Originally published in: Proceedings of the 1st Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA), August 17 2010, Lisbon, Portugal. - Alicante: Universidad de Alicante, 2010. Pp. 59-66.

Bootstrapping Supervised Machine-learning Polarity Classif ers with Rule-based Classif cation

Michael Wiegand and **Dietrich Klakow**[1]

Abstract. In this paper, we explore the effectiveness of bootstrapping supervised machine-learning polarity classif ers using the output of domain-independent rule-based classif ers. The benef t of this method is that no labeled training data are required. Still, this method allows to capture in-domain knowledge by training the supervised classif er on in-domain features, such as bag of words.

We investigate how important the quality of the rule-based classif er is and what features are useful for the supervised classif er. The former addresses the issue in how far relevant constructions for polarity classif cation, such as word sense disambiguation, negation modeling, or intensif cation, are important for this self-training approach. We not only compare how this method relates to conventional semi-supervised learning but also examine how it performs under more diff cult settings in which classes are not balanced and mixed reviews are included in the dataset.

1 Introduction

Recent years have seen a growing interest in the automatic text analysis of opinionated content. One of the most popular subtasks in this area is polarity classif cation which is the task of distinguishing between positive utterances (Sentence 1) and negative utterances (Sentence 2).

1. The new iPhone looks *great* and is *easy* to handle.
2. London is *awful*; it's *crime-ridden*, *dirty* and full of *rude* people.

Various supervised classif cation approaches, in particular classif ers using bag of words, are heavily domain-dependent [2], i.e., they usually generalize fairly badly across different domains. Yet the costs to label data for any possible domain are prohibitively expensive.

Semi-supervised learning tries to solve this issue by reducing the size of the labeled dataset. The lack of labeled training data is compensated by a large unlabeled dataset of the target domain. The latter is much cheaper to obtain.

Rule-based classif cation does not require any labeled training data. In polarity classif cation, the rule-based classif er relies on domain-independent polar expressions. Polar expressions are words containing a prior polarity, such as *great* and *awful*. One typically counts the number of positive and negative polar expressions in a test instance and assigns it the polarity type with the majority of polar expressions. Since the classif er is restricted to domain-independent polar expressions, it lacks the knowledge to recognize domain-specif c polar expressions, such as *crunchy*$^{+}$ in the food domain or *buggy*$^{-}$ in the computer domain.

[1] Spoken Language Systems, Saarland University, Germany, email: {michael.wiegand, dietrich.klakow}@lsv.uni.saarland.de

In this paper, we explore the effectiveness of an alternative, which like most semi-supervised learning algorithms is based on *self-training*, i.e., the process of labeling the unlabeled data with a preliminary classif er and then training another (more robust) classif er by using the expanded annotated dataset. Unlike traditional semi-supervised learning, we do not use an initial classif er trained on a small labeled dataset but the output of a domain-independent rule-based classif er. (For reasons of simplicity, we will often refer to this specif c version as plain *self-training* in the following sections.) While the rule-based classif er is restricted to the knowledge of (domain-independent) polar expressions, the supervised classif er trained on in-domain data labeled by the rule-based classif er can make use of domain-specif c features, such as bag of words. Hopefully, the supervised classif er can effectively use this domain-specif c knowledge and thus outperform the rule-based classif er.

Though this kind of self-training has already been applied to tasks in opinion mining, including polarity classif cation, there are certain aspects of this method which have not yet been fully examined:

Firstly, what are good features for the (pseudo-)supervised polarity classif er which is trained on the data labeled by the rule-based classif er? Do the insights hold from common supervised learning or semi-supervised learning?

Secondly, what is the impact of the robustness of the rule-based classif er on the f nal classif ers, i.e., does the supervised classif er improve when the rule-based classif er improves? This addresses the issue of in how far relevant constructions for polarity classif cation that can be incorporated into a rule-based classif er, such as word disambiguation, negation modeling, or intensif cation, are important for this kind of self-training approach.

Thirdly, how does this type of self-training compare to state-of-the-art semi-supervised learning algorithms?

Finally, does this method work in realistic settings in which – in addition to def nite polar reviews – also mixed polar reviews are part of the dataset and the distribution of the classes is imbalanced?

2 Related Work

There has been much work on document-level polarity classif cation using supervised machine learning methods. Various classif ers and feature sets have been explored [10, 11]. Support Vector Machines (SVMs) [5] usually provide best results [11]. Unigram and bigram features outperform complex linguistic features [10].

Rule-based polarity classif cation usually requires an open-domain polarity lexicon with polar expressions. One typically counts the number of positive and negative polar expressions occurring in a test document and assigns it the polarity type with most polar expressions. This method can be enhanced by disambiguating polar expressions in their respective contexts. A framework in which scores are

Publikationsserver des Instituts für Deutsche Sprache
URN: http://nbn-resolving.de/urn:nbn:de:bsz:mh39-84473

heuristically assigned to polar expressions depending on their individual contexts is proposed in [12]. The contextual modeling mainly focuses on *negation* and *intensif cation*. Implementations inspired by that formalism have been empirically proven effective [7, 8, 9].

Semi-supervised learning for polarity classif cation has been shown to be effective on inducing polarity lexicons from lexical resources [3, 14] but on text classif cation, the effectiveness is heavily dependent on the parameter settings. Signif cant improvement over supervised classif cation can usually only be achieved in presence of few labeled training data and a predictive feature set, such as in-domain adjectives or polar expressions from a polarity lexicon [17]. Another effective semi-supervised approach suggests to apply unsupervised learning (i.e., clustering) to classify unambiguous data instances and restrict manual annotation to hard data instances [4].

