Language-Independent Twitter Sentiment Analysis

Sascha Narr, Michael Hülfenhaus and Sahin Albayrak
DAI-Labor, Technical University Berlin, Germany
{vascha.narr, michael.huelfenhaus, sahin.albayrak}@dai-labor.de

Abstract

Millions of tweets posted daily contain opinions and sentiment of users in a variety of languages. Sentiment classification can benefit companies by providing data for analyzing customer feedback for products or conducting market research. Sentiment classifiers need to be able to handle tweets in multiple languages to cover a larger portion of the available tweets. Traditional classifiers are however often language specific and require much work to be applied to a different language. We analyze the characteristics and feasibility of a language-independent, semi-supervised sentiment classification approach for tweets. We use emoticons as noisy labels to generate training data from a completely raw set of tweets. We train a Naïve Bayes classifier on our data and evaluate it on over 10000 tweets in 4 languages that were human annotated using the Mechanical Turk platform. As part of our contribution, we make the sentiment evaluation dataset publicly available. We present an evaluation of the performance of classifiers for each of the 4 languages and of the effects of using multilingual classifiers on tweets of mixed languages. Our experiments show that the classification approach can be applied effectively for multiple languages without requiring extra effort per additional language.

1 Introduction

Twitter is one of the biggest microblogging services on the internet. Microblogs are short text messages that people use to share all kinds of information with the world. On Twitter, these microblogs are called “tweets”, and over 400 million of them are posted every day. They can contain news, announcements, personal affairs, jokes, opinions and more.

In this paper we analyze methods of extracting people’s opinions from tweets. A single piece of opinion from one person may not seem important, but among the billions of tweets the collection of opinions can form a comprehensive picture. People’s opinions and sentiments about products, other people and events in large numbers are invaluable. Companies can use them to gain feedback on their products, to do market research and analyze customer demands, and for a wide variety of other applications.

There has been a large amount of research to find ways to automatically extract sentiment from text. Most research is conducted only on tweets of one language. However, Twitter is an internationally popular service and a large part of the tweets are authored in a wide variety of languages. Therefore analyzing tweets in only one language covers only a part of the available content. Often, much work is put into hand-crafting features specific to that language. While using these features increases the quality of the sentiment classifications for that language, classifiers perform worse for other languages and readjusting the features requires much effort. Traditional multilingual sentiment classification approaches, like described by Mihalcea et al. [Mihalcea et al., 2007] require resources like parallel corpora for every language that should be classified. Additionally, language specific classifiers require tweets to be sorted out by their language. Since tweets do not always have any explicit language information, their language needs to be detected, which is not a trivial task given their frequent brevity.

A classification approach that is not restricted to analyzing only one language would be able collect much more sentiment information from the hugely multilingual twitterverse than a language-specific approach. Such an approach would ideally require as little effort as possible to be applied to additional languages. Multilingual sentiment information can help companies get feedback from people of a variety of languages simultaneously. It can illuminate how a product is received around the world.

In this paper, we examine a language-independent sentiment classification approach. We train a classifier to label the sentiment polarity specifically of tweets. We use a semi-supervised emoticon heuristic to generate labelled training data. For any language, our approach requires only raw tweets of that language for training and no additional adjustments or intervention.

We train classifiers on tweets of 4 different languages: English, German, French and Portuguese. For our evaluation, we collected thousands of human-annotated tweets in these 4 languages using Amazon’s Mechanical Turk². As part of our contribution, we make this evaluation dataset publicly available. We report on the differing outcomes between classifiers trained on different languages. We also report on our experiments on mixed-language sentiment classifiers.

Our approach assumes that any one tweet only contains one opinion at a time, so it can not separate between multiple sentiments in the same tweet. For our language-independent approach we assume that words are distinctly

¹Source: http://news.cnet.com/8301-1023_3-57448388-93/twitter-hits-400-million-tweets-per-day-mostly-mobile/

²Amazon Mechanical Turk: http://mturk.com
separated. Thus, our approach can only be applied to languages that use spaces as word separators.

We find that our approach performs well on the sentiment classification task, especially given its independence of specific languages. Classification accuracies tend to vary significantly between different languages. We also find that using a mixed-language classifier decreases the classification performance only slightly, but can greatly increase the ease of application in a practical setting.

Our findings help explore the use of sentiment classifiers in a multilingual setting. We give insight on how to create classifiers that function for different languages without additional overhead. Our evaluation dataset provides a basis for the evaluation of a wide range of multilingual sentiment classification approaches for microblogs.

