Naïve Bayes in Sentiment Classification

Adrian Barfuss
Seminar AI: past, present, future
Applied Computer Science, University of Bamberg

Abstract. In this report, I will present and discuss the work by Pak and Paroubek (2010). To do so, I will present the field of Sentiment Analysis, the specifics of Naïve Bayes Classifiers and the application of this approach by Pak and Paroubek (2010). I will then discuss the results and problems arising from this application. In particular, I will identify the exclusion of n-grams by Salience as an promising preprocessing technique suggested by the authors. Finally, I will point out further research that could help to further advance and generalize these findings.

Keywords: Information Systems, Information Retrieval, Sentiment Analysis, Natural Language Processing, Bayesian Analysis

1 Introduction

The analysis of textual is a very interesting field of research for many organizations. The reason for this is the huge amount of textual data available from web pages, the wide spread usage of micro blogging services (such as Twitter), social media platforms and messengers. Sentiment Analysis, the classification of texts according to the emotions and/or opinions expressed in them, is a particularly interesting field of research. And in this report I will present a Naïve Bayes approach for the Sentiment Analysis of textual data. In detail, I will present the results obtained by Pak and Paroubek (2010) who used a Naïve Bayes classifier to classify tweets according to the emotions contained in them. Furthermore, I will discuss the different preprocessing techniques the authors applied with regard to their impact on the performance of the classifier.

As a start, I will give a short introduction into the field of sentiment analysis and the theory behind Naïve Bayes classification in section 2. In next section 3 I will give a detailed outline of the methodology used by Pak and Paroubek (2010) which will be followed by an evaluation of its results in 4. In the last section 5 I will draw a short conclusion and point out further research topics.

2 State of Research

In this section I will give a cursory overview over the domain of Sentiment Analysis and the most popular approaches used in this field of study 2.1. In the following, I will present the Naïve Bayes approach as a particularly popular approach which is deployed by Pak and Paroubek (2010) whose paper I am going to present.
2.1 Sentiment Analysis

Application Domains and Motivation  Sentiment analysis can be interesting to companies who want to aggregate user reviews of their product, to monitor the image of the company itself as it is discussed in social media or micro blogging services such as Twitter or to monitor satisfaction of their employees. Very similar applications make the topic interesting to political organizations, social scientists, parties and governments, as they provide the means to analyse how these organizations, or particular political topics and policies are discussed in the media and social media (Procter, Vis, & Voss, 2013). These applications are meant as examples to motivate the this paper. For a more holistic overview, I invite you to consult Liu (2015, p. 4).

Definitions and Problem

"Sentiment analysis, also called opinion mining, is the field of study that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities[.]" (Liu, 2012, p. 7)

In a general attempt, Liu (2015, pp. 18) defines an opinion $o$ as a quintuple:

$$o \equiv (g, s, h, t)$$  \hspace{1cm} (1)

Where $g$ represents the target entity with regard to which an opinion holder $h$ expresses a sentiment $s$ at a given time $t$. The sentiment $s$ itself can be further differentiated into three components (Liu, 2015, p. 20):

$$s \equiv (y, o, i)$$  \hspace{1cm} (2)

With $y$ as the type of the sentiment, $o$ representing the orientation of $s$ (e.g. positive or negative) and $i$ expressing the intensity of the sentiment.

The different components of an opinion in this definition can be seen as subproblems of the greater problem of sentiment analysis. More recent approaches in the field of aspect oriented sentiment analysis try to tackle all these problems in parallel to obtain more meaningful and realistic results. But at least until recently, document-level sentiment analysis was one of the most studied approaches (Liu, 2015, p. 47).

Document-level Sentiment Analysis assumes that each document $x$ only contains one sentiment $s$ expressed by one opinion holder $h$ with regard to one entity $e$. By this simplification, document-level sentiment analysis reduces the problem to the classification of one sentiment $s$ for the whole document $x$ (Liu, 2015, pp. 47f.). It is obvious that this assumption is misleading for larger texts such as newspaper articles that discuss different aspects of an entity or even whole sets of entities and might include quotes or paraphrased opinions of organizations or individuals other than the author him- or herself. But it might be less problematic for other types of texts such as customer reviews or tweets which are both, more limited in scope and strongly focussed on a particular entity or topic (Liu, 2015, p. 48).
Current Approaches to Sentiment Analysis Many different approaches from the domains of statistics and machine learning are applied to the problem of sentiment classification in the field of document-level sentiment analysis. An overview of these approaches is given in figure 1.

