Twitter Sentiment Detection via Ensemble Classification Using Averaged Confidence Scores

Matthias Hagen, Martin Potthast, Michel Büchner, and Benno Stein

Bauhaus-Universität Weimar
<first name>.<last name>@uni-weimar.de

Abstract We reproduce three classification approaches with diverse feature sets for the task of classifying the sentiment expressed in a given tweet as either positive, neutral, or negative. The reproduced approaches are also combined in an ensemble, averaging the individual classifiers' confidence scores for the three classes and deciding sentiment polarity based on these averages. Our experimental evaluation on SemEval data shows our re-implementations to slightly outperform their respective originals. Moreover, in the SemEval Twitter sentiment detection tasks of 2013 and 2014, the ensemble of reproduced approaches would have been ranked in the top-5 among 50 participants. An error analysis shows that the ensemble classifier makes few severe misclassifications, such as identifying a positive sentiment in a negative tweet or vice versa. Instead, it tends to misclassify tweets as neutral that are not, which can be viewed as the safest option.

1 Introduction

We reproduce three state-of-the-art approaches to classifying the sentiment expressed in a given tweet as either positive, neutral, or negative, and combine the three approaches to an ensemble based on the individual classifiers' confidence scores.

With about 271 million active users per month and about 350,000 tweets per minute, Twitter is one of the biggest social networks that can be mined for opinions. It is used as a communication platform by individuals, but also companies and organizations. The short text messages, or tweets, shared on Twitter cover a range of topics like information and comments on ongoing events but also opinions on products, brands, etc. Making the latter piece of information accessible to automatic analysis is not straightforward. A central tool required for this is sentiment detection, which determines whether a given tweet is rather positive or rather negative. However, sentiment detection for tweets is a challenge in itself. Unlike Amazon reviews, for instance, that typically have a length of several sentences or even paragraphs, the tweets come with a length limit of just 140 characters, which forces people to use abbreviations, slang, and genre-typical expressions. The language used in tweets often differs significantly from what is observed in other large text collections.

To facilitate the development of effective approaches for the analysis of tweets, corresponding shared tasks have been organized at SemEval from 2013 onwards. The goal of these tasks is to grasp the opinions expressed in microblogging forums like Twitter and to thus gain a better understanding of what matters to the users, e.g., what they like and what they do not like. Platforms such as Twitter, with their wealth and diversity

of users, have often been found to be an accurate source for tracking opinions in societies that increasingly express themselves online. People share their reviews of books or movies, express pros and cons on political topics, or just give feedback to companies or restaurants. Such expressions are meaningful to the respective reviewed subjects, for instance, to design new products, but they are also meaningful to the general public to get a better idea whether a specific product is useful. Especially for information retrieval, incorporating the sentiment of a piece of text is an important signal for diversification of retrieval results.

In particular, we focus on subtask B in SemEval 2013's task 2 and SemEval 2014's task 9 "Sentiment Analysis in Twitter," where the goal is to classify the whole tweet as either positive, neutral, or negative. Note that this is a slightly different task than classifying the sentiment of a tweet expressed for or against a given target topic (e.g., a product or a brand). In the setting we address, the tweet as a whole could express several sentiments (positive for topic A but negative against topic B) and the goal is to identify the sentiment that dominates. Besides the aforementioned SemEval tasks, this variant of the Twitter sentiment detection problem has attracted quite some research interest.

Since notebook descriptions accompanying submissions to shared tasks are understandably very terse, it is often a challenge to reproduce the results reported. Therefore, we attempt to reproduce three state-of-the-art Twitter sentiment detection algorithms that have been submitted to the aforementioned tasks. Furthermore, we combine them in an ensemble classifier. Since the individual approaches employ diverse feature sets, the goal of the ensemble is to combine their individual strengths. Our experimental evaluation shows that our re-implementations of the three selected approaches outperform their originals in two cases, whereas one achieves the same results as its original. Furthermore, our ensemble approach outperforms its components. The ensemble would have been ranked in the top-5 ranks among all the 50 participants of SemEval 2013 and 2014. An error analysis of our approach reveals that there are hardly any misclassifications of a positive tweet as negative or vice versa. Instead, misclassifications of positive or negative tweets usually result in a neutral classification. This could be viewed as the safest option when the classifier is in doubt.

The remainder of the paper is organized as follows. In Section 2 we briefly review related work on sentiment detection with a focus on Twitter. The detailed description of the three individual approaches as well as our ensemble approach follows in Section 3. Our experimental evaluation within the SemEval task's setting is described in Section 4. Some concluding remarks and an outlook on future work close the paper in Section 5.