Bootstrapping supervised machine-learning classif ers with the help of rule-based classif cation has already been effectively applied to subjectivity detection of sentences [16]. The method has also been applied to polarity classif cation, but so far only on Chinese data [13, 15]. While the performance with out-of-domain supervised classif ers is compared in [15], this method is embedded into a complex bootstrapping system which also extends the vocabulary (or feature set) of the rule-based classif er in [13]. Neither of these works examine the impact of the rule-based classif er on the f nal result, the relation towards semi-supervised learning, nor discusses various settings of the self-training algorithm, in particular, different feature sets for the supervised classif er.

3 Data

In this paper, we use both the dataset of *IMDb* movie reviews [11] and reviews extracted from *Rate-It-All*[2]. We evaluate on the former because it is considered a benchmark dataset for polarity classif cation. The additional data are used to show that our f ndings are valid throughout different domains. Moreover, they have also been used in prior work on semi-supervised learning [17] which we also make use of in our experiments. Table 1 lists the properties of the corpora from the different domains. Note that on the *Rate-It-All* datasets we labeled 1 and 2 star reviews as *negative* and 4 and 5 star reviews as *positive*. 3 star reviews are labeled *mixed*. The actual class of these reviews is unknown. Usually a 3 star review should be neutral in the sense that it equally enumerates both positive and negative aspects about a certain topic, so that a def nite verdict in favor or against it is not possible. That is also why we cannot assign these instances to either of the other two groups previously mentioned, i.e., *positive* and *negative*. During a manual inspection of some randomly chosen instances, however, we also found def nite positive and negative reviews among 3 star reviews. For this work, we leave these instances in the category of mixed reviews.

4 Method

4.1 Rule-based Classif er

In the following, we describe how a polarity lexicon is converted to a rule-based polarity classif er. The polarity lexicon, the list of other important word classes being intensif ers, negation expressions (including the rules to disambiguate them) and polarity shifters are taken from the *MPQA* project [18].

[2] `http://www.rateitall.com`

4.1.1 Feature Extraction

Any word in a review that is not included in a polarity lexicon is discarded. Positive words (e.g., *excellent*) are assigned the value $+1$, negative words (e.g., *awful*) -1, respectively.

4.1.2 Basic Word Sense Disambiguation with Part-of-speech Tags

The polarity lexicon we use has part-of-speech tags attached to polar expressions in order to disambiguate them, e.g., the word *like* is either a polar verb or a preposition (in which case it is meaningless for polarity classif cation). We identify words as polar expressions only if their part-of-speech tags also match the specif cation in the lexicon. This can be considered as some basic form of word sense disambiguation. For part-of-speech tagging we use the *C&C* tagger[3].

4.1.3 Negation Modeling

If a polar expression occurs within the scope of a negation, its polarity is reversed (e.g., $[\textit{not}\ \text{nice}^{+}]^{-}$). By scope, we def ne the f ve words immediately preceding the polar expression in the same sentence. Since some negation words are ambiguous and do not express negations when used in certain constructions, such as *not* in *not only . . . but also*, we also apply some rules disambiguating negation words.

In addition to common negation expressions, such as *not*, we also consider *polarity shifters*. Polarity shifters are weaker than ordinary negation expressions in the sense that they only reverse a particular polarity type. For example, the shifter *abate* only modif es negative polar expressions as in $[\textit{abate the damage}^{-}]^{+}$.

4.1.4 Heuristic Weighting

So far, all polar expressions contained in the polarity lexicon are assigned the same absolute weight, i.e., $(\pm)1$. This does not ref ect reality. Polar expressions differ in their individual polar intensity or, in case of ambiguous words, in their likelihood to convey polarity. Therefore, they should not obtain a uniform weight.

The polarity lexicon we use [18] includes a binary feature expressing the prior intensity of a polar expression. It distinguishes between *weak* polar expressions, such as *disordered*, and *strong* polar expressions, such as *chaotic*. Intuitively, strong polar expressions should obtain a higher weight than weak polar expressions.

When a polar expression is modifed by a so-called *intensif er*, such as *very* or *extremely*, its polar intensity is also increased. An ordinary weak polar expression has a similar polar intensity when it is modif ed by an intensif er as a strong polar expression, e.g., *extremely disordered* and *chaotic*.

The part of speech of a polar expression usually sheds light on the level of ambiguity of the word. If a polar expression is an *adjective*, its prior probability of being polar is much higher than the one of polar expressions with other parts of speech, such as verbs or nouns [11, 17]. Therefore, polar adjectives should obtain a larger weight than polar expressions with other parts of speech.

Since there are no development data in order to adjust the weights for the previously mentioned properties, we propose to simply *double* the value of a polar expression if either of these properties apply. If n of these properties apply for a polar expression, then its value is

[3] `http://svn.ask.it.usyd.edu.au/trac/candc`

Table 1. Properties of the different domain corpora (†only relates to the *Rate-It-All* data).

Domain	Source	Positive (4 & 5 Stars†)	Mixed (3 Stars†)	Negative (1 & 2 Stars†)	Vocabulary Size
computer	*Rate-It-All*	952	428	1253	15083
products	*Rate-It-All*	2292	554	1342	21975
sports	*Rate-It-All*	4975	725	1348	24811
travel	*Rate-It-All*	9397	1772	3289	38819
movies	*IMDb*	1000	0	1000	50920

doubled n times. For instance, an intensif ed adjective is assigned the value of 4, i.e., $2 \cdot 2$.

The properties considered for heuristic weighting have already been motivated and proven effective in previous work [7, 11].

