Originally published in: Language Resources and Evaluation
vol. 47 (2013) no. 4, pp. 1049-1088.
DOI: *https://doi.org/10.1007/s10579-013-9218-3*

Bootstrapping polarity classifiers with rule-based classification

Michael Wiegand · Manfred Klenner · Dietrich Klakow

Abstract In this article, we examine the effectiveness of bootstrapping supervised machine-learning polarity classifiers with the help of a domain-independent rule-based classifier that relies on a lexical resource, i.e., a polarity lexicon and a set of linguistic rules. The benefit of this method is that though no labeled training data are required, it allows a classifier to capture in-domain knowledge by training a supervised classifier with in-domain features, such as bag of words, on instances labeled by a rule-based classifier. Thus, this approach can be considered as a simple and effective method for domain adaptation. Among the list of components of this approach, we investigate how important the quality of the rule-based classifier is and what features are useful for the supervised classifier. In particular, the former addresses the issue in how far linguistic modeling is relevant for this task. We not only examine how this method performs under more difficult settings in which classes are not balanced and mixed reviews are included in the data set but also compare how this linguistically-driven method relates to state-of-the-art statistical domain adaptation.

Keywords Polarity classification · Sentiment analysis · Bootstrapping methods · Feature engineering · Text classification

M. Wiegand (✉) · D. Klakow
Spoken Language Systems, Saarland University, Building C7.1, 66123 Saarbrücken, Germany
e-mail: Michael.Wiegand@lsv.uni-saarland.de

M. Klenner
Institute of Computational Linguistics, Zürich University, Binzmühlestrasse 14, 8050 Zürich, Switzerland

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 classification which is the task of distinguishing between positive utterances (1) and negative utterances (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 classification approaches, in particular classifiers using bag of words, are heavily domain dependent (Aue and Gamon 2005), that is, they usually generalize poorly across different domains. This is mostly due to the fact that the words employed to convey polarity can vary from domain to domain. One solution to this problem would be to provide labeled training data for every possible domain. However, this is impractical as the costs for that endeavor are prohibitively expensive.

Semi-supervised learning tries to solve the problem of domain dependence by reducing the size of the labeled data set of the target domain or using labeled out-of-domain data. The lack of sufficient labeled in-domain training data is compensated by a large unlabeled data set of that domain. The latter is much cheaper to obtain.

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

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

Consider, for example, the two negative sentences (3) and (4) from the movie and the computer domain. With the knowledge that *poorly* is a domain-independent polar expression, one could label these sentences as negative opinions. Considering these sentences as labeled training data and applying supervised learning, a classifier may learn that *predictable* and *defective* are polar expressions of these particular domains.

(3) Stop giving us these poorly^{-} written thrillers with plots as predictable^{-} as the sunset.
(4) The system was deemed defective^{-} and poorly^{-} designed.

Although this kind of self-training has already been applied to tasks in opinion mining (Wiebe and Riloff 2005), including polarity classification (Tan et al. 2008; Qiu et al. 2009), there are certain aspects of this method that have not yet been fully examined:

Firstly, what is the impact of the robustness of the rule-based classifier on the final classifier, that is, does the supervised classifier improve when the rule-based classifier improves? This addresses the issue to what extent the analysis of linguistic phenomena that are relevant for polarity classification and can be incorporated into a rule-based classifier, such as word disambiguation, negation modeling, modality, or intensification, is important for this kind of self-training approach. In this article, we take a much more detailed look at the optimization and effectiveness of individual features than in previous work.

Secondly, how can a good labeled training set for self-training be acquired with the help of the rule-based classifier? A contribution of this article is that we compare different data selection criteria with regard to this bootstrapping method.

Thirdly, what are typical features that can be learnt with this bootstrapped approach that are not contained within rule-based classifier? For the first time, we provide some detailed illustration of what features are learnt.

Fourthly, how does this bootstrapping approach compare to compositional rule-based classification? Are there differences in effectiveness between these two approaches with regard to the levels of granularity that are considered (i.e., document and sentence level)?

Fifthly, how does this type of self-training, in which a model is mainly bootstrapped with the help of linguistic information, compare to state-of-the-art statistical domain adaptation methods using out-of-domain labeled training data and hardly any linguistic knowledge?

Finally, does this method work in realistic settings in which—in addition to definite polar reviews—also mixed polar reviews are part of the data set and the distribution of the classes is imbalanced?

The remainder of this article is structured as follows. Section 2 describes the data we use. Section 3 describes in detail the set of rule-based polarity classifiers we consider for self-training along the performance they achieve on our given data sets. In Sect. 4, we present the different configurations for self-training and evaluate them. Section 5 compares the standard rule-based classification from Sect. 3 with compositional rule-based polarity classification. These classifiers are evaluated on document-level data and sentence-level data also taking self-training into account. Section 6 compares self-training with statistical domain adaptation, while Sect. 7 discusses the impact of natural class distribution and mixed reviews on self-training. In Sect. 8, we discuss related work, and we conclude in Sect. 9.

2 Data

In this article, we carry out most experiments on a multi-domain data set that consists of *IMDb* movie reviews (Pang et al. 2002) and reviews extracted from *Rate-It-All*[1] covering the domains *Computer, Products, Sports*, and *Travel*. We evaluate on the IMDb movie reviews because they are considered benchmark data for polarity classification. The additional data are used to show that our findings are valid throughout different domains. Moreover, they have also been used in previous work on polarity classification (Wiegand and Klakow 2009a, 2010).

Table 1 lists the properties of the corpora from the different domains. It lists the individual class distributions, the size of the vocabulary, and the average number of sentences per document. The vocabulary is computed on stemmed word forms (as stemmed word forms will be the basis for text processing).[2] The table shows that with regard to these dimensions the domains differ among each other. With regard to the average number of sentences per document there is a consistent difference between the *Rate-It-All* corpora and the *IMDb* corpus. The documents of the *Movies* domain are much longer.

On all data sets, the labels are automatically derived from the ratings. 1 and 2 star reviews are labeled as *negative* and 4 and 5 star reviews as *positive*. Only the *Rate-it-All* data sets include 3 star reviews. They are labeled as *mixed* reviews. 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 definite verdict in favor or against it is not possible. That is also why we cannot assign these instances to either *positive* and *negative*. During a manual inspection of some randomly chosen instances, however, we also found definite positive and negative reviews among 3 star reviews. For this work, we leave these instances in the category of mixed reviews.

3 Rule-based polarity classification

In this section, we describe how a rule-based polarity classifier can be designed with the help of a polarity lexicon. A polarity lexicon comprises a list of polar expressions, that is, words containing a prior polarity, such as *great* and *awful*, along their respective polarity type (i.e., *positive* or *negative*). We use the Subjectivity