2 Related Work

There has been a large amount of research to find ways to automatically extract sentiment from text. While traditional work [Pang et al., 2002] focused on text sources like movie reviews, more recent research has explored microblogs for sentiment analysis. The methods involved differ somewhat since texts like tweets have a different purpose and a more colloquial linguistic style [Han and Baldwin, 2011]. Notable works have been Go et al. [Go et al., 2009] who have trained a sentiment classifier to label tweets’ sentiment polarities as “positive” or “negative”. Pak et al. [Pak and Paroubek, 2010] trained classifiers to also detect “neutral” tweets that do not contain sentiment.

Sentiment classifiers require a large amount of training data, but obtaining it by having humans annotate tweets is very costly. To gather training data, Go, Pak and others used a heuristic method introduced by Read [Read, 2005] to assign sentiment labels to tweets based on emoticons instead. Test data is usually hand-labelled and consists only of a small amount of data.

To train a sentiment classifier, the source texts first have to be converted into some type of features. The most prominent features are n-grams, values from sentiment lexicons, part-of-speech tags and special microblogging features [Barbosa and Feng, 2010; Kouloumpis et al., 2011] that include among other emoticons, hashtags, punctuation and character repetitions and words in capital letters. Because the language used on Twitter is often informal and differs from traditional text types [Han and Baldwin, 2011], most approaches include a preprocessing step. Usually emoticons are detected, URLs removed, abbreviations expanded and twitter markup is removed or replaced by markers.

Go et al. [Go et al., 2009] compared different machine learning methods, Naïve Bayes, Maximum Entropy and Support Vector Machines (SVM), with unigram, bigram and part-of-speech features. Their training data consisted of 1.6M tweets equally split between positive and negative classes. The evaluation was performed on 359 hand-annotated tweets (182 positive and 177 negative tweets). Pak and Paroubek [Pak and Paroubek, 2010] collected 300k tweets as training data and performed evaluation on 216 hand-annotated tweets. The explored features included unigrams, bigrams and trigrams. The group conducted experiments with multilingual Naïve Bayes, SVM and Conditional Random Field classifiers; multinomial Naïve Bayes using bigrams performed best.

While most groups used a single classifier, Barbosa and Feng [Barbosa and Feng, 2010] followed a two-step approach with a neutral/subjective classifier and a positive/negative classifier. Instead of n-gram features they used part-of-speech tags, lexical and microblogging features to build SVM classifiers. Zhang et al. [Zhang et al., 2011] combined a lexicon-based classifier with an SVM to increase the recall of the classification.

Other research in sentiment classification focused beyond simple classification. Jiang et al. [Jiang et al., 2011] tried to perform a more fine-grained classification and assigned values from 1 to 5 to reflect the strength of expressed sentiments. Thelwall et al. [Thelwall et al., 2010] built a classifier that takes into account the targets of sentiment expressions and also considers related tweets.

Multilingual sentiment classification approaches often classify sentiment by using cross-lingual training [Wan, 2009; Mihalcea et al., 2007] with classification approaches for English texts. This however requires resources, like parallel corpora, that bridge between English and every language that should be classified. Many other approaches rely on machine-translation to first translate texts into English, and applying English classification techniques [Dencke, 2008] to the translated texts. Others use language-specific features to accommodate different languages [Boiy and Moens, 2009].

Multilingual sentiment classification approaches on tweets have before made use of emoticons as noisy labels for text of varying languages. However to the best of our knowledge, no work has methodically compared classifications among several languages. Davies and Ghaframani [Davies and Ghaframani, 2011] analyze bayesian learning on a set of mixed-language tweets, but report only test error on their training data. Cui et al. [Cui et al., 2011] use a graph propagation algorithm on a few seed tokens to build a multilingual sentiment lexicon, but do not report on outcomes between individual languages.

3 Methods

We use a two-class Naïve Bayes classifier with n-gram features for our sentiment classification. Training data is labeled by a semi-supervised approach that uses emoticons as noisy labels. Our training method is applicable as described for almost any language that uses spaces as word separators.