2 Related Work

Sentiment detection in general is a classic problem of text classification. Unlike other text classification tasks, the goal is not to identify topics, entities, or authors of a text but to rate the expressed sentiment typically as positive, negative, or neutral. Most approaches used for sentiment detection have also been useful for other text classification tasks and usually involve methods from machine learning, computational linguistics, and statistics. Typically, several approaches from these fields are combined for sentiment detection [32, 41, 14]. Linguistic considerations range from tokenizing the to-be-classified texts to other syntactic analyses. Statistical considerations typically involve

frequencies of tokens or phrases, e.g., the occurrence of many "positive" words in a text, or similar statistics. The respective features then usually are combined by machine learning algorithms to classify the sentiment of arbitrary texts.

Machine learning methods are applied to sentiment detection as a matter of course, both supervised or unsupervised. Without training data available, one of the earliest unsupervised sentiment detection methods is based on word polarity dictionaries and pointwise mutual information (PMI) of part-of-speech (POS) sequences [41]. PMI is a measure of the statistical dependency of two terms. First, the PMI scores of POS-tagged noun phrases like "long movie" with positive and negative words from the polarity dictionary like "excellent" or "poor" are computed. The two PMI scores are then subtracted and, based on the result the original noun phrase, tagged as either positive or negative. For a given text, the polarity scores of all phrases are added and the sum's algebraic sign "detects" the text's overall sentiment. Originally, the PMI scores were determined via search engine requests but also large text corpora such as the ClueWeb or similar can be applied. The accuracy of the PMI method is not too impressive and can usually be improved when labeled data is available for training.

Supervised methods are trained on labeled data (i.e., texts with known polarity). In case of reviews, the actual assessment like "5 stars" or "1 star" can easily be translated to a sentiment. However, in case of sentiment detection in tweets, acquiring training data typically is a laborious and costly manual process. Typical methods of supervised learning for sentiment detection involve features like unigrams, bigrams, trigrams, polarity dictionaries, etc. Standard learning approaches range from Naive Bayes or Maximum Entropy to Support Vector Machines that learn the actual classifier from labeled training data [32]. Our approach is based on reproducing three supervised approaches, which are trained on data obtained from the SemEval Twitter sentiment detection task.

Several state-of-the-art methods for sentiment detection in texts exist but the important question then is for what scenarios the sentiment detection is actually useful [22]—besides determining the polarity of a text. Since we deal with Twitter data, the question is about use cases of detecting the sentiment of tweets: corresponding papers apply state-of-the-art sentiment detection on Twitter data to identify the general public's mood on given events from media, politics, culture, or economics [6]. This way, sentiment detection enables sociological studies at scale and almost in real time. Another paper studies the evaluation of politicians' TV debate performance based on sentiments expressed on Twitter [12]. This gives direct feedback to political election campaigns on what specific topics the voters are interested in and how the candidate's perceived performance is on that topic. Similarly, the general sentiment, or rather opinion, on products or events can be extracted from the Twitter stream [5], and aid in economic or sociological studies. As for companies, besides detecting sentiment for products or for marketing campaigns, also identifying the employees' mood can be beneficial for employee development programs and the like [27]. Of course, in this case the work force should be rather big and Twitter-savvy to get meaningful results.

As for more retrieval-oriented tasks, the ranking of products and reviews benefits from sentiment detection [10]: by identifying categories important to the users from sentiments expressed on Twitter, products ca be re-ranked accordingly. Moreover cross-language retrieval and ranking can incorporate sentiments and their respective transla-

tions [19]. Finally, annotating search results with the expressed general sentiment can be helpful as a facet in result presentation [11].

Due to the different applications in mining and retrieval, and since Twitter is one of the richest sources of opinion, a lot of different approaches to sentiment detection in tweets have been proposed. Different approaches use different feature sets ranging from standard word polarity expressions or unigram features also applied in general sentiment detection [17, 23], to the usage of emoticons and uppercases [4], word lengthening [8], phonetic features [13], multi-lingual machine translation [3], or word embeddings [40]. The task usually is to detect the sentiment expressed in a tweet as a whole (also focus of this paper). But it can also be to identify the sentiment in a tweet with respect to a given target concept expressed in a query [21]. The difference is that a generally negative tweet might not say anything about the target concept and must thus be considered neutral with respect to the target concept.

