying
data from social media, therefore the following secondary questions are also asked
to examine their effects on classification performance:

- How do emoticons affect classification performance for different lexicon and
  learning-based classifiers?

- Is preprocessing a successful method to improve classification independent of
  implementation approach?

1.3 Hypothesis

The machine-learning classifiers are expected to outperform the lexicon-based ap-
proach due to their performance in pattern recognition when trained correctly. This
attribute should prove useful in recognizing strong sentiment-laden properties in the
informal language and lead to improved classification results.

In addition, the better prepared data sets are expected to improve classification
performance. More specifically, classification of preprocessed sets should outperform
their unprocessed counterparts, considering that a generalized vocabulary is easier
to process. Though emoticons provide strong emotional connotation, the noise they
generate lessens the ease of classification.

1.4 Scope

The scope of the report is limited to investigating the performance of classifiers on
differently processed data sets of Tweets1 and examining the effects of emoticons on
sentiment classification. The classifiers include a lexicon-based and a learning-based
approach, limited to three different algorithms for the learning-based classifier. The
algorithms explored are naive Bayes, Maximum Entropy and Support Vector Ma-
chine. A single sentiment lexicon is utilized for the lexicon-based classifier VADER.
Classification is tested on data set consisting of 4200 manually annotated Tweets.
Two variations of this data set are used: an unprocessed set with emoticons included,
and a preprocessed set where emoticons have been replaced with sentiment-laden
words corresponding to the sentiment value of the emoticon.

1https://twitter.com
1.5 Structure

The report is structured into six main sections consisting of Background, Method, Results, Discussion, Future Research and Conclusion. Background details all the essential information surrounding the theory and state-of-the-art of the topic in order to assist with understanding the following sections of the report. It also includes a subsection dedicated to related work outlining similar or related research on the topic. Method describes the necessary procedure of preparing and implementing the sentiment classifiers with the accompanied data set. The performance of the classifiers on the different data sets is presented in Results. An attempt to explain the results, propose possible improvements and follow-up investigations is included in the Discussion. Following the discussion is a section detailing Future Research that could be pursued. Finally, the Conclusion answers the research questions based on the results of the investigation.
Chapter 2

Background

2.1 Sentiment Classification

Text classification refers to the automated process of dividing and labeling of units of texts into separate, predefined categories, also known as classes. Text classification can be used to extract the topic from a text but can also include sentiment classification. Sentiment classification, or sentiment analysis, relates to the polarity classification of a text, i.e. deciding whether the text is positive, negative or neutral. NLP is used to systematically examine and evaluate the sentiment conveyed in text and label it with a corresponding sentiment class.

The popularity of social media has increased the interest and importance of sentiment classification (Kiritchenko et al 2014). The substantial amounts of data which they produce increase the need for an automated process of structuring and categorizing the data, which has potential for a multitude of commercial, political and social applications. Seeing as 97% of comments on MySpace contain non-standard formal written English (Thelwall 2009), being able to correctly classify informal text is becoming increasingly important. These areas include, but are not limited to, trend recognition, market prediction, spam and flame detection, decision making and popularity analysis. The availability of the data can allow businesses to analyze their customers in larger volumes and detect if their product is in the target domain, while consumers can gain access to information about their interests allowing for more informed choices (Pang, Lee, 2008).

Certain approaches to sentiment analysis separates the problem into a two-step process (Wiebe et al, 2005; Yang & Cardie 2013). First the text is analyzed to determine if it is objective or subjective. The subjective texts are then classified as positive, negative or neutral (Yu & Hatzivassiloglou, 2003). However, this approach can cause additional errors in classification in cases where a text is mislabeled as either objective or subjective. In addition, objective texts are not necessarily free of sentiment-laden statements and could thus possibly be handled by the subjective classifier (Kiritchenko et al., 2014). The neutral class has often been omitted in research or included in the positive and negative classes, but results have shown
that distinguishing neutral cases from the positive and negative classes can yield increased classification accuracy (Koppel and Schler 2006).

2.2 Methods of Sentiment Classification

There are two main approaches to sentiment classification: lexicon-based and machine-learning. A lexicon-based approach tokenizes data into individual words which are checked with a sentiment lexicon containing a polarity value for individual words. The sum of the polarities are passed to an algorithm that determines the overall polarity of the sentence. A machine-learning approach utilizes a labeled training set to adapt a classifier to the data domain of the training set. The trained classifier can then predict the outcome of the problem and the success rate of the prediction depends on how well the problem is contained within the same domain.