4.1.5 Classif cation

For each data instance the *contextual* scores assigned to the individual polar expressions are summed. If the sum is positive, then the instance is classif ed as positive. It is classif ed as negative, if the sum is negative. We assign to all cases in which the sum is 0 the polarity type which gives best performance on that individual dataset (which is usually negative polarity). Thus, we have a stronger baseline that is to be beaten by self-training.

Note that the prediction score of a data instance, i.e., the sum of contextual scores of the polar expressions, can also be interpreted as a conf dence score. This property is vital for effectively using this rule-based classif er in self-training. Thus, previously mentioned instances with a score of 0, for example, are unlikely to occur in the labeled training set since it only includes instances labeled with a high conf dence score. The sum of contextual scores is normalized by the overall number of tokens in a test instance. This normalization additionally encodes the density of polar expressions within the instance. The greater the density of polar expressions of a particular type is in a text, the more likely the text conveys that polarity.

Figure 1 summarizes all steps of the rule-based classif er.

1. Lexicon loading, i.e., polar expressions, negation words, and intensif ers
2. Preprocessing:
   (i) Stem test instance.
   (ii) Apply part-of-speech tagging to test instance.
3. Polar expression marking:
   (i) Check whether part-of-speech tag of potential polar expression matches lexical entry (*basic word sense disambiguation*).
   (ii) Mark strong polar expressions.
4. Negation modeling:
   (i) Identify potential negation words (including polarity shifters).
   (ii) Disambiguate negation words.
   (iii) Reverse polarity of polar expression in scope of (genuine) negation.
5. Intensif er marking
6. Heuristic weighting: double weight in case polar expression is:
   (i) a strong polar expression
   (ii) an intensif ed polar expression
   (iii) a polar adjective.
7. Classif cation: assign test instance the polarity type with the largest (normalized) sum of scores.

Figure 1. Rule-based classif er.

4.1.6 Different Versions of Classif ers

We def ne four different types of rule-based classif ers. They differ in complexity. The simplest classif er, i.e., RB_{Plain}, does not contain word sense disambiguation, negation modeling or heuristic weighting. RB_{bWSD} is like RB_{Plain} but also contains basic word sense disambiguation. RB_{Neg} is like RB_{bWSD} but also contains negation modeling. The most complex classif er, i.e., RB_{Weight}, is precisely the algorithm presented in the previous sections. Table 2 summarizes the different classif ers with their respective properties.

4.2 Semi-Supervised Learning

Semi-supervised learning is a class of machine learning methods that makes use of both labeled and unlabeled data for training, usually a small set of labeled data and large set of unlabeled data. A classif er using unlabeled and labeled training data can produce better performance than a classif er trained on labeled data alone. This is usually achieved by harnessing correlations between features in labeled and unlabeled data instances and thus making inferences about the label of these unlabeled instances. Since labeled data are expensive to produce, semi-supervised learning is an inexpensive alternative to supervised learning.

In this paper, we exclusively use Spectral Graph Transduction (SGT) [6] as a semi-supervised algorithm since it produced consistently better results than other algorithms on polarity classif cation in previous work [17]. In SGT, all instances of a collection (i.e., labeled and unlabeled) are represented as a k nearest-neighbor graph. The graph is transformed to a lower-dimensional feature space, i.e., its spectrum, and then divided into two clusters by minimizing the graph cut. The two clusters that are chosen should preserve the highest possible connectivity of edges within the graph.

4.3 Self-Training a Polarity Classif er using the Output of a Rule-based Classif er

The idea of this bootstrapping method is that a domain-independent rule-based classif er is used to label an unlabeled dataset. Unlike in semi-supervised learning (Section 4.2), no labeled training data are used. The only available knowledge is encoded in the rule-based classif er. The data instances labeled by the rule-based classif er with a high conf dence serve as labeled training data for a supervised machine-learning classif er. Ideally, the resulting supervised classif er is more robust on the domain on which it was trained than the rule-based classif er. The improvement can be explained by the fact that the rule-based classif er only comprises domain-independent knowledge. The supervised classif er, however, makes use of domain-specif c features, i.e., words such as *crunchy*$^+$ (food domain) or *buggy*$^-$ (computer domain), which are not part of the rule-based classif er. It may also learn to correct polar expressions that are specif ed in the polarity lexicon but have a wrong polarity

Table 2. Properties of the different rule-based classifiers.

Properties	RB_{Plain}	RB_{bWSD}	RB_{Neg}	RB_{Weight}
basic word sense disambiguation		✓	✓	✓
negation modeling			✓	✓
heuristic weighting				✓

type on the target domain. A reason for a type mismatch may be that a polar expression is ambiguous and contains different polarity types throughout the different domains (and common polarity lexicons usually only specify one polarity type per entry). For instance, in the movie domain the polar expression *cheap* is predominantly negative, as it can be found in expressions, such as *cheap flms*, *cheap special-effects* etc. In the computer domain, however, it is predominantly positive as it appears in expressions such as *cheap price*. If such a polar expression occurs in sufficient documents which the rule-based classifier has labeled correctly, then the supervised learner may learn the correct polarity type for this ambiguous expression on that domain despite the fact that the opposed type is specified in the polarity lexicon.

We argue that using a rule-based classifier is more worthwhile than using few labeled (in-domain) data instances – as it is the case in semi-supervised learning – since we thus exploit two different types of features in self-training being domain-independent polar expressions and domain-specific bag of words which are known to be complementary [1]. The traditional semi-supervised approach usually just comprises one homogeneous feature set.

Figure 2 illustrates both semi-supervised learning and self-training using a rule-based classifier for bootstrapping.