3.1 Tokenization and Features

We use a Naïve Bayes classifier on n-gram features to classify sentiment in tweets. To obtain n-gram features, we first have to tokenize the text input. Tweets pose a problem for standard tokenizers designed for formal and regular text. Tweet texts have unique characteristics like missing whitespaces, irregular spelling and special tokens such as hashtags (“#tag”), mentions of twitter users (“@username”) and retweet markers (“RT”). Therefore the data is tokenized using an extended version of O’Connor’s regular-expression-based tweet tokenizer [O’Connor et al., 2010]. It employs rules to define strings that form a token, e.g. emoticons, urls and strings of punctuation, and splits the rest of the strings at whitespaces.

We extended O’Connor’s tokenizer to match a great variety of emoticons used in different languages, including eastern style emoticons (e.g. 😃). In addition, emoticons can sometimes be attached directly to words, but they can also contain letters. Therefore it was important to account for cases where wrong tokenization would separate words (e.g. the emoticon 😊 should
be found in "He finally asked:D", but not in "He asked: Did you see the new movie?". A further extension handles emoticons with character repetitions, which usually expresses a stronger sentiment, e.g.: (((.

We used n-grams count of tokens as basic features for training the classifier. We reduced the feature space by reducing multiple letter repetitions to at most 2 consecutive letters and lower-casing all tokens after tokenization. We replaced uninformative tokens, e.g. URLs, usernames, numbers and single non-letter characters, by placeholder tokens.

3.2 Emoticon Heuristic

To generate training data, we used a method similar to the one used by Pak and Paroubek [Pak and Paroubek, 2010]. We assign noisy polarity class labels to tweets based on the existence of positive or negative emoticons. If a tweet contains one or more happy emoticons, and no negative ones, we assign it to the positive class, and vice versa. Because of the shortness of tweets, we assume here that the sentiment of a smiley applies to the whole text of the tweet. Although this may be wrong in some cases, it simplifies the analysis significantly. We used a range of emoticons that are very distinctly positive or negative as noisy labels. Below are shown some examples of the emoticons we used:

Examples of positive emoticons used:
:}iação }=} ;} } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ }

Examples of negative emoticons used:
:}iação }=} ;} } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ } ;{ };
Table 1: Our sentiment evaluation dataset: tweets annotated using Amazon Mechanical Turk. Each tweet was labeled by 3 human annotators. Shown are the total number of tweets for each language, the Fleiss’ kappa inter-annotator agreement, and the number of tweets for each sentiment class whose label 2 or more, as well as all 3 annotators agreed on.

<table>
<thead>
<tr>
<th>Language</th>
<th>Tweets</th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>en</td>
<td>7200</td>
<td>0.430</td>
<td>1595</td>
</tr>
<tr>
<td>de</td>
<td>1800</td>
<td>0.419</td>
<td>333</td>
</tr>
<tr>
<td>fr</td>
<td>1797</td>
<td>0.244</td>
<td>341</td>
</tr>
<tr>
<td>pt</td>
<td>1800</td>
<td>0.408</td>
<td>626</td>
</tr>
<tr>
<td>fr+en+de+pt</td>
<td>12597</td>
<td>0.407</td>
<td>2915</td>
</tr>
</tbody>
</table>

For our evaluation, we used the tweets from the Mechanical Turk set that had been classified as “positive” or “negative”. We ignored tweets annotated as “neutral”. We split the data into two evaluation sets. In the agree-3 set, we only used tweets that all 3 annotators had agreed on to belong to the same sentiment class. In the agree-2 set, we used tweets that at least 2 annotators had agreed on, thus including the agree-3 set. Each evaluation set contains tweets of the 4 languages: English (“en”), German (“de”), French (“fr”) and Portuguese (“pt”), and a set of the combined tweets of languages fr, en, de, pt. Table 3 gives details about the different evaluation set sizes.

Table 2: Our sentiment evaluation dataset: tweets annotated using Amazon Mechanical Turk. Each tweet was labeled by 3 human annotators. Shown are the total number of tweets for each language, the Fleiss’ kappa inter-annotator agreement, and the number of tweets for each sentiment class whose label 2 or more, as well as all 3 annotators agreed on.