There are good arguments to model a sentiment $s(x)$ expressed as a continuous variable. But according to Medhat et al. (2014), most approaches model it as a classification task where each $x$ is assigned one of three classes:

$$s(x) : x \rightarrow S = \{ \text{positive}, \text{neutral}, \text{negative} \}$$ (3)

Among the most popular techniques used to solve this kind of classification problem are Support Vector Machines and Naïve Bayes Classifiers (Liu 2015, p. 49). The latter will be presented in section 2.2 as it is used in the study of Pak and Paroubek (2010) which will be discussed in the following.

2.2 Naïve Bayes

Foundations: Bayes’ Theorem The foundation of Naïve Bayes classifiers is Bayes’ Theorem. In general, Bayes’ Theorem allows for the calculation of the posterior probability $P(h|D)$ that a hypothesis $h$ holds given we already observed training data $D$. This is done on the basis of the prior probabilities $P(D)$—the probability that we observe the training data $D$—and $P(h)$—the probability
that hypothesis $h$ holds—in combination with the probability $P(D|h)$—the probability that we observe $D$ in a world where we know that hypothesis $h$ holds:

**Definition 1. Bayes’ Theorem**

\[
P(h|D) = \frac{P(D|h)P(h)}{P(D)}
\]

(4)

Since $P(D)$ is independent of $P(h)$, it can be dropped to simplify the equation (5). In many cases it is further reasonable to assume, that all hypotheses $h$ are equally probable. In such a situation, $P(h)$ can be dropped as well to further simplify the equation term (6).

\[
P(h|D) = P(D|h)P(h) = P(D|h), \text{ iff } \forall i, j : P(h_i) = P(h_j)
\]

(6)

**Naïve Bayes Classifiers** The Naïve Bayes classifier as described by Mitchell (1997, pp 177-180) is a probabilistic approach that—as the second part of its name already suggests—relies heavily on Bayes’ Theorem. At the core of Naïve Bayes classifiers is the idea of determining the most probable a posteriori-class $c_{MAP} \in C$ for a given instance $x$ where $C$ represents the set of all possible classes. And where the instance $x$ itself consists of a conjunction of attribute values $x = < a_1, a_2, \ldots , a_n >$. More formally, this task can be rewritten as:

\[
c_{MAP} = \arg\max_{c \in C} [P(x|c)P(c)]
\]

(7)

\[
= \arg\max_{c \in C} [P(a_1, a_2, \ldots , a_n|c)P(c)]
\]

(8)

\[
= \arg\max_{c \in C} [P(a_1, a_2, \ldots , a_n|c)], \text{ iff } \forall i, j : P(c_i) = P(c_j)
\]

(9)

Once the problem has been reformulated in this way, the naïve assumption that is also indicated by the name takes effect: We assume that conditioned on a known value for the target class $c$, all attribute values $a_i \in x$ are stochastically independent from each other. This strong assumption in combination with the chain rule of probability makes it possible to rewrite $P(x|c)$ in a much much more convenient way:

\[
P(x|c) = P(c)P(a_1, a_2, \ldots , a_n|c)
\]

(10)

\[
= P(c) \prod_i P(a_i|c)
\]

(11)

\[
\implies c_{MAP} = \arg\max_{c \in C} [P(c)P(x|c)] = \arg\max_{c \in C} \left[ P(c) \prod_i P(a_i|c) \right]
\]

(12)

\[
= \arg\max_{c \in C} \left[ P(c) \prod_i \frac{P(a_i \cap P(c))}{P(c)} \right]
\]

(13)
with

\[ P(a = a_x) \approx \frac{|D_{a=a_x}|}{|D|} \quad (15) \]

\[ P(c) \approx \frac{|D_c|}{|D|} \quad (16) \]

After this simplification, it becomes rather easy to estimate the \textit{a priori} probabilities needed to determine \( c_{MAP} \). This can be done by calculating the share \( P(c) \) of all training examples \( D_c \subseteq D \) for which \( c \) holds. And the share \( P(a = a_x) \) of all training examples \( D_{a=a_x} \subseteq D \) for which the attribute \( a \) takes the value \( a_x \) as observed for \( x \) \cite{Bishop} p. 380).