4.4 Feature Sets

Table 3 lists the different feature sets we examine for the supervised classifier (within self-training) and the semi-supervised classifiers. We list the feature sets along their abbreviation with which they will henceforth be addressed. The first three features (i.e., Top2000, Adj600, and MPQA) have been used in previous work on semi-supervised learning [17]. They all remove noise contained in the overall vocabulary of a domain corpus. The last two features (i.e., Uni and Uni+Bi) are known to be effective for supervised polarity classification [10]. Bigrams can be helpful in addition to unigrams since they take into account some context of polar expressions. Thus, crucial constructions, such as negation (*[not nice]*$^{-}$) or intensification (*[extremely nice]*$^{++}$), can be captured. Moreover, multiword polar expressions, such as *[low tax]*$^{+}$ or *[low grades]*$^{-}$, can be represented as individual features. Unfortunately, bigram features are also fairly sparse and contain a considerable amount of noise.

Table 3. Description of the different feature sets.

Feature Set	Abbrev.
the 2000 most frequent non-stopwords in the domain corpus	Top2000
the 600 most frequent adjectives and adverbs in the domain corpus	Adj600
all polar expressions within the polarity lexicon	MPQA
all unigrams in the domain corpus	Uni
all unigrams and bigrams in the domain corpus	Uni+Bi

5 Experiments

For the following experiments – with the exception of those presented in Section 5.4 – we mainly adhere to the settings of previous work [17]. We deliberately chose these settings in favor of semi-supervised learning in order to have a strong baseline for the proposed self-training method. We use a balanced subset (randomly generated) for each domain. The *Rate-It-All* dataset consists of 1800 data instances per domain, whereas the *IMDb* dataset consists of 2000 data instances. We just consider (definite) positive and (definite) negative reviews. The rule-based classifiers and the self-trained classifiers (bootstrapped with the help of rule-based classification) are evaluated on the entire domain dataset. The 1000 most highly-ranked data instances (i.e., 500 positive and 500 negative instances) are chosen as training data for the supervised classifier. This setting, which is similar to the one used for semi-supervised learning [17], provided good performance in our initial experiments. For the supervised classifier, we chose SVMs. As a toolkit, we use *SVMLight*[4]. Feature vectors were always normalized to unit length and additionally weighted with *tf-idf* scores. All words are stemmed. We report statistical significance on the basis of a paired t-test using 0.05 as the significance level.

5.1 Comparison of Different Rule-based Classifiers

Table 4 shows the results of the different rule-based classifiers across the different domains. On average, the more complex the rule-based classifier gets, the better it performs. The only notable exceptions are the *products* domain (from RB_{Neg} to RB_{Weight}) and the *sports* domain (from RB_{Plain} to RB_{bWSD}). On average (i.e., considering all domains), however, the improvements are statistically significant.

5.2 Self-Training with Different Rule-based Classifiers and Different Feature Sets

Table 5 compares self-training (SelfTr) using different rule-based classifiers and different feature sets for the embedded supervised classifier. In addition to accuracy, we also listed the F(1)-scores of the two different classes. The results are averaged over all domains. With the exception of RB_{Neg} in combination with Top2000 and MPQA, there is always a significant improvement from a rule-based classifier to the corresponding self-trained version. If Top2000 or MPQA is used, there is a drop in performance from RB_{Neg} to SelfTr in the *sports* domain. Improving a rule-based classifier also results in an improvement of the self-trained classifier. With exception of SelfTr(RB_{Plain}) to SelfTr(RB_{bWSD}) this is even significant.

The feature set producing the best results is Uni+Bi. Uni+Bi is statistically significantly better than Uni. This means that, as far as feature design is concerned, the supervised classifier within self-training behaves similar to ordinary supervised classification [10]. Unlike in semi-supervised learning [17], a noiseless feature set is not necessary. Best performance of SelfTr using a large set of polar

[4] `http://svmlight.joachims.org`

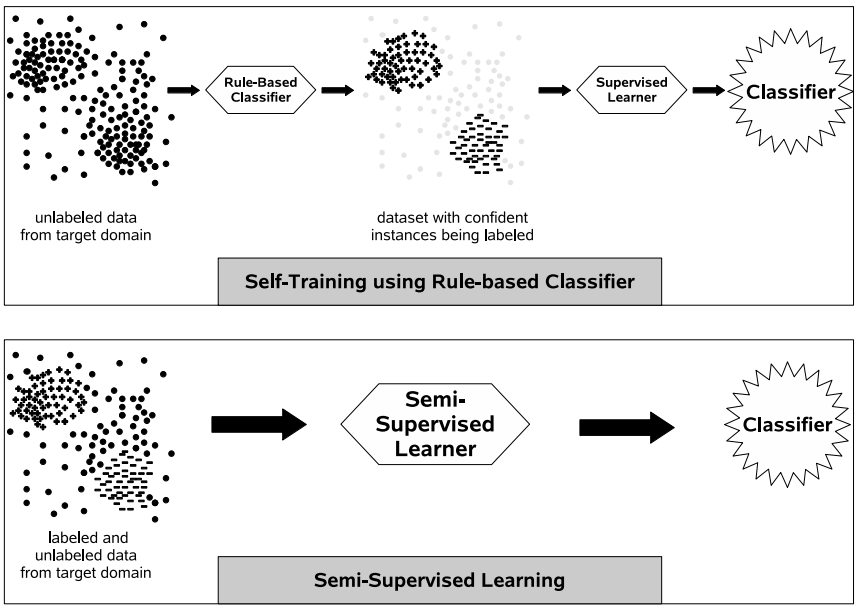

Figure 2. Comparison of semi-supervised learning and self-training using a rule-based classif er for bootstrapping.