2.2.1 Learning-based Classifiers

Learning-based sentiment classifiers have their foundation in the machine learning branch of artificial intelligence. These classifiers require preliminary work to train the classifier with a training set, necessitating manual annotation of the features and sometimes overall sentiment of each sentence in the set. NB, MaxEnt and SVM are three standard algorithms shown to be effective in learning-based classification of text. Additional features, e.g. unigrams, bigrams and feature frequency, can also be implemented alongside the algorithms and have proved to be successful in increasing the classification accuracy (Pang, Lee, 2002).

Learning-based classifiers depend on learning domain specific knowledge in order to correctly classify text. These are all supervised algorithms, meaning that they need to be pre-trained with annotated training sets in order to correctly and accurately classify texts. In order to accurately predict text with the correct class, the classifier needs to be trained with data from that particular domain. Sentiment classifiers trained in one domain do therefore not perform well in other domains (Aue, Gamon, 2005). Therefore, learning-based classifiers might not be as suitable for analyzing newly created or discovered areas which limits their applicability.

Learning-based classifiers perform better than the lexicon-based counterparts when used in the domain they are trained for. (Musto et al, 2014; Pang, Lee, 2004). Therefore, using a learning-based classifier is favorable if the use case and domain of analysis is known beforehand.

Feature Extraction

Feature extraction refers to the process of extracting features (e.g. words, sequences of words) from text. In learning-based algorithms these features are extracted and put into a bag-of-features, a data set suitable for machine learning algorithms.

Term frequency is another feature that often is an accurate indicator of class belonging, e.g. text containing many occurrences of “happy” is likely of positive
2.2. METHODS OF SENTIMENT CLASSIFICATION

sentiment. However, longer texts naturally have larger amounts of word occurrences, and the terms occurring very frequently are less informative than features that occur rarely. These features need to be weighed accordingly which is known as tf-idf (term frequency times the inverse document frequency).

Naive Bayes

A naive Bayes classifier utilizes a simple methodology to calculate the probability of a class belonging to a text. Its basis lies on the naive assumption that the features of a text are independent. This assumption and Bayes’ theorem form the basis of an NB classifier. In the following equation $C_k$ is a class and $x$ is a feature vector.

\[
p(C_k|x) = \frac{p(C_k)p(x|C_k)}{p(x)} \tag{2.1}
\]

Despite the classifiers’ simplicity it can compete with other state of the art sentiment classifiers, and can be considered a strong competitor in the field as it is easy to implement and achieves high performance scores (Rennie et al., 2003).

Given the assumption of independence between features given the class $C_k$, the following proportionality is obtained:

\[
p(C_k|x_1,\ldots,x_n) \propto p(C_k) \prod_{i=1}^{n} p(x_i|C_k) \tag{2.2}
\]

Under the above independence assumptions, the probability of features belonging to class $C_k$ can be calculated as follows:

\[
p(C_k|x_1,\ldots,x_n) = \frac{1}{Z} p(C_k) \prod_{i=1}^{n} p(x_i|C_k) \tag{2.3}
\]

From the calculations of probabilities above, a classifier is created:

\[
y = \arg\max_{k \in 1,\ldots,K} p(C_k) \prod_{i=1}^{n} p(x_i|C_k) \tag{2.4}
\]

A number of different NB classifiers exist, and the ones most suitable for classifying texts are Bernoulli NB and multinomial NB. The two classifiers differ in how the features are represented. In Bernoulli classifiers features are independent binary variables (booleans), whereas a multinomial NB classifier uses feature vectors representing the frequencies of feature occurrences.

Bernoulli classifiers have been shown to outperform the multinomial classifiers when the vocabulary of the data set is small (McCallum et al, 1998). However, term frequency is of greater importance when classifying longer texts with a larger vocabulary, and so for longer texts the multinomial variant is preferred (McCallum et al, 1998).
Support Vector Machine

Support Vector Machine is a supervised learning model that analyzes data and is widely used in text classification (Taboada et al., 2011). SVMs evaluate text by separating data linearly in a high dimension feature space, i.e. given training data, annotated with the data’s class, an SVM algorithm builds a model that represents the data as points in space. The points are mapped so that the gap, represented by a vector hyperplane, between the distinct classes is as large as possible. New test data is mapped to the same space and classified by determining on which side of the hyperplane they fall.