<table>
<thead>
<tr>
<th>Language</th>
<th>Tweets</th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>en</td>
<td>148.6M</td>
<td>7.1% (10.6M)</td>
<td>1.7% (2.5M)</td>
</tr>
<tr>
<td>pt</td>
<td>50.6M</td>
<td>6.6% (3.3M)</td>
<td>3.2% (1.6M)</td>
</tr>
<tr>
<td>fr</td>
<td>2.9M</td>
<td>13.2% (385k)</td>
<td>1.4% (40k)</td>
</tr>
<tr>
<td>de</td>
<td>2.1M</td>
<td>15.8% (330k)</td>
<td>2.4% (49k)</td>
</tr>
</tbody>
</table>

Table 3: Datasets used to evaluate our experiments: The agree-3 set contains tweets all three annotators agreed on, agree-2 contains also tweets agreed upon by two.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Language</th>
<th>Positive</th>
<th>Negative</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>agree-2</td>
<td>en</td>
<td>1595</td>
<td>998</td>
<td>2593</td>
</tr>
<tr>
<td></td>
<td>de</td>
<td>353</td>
<td>239</td>
<td>592</td>
</tr>
<tr>
<td></td>
<td>fr</td>
<td>341</td>
<td>321</td>
<td>662</td>
</tr>
<tr>
<td></td>
<td>pt</td>
<td>626</td>
<td>414</td>
<td>1040</td>
</tr>
<tr>
<td></td>
<td>combined</td>
<td>2915</td>
<td>1972</td>
<td>4887</td>
</tr>
<tr>
<td>agree-3</td>
<td>en</td>
<td>739</td>
<td>488</td>
<td>1227</td>
</tr>
<tr>
<td></td>
<td>de</td>
<td>143</td>
<td>95</td>
<td>238</td>
</tr>
<tr>
<td></td>
<td>fr</td>
<td>159</td>
<td>160</td>
<td>319</td>
</tr>
<tr>
<td></td>
<td>pt</td>
<td>297</td>
<td>213</td>
<td>510</td>
</tr>
<tr>
<td></td>
<td>combined</td>
<td>1338</td>
<td>956</td>
<td>2294</td>
</tr>
</tbody>
</table>

Table 4: Overview of the training dataset: raw tweets and the number of tweets labeled as containing sentiment by our semi-supervised emoticon heuristic.

5.2 Evaluation Dataset

We evaluated the classifier on a subset of tweets taken from our evaluation dataset of manually annotated tweets (see Section 4).

For our evaluation, we used the tweets from the Mechanical Turk set that had been classified as “positive” or “negative”. We ignored tweets annotated as “neutral”. We split the data into two evaluation sets. In the agree-3 set, we only used tweets that all 3 annotators had agreed on to belong to the same sentiment class. In the agree-2 set, we used tweets that at least 2 annotators had agreed on, thus including the agree-3 set. Each evaluation set contains tweets of the 4 languages: English (“en”), German (“de”), French (“fr”) and Portuguese (“pt”), and a set of the combined tweets of languages fr, en, de, pt. Table 3 gives details about the different evaluation set sizes.

5.3 Experimental Setup

We trained one classifier per language using an equal amount of tweets for both polarity classes from our training dataset. We trained a Naïve Bayes classifier separately, each using 1-grams, 2-grams and combined 1-grams and 2-grams.

We optimized each classifier using the process described in Section 3, with a threshold for infrequent n-grams of 2 or less. We also trained a fifth classifier on the combined tweets of a mixture of the 4 languages. Here we used an equal number of tweets for each language, as many as available for the smallest language set.

We trained the classifiers in multiple steps to gain insight on the effects of the amount of training data on the classification performance. We had fewer German and French training tweets available than for the other languages. Therefore we could only evaluate those two languages on about 100k training tweets. During experimenting, we used 10-fold cross-validation testing on the training data for validation and parameter tuning.

We ran experiments on our 5 classifiers using the 2 different evaluation datasets detailed in Section 5.2. Each classifier was evaluated using the tweets of the appropriate language. The mixed-language classifier was evaluated on the combined language sets (“frendept”).
5.4 Results & Discussion

The results of our experiments for the different classifier variations, grouped by each of the 4 languages, are displayed in Figures 1a through 1d. Figure 1e displays the results for the mixed classifier of all 4 languages. An overview of the performance of only the unigram classifier on a single evaluation set, compared between all languages, is shown in Figure 1f.

As a reference point for the classifiers, we added as a baseline the accuracy of a mock-classifier that always guesses a “positive” sentiment. The baseline accuracies were calculated individually for each language, using the number of positive and negative tweets in the agree-3 evaluation set as shown in Table 3. Because we had more positive than negative test tweets in all but one evaluation set, we decided to use the positive guess as a baseline instead of discarding tweets to balance the test sets.