Table 4. Comparison of accuracy between different rule-based classif ers (RB) and self-trained classif ers (SelfTr) trained on best feature set (Uni+Bi) on different domains (for each domain, performance is evaluated on a balanced corpus).

	$\mathbf{RB}_{Plain}$		$\mathbf{RB}_{bWSD}$		$\mathbf{RB}_{Neg}$		$\mathbf{RB}_{Weight}$	
Domain	**RB**	**SelfTr**	**RB**	**SelfTr**	**RB**	**SelfTr**	**RB**	**SelfTr**
computer	64.11	80.22	70.61	81.72	73.56	83.67	74.28	83.50
products	60.78	70.78	66.06	73.89	71.06	77.00	70.94	77.00
sports	64.33	66.44	64.39	64.94	67.50	68.89	68.89	72.78
travel	64.61	69.56	67.39	69.83	70.72	73.33	72.61	76.89
movies	61.75	72.70	64.80	72.45	67.85	73.55	71.30	77.75
average	63.12	71.94	66.65	72.57	70.14	75.29	71.60	77.58

expressions is reported in [13]. The feature set comprises an open-domain polarity lexicon and is automatically extended by domain-specif c expressions. Our results suggest a less complex alternative. Using SelfTr with unigrams and bigrams (i.e., SelfTr_{Uni+Bi}) already provides better classif ers than SelfTr with a polarity lexicon (i.e., SelfTr_{MPQA}). The increase is approx. 3%.

It is also worth pointing out that the gain in performance that is achieved by improving a basic rule-based classif er (i.e., RB_{Plain}) by modeling constructions (i.e., RB_{Weight}) is the same as is gained by just self-training it with the best feature set (i.e., SelfTr_{Uni+Bi}).

The relation between the F-scores of the two different classes differs between RB and SelfTr. In RB, the score of the positive class is always signif cantly better than the score of the negative class. This is consistent with previous f ndings [1]. The gap between the two classes, however, varies depending on the complexity of the classif er. In RB_{Plain}, the gap is 17.45%, whereas it is less than 6% in RB_{Neg} and RB_{Weight}. In SelfTr, the F-score of the negative class is usually better than the score of the positive class[5]. This relation between the two classes is typical of learning-based polarity classif ers [1]. However, it should also be pointed out that the gap is much smaller (usually not greater than 2%). Moreover, the size of the gap does not bear any relation to the gap in the original RB, i.e., though there is a considerable difference in size between the gaps of RB_{Plain} and RB_{Neg}, the size of the gaps in the self-trained versions is fairly similar.

We also experimented with a combination of bag of words and the knowledge encoded in the rule-based classif er, i.e., the two features: the number of positive and negative polar expressions within a data instance. The performance of this combination is worse than a classif er trained on bag of words. The correlation between the two class labels and the two polarity features is disproportionately high since the polarity features essentially encode the prediction of the rule-based classif er. Consequently, the supervised classif ers develop a strong bias towards these two features and inappropriately downweight the bag-of-words features.

Table 4 compares rule-based classif cation and self-training on individual domains. In some domains self-training does not work. This is most evident in the *sports* domain using self-training on RB_{bWSD}.

[5] The only exception where the reverse is always true is SelfTr_{MPQA}. This does not come as a surprise since this feature set resembles RB most.

Apparently, the better the rule-based classif er is, the more likely a notable improvement by self-training can be obtained. Note that in the *sports* domain the self-trained classif er using the most complex rule-based classif er, i.e., SelfTr(RB$_{Weight}$), achieves the largest improvement compared to the rule-based classif er. These observations are also representative for the remaining feature sets examined but not displayed in Table 4.

5.3 Self-Training using Rule-based Classif ers Compared to Semi-Supervised Learning

In the following experiments, we use Spectral Graph Transduction (SGT) [6] as a semi-supervised classif er, since it provided best performance in previous work [17]. As a toolkit, we use *SGTLight*[6]. For each conf guration (i.e., training and test partition) we randomly sample 20 partitions from the corpus. Labeled training and test data are always mutually exclusive but the test data (500 positive and 500 negative instances) can be identical to the unlabeled training data.

Figure 3 compares self-training bootstrapped on the output of rule-based classif cation (SelfTr) to supervised learning (SL) and semi-supervised learning (SSL). We compare two variations of SelfTr. SelfTr-A, as SSL, uses the same 1000 randomly sampled data instances for both unlabeled training and testing[7]. (Again, we report the averaged result over 20 samples.) SelfTr-B (like in previous sections) selects 1000 training instances by conf dence from the entire dataset. The test data are, however, the same as in SelfTr-A. Unlike in previous work in which Top2000 is used for SL [17], we chose Uni+Bi as a feature set. It produces better results than Top2000 on classif ers trained on larger training sets (i.e., $\geq$ 400)[8]. For SSL, we consider Uni+Bi and Adj600, which is the feature set with the overall best performance using that learning method. For SelfTr, we consider the best classif er, i.e., SelfTr$_{Uni+Bi}$.

Though SSL gives a notable improvement on small labeled training sets (i.e., $\leq$ 100), it produces much worse performance than SL on large training sets (i.e., $\geq$ 200). Adjectives and adverbs are a very reliable predictor. However, the size of the feature set is fairly small. Too little structure can be learned on large labeled training sets using such a small feature set. Using larger (but also noisier) feature sets for SSL, such as Uni+Bi, improves performance on larger labeled training sets. However, even with Uni+Bi SSL does not reach a performance comparable to SL on large training sets and it is signif cantly worse than Adj600 on small training sets.