Maximum Entropy

Maximum Entropy is an alternative method of calculating probability of class belonging. The method’s underlying principle is that distributions that are uniform should be preferred (Nigam et al., 1999). As opposed to the NB method, MaxEnt makes no assumptions about feature independence. This is more in line with intuition and MaxEnt has been shown to be effective in a number of different NLP applications (Berger et al, 1996), and sometimes outperforms NB in standard text classification (Nigam et al., 1999). The probability of data \( d \) belonging to class \( c \) is in a MaxEnt estimation is as follows:

\[
P_{ME}(c|d) := \frac{1}{Z(d)} \exp\left(\sum_{i=1}^{\lambda_i c F_{i,c}(d,c)}\right)
\]  

where \( F_{i,c}(d,c) \) is a feature function, \( Z(d) \) is a normalization function and \( \lambda_i \) is the feature weight. A large \( \lambda_i \) indicates that \( f_i \) is considered a strong indicator for class \( c \). The parameter values are set to maximize the entropy of the distribution while still following the constraints set on the distribution by the training set. The parameters that yield the maximum entropy given a set of constraints are calculated using a hillclimbing algorithm (Berger et al, 1996).

\[\text{Source: } \text{http://scikit-learn.org/stable/modules/svm.html}\]
2.2. METHODS OF SENTIMENT CLASSIFICATION

2.2.2 Lexicon-based Classifiers

A lexicon-based classifier only needs a polarity document, containing words and their semantic orientation, and do not need to be trained or otherwise processed before use. The polarity document is a list of words and their respective semantic orientation (Taboada et al 2011). The sentiment classification result is presented as binary positive or negative score, a neutral score at times also included. The lexicon-based classifiers are simpler to create and implement than the learning-based, because the need of a pre-annotated training set is not required. Although they do not perform as well as the learning-based classifiers, there are still benefits of using a lexicon-based classifier. In contrast to a learning-based classifier, lexicon-based classifiers do not have any domain specific knowledge. Learning based classifiers’ performance drops significantly when used in a different domain than that for which they are trained (Aue, Gamon, 2005), whereas a lexicon-based approach remains unaffected.

Negation Words

One of the major drawbacks of using a lexicon-based classifier is that the context in which the text is found is often neglected. Consider the following example:

This restaurant was actually neither that good, nor super trendy.

Even though the sentence carries an overall negative sentiment, the words that will carry the most weight are “good” and “trendy”, and hence the sentence’s class will be predicted as positive. The problem is that the classifier has no concept of negation and intensification. One possible solution is to invert the polarity of a word when it is found next to a negation word, e.g. not. However, one problem is that negation words are often placed long before the word they are negating. In the example above, neither “neither” nor “nor” is next to the word they are negating. A possible improvement that might fix this is looking for negation words before the word until a clause boundary marker is found (Taboada et al., 2011).

Intensifiers

Intensifiers are words that alter the polarity value of another a phrase, e.g. “super”, “slightly” and are largely ignored in lexicon-based approaches. These words are important for classifying the sentiment value of a phrase, seeing as “very good” has a higher polarity value than “good”. A possible solution is to increase the polarity value of a word when placed next to an amplifier (e.g. “very”, “really”), and decrease it when found next to a downtoner (e.g. “slightly”, “somewhat”).

Increasing a lexicon-based classifier’s ability to handle negations and intensifiers greatly improves on its weaknesses, and consistently scores high accuracy on texts from different domains, despite the classifier’s simplicity (Taboada et al., 2011).
2.3 Measuring Performance

The simplest measure of performance regarding text classification is accuracy. Accuracy is calculated as the ratio of correct classifications divided by total classifications. However, accuracy is not a good indicator of performance if data is unbalanced. A high accuracy can be achieved by predicting according to how the data is skewed, e.g. if a set contains 95% negative Tweets, an accuracy of 95% can be achieved trivially by predicting that all the Tweets are negative.

Recall and Precision are alternative measures to accuracy for determining classification performance. In contrast to accuracy, they are defined in terms of predicted and actual classes.