Both tasks, namely sentiment detection in a tweet, and sentiment detection with respect to a specific target concept, are part of the SemEval sentiment analysis tasks since 2013 [28, 38]. SemEval thus fosters research on sentiment detection for short texts in particular, and gathers the best-performing approaches in a friendly competition. The problem we are dealing with is formulated as subtask B in the SemEval 2013 task 2 and in the SemEval 2014 task 9: given a tweet, decide whether its message is positive, negative, or neutral. A few examples from the annotated SemEval 2013 training set give the gist of the task:

Positive: `Gas by my house hit $3.39!!!! I'm going to Chapel Hill on Sat. :)`
Negative: `Dream High 2 sucks compared to the 1st one.`
Neutral: `Battle for the 17th banner: Royal Rumble basketball edition`

State-of-the-art approaches have been submitted to the SemEval tasks. However, the organizers never trained a meta classifier based on the submitted approaches to determine what can be achieved when combining them, whereas each participating team only trains their individual classifier using respective individual feature set. Our idea is to combine three of the best-performing approaches with different feature sets, and to form an ensemble classifier that leverages the individual classifiers' strengths.

Ensemble learning is a classic approach of combining several weak classifiers to a more powerful ensemble [30, 33, 36]. The classic approaches of Bagging [7] and Boosting [39, 15] try to either combine the outputs of different classifiers trained on different random instances of the training set or on training the classifiers on instances that were misclassified by the other classifiers. Both rather work on the final predictions of the classifiers just as for instance averaging or majority voting on the predictions [1] would do. In our case, we employ the confidence scores of the participating classifiers. Several papers describe different ways of working with the classifiers' confidence scores, such as learning a dynamic confidence weighting scheme [16], or deriving a set cover with averaging confidences [37]. Instead, we simply average the three confidence scores of the three classifiers for each individual class. This straightforward approach performs superior to its individual parts and performs competitive in the SemEval competitions. Thus, its sentiment detection results can be directly used in any of the above use cases for Twitter sentiment detection.

3 Approaches

We select three state-of-the-art approaches for sentiment detection among the 38 participants of subtask B of the SemEval 2013 sentiment detection task. To identify worthy candidates—and to satisfy the claim "state of the art"—we picked the top-ranked approach by team NRC-Canada [26]. However, instead of simply picking the approaches on ranks two and three to complete our set, we first analyzed the notebooks of the top-ranked teams in order to identify approaches that are significantly dissimilar from the top-ranked approach. We decided to handpick approaches this way so they complement each other in an ensemble. As a second candidate, we picked team GU-MLT-LT [18] since it uses some other features and a different sentiment lexicon. Incidentally, it was ranked second. As a third candidate, we picked team KLUE [35], which was ranked fifth. We discarded the third-ranked approach as it is using a large set of not publicly available rules, whereas the fourth-ranked system seemed too similar to NRC and GU-MLT-LT to add something new to the planned ensemble.

This way, reproducing three approaches does not deteriorate into reimplementing the feature set of one approach and reusing it for the other two. Moreover, combining the three approaches into an ensemble classifier actually makes sense, since, due to the feature set diversity, they tap sufficiently different information sources. In what follows, we first briefly recap the features used by the individual classifiers and then explain our ensemble strategy.

3.1 NRC-Canada

Team NRC-Canada [26] used a classifier with a wide range of features. A tweet is first preprocessed by replacing URLs and user names by some placeholder. The tweets are then tokenized and POS-tagged. An SVM with linear kernel is trained using the following feature set:

N-grams The occurrences of word 1-grams up to word 4-grams are used as features as well as occurrences of pairs of non-consecutive words where the intermediate words are replaced by a placeholder. No term-weighting like $tf{\cdot}idf$ is used. Similarly, character 3-grams up to character 5-grams are used as features.

ALLCAPS The number of words written all capitalized is used as a feature.

Parts of speech The occurrences of part-of-speech tags is a feature.

Polarity dictionaries In total, five polarity dictionaries are used. Three of these were manually created: the NRC Emotion Lexicon [24, 25] with 14,000 words, the MPQA Lexicon [42] with 8,000 words, and the Bing Liu Lexicon [20] with 6,800 words. Two other dictionaries were created automatically. For the first one, the idea is that several hash tags can express sentiment (e.g., #good). Team NRC crawled 775,000 tweets from April to December 2012 that contain at least one of 32 positive or 38 negative hash tags that were manually created (e.g., #good and #bad). For word 1-grams and word 2-grams in the tweets, PMI-scores were calculated for each of the 70 hash tags to yield a score for the n-grams (i.e., the ones with higher positive hash

tag PMI are positive, the others negative). The resulting dictionary contains 54,129 unigrams, 316,531 bigrams, and 308,808 pairs of non-consecutive words. The second automatically created dictionary is not based on PMI for hash tags but for emoticons. It was created similarly to the hash tag dictionary and contains 62,468 unigrams, 677,698 bigrams, and 480,010 pairs of non-consecutive words.