Whenever SSL outperforms SL, every variation of SelfTr also outperforms SSL. SelfTr-B is signif cantly better than SelfTr-A which means that the quality of labeled instances matters and SelfTr is able to select more meaningful data instances than are provided by random sampling. Unfortunately, SSL-methods, such as SGT, do not incorporate such a selection procedure for the unlabeled data. Further exploratory experiments using the *entire* dataset as unlabeled data for SSL produced, on average, results similar to those using 1000 instances. This proves that SSL cannot internally identify as meaningful data as SelfTr-B does. Whereas SSL signif cantly outperforms SL on training sets using less than 200 training instances, the best variation of SelfTr, i.e., SelfTr-B, signif cantly outperforms SL on training sets using less than 400 instances. This difference is, in particular, remarkable since SelfTr does not use any labeled training data at all whereas SSL does.

[6] `http://sgt.joachims.org`
[7] We use this conf guration since it is required by *SGTLight*.
[8] Note that previous work in particular focused on small training sets [17].

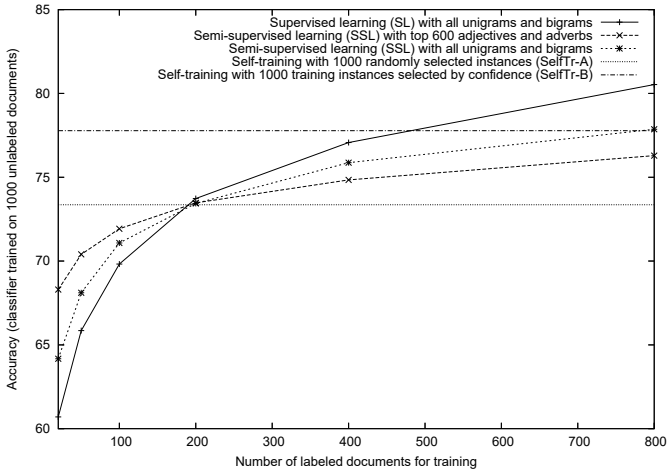

Figure 3. Comparison of self-training and semi-supervised learning (performance is evaluated on balanced corpus and results are averaged over all domains).

5.4 Natural Class Imbalance and Mixed Reviews

In this section, we want to investigate what impact natural class imbalance has on bootstrapping polarity classif ers with a rule-based classif er since this aspect has only been marginally covered in previous work [13, 15]. In those works, different class ratios on the test set are evaluated. However, the same amount of positive and negative reviews is always selected for training. We assume that the optimal performance of self-training can be achieved when the class distribution of training and test set is identical and we will provide evidence for that. Moreover, we want to explore what impact different distributions between the two sets have on the accuracy of the classif er and how different class-ratio estimation methods perform.

Previous work dealing with bootstrapping polarity classif ers using unlabeled data also focuses on datasets exclusively consisting of def nite positive and negative reviews [4, 13, 15, 17]. In this section, the unlabeled dataset will also include mixed reviews, i.e., 3 star reviews (see Section 3). Due to the availability of such data the experiments are only carried out on the *Rate-It-All* data. We also add the constraint that the test data must be disjoint from the unlabeled training data[9].

Test data are exclusively (def nite) positive reviews (i.e., 4 & 5 star reviews) and (def nite) negative reviews (i.e., 1 & 2 star reviews). From each domain, we randomly sample 200 data instances 10 times. We state the results averaged over these different test sets. The class ratio on each test set corresponds to the distribution of def nite polar reviews, i.e., 3 star reviews are ignored.

The unlabeled training dataset is the dataset of a domain excluding the test data. As labeled training data for the embedded supervised classif er within self-training, we use 70% of data instances labeled by the rule-based classif er ranked by conf dence of prediction (across all domains/conf gurations, this size provided best results). Hopefully, most mixed reviews are among the remaining 30%.

[9] We can include this restriction in this section since we will not consider the semi-supervised learning algorithm SGT in this section.

Table 5. Performance of self-trained classifiers with different feature sets (experiments are carried out on a balanced corpus and results are averaged over all domains).

	RB_{Plain}			RB_{bWSD}			RB_{Neg}			RB_{Weight}		
Type	**F1+**	**F1−**	**Acc**	**F1+**	**F1−**	**Acc**	**F1+**	**F1−**	**Acc**	**F1+**	**F1−**	**Acc**
RB (*Baseline*)	69.81	52.36	63.12	70.39	61.79	66.65	72.42	67.40	70.14	74.26	68.30	71.60
SelfTr$_{Top2000}$	70.15	70.88	70.53	70.26	71.55	70.92	72.78	73.88	73.40	74.79	74.18	75.73
SelfTr$_{Adj600}$	68.94	69.92	69.44	70.08	71.41	70.76	72.46	73.90	73.20	74.34	75.82	75.10
SelfTr$_{MPQA}$	69.18	67.85	68.55	70.03	69.46	69.75	72.50	72.19	72.15	74.57	75.47	75.04
SelfTr$_{Uni}$	69.82	71.16	70.51	70.53	72.41	71.50	73.17	74.87	74.05	75.73	77.67	76.74
SelfTr$_{Uni+Bi}$	71.14	74.69	71.94	71.41	73.64	72.57	74.39	76.12	75.29	76.43	78.62	77.58