This approach can be further elaborated as explained in \cite{Mitchell} p. 174). But for the purpose of this report, we only need knowledge of the classical Naïve Bayes approach.

## 3 Twitter as a Corpus for Sentiment Analysis

In this section, I will present the study by Pak and Paroubek \cite{Pak2010}. I will first outline the general problem of the study in section \[3.1\] before I present the training data and preprocessing steps applied to it by the authors \[3.2\]. After that, I will give an overview of the final model built on this basis \[3.3\] and the results obtained with it \[3.4\].

### 3.1 Problem Outline

The goal of the authors was to train and evaluate a Naïve Bayes classifier for \textit{document-level sentiment classification} of tweets. In equation (2) in section 2 I defined a sentiment as a triple \((y, o, i)\) consisting of a sentiment type \( y \), an orientation \( o \) and an intensity \( i \). As such, the sentiment was further defined as one dimension of an opinion \( [1] \). With regard to these definitions, Pak and Paroubek \cite{Pak2010} are trying to determine the sentiment \( s \) expressed in a tweet \( x \) out of the space of possible tweets \( X \). They further simplified the task to the classification of the orientation \( o \) of the sentiment while assuming a uniform \( i \) for every sentiment and without determining between the sentiment’s type \( y \). The three classes of orientations that make up the set \( S \) of the classes comprised by concept were \textit{positive} \((s_+)\), \textit{neutral} \((s_0)\) and \textit{negative} \((s_-)\) sentiments. The goal was thus to find a hypothesis \( h \) such that

\[ \forall x \in X : h(x) = c(x) \text{ with } c : X \rightarrow S = \{s_+, s_0, s_-\} \quad (17) \]

The approach the authors chose was a Naïve Bayes classifier. As explained in 2.2 this is a probabilistic approach that looks for the hypothesis \( h_{MAP} \) with the \textit{maximum a posteriori} (MAP) probability for an instance \( x \) given the training data \( D \). In this particular case, this is equivalent to finding the most probable
sentiment orientation $s \in S$ for a given tweet $x$. So we can reformulate the goal defined in (17) as follows:

$$\forall x \in X : \argmax_{s \in S} [P(s)P(x|s)] = c(x)$$  \hfill (18)

### 3.2 Training Data and Preprocessing

In order to acquire the training data, the authors randomly selected 100 000 tweets for each of the three classes by different criteria.

**Positive tweets** were identified by searching for the presence of positively connoted emoticons (:%D, =), ;),....)

**Negative tweets** where selected using the same strategy, but this time only tweets containing negatively connoted emoticons (:(, =(,...) were selected

**Neutral tweets** were selected by sampling tweets from accounts of large newspapers whose tweets were assumed to express no positive or negative emotions

Furthermore, the authors assumed a uniform probability distribution over $S$ which allows to drop $P(s)$ from equation (18):

$$\forall x \in X : \argmax_{s \in S} [P(x|s)] = c(x)$$  \hfill (19)

This assumption is not directly justified by the balanced training set since the training set was not acquired by random sampling. This assumption is not intrinsically problematic, but I will have to come back to it when I discuss the characteristics of the training set.

**Preprocessing and representation of the tweets** As stated in 2.2, a Naïve Bayes classifier requires the representation of an instance in terms of a collection of attribute-value pairs. For this purpose, the authors chose a representation of each tweet $x \in X$ as a set of two sets $G_x$ and $K_x$.

$$x \xrightarrow{\text{preprocessing}} \{G_x, K_x\}$$  \hfill (20)

The set $G_x$ is a so-called bag of words-representation of the tweet as an unordered set of words or sequences of words. The single steps of preprocessing needed to map every $x$ to a set $G_x$ were

1. In a first step, the authors filtered the tweets, sorting out user names, emoticons, hyperlinks and twitter special words such as “RT” which indicates a retweet.