For each entry of the five dictionaries, the dictionary score is either positive, negative, or zero. For a tweet and each individual dictionary, several features are computed: the number of dictionary entries with a positive score and the number of entries with a negative score, the sum of the positive scores and the sum of the negative scores of the tweet's dictionary entries, the maximum positive score and minimum negative score of the tweet's dictionary entries, and the last positive score and negative score.

Punctuation marks The number of non-single punctuation marks (e.g., !! or ?!) is used as a feature and whether the last one is an exclamation or a question mark.

Emoticons The emoticons contained in a tweet, their polarity, and whether the last token of a tweet is an emoticon are employed features.

Word lengthening The number of words that are lengthened by repeating a letter more than twice (e.g., cooooolll) is a feature.

Clustering Via unsupervised Brown clustering [9] a set of 56,345,753 tweets by Owoputi [31] clustered into 1,000 clusters. The IDs of the clusters in which the terms of a tweet occur are also used as features.

Negation The number of negated segments is another feature. According to Pang et al. [32] a negated segment starts with a negation (e.g., shouldn't) and ends with a punctuation mark. Further, every token in a negated segment (words, emoticons) gets a suffix NEG attached (e.g., perfect_NEG).

3.2 GU-MLT-LT

Team GU-MLT-LT [18] was ranked second in the SemEval 2013 ranking and trains a stochastic gradient decent classifier on a much smaller feature set compared to NRC. For feature computation, they use the original raw tweet, a lowercased normalized version of the tweet, and a version of the lowercased tweet where consecutive identical letters are collapsed (e.g., helllo gets hello). All three versions are tokenized. The following feature set is used:

Normalized unigrams The occurrence of the normalized word unigrams is one feature set. Note that no term weighting like for instance $tf \cdot idf$ is used.

Stems Porter stemming [34] is used to identify the occurrence of the stems of the collapsed word unigrams as another feature set. Again, no term weighting is applied.

Clustering Similar to the NRC approach, the cluster IDs of the raw, normalized, and collapsed tokens is a feature set.

Polarity dictionary The SentiWordNet assessments [2] of the individual collapsed tokens and the sum of all tokens' scores in a tweet are further features.

Negation Normalized tokens and stems were added as negated features similarly to the NRC approach.

3.3 KLUE

Team KLUE [35] was ranked fifth in the SemEval 2013 ranking. Similarly to NRC, team KLUE first replaces URLs and user names by some placeholder and tokenizes the lowercased tweets. A maximum entropy-based classifier is trained on the following features.

N-grams Word unigrams and bigrams are used as features but in contrast to NRC and GU-MLT-LT not just by occurrence but frequency-weighted. Due to the short tweet length this however often boils down to a simple occurrence feature. To be part of the feature set, an n-gram has to be contained in at least five tweets. This excludes some rather obscure and rare terms or misspellings.

Length The number of tokens in a tweet (i.e., its length) is used as a feature. Interestingly, NRC and GU-MLT-LT do not explicitly use this feature.

Polarity dictionary The employed dictionary is the AFINN-111 lexicon [29] containing 2,447 words with assessments from -5 (very negative) to $+5$ (very positive). Team KLUE added another 343 words. Employed features are the number of positive tokens in a tweet, the number of negative tokens, the number of tokens with a dictionary score, and the arithmetic mean of the scores in a tweet.

Emoticons and abbreviations A list of 212 emoticons and 95 colloquial abbreviations from Wikipedia was manually scored as positive, negative, or neutral. For a tweet, again the number of positive and negative tokens from this list, the total number of scored tokens, and the arithmetic mean are used as features.

Negation Negation is not treated for the whole segment as NRC and GU-MLT-LT do but only on the next three tokens except the case that the punctuation comes earlier. Only negated word unigrams are used as an additional feature set. The polarity scores from the above dictionary are multiplied by -1 for terms up to 4 tokens after the negation.