<table>
<thead>
<tr>
<th>Actual Class</th>
<th>Predicted Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pos</td>
<td>True Positive (TP)</td>
</tr>
<tr>
<td>Neg/Neu</td>
<td>False Positive (FP)</td>
</tr>
</tbody>
</table>

Figure 2.2. Confusion matrix showing the relationship between true positives, false negatives, false positives and true negatives

2.3.1 Recall

Recall is measured as the fraction of the relevant instances retrieved. However, since it does not take false positives into account, a perfect recall can be achieved by classifying every text as that class.

\[
\text{recall} = \frac{TP}{TP + FN}
\]  \hspace{1cm} (2.6)

2.3.2 Precision

Precision is measured as the probability of a predicted class being correct. It can be used in conjunction with recall to fill in the holes left by only considering a recall measure, i.e. to distinguish where data has been predicted incorrectly.

\[
\text{precision} = \frac{TP}{TP + FP}
\]  \hspace{1cm} (2.7)

2.3.3 F-measure

F-measure (or F-score) is a measure of performance that combines precision and recall by calculating their weighted harmonic mean that covers the flaws of accuracy with skewed data.

\[
F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}
\]  \hspace{1cm} (2.8)
2.4. RELATED WORK

Equation 2.8 is also known as the $F_1$-score because the relative weight of recall and precision is set to 1. Unless the experiments explicitly necessitate changing the relative weights, $F_1$ score is the default measurement for comparing performance in sentiment classification. A higher $F_1$ score, henceforth known as F-measure, indicates better performance than a lower score.

2.4 Related Work

The machine-learning methods mentioned in this thesis have normally been used for topic classification, i.e. determining whether a text is about politics or sports. Pang et al. (2002) adopted these techniques and regarded positive or negative sentiment as a topics of their own. With this approach they managed to achieve performances hovering around 80% when analyzing movie reviews. However, the experiments conducted only contained reviews that were considered positive or negative.

Agarwal et. al (2011) conducted sentiment analysis on Tweets using a SVM and reached an accuracy hovering around 60% depending on the features used. Moreover, they preprocessed the Tweets by replacing acronyms with their full meaning, and replacing emoticons with their emotional state. It is worth noting that the same method reached accuracies around 75% when conducting binary classification, ignoring the neutral class and only having positive or negative classes.

Bayani et al (2009) conducted experiments that utilized distant supervision on data extracted from Twitter. Distant supervision relates to gathering and labeling training data automatically. They worked under the assumption that any Tweet containing a positive emoticon, e.g. :) and :-), also contains positive sentiment, while any Tweet containing a negative emoticon contains negative sentiment. With these assumptions they could create an annotated classifier training set without hand-labeling data. However, they considered emoticons to be noise and stripped them off in the data set. Considering that emoticons are important for expressing moods, as reported by Gilbert et al. (2013) that they are largely used for assessing sentiment, a possible improvement to this approach would correctly include emoticons and their respective emotional connotation.

By using distant supervision to collect an annotated training set, training data that crosses over a large number of different domains can be collected. Aue and Gamon (2005) showed that learning-based classifiers’ accuracy dropped when used in a different domain than that for which they are trained, but by using distant supervision the classifiers can be trained with data sets large enough to include texts from many different domains, thus reducing the drop in accuracy and need for manual annotation.

Furthermore, text preprocessing has been utilized in order to more accurately classify sentiment in texts from social media. This is done to normalize the language and generalize the vocabulary with the intent of creating data that is easier for classifiers to process. Balahur (2013) employed the following methods to preprocess the Tweets:

11
• Removing repeated punctuation - Informal texts written on social media often contain multiple punctuation symbols, e.g. '!!!' or '??'. These were normalized to a single occurrence of the punctuation symbol, e.g. '!' and '?'.

• Emoticon replacement - By comparing the emoticon found in the Tweet with an emoticon sentiment dictionary, Balahur replaced emoticons conveying positive sentiment with the word “positive”, and emoticons conveying negative sentiment with “negative”. Emoticons considered neutral were removed.

• Slang replacement - Replacing slang is done with the intention of normalizing the language used in the text. This was done using a specialized site with replacements for slang words.

• Word normalization - Texts in social media often contain words that have been stressed by repeating some of the letters in the word. For example, querying Twitter for the word “haaate” generates a large amount of results. In order to deal with this, Balahur checked for the existence of the word in a dictionary, and if none were found, the stressed letters were removed until a word was found. For example “haaate” would become “haate”, and finally “hate”.

By utilizing these strategies of vocabulary normalization, Balahur (2013) reached an accuracy of 85% using a SVM that had been trained with data from social media. However, this was measured when conducting binary classification.
Chapter 3

Method

3.1 Programming Frameworks

The programming language of choice is Python 2.7.1 because of its wide adoption in the software development industry and the scientific world, as a result the language has well-supported libraries providing NLP tools. The NLTK\(^2\) and Scikit-learn\(^3\) libraries offer implementations of lexicon-based and machine-learning classifiers, including NB, SVM and MaxEnt and techniques for feature extraction.