5.4.1 *Class Imbalance and Rule-based Classif cation*

In the f rst experiment, we just focus on class imbalance (i.e., 3 star reviews are excluded). We examine a self-trained classif er using the class-ratio estimate of a rule-based classif er as it is the most obvious estimate since the rule-based classif er is also used for generating the labeled training data. In particular, we want to explore whether there is a systematic relationship between the class distribution, the class-ratio estimate of the rule-based classif er and the resulting self-trained classif er. Table 6 lists the actual distribution of classes on the test set, the deviation between the distribution as it is predicted by the rule-based classif er and the actual distribution along the information towards which class the rule-based classif er is biased. Finally, we also list the absolute improvement/deterioration of the self-trained classif er in comparison to the rule-based classif er. We will only consider the best rule-based classif er, i.e., RB_{Weight}, and for self-training, we will exclusively consider the best conf guration from the previous experiments, i.e., SelfTr_{Uni+Bi}. The table shows that the quality of class-ratio estimates of rule-based classif ers varies among the different domains. The deviation is greatest on the *computer* domain. This is also the only domain in which the majority class are the negative reviews. With exception of the *sports* domain, the rule-based classif er always overestimates the amount of positive reviews. This overestimation is surprising considering that the polarity lexicon we use contains almost twice as many negative polar expressions as positive polar expressions. This f nding, however, is consistent with our observation from Section 5.2 that rule-based classif ers have a bias towards positive reviews, i.e., they achieve a better F-score for positive reviews than for negative reviews[10]. Table 6 also clearly shows that the deviation negatively correlates with the improvement of the self-trained classif er towards the rule-based classif er. The improvement is greatest on the *sports* domain where the deviation is smallest and the greatest deterioration is obtained on the *computer* domain where the deviation is largest.

In summary, the class distribution of the data has a signif cant impact on the f nal self-trained classif er. In case there is a heavy mismatch between actual and predicted class ratio, the self-training approach will not improve the rule-based classif er.

5.4.2 *Class Imbalance, Class Ratio Estimates and 3 Star Reviews*

In the following experiment we will compare how alternative class-ratio estimates relate to each other when applied to self-training. We compare the actual distribution (Ratio-Oracle) with the balanced class ratio (Ratio-Balanced), the class ratio as predicted by the rule-based classif er over the entire dataset (Ratio-RB) and estimates gained from a small amount of randomly sampled data instances from the dataset. We randomly sample 20 (Ratio-20), 50 (Ratio-50) and 100 (Ratio-100) instances. For each conf guration (i.e., 20, 50, and 100), we sample 10 times, run SelfTr for each sample and report the averaged result. We compare the self-trained classif er with a classif er always assigning a test instance to the majority class (Majority-Cl) and the rule-based classif er (RB_{Weight}). This time, we also include the 3 star reviews in the unlabeled dataset.

Table 7 displays the results. We also display results of the datasets without using 3 star reviews in brackets. SelfTr using Ratio-Balanced produces the worst results among the self-training classif ers. This was the only method used in previous work (in Chinese) [13, 15]. Apparently, English data are more diff cult than Chinese and, in English, SelfTr is more susceptible to deviating class-ratio estimates since in [13, 15] SelfTr with Ratio-Balanced scores rather well. Ratio-Oracle produces best results which comes to no surprise since the class distribution in training and test set is the same. On average, Ratio-100 produces the second best result as it also gives fairly reliable class-ratio estimates (the deviation is 3.3% on average, whereas the deviation of Ratio-Balanced is 18.16%). Both Ratio-50 and Ratio-100 produce results which are signif cantly better than Majority-Cl and RB_{Weight}.

As Ratio-Oracle, Ratio-Balanced, Ratio-20, Ratio-50, and Ratio-100 suggest, the presence of mixed polar reviews does not produce signif cantly different results. It is very striking, however, that the results of Ratio-RB are better using the 3 star reviews which seems counter-intuitive. We found that this is a corpus artifact. As already stated in Section 3, 3 star reviews do not only contain indef nite polar reviews but also positive and negative reviews. We also noted that Ratio-RB has a bias towards predicting too many positive instances. The bias is stronger if 3 star reviews are not included in the ratio-prediction (deviation of 8.5% instead of 6%). We, therefore, assume that among the 3 star reviews the proportion of negative-like reviews is greater than among the remaining part of the dataset and RB within SelfTr detects them as such. Thus, the bias towards positive polarity is slightly neutralized.

In summary, using small samples of labeled data instances is the most effective way for class ratio estimation enabling SelfTr to consistently outperform Majority-CL and RB_{Weight}. Mixed reviews only have a marginal impact on the f nal overall result of SelfTr.

6 Conclusion

In this paper, we examined the effectiveness of bootstrapping a supervised polarity classif er with the output of an open-domain rule-based classif er. The resulting self-trained classif er is usually signif cantly better than the open-domain classif er since the supervised classif er

[10] We also observed that this bias is signif cantly larger on simple classif ers, such as RB_{Plain}, which is plausible since on this classif er the gap between F-scores of positive and negative reviews is also largest (see Table 5).

Table 6. Class imbalance and its impact on self-training.

Domain	Class distribution ($+ : -$)	Deviation of predicted distribution from actual distribution	Class towards which predicted distribution is biased	Difference in Accuracy between RB and SelfTr(RB)
computer	43.17 : 56.83	16.30	$+$	-3.60
products	63.07 : 36.93	6.65	$+$	-0.25
sports	78.68 : 21.32	2.10	$-$	$+3.15$
travel	74.07 : 25.93	3.71	$+$	$+1.30$

Table 7. Accuracy of different classif ers tested on naturally imbalanced data: for self-trained classif ers the unlabeled data also contain 3 star reviews; numbers in brackets state the results on a dataset which excludes 3 star reviews.