3.4 Remarks on Reimplementing the Original Approaches

As was to be expected, it turned out to be impossible to re-implement all features precisely as the original authors did. Either not all data was publicly available, or the features themselves were not sufficiently explained in the notebooks. We deliberated to contact the original authors to give them a chance to supply missing data as well as to elaborate on missing information. However, we ultimately opted against doing so for the following reason: our goal was to reproduce their results, not to repeat them. The difference between reproducibility and repeatability is subtle, yet important. If an approach can be re-implemented with incomplete information and if it then achieves a performance within the ballpark of the original, it can be considered much more robust than an approach that must be precisely the same as the original to achieve its expected performance. The former hints reproducibility, the latter only repeatability. This is why we have partly re-invented the approaches on our own, wherever information or data was missing. In doing so, we sometimes found ourselves in a situation where departing from the original approach would yield better performance. In such cases, we decided

Table 1. F1-scores of the original and reimplemented classifiers on the SemEval 2013 test data.

Classifier	Original SemEval 2013	Reimplemented Version
NRC	69.02	69.44
GU-MLT-LT	65.27	67.27
KLUE	63.06	67.05

to maximize performance rather than sticking to the original, since in an evaluation setting, it is unfair to not maximize performance wherever one can.

In particular, the emoticons and abbreviations added by the KLUE team were not available, such that we only choose the AFINN-111 polarity dictionary and re-implemented an emoticon detection and manual polarity scoring ourselves. We also chose not to use the frequency information in the KLUE system but only Boolean occurrence like NRC and GU-MLT-LT, since pilot studies on the SemEval 2013 training and development sets showed that to perform much better. For all three approaches, we unified tweet normalization regarding lowercasing and completely removing URLs and user names instead of adding a placeholder. As for the classifier itself, we did not use the learning algorithms used originally but L2-regularized logistic regression from the LIBLINEAR SVM library for all three approaches. In our pilot experiments on the SemEval 2013 training and development set this showed a very good trade-off between training time and accuracy. We set the cost parameter to 0.05 for NRC and to 0.15 for GU-MLT-LT and KLUE.

Note that neither of our design decisions hurt the individual performances but instead improve the accuracy for GU-MLT-LT and KLUE on the SemEval 2013 test set. Table 1 shows the performance of the original SemEval 2013 ranking and that of our re-implementations. Corresponding to the SemEval scoring, we report the averaged F1-score for the positive and negative class only. As can be seen, the NRC performance stays the same while GU-MLT-LT and KLUE are improved.

Altogether, we conclude that reproducing the SemEval approaches was generally possible but involved some subtleties that lead to difficult design decisions. As outlined, our resolution is to maximize performance rather than to dogmatically stick to the original approach. Our code for the three reproduced approaches as well as that of the ensemble described in the following section is publicly available.[1]

3.5 Ensemble Combination

In our pilot studies on the SemEval 2013 training and development sets, we tested several ways of combining the above three classifiers to an ensemble method. One of the main observations was that each individual approach classifies some tweets correctly that others do fail for. This is not too surprising given the different feature sets but also supports the idea of using an ensemble to combine the individual strengths. Although we briefly tried different ways of bagging and boosting the three classifiers, it soon turned out that some simpler combination performs better. A problem, for instance, was that some misclassified tweets are very difficult (e.g., the positive `Cant wait for`

[1] `http://www.uni-weimar.de/medien/webis/publications/by-year/#stein_2015b`

`the UCLA midnight madness tomorrow night`). Since often at least two classifiers fail on a hard tweet, this rules out some basic combination schemes, such as the majority vote among the three systems (the majority vote turned out to perform worse on the SemEval 2013 development set than NRC alone).

The solution that we finally came up with is motivated by observing how the three classifiers trained on the SemEval 2013 training set behave for tweets in the development set. Typically, not the three final decisions but the respective confidences or probabilities of the individual classifiers give a good hint on uncertainties. If two are not really sure about the final classification, sometimes the remaining third one favors another class with high confidence. Thus, instead of looking at the classifications, we decided to use the confidence scores or probabilities to build the ensemble. This approach is also motivated by old and also more recent papers on ensemble learning [1, 16, 37]. But instead of computing a weighting scheme for the different individual classifiers or learning the weights, we decided to simply compute the average probability of the three classifiers for each of the three classes (positive, negative, neutral).

Our ensemble thus works as follows. The three individual re-implementations of the NRC, the GU-MLT-LT, and the KLUE classifier are individually trained on the SemEval 2013 training set as if being applied individually—without boosting or bagging. As for the classification of a tweet, the ensemble ignores the individual classifiers' classification decisions but requests the classifiers' probabilities (or confidences) for each class. The ensemble decision then chooses the class with the highest average probability—again, no sophisticated techniques like dynamic confidence weighting [16] or set covering schemes [37] are involved. Thus, our final ensemble method is a rather straightforward system based on averaging confidences instead of voting schemes on the actual classifications of the individual classifiers. It can be easily implemented on top of the three classifiers and thus incurs no additional overhead. It also proves very effective in the following experimental evaluation.