3.2 The Data

A data set is required for testing the classifiers and measuring their performance. The classifiers in this report are tested on the VADER data set\(^4\) consisting of 4200 Tweets that have been manually annotated by trained individuals and represent the gold standard in sentiment annotation (Hutto, Gilbert, 2014).

The data set was chosen because of its relevance to the subject of sentiment classification in social media, while retaining the oft omitted informal features that are important in conveying sentiment (Davidov et al., 2010). Tweets are also characterized by their short length (maximum 140 characters), which imposes additional challenges for determining the sentiment of the Tweet (Kiritchenko et al., 2014). Moreover, the data set has already been manually annotated. This eliminates the need for distant supervision and the baseline to which the results are compared will be highly reliable. However, this also means that the tested classifiers are only tested in the specific domain of these Tweets.

The data set contains Tweets annotated with intensity scores ranging from -4 (extremely negative) to +4 (extremely positive) for each Tweet, whereas NLTK

---

\(^1\)https://python.org
\(^2\)version 3.2, http://nltk.org
\(^3\)version 0.17.1, http://scikit-learn.org
\(^4\)http://comp.social.gatech.edu/papers/
classifiers return sentiment values ranging from -1 to +1. In order to account for this, the polarity values are normalized to the [-1, 1] range.

Thresholds were set for positive, negative and neutral classes after inspecting the data set. A sentiment polarity value \( \geq 0.2 \) is considered positive, whereas a value \( \leq -0.2 \) is considered negative. Any value in the range \([-0.2, 0.2]\) is considered to be of neutral polarity. The thresholds were necessitated due to the classifiers results being reported as simply positive, negative or neutral but had to be compared to the values in the data set for validation.

The data set contains 4200 Tweets distributed over the three sentiment classes of positive, negative and neutral as follows:

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
<th>Neutral</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>1998</td>
<td>917</td>
<td>1285</td>
</tr>
<tr>
<td>%</td>
<td>47.6%</td>
<td>21.8%</td>
<td>30.6%</td>
</tr>
</tbody>
</table>

Table 3.1. The number of Tweets of each class and their relative size

3.3 Preprocessing

In order to determine the importance of emoticons when conveying sentiment, the classifiers will also be tested with a preprocessed set. The data will be preprocessed by replacing emoticons in the data set with “happy”, “neutral” or “sad”. These words were chosen for representing the average sentiment value of each class.

The lexicon used in the lexicon-based experiments contains over 200 emoticons, and they are all manually annotated with a sentiment score in the range \([-4, 4]\], normalized to the range \([-1, 1]\). Every Tweet in the data set is then scanned for emoticons. If an emoticon is found it is replaced with “happy” for sentiment values \( \geq 0.2 \), “sad” for values \( \leq -0.2 \) and neutral for the \([-0.2, 0.2]\) range.

3.4 Lexicon-based classifiers

In order to accurately test the lexicon-based classifiers NLTK provides, a lexicon containing words and their respective sentiment orientation is required. The lexicon used in the following experiments is the Valence Aware Dictionary for sEntiment Reasoning (VADER). The lexicon contains 7517 words and emoticons and their respective sentiment polarity. The VADER-lexicon was created by collecting intensity ratings on 9 000 words from 10 independent human raters, for a total of over 90 000 ratings. The human raters’ reading comprehension was required to be above 80% at a standardized college-level test, giving credibility to their intensity ratings.

The library for NLTK VADER classifier utilizes an extensive rule set in its analysis, which introduces functionality for handling context and other syntactical
3.5. LEARNING-BASED CLASSIFIERS

considerations in text. This allows the handling of features, such as negation and intensifiers, which improves the performance of VADER compared to other lexicon-based classifiers (Hutto, Gilbert, 2014). The lexicon was created with the incoherent nature of social media content in mind. Therefore, VADER is considered suitable for a lexicon-based classifier for social media classification.

The test data is stripped of any labels or annotation and fed to the classifier. The classifier returns a score in the range of [-1, 1] for every Tweet. Finally, the label is validated with the label in the the original data set. A neutrality threshold needs to be set up for the comparison to signify anything.

3.4.1 Neutrality Thresholds

In order to maximize the performance of the lexicon-based classifiers the neutrality thresholds will have to be carefully chosen. Determining the maximum performance of the lexicon-based classifiers will therefore be done by testing different neutrality thresholds. The maximum F-score achieved will be the one used to compare the performance of the lexicon-based classifier to the other classifiers.