			SelfTr					
Domain	**Majority-Cl**	**RB$_{Weight}$**	**Ratio-Oracle**	**Ratio-Balanced**	**Ratio-RB**	**Ratio-20**	**Ratio-50**	**Ratio-100**
computer	56.83	73.80	82.80 (83.35)	**83.25 (82.95)**	75.95 (70.20)	77.36 (77.95)	80.43 (80.91)	80.96 (81.47)
products	63.07	76.00	80.90 (81.70)	75.40 (76.05)	77.50 (75.75)	77.61 (78.10)	80.45 (80.86)	**80.69 (81.27)**
sports	78.68	77.35	81.25 (81.10)	62.55 (60.30)	**80.75 (80.50)**	79.10 (79.01)	79.94 (79.94)	80.62 (**80.50**)
travel	74.07	79.50	81.70 (81.60)	66.95 (66.10)	**81.15 (80.80)**	77.96 (76.59)	80.64 (80.52)	80.76 (80.58)
average	68.16	76.66	81.66 (81.94)	72.04 (71.35)	78.84 (76.81)	78.01 (77.91)	80.37 (80.56)	**80.76 (80.96)**

exploits in-domain features. As far as the choice of the feature set is concerned, the supervised classif er within self-training behaves very much like an ordinary supervised classif er. The set of all unigrams and bigrams performs best.

The type of rule-based classif er has an impact on the performance of the f nal classif er. Usually, the more accurate the rule-based classif er is, the better the resulting self-trained classif er is. Therefore, modeling open-domain constructions relevant for polarity classif cation is important for this type of self-training. It also suggests that further improvement of rule-based polarity classif ers by more advanced linguistic modeling is likely to improve self-training as well.

In cases in which semi-supervised learning outperforms supervised learning, self-training at least also performs as well as the best semi-supervised classif er. A great advantage of self-training is that it chooses instances to be added to the labeled training set by using conf dence scores whereas in semi-supervised learning one has to resort to random sampling. The resulting data from self-training are usually much better.

Self-training also outperforms a rule-based classif er and a majority-class classif er in more diff cult settings in which mixed reviews are part of the dataset and the class distribution is imbalanced, provided that the class-ratio estimate does not deviate too much from the actual ratio on the test set. A class-ratio estimate can be obtained by the output of the rule-based classif er but, on average, using small samples from the data collection produces more reliable results.

Since this self-training method works under realistic settings, it is more robust than semi-supervised learning, and its embedded supervised classif er only requires simple features in order to produce reasonable results, it can be considered an effective method to overcome the need for many labeled in-domain training data.

Acknowledgements

Michael Wiegand was funded by the German research council DFG through the International Research Training Group “IRTG” between Saarland University and University of Edinburgh and the BMBF project NL-Search under contract number 01IS08020B. The authors would like to thank Alexandra Balahur for insightful comments.

REFERENCES

[1] A. Andreevskaia and S. Bergler, ‘When Specialists and Generalists Work Together: Overcoming Domain Dependence in Sentiment Tagging’, in *Proc. of ACL/HLT*, (2008).
[2] A. Aue and M. Gamon, ‘Customizing Sentiment Classif ers to New Domains: a Case Study’, in *Proc. of RANLP*, (2005).
[3] 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).
[4] S. Dasgupta and V. Ng, ‘Mine the Easy, Classify the Hard: A Semi-Supervised Approach to Automatic Sentiment Classif cation’, in *Proc. of ACL/IJCNLP*, (2009).
[5] T. Joachims, ‘Making Large-Scale SVM Learning Practical’, in *Advances in Kernel Methods - Support Vector Learning*, (1999).
[6] T. Joachims, ‘Transductive Learning via Spectral Graph Partitioning’, in *Proc. of ICML*, (2003).
[7] A. Kennedy and D. Inkpen, ‘Sentiment Classif cation of Movie Reviews Using Contextual Valence Shifters’, in *Computational Intelligence (Special Issue)*, volume 22, (2006).
[8] M. Klenner, S. Petrakis, and A. Fahrni, ‘Robust Compositional Polarity Classif cation’, in *Proc. of RANLP*, (2009).
[9] K. Moilanen and S. Pulman, ‘Sentiment Construction’, in *Proc. of RANLP*, (2007).
[10] V. Ng, S. Dasgupta, and S. M. N. Arif n, ‘Examining the Role of Linguistic Knowledge Sources in the Automatic Identif cation and Classif cation of Reviews’, in *Proc. of COLING/ACL*, (2006).
[11] B. Pang, L. Lee, and S. Vaithyanathan, ‘Thumbs up? Sentiment Classif cation Using Machine Learning Techniques’, in *Proc. of EMNLP*, (2002).
[12] L. Polanyi and A. Zaenen, ‘Context Valence Shifters’, in *Proc. of the AAAI Spring Symposium on Exploring Attitude and Affect in Text*, (2004).
[13] L. Qiu, W. Zhang, C. Hu, and K. Zhao, ‘SELC: A Self-Supervised Model for Sentiment Classif cation’, in *Proc. of CIKM*, (2009).
[14] D. Rao and D. Ravichandran, ‘Semi-Supervised Polarity Lexicon Induction’, in *Proc. of EACL*, (2009).
[15] S. Tan, Y. Wang, and X. Cheng, ‘Combining Learn-based and Lexicon-based Techniques for Sentiment Detection with Using Labeled Examples’, in *Proc. of SIGIR*, (2008).
[16] J. Wiebe and E. Riloff, ‘Creating Subjective and Objective Sentence Classif ers from Unannotated Texts’, in *Proc. of CICLing*, (2005).
[17] M. Wiegand and D. Klakow, ‘Predictive Features in Semi-Supervised Learning for Polarity Classif cation and the Role of Adjectives’, in *Proc. of NoDaLiDa*, (2009).
[18] T. Wilson, J. Wiebe, and P. Hoffmann, ‘Recognizing Contextual Polarity in Phrase-level Sentiment Analysis’, in *Proc. of HLT/EMNLP*, (2005).