2. The tweets were tokenized, this included the splitting of the tweets along white spaces and punctuation marks while trying to ensure that semantically interdependent tokens like “I’ll” remained intact.
3. In a third step, so called stop words were removed. Stop words are words that are known to be evenly distributed throughout texts (such as articles and pronouns) and whose removal decreases the necessary computational resources and should improve accuracy of the final model.

4. The fourth step encoded each tweet $x$ as an unordered set of $n$-grams $G_x$. An $n$-gram is a sequence of $n$ words encoded as a single entity. It can be thought of as shifting a window of size $n$ sequentially through the text, saving all unique combinations encountered as a single n-gram.

An illustrative overview of the process is given in figure 2.

![Fig. 2: Preprocessing for a bag of words-representation](image)

The set $K_x$ represents a set of tags for $x$ which was acquired by applying the TreeTagger, a decision tree based text tagging tool. TreeTagger tags grammatical structures and types of vocabulary. The tags are shown to correlate with different types of sentiments and thus thought to increase the accuracy of the classifier [Pak & Paroubek 2010, p. 1323].

The authors also experimented with different parameters and preprocessing steps and measured their impact on the performance of the classifier. These
variations included excluding ngrams with high entropy or low salience—an alternative measure for the distribution of ngrams among target classes. \(\text{Salience} \) is calculated as

\[
\text{salience}(g) = \frac{1}{N} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} 1 - \frac{\min[P(g|s_i), P(g|s_j)]}{\max[P(g|s_i), P(g|s_j)]}
\] (21)

And it averages the spread of probabilities \(P(g|s)\) for an ngram \(g\) given the different target classes and normalizes the values. Values close to 1 thus indicate that \(g\) is strongly associated with a particular sentiment while occurring much less often with other values for \(s\). In addition, the authors varied the number of tokens to include in an ngram and the attachment of negation words (such as “not”) to an ngram if they occurred in its context. I will shortly discuss the effects of these modifications in section 4.

3.3 The final classifier

Based on this representation of a tweet \(x\) as a set of tags \(T_x\) and a set of ngrams \(G_x\) as well as on the conditional independence-assumption of the Naïve Bayes approach, the final classifier determined the sentiment \(s\) expressed in \(x\) by maximizing the probability that its individual tags and ngrams occurred in combination with a sentiment in the training data. For this calculation, the \(a-priori\) probabilities for the tags and ngrams where given equal weight:

\[
s_{MAP_x} = \arg\max_{s \in S} \left[ \prod_{t \in T_x} P(t|s) \cdot \prod_{g \in G_x} P(g|s) \right]
\] (22)

3.4 Results

The performance of the classifier was evaluated on the basis of a set of 216 hand-labelleled and hand-collected tweets. The exact structure of the test set is given in table 1. It is worth noting that the distribution of the target classes in the test set does not match the distribution in the training data. The authors refer to Go, Bhayani, and Huang (2009) for the specifics of the collection process of the

<table>
<thead>
<tr>
<th>Sentiment</th>
<th>(N)</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive ((s_+))</td>
<td>108</td>
<td>50%</td>
</tr>
<tr>
<td>Neutral ((s_0))</td>
<td>75</td>
<td>34.72%</td>
</tr>
<tr>
<td>Negative ((s_-))</td>
<td>33</td>
<td>15.28%</td>
</tr>
<tr>
<td>Total</td>
<td>216</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table 1: Structure of the test set
test set. But Go et al. (2009) do not give a clear methodology for the collection process even though they indicate that the set is no representative sample. I will discuss the resulting inconsistency of the distribution of target classes in the training and the test set in section 4.