4 Evaluation

To evaluate our ensemble approach, we employ the data sets provided for the SemEval 2013 and 2014 Twitter sentiment analysis tasks. More precisely, our setting is that of the subtask B (detect the sentiment of a whole tweet) while subtask A asks to detect the sentiment in a specific part of a tweet.

4.1 Evaluation Setup

The datasets used for the SemEval Twitter sentiment detection subtask B consist of a training set of 9,728 tweets (3,662 positive, 1,466 negative, 4,600 neutral), a developer set of 1,654 tweets (575 positive, 340 negative, 739 neutral), and two test sets. The test set from 2013 contains 3,813 tweets (1,572 positive, 601 negative, 1,640 neutral) while the smaller test set from 2014 contains 1,853 tweets (982 positive, 202 negative, 669 neutral). The tweets were crawled by the task organizers with a focus on topics relevant in the crawling period of January 2012 to January 2013 (the test set of 2014 was added later), including entities (e.g., Gaddafi, Steve Jobs), products (e.g., Kindle, Android phone), or events (e.g., Japan earthquake, NHL playoffs) [28, 38].

Table 2. Ranking and classification results on the SemEval Twitter data based on the F1-scores. The left part shows the original top ranks and the average score of 38 participants of SemEval 2013 with our ensemble included. The right part shows the results for SemEval 2014 (F1-scores for the 2014 and 2013 test data): the top ranks and average of 50 participants with our ensemble included according to its weaker rank on the 2014 data.

Ranking SemEval 2013		**Ranking SemEval 2014**		
Team	F1-score	Team	F1 on 2014	F1 on 2013
Our ensemble	**71.09**	TeamX	70.96	72.12
NRC-Canada	69.02	coooolll	70.14	70.40
GU-MLT-LT	65.27	RTRGO	69.95	69.10
teragram	64.86	NRC-Canada	69.85	70.75
BOUNCE	63.53	**Our ensemble**	**69.79**	**71.09**
KLUE	63.06	TUGAS	69.00	65.64
AMI&ERIC	62.55	CISUC KIS	67.95	67.56
FBM	61.17	SAIL	67.77	66.80
AVAYA	60.84	SWISS-CHOCOLATE	67.54	64.81
SAIL	60.14	Synalp-Empathic	67.43	63.65
Average	53.70	Average	60.41	59.78

The evaluation and ranking at SemEval 2013 and SemEval 2014 is based on the F1-score for the positive and negative tweets only. To compute the positive precision $prec_{\text{pos}}$, the number of tweets that were correctly classified as positive by the system is divided by the total number of tweets classified as positive by the system. Likewise, positive recall rec_{pos} is computed by dividing the number of tweets that were correctly classified as positive by the number of positive tweets in the gold standard test data. The F1-score for the positive class is computed as usual: $F_{\text{pos}} = \frac{2(prec_{\text{pos}}+rec_{\text{pos}})}{prec_{\text{pos}} \cdot rec_{\text{pos}}}$. Similarly, F_{neg} is computed from the negative precision and recall and the final overall score is the average F1-score $F = (F_{\text{pos}} + F_{\text{neg}})/2$.

4.2 General Performance

We use the SemEval 2013 training set to train the individual classifiers of our system. The SemEval 2013 development set is used to obtain the ensemble combination as described in Section 3.5. The ensemble is then tested on the 2013 test set against the participants of SemEval 2013 and on the 2013 and 2014 test sets against the participants of SemEval 2014. The results are shown in Table 2.

As can be seen on the 2013 participants (left part of Table 2), our ensemble method outperforms all 2013 participants and thus also the individual classifiers forming the ensemble. On the 2013 test data, our ensemble still takes the second place among the participants of SemEval 2014 while on the 2014 test data, our ensemble is ranked fifth (right part of Table 2). This places our ensemble system among the top-5 approaches in the two years of Twitter sentiment detection at SemEval. It would be an interesting direction for future research to try including new top-performing systems in our ensemble and to identify approaches among the top-performing 2014 participants that implement different paradigms to complement our ensemble. In a further analysis of the evaluation results, we examine the influence of the different classifiers in the ensemble and the characteristics of tweets with classification errors.

Table 3. F1-scores of the ensemble and without each individual classifier on the 2013 test data.