3.5 Learning-based classifiers

In order to train our classifiers, the Tweets were split up into two parts: a training set and the testing set. The training set consisted of 80% of the collected Tweets, and the remaining a 20% was used to test the accuracy of the classifiers. The following methods of learning-based classification will be tested:

3.5.1 Naive Bayes

The maximum performance of the Bernoulli NB classifier is tested by varying the minimum feature occurrence count, e.g. if the minimum feature occurrence is 2, the features that only occur once are not counted. This is done in order to avoid giving incorrect weights to features.

3.5.2 Support Vector Machine

The SVM used is a linear SVM with word frequencies as features together with a tf-idf transformer. The maximum performance is found by varying the regularization parameter $\alpha$, controlling the structural risk minimization. This is done to find the optimal trade-off between achieving a small error on the training data and be able to correctly generalize the classifier to unseen data.

3.5.3 Maximum Entropy

The MaxEnt classifier uses unigrams as features and is trained with unigrams that occur at least 4 times in the data set as default. The classifier is trained with the
unigrams occurring most frequently in the data set. The number of unigrams chosen is varied in the range [10, 20, 50, 100, 200, 300], in search of which parameters yield the maximum performance.
Chapter 4

Results

This section follows up on the method outlined in the previous sections and presents the results that were achieved. A range of results for different parameters are shown for each classifier, followed by their preprocessed variant. The lexicon-based VADER is presented first, which is succeeded by the machine learning approaches of NB, SVM and MaxEnt respectively. Finally, the results of the optimal parameters are aggregated and compared in one table.

4.1 VADER

Varying the positive neutrality threshold from 0.1 to 0.9 and the negative threshold from $-0.1$ to $-0.9$ yielded a maximum F-measure of 72.3% given a negative threshold of $-0.3$ and a positive threshold of 0.4

![Figure 4.1. Performance of the lexicon-based classifier for different neutrality thresholds](image-url)
4.2 VADER preprocessed

Varying the neutrality thresholds from 0.1 to 0.9 and −0.1 to −0.9 yielded a maximum F-measure of 74.9% given a positive threshold of 0.5 and negative threshold of −0.3.

![Figure 4.2](image.png)

**Figure 4.2.** Performance of the lexicon-based classifier on preprocessed data for different neutrality thresholds
4.3 Naive Bayes

The maximum F-measure was achieved when the minimum feature occurrence was 4, and this yielded an F-measure of 58.2%.

Figure 4.3. Performance of the Bernoulli NB classifier on unprocessed data for different values of minimum feature occurrences

4.4 Naive Bayes preprocessed

The maximum F-measure was achieved when the minimum feature occurrence was 4, and this yielded a maximum F-measure of 58.3%.

Figure 4.4. Performance of the Bernoulli NB classifier on preprocessed data for different values of minimum feature occurrences
CHAPTER 4. RESULTS

4.5 Support Vector Machine

A maximum F-measure of 62.7% was achieved with an $\alpha$ of 0.0003.

![Figure 4.5. Performance of the SVM classifier on unprocessed data for different $\alpha$ values](image)

4.6 Support Vector Machine preprocessed

A maximum F-measure of 65.5% was encountered with an $\alpha$ of 0.0002.

![Figure 4.6. Performance of the SVM classifier on preprocessed data for different $\alpha$ values](image)
4.7 Maximum Entropy

The maximum performance was measured when the 50 most frequently used features were used for classification which yielded an F-measure of 47.7%.

![Figure 4.7. Performance of the MaxEnt classifier on unprocessed data for different numbers of the most frequent features used](image)

4.8 Maximum Entropy preprocessed

The maximum performance was measured when the 100 most frequently used features were used for classification which yielded an F-measure of 50.2%.

![Figure 4.8. Performance of the MaxEnt classifier on preprocessed data for different numbers of the most frequent features used](image)
4.9 Comparison of the Results

Following is a collection of the maximum measured performance for each classifier and data set.

![Bar chart showing F-measure comparison for VADER, VADER PP, NB, NB PP, SVM, SVM PP, MaxEnt, and MaxEnt PP.]