The performance of the classifier was measured using accuracy (% of correctly classified tweets) against decision (% of the ngrams of an instance classified). Furthermore, the authors evaluated the F0.5-score against the percentage of the training data used for training. The F0.5-score is calculated as

$$F_{0.5} = \left(1 + \beta^2\right) \frac{\text{accuracy} \cdot \text{decision}}{\beta^2 \cdot \text{accuracy} \cdot \text{decision}} \text{ with } \beta = 0.5$$

and is meant to represent a harmonic average over the accuracy and decision. The results are given in figure 3.

![Graphs showing the performance of the classifier using different parameters.](image)

(a) Varying size of ngrams  
(b) Exclusion by entropy / salience  
(c) Attachment of negation words  
(d) F0.5-score (bigrams, no filter)

Fig. 3: Performance of the classifier using different parameters. Source: XXX
Regardless of the parameters used, the classifier always reached accuracies greater than 0.6. The strongest improvement can be observed when ngrams are excluded by salience, even if compared to exclusion of ngrams by high entropy. Further findings are from the variation of preprocessing parameters are:

- bi-grams are outperforming unigrams and trigrams when used to train the classifier (fig. 3a)
- attaching negation words to ngrams slightly increases the performance of the classifier (fig. 3c)
- a larger training set improves performance in all settings—this is especially obvious for the F0.5-score, the harmonic average over accuracy and decision (fig. 3d)

4 Evaluation and further Work

As Liu (2015, p. 17) points out, accuracy of document-level sentiment classification for tweets usually achieves higher accuracies as if applied on other sorts of texts. This is due to the fact that tweets have a very limited size of 140 characters. Under these circumstances, the core assumption of document level sentiment analysis—namely, that a text expresses only one sentiment at once—is much more plausible as for larger texts. Against this background, the performance achieved by the classifier presented by Pak and Paroubek (2010) seems to stay within the expected range.

The evaluation of different preprocessing techniques is of greater interest, especially the usage of salience to exclude less expressive ngrams from the training set. Unfortunately, the authors do neither mention the exact threshold they used nor how different thresholds affected the results. That being said, salience still seems to be an interesting option for preprocessing of textual data that is not mentioned in introductory works such as Liu (2015).

4.1 Problems

A question not answered by the paper and which I could not resolve by further research concerns the consistency of the test and training data. A first inconsistency arises from the fact that the training data is acquired by selecting 300,000 tweets based on the emoticons contained in them while the test set consists only of 216 hand-labelled tweets. This raises the question to which extend the occurrence of emoticons in tweets captures the concept of emotions applied for the labelling of the test set. Discrepancies in this respect could represent a “glass ceiling” for the performance of the classifier on the test set that is not measured or mentioned in the paper.

The second inconsistency lies with the different distributions of sentiments in the test and training set. Although the different sentiments were equally represented in the training set, the test set was very imbalanced—it contained 50% positive emotions and only about 15% of negative ones. Unfortunately, the authors do neither provide a justification for the distribution of sentiments in the
training set nor for the distribution in the test set. This being said, it could either be that the test set is imbalanced or that the uniform \textit{a priori} probabilities for the different sentiments used for the model are not realistic. In any case, the inconsistent target class distributions and the small size of the test set strongly limit the extent to which the results obtained can be generalized.

5 Conclusions and further work

The suggestion of Pak and Paroubek (2010) to use salience for the filtering of relevant tokens seems to be a promising approach that is underpinned by their results. One very obvious advancement of the study would include running it on a larger test set and including explicit information on the \textit{a priori} probability of different sentiments into the model. Further research topics could be:

\textbf{Interaction Effects of Preprocessing parameters} It would be interesting to explicitly examine the interaction effects of different parameter settings. Possible research questions could be: How does the exclusion by salience affect the performance using different ngrams? Do different ngrams cause the classifier to generalize better / worse?

\textbf{Comparison of different Classifiers} Exclusion by salience seems to be a promising preprocessing technique for a Naïve Bayes approach to sentiment classification. It would be interesting to compare the impact of this technique between multiple classifiers. An plausible first choice for comparison would be Support Vector Machines, since they belong to the most prominent approaches in the field.

References