Unigrams

The unigram Naive Bayes classifier provides a simple and very lightweight classification method. The best results we obtained were 81.3% accuracy for the English agree-3 test set. Generally, the classifiers’ performance improves with an increased number of tweets used for training. Using too few training tweets leads to a lower accuracy because less of the highly diverse topics present in tweets can be captured by the classifier.

For the maximum number of available training tweets, the performance of all classifiers is significantly better than the best-guess baseline “baseline-all-pos”. Even though the results are not directly comparable because of the different training and evaluation datasets, they are close in range to the results for the mixed classifier of all 4 languages.

Unigrams + Bigrams

The 1+2-gram classifier adds bigram features to the classification. Bigram features tend to be very sparse, even more so for datasets with more, irregular word tokens as found in tweets. Therefore the 1+2-gram classifier requires both more runtime memory and more training time than the unigram classifier. In our evaluations we could not observe a stable increase in performance for the 1+2-gram classifier across languages. While it tended to outperform the unigram classifier for both test sets in English and Portuguese, it performed worse than the unigram classifier for German and French.

Performances Between Evaluation Sets

Figure 1a shows significant discrepancies in the quality of classifications between the agree-3 and the agree-2 evaluation dataset. Similar observations can be made for the other languages in Figures 1b, 1c and 1d. This is likely the case because the tweets in the agree-2 set were already evidently harder to classify accurately for the human annotators. Since there is less agreement between humans, the data poses a higher difficulty for a classification and thus leads to lower classifier performance.

Language-Specific Differences

Overall, the best accuracy performances measured on the agree-3 set were 81.3% for English, 64.9% for Portuguese, 79.8% for German and 74.9% for French. All of these best performances were achieved using the unigram classifier.

Nevertheless, these experiments show that it is indeed feasible to train a classifier on the combination of multiple languages. Though the performance degrades slightly, it can still be useful to employ such a combined classifier. If for example the tweets that should be classified come from a set of several possible languages, the degradation in performance could be acceptable, as a combined classifier saves the need to detect the tweets’ language beforehand. Knowing that a combined classifier performs only slightly worse is valuable, as English is a ubiquitous language in microtexts, and tweets of any language are often intermined with English ones.

6 Conclusion

In this paper we presented a language-independent classification approach to detect the sentiment polarity in tweets. Our approach uses a semi-supervised heuristic labeling scheme to acquire large amounts training data in a variety of languages, and content-based features that work well across languages. In our experiments, we trained a Naive Bayes classifier on source sets of millions of tweets in English, German, French and Portuguese. We compared the results between languages using tweets that were human-annotated using the Amazon Mechanical Turk service. We make this multilingual evaluation dataset publicly available.

We showed that the used algorithm achieves a good performance for tweets of multiple languages, especially given its efficient applicability to new languages. The classifier can be trained effortlessly on new languages, given only raw training data. We have not dealt with classifying sub

---

We provide a detailed list of the classification performances for each classifier, language and training set size on our website. See http://www.dai-lab.de/~narr/paper-sentiment
Figure 1: Classification accuracies for different Naïve Bayes sentiment polarity classifiers. We compare the classifier performance for each of 5 different languages using an increasing number of heuristically labeled training tweets. Test data are human annotated tweets; one set of at least 2 and one set of 3 mutual annotator agreements (test set size varies with language). The “baseline-all-pos” baseline indicates the best-guess performance of a mock classifier for each language that always guesses a positive sentiment (the majority class in the evaluation dataset).

(a) Accuracy of the English 1-gram and 1+2-gram classifiers, on the agree-2 and agree-3 sets.

(b) Accuracy of the Portuguese 1-gram and 1+2-gram classifiers, on the agree-2 and agree-3 sets.

(c) Accuracy of the German 1-gram and 1+2-gram classifiers, on the agree-2 and agree-3 sets.

(d) Accuracy of the French 1-gram and 1+2-gram classifiers, on the agree-2 and agree-3 sets.

(e) Accuracy of the 4-language mixed 1-gram and 1+2-gram classifiers, on the agree-2 and agree-3 sets combined for the 4 languages.

(f) Overview of classifier accuracies for each language, using a unigram classifier on the agree-3 set.
jectivity in tweets in this work. This is something we hope to be able to accomplish in future work.

Acknowledgments

We would like to thank SearchMetrics for providing us with the raw tweet dataset, and for helping us set up the Mechanical Turk annotation task.

References