**Table 4.1.** Performance of every classifier with optimal parameters with and without preprocessed data, where PP stands for preprocessed

<table>
<thead>
<tr>
<th>Classification type</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>VADER</td>
<td>72.3%</td>
</tr>
<tr>
<td>VADER PP</td>
<td>74.9%</td>
</tr>
<tr>
<td>NB</td>
<td>58.2%</td>
</tr>
<tr>
<td>NB PP</td>
<td>58.3%</td>
</tr>
<tr>
<td>SVM</td>
<td>59.5%</td>
</tr>
<tr>
<td>SVM PP</td>
<td>62.5%</td>
</tr>
<tr>
<td>MaxEnt</td>
<td>47.7%</td>
</tr>
<tr>
<td>MaxEnt PP</td>
<td>50.3%</td>
</tr>
</tbody>
</table>
Chapter 5

Discussion

As shown in the aggregate results, the method yielding the best F-measure is the lexicon-based VADER classifier. The scores are the highest when the Tweets have been preprocessed, i.e. all emoticons in the Tweets have been replaced with their respective label.

The results also show that the more advanced machine-learning based approaches performed worse than their lexicon-based counterparts. This section attempts to cover the reasons for this trend, criticizes and identifies faults in the proposed method, and suggests improvements that could be included in future research.

5.1 Size of Data Set and Domain-Specific Knowledge

Performance of the learning-based classifiers NB, SVM and MaxEnt tend to hover around the 60% range for social media texts (Hutto, Gilbert, 2014). Learning-based methods of classification tend to perform poorly when used outside the domain they are trained for (Aue, Gamon, 2005). This could explain the performance of the learning-based classifiers in the experiments considering that the training and test set included text from many different domains.

Specifically, the performance of the classifiers is likely worse in less explored domains, areas of discussion that were not included in the training set. This could perhaps be improved by utilizing a larger data set, though this is a labor-intensive process for which resource limitations might not allow. An alternative would be to utilize distant supervision to train the classifier with larger amounts of data from different domains. However, the data set that was used to achieve the results was manually annotated by trained individuals and therefore highly reliable. The data set collected by distant supervision would be less accurate in terms of labeling, but the data set would have the potential to be much bigger since the resource-intensive process of manual labelling is omitted.

It is uncertain if the results for the lexicon-based classifier would scale with a larger data set and wider domain, but by using distant supervision and automatically labeling data for training, a larger data set could be collected, spanning a
wide range of domains. The Tweets used in the experiment spanned many different areas but was lacking in size. Therefore, it is possible that a less reliable but larger data set could improve the performance of the learning-based classifiers.

### 5.2 Imbalanced Data Set

The data set used in the tests was not balanced, meaning it did not contain equal amounts of negative, positive and neutral Tweets. This does not affect the performance of VADER, as it is lexicon-based, whereas the machine-learning based classifiers’ performance suffer because they have not been evenly trained to predict the different classes. This is evident through the difference in performance between the positive classifications and the other two cases for the NB classifier. While the size of the data set is probably a greater factor in determining the overall performance of the classifier, a balanced set should lead to a more even performance between the different classifications and, therefore, be more indicative of the overall performance of the classifier.

From the achieved results alone, it is impossible to say with certainty if the machine-learning classifiers should perform closer to the positive classifications, or the negative and neutral classifications. However, when considering results from previous research done on machine-learning classifiers (Hutto, Gilbert, 2014) and the performance of the positive classifications with a relatively small increase in the training set, balancing and increasing the size of the data set should yield an overall performance that is closer to or better than that of the positive classifications.

### 5.3 Cross-validation

One method through which the imbalanced data set could have been amended to some degree would have been to use cross-validation, i.e 5-fold cross-validation, to gain a better distribution of the negative, positive and neutral training data. This would not solve the problems caused by the size of the data set and domain-specific knowledge, but could improve the distribution of the sentiments in the training data, and consequently allow for a more balanced performance between the positive, negative and neutral classifications.

### 5.4 Preprocessing

The increase in performance when the data set was stripped of emoticons was measured up to 5%. This might not be in line with intuition, considering that emoticons and other informal features are important for conveying sentiment in social media (Davidov et al., 2010). However, an explanation for this could be that the words “happy”, “neutral” and “sad” occurred more frequently in the data set after preprocessing, and as a result, allowed the classifiers to more accurately weigh these unigrams.
5.4. PREPROCESSING

The data set contained many of the informal features that are frequent in social media text. The features were not removed due to being considered important for conveying statement (Davidov et al., 2010). The task of generalizing the language through preprocessing, in order to improve classifier performance, has been used before (Balahur, 2013). The preprocessing removed repeated punctuation, replaced and normalized capitalization. In other words, the generalization was done to such an extent that the preprocessed data had removed most of the informal sentiment-laden features and generally remodeled it to a formal text. However, as is evident of the results in this report, removing detail and abstracting the problem through preprocessing can be a suitable method for improving classification performance in social media.