Ensemble	**F1-score**	$prec_{\text{pos}}$	rec_{pos}	$prec_{\text{neg}}$	rec_{neg}
All	71.09	72.61	79.60	65.73	66.72
All - GU-MLT-LT	70.67 (-0.42)	72.83	78.78	67.31	64.06
All - KLUE	70.56 (-0.53)	73.39	78.59	65.22	65.22
All - NRC	68.80 (-2.29)	69.15	78.71	57.97	71.38

4.3 Component influence

To check the influence of each individual classifier in our ensemble, we compare the ensemble of three classifiers to the ensembles with just two approaches—again, classification always is done by averaging the confidence scores. As can be seen from Table 3, each component is important for the overall score of the system. Leaving out the individually best classifier NRC reduced the performance the most but still results in an ensemble better than the two individual approaches. This is not too surprising since the observations on the development set showed that each individual classifier's confidence scores can sometimes help to avoid misclassifications. Hence, none of the individual and different classifiers should be removed from the ensemble. Adding appropriate other approaches is an interesting direction for future work.

4.4 Error analysis

Analyzing the misclassifications of our ensemble sheds light on its robustness. The confusion matrices in Table 4 show a particularly nice feature of our ensemble. There are hardly severe misclassifications like classifying a positive tweet to be negative and vice versa. Most of the misclassifications put a positive or negative tweet in the neutral class which can be viewed as a rather safe option: in doubt it is often better to leave something without clear classification.

Altogether, the experimental results show our ensemble system to be able to robustly classify sentiment in tweets across different data sets and to rank among the top-performing approaches. The system itself is build straightforward from the individual approaches, each having their share in the achieved detection scores. Most of the errors observed concerns the misclassification of a tweet as neutral which is not the worst misclassification possible.

5 Conclusion and Outlook

We have reproduced three state-of-the-art approaches to sentiment detection for Twitter tweets. Our findings include that not all aspects of the approaches could be reproduced precisely, but that missing data, missing information, as well as opportunities to improve the approaches' performances lead us to re-invent them and to depart to some extent from the original descriptions. All of our changes have improved the performances of the original approaches. Moreover, we have demonstrated that the approaches can be reproduced even with incomplete information about them, which is a much stronger property than being merely repeatable.

Table 4. Confusion matrices of gold and computed classes for SemEval 2013 and 2014 test data.

		Computed classification of our ensemble					
		SemEval 2013			**SemEval 2014**		
		positive	neutral	negative	positive	neutral	negative
Gold label	positive	1,249	234	86	768	184	30
	neutral	394	1,123	123	177	436	56
	negative	77	123	401	41	33	128

In addition, we investigated a combination of confidence scores of the three approaches within an ensemble that altogether yields a top-performing Twitter sentiment detection system. The ensemble computation is as efficient as its components, and its effectiveness was shown on the SemEval Twitter sentiment evaluation data where our system ranks among the top-5 approaches for various data sets.

Promising directions for future research are the analysis of further potential classifiers useful as ensemble components. Following our philosophy of selecting approaches that are significantly different from each other, it will be interesting to observe how much new approaches can improve our existing ensemble. A further error analysis from a retrieval perspective might also be promising. We found that our ensemble classifier makes few severe errors, such as misclassifying a negative tweet as positive or vice versa. From a retrieval perspective, it would be nice to examine the influence of such misclassifications on the perceived usefulness of a system that incorporates sentiment detection results into search results, e.g., for diversification.

Bibliography

[1] L. Asker and R. Maclin. Ensembles as a sequence of classifiers. In *Proc. of IJCAI 1997*, pp. 860–865.
[2] S. Baccianella, A. Esuli, and F. Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proc. of LREC 2010*.
[3] A. Balahur and M. Turchi. Improving sentiment analysis in twitter using multilingual machine translated data. In *Proc. of RANLP 2013*, pp. 49–55.
[4] L. Barbosa and J. Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proc. of COLING 2010*, pp. 36–44.
[5] A. Bifet and E. Frank. Sentiment knowledge discovery in twitter streaming data. In *Proc. of DS 2010*, pp. 1–15.
[6] J. Bollen, H. Mao, and A. Pepe. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *Proc. of ICWSM 2011*.
[7] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
[8] S. Brody and N. Diakopoulos. Cooooooooooooooollllllllllll!!!!!!!!!!!!!! Using word lengthening to detect sentiment in microblogs. In *Proc. of EMNLP 2011*, pp. 562–570.
[9] P. F. Brown, V. J. Della Pietra, P. V. de Souza, J. C. Lai, and R. L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.
[10] J. Carrillo de Albornoz, L. Plaza, P. Gervás, and A. Díaz. A joint model of feature mining and sentiment analysis for product review rating. In *Proc. of ECIR 2011*, pp. 55–66.
[11] G. Demartini. ARES: A retrieval engine based on sentiments - sentiment-based search result annotation and diversification. In *Proc. of ECIR 2011*, pp. 772–775.
[12] N. Diakopoulos and D. A. Shamma. Characterizing debate performance via aggregated twitter sentiment. In *Proc. of CHI 2010*, pp. 1195–1198.
[13] S. Ermakov and L. Ermakova. Sentiment classification based on phonetic characteristics. In *Proc. of ECIR 2013*, pp. 706–709.
[14] R. Feldman. Techniques and applications for sentiment analysis. *CACM*, 56(4):82–89, 2013.

[15] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proc. of ICML 1996*, pp. 148–156.
[16] G. P. C. Fung, J. X. Yu, H. Wang, D. W. Cheung, and H. Liu. A balanced ensemble approach to weighting classifiers for text classification. In *Proc. of ICDM 2006*, pp. 869–873.
[17] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. Project Report CS224N, Stanford University, 2009.
[18] T. Günther and L. Furrer. GU-MLT-LT: Sentiment analysis of short messages using linguistic features and stochastic gradient descent. In *Proc. of SemEval 2013*, pp. 328–332.
[19] Y. He. Latent sentiment model for weakly-supervised cross-lingual sentiment classification. In *Proc. of ECIR 2011*, pp. 214–225.
[20] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proc. of KDD 2004*, pp. 168–177.
[21] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao. Target-dependent twitter sentiment classification. In *Proc. of HLT 2011*, pp. 151–160.
[22] J. Karlgren, M. Sahlgren, F. Olsson, F. Espinoza, and O. Hamfors. Usefulness of sentiment analysis. In *Proc. of ECIR 2012*, pp. 426–435.
[23] E. Kouloumpis, T. Wilson, and J. D. Moore. Twitter sentiment analysis: The good the bad and the OMG! In *Proc. of ICWSM 2011*.
[24] S. M. Mohammad and P. D. Turney. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proc. of HLT 2010 Workshop CAAGET 2010*, pp. 26–34.
[25] S. M. Mohammad and P. D. Turney. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.
[26] S. M. Mohammad, S. Kiritchenko, and X. Zhu. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proc. of SemEval 2013*, pp. 321–327.
[27] A. Moniz and F. de Jong. Sentiment analysis and the impact of employee satisfaction on firm earnings. In *Proc. of ECIR 2014*, pp. 519–527.
[28] P. Nakov, Z. Kozareva, A. Ritter, S. Rosenthal, V. Stoyanov, and T. Wilson. Semeval-2013 task 2: Sentiment analysis in Twitter. In *Proc. of SemEval 2013*, pp. 312–320.
[29] F. Å. Nielsen. A new ANEW: evaluation of a word list for sentiment analysis in microblogs. In *Proc. of ESWC 2011 Workshop MSM 2011*, pp. 93–98.
[30] D. W. Opitz and R. Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.
[31] O. Owoputi, B. O'Connor, C. Dyer, K. Gimpel, N. Schneider, and N. A. Smith. Improved part-of-speech tagging for online conversational text with word clusters. In *Proc. of HLT 2013*, pp. 380–390.
[32] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proc. of EMNLP 2002*, pp. 79–86.
[33] R. Polikar. Ensemble based systems in decision making. *IEEE CASS Mag.*, 6(3):21–45, 2006.
[34] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
[35] T. Proisl, P. Greiner, S. Evert, and B. Kabashi. Klue: Simple and robust methods for polarity classification. In *Proc. of SemEval 2013*, pp. 395–401.
[36] L. Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39, 2010.
[37] L. Rokach, A. Schclar, and E. Itach. Ensemble methods for multi-label classification. *Expert Systems with Applications*, 41(16):7507–7523, 2014.
[38] S. Rosenthal, A. Ritter, P. Nakov, and V. Stoyanov. Semeval-2014 task 9: Sentiment analysis in twitter. In *Proc. of SemEval 2014*, pp. 73–80.
[39] R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.
[40] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proc. of ACL 2014*, pp. 1555–1565.
[41] P. D. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proc. of ACL 2002*, pp. 417–424.
[42] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proc. of EMNLP 2005*, pp. 347–354.