For the learning-based classifiers, in particular MaxEnt, it is important to note that features that occur rarely in the data set, i.e. less than the minimum number of occurrences required to be considered for training, would not have an impact on the classifier’s performance. By replacing the occurrences of emoticons with the words “happy”, “sad”, or “neutral”, and thus increase the frequency of these words, the replaced emoticons would be correctly accounted for by the classifier. In addition, features that were not previously considered could have been included in training and evaluated accurately by the classifiers.

Preprocessing had the least effect on the NB classifier. An explanation for this could be that the NB classifier was a Bernoulli NB classifier. In other words, the classifier only weighed the features based on their presence rather than their frequency. Therefore, when the emoticons were replaced and certain words became more frequent, it did not have a significant effect on how the NB classifier evaluated the data. An alternative would have been to use a Multinomial NB classifier which evaluates features based on their frequency, in which case, preprocessing could have had a more noticeable impact on the classifier’s performance.

VADER’s performance differed by 2.6 percentage points when comparing the results between the unprocessed and preprocessed data sets. The lexicon-based classifier evaluates the sentiment according to the weights contained in its sentiment lexicon. While it may seem less accurate than evaluating each emoticon with its specific weight, the performance difference could be explained by the fact that the emoticons were replaced with words corresponding to a value closer to the average of that class. In other words, emoticons with values close to the thresholds for each class increased in value and thus conveyed sentiment stronger. Simultaneously, the more sentiment-laden emoticons decreased in value, but the performance gain could be attributed to the overall increase in words conveying strong sentiment.

Preprocessing might affect differently sized data sets in varying ways. If the data set used for training is larger and contains enough occurrences of informal features for the classifiers to be trained on, generalizing the language could penalize the performance. However, if the data set is smaller and does not contain enough of these informal features, it is likely that preprocessing the language improves the performance of the classifier. The accuracy of this statement and the effects of more sophisticated preprocessing methods could be investigated further.
CHAPTER 5. DISCUSSION

5.5 Twitter Data Set

The chosen data set contained features that are prevalent on Twitter. However, this data might not represent features that are standard on other social media platforms. Therefore, the results might not be indicative of classification performance when applied to data from other social media platforms such as Facebook, Instagram or WhatsApp.

5.6 Finding the Optimal Parameters

The quality of the optimal parameter search methodology is worth examining. The chosen parameters for changing was based upon their observed impact on performance, in other words, almost arbitrarily, but was done due to time and resource limitations. The parameters that were observed to have the best immediate influence on the performance were varied, while other parameters were left with default values. Thus, investigating a wider range of parameter combinations could improve the performance of the classifiers further.
Chapter 6

Future Research

6.1 Larger Data Set

The results presented in this thesis give reason to believe that lexicon-based methods of sentiment classification perform better than their machine-learning based counterparts when classifying short texts containing informal features. These results may differ greatly if the training would have been done with a larger data set. Comparing performances of NB, SVM, MaxEnt and VADER with a data set that has been collected and annotated using distant supervision would likely yield different results than those which have been presented in this report.

6.2 Neural Networks

Neural Networks have been shown to perform well when classifying text with correct punctuation and correctly written English (Socher et al., 2013), and pushed state of the art sentiment analysis of single sentences forward by 5.4%. The classifier’s performance on single sentences gives reason to believe that it could perform well on classification on social media text if used in conjunction with preprocessing.

6.3 Preprocessing

Preprocessing was shown to have a positive impact on the performance of the classifiers. While the effects of extensive preprocessing has been explored before (Balahur, 2013), the extent of the preprocessing in this report was relatively minimal. A more balanced approach to avoid removing all informal features could be explored, in other words normalize these features to retain the difference in sentiment conveyed by e.g. “!” and “!!!!!!”. The chosen approach only evaluated text based on three classes, therefore the importance of retaining these features could be more evident if additional classes were introduced.
Chapter 7

Conclusion

The results give reason to believe that a lexicon-based approach is the best choice for sentiment classification in social media. The simplicity of the lexicon-based classifier coupled with not requiring resource-costly training data makes it a strong contender for social media sentiment classification. A generalized vocabulary improves the performance of the classifiers, which proposes that further language abstraction enhances classification performance in social media. Preprocessing the data set is therefore a successful method for improving sentiment classification results.
Bibliography


Balahur, A., 2013, June. Sentiment analysis in social media texts. In 4th workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (pp. 120-128). http://www.aclweb.org/anthology/W13-1617 (visited on 1/4/2016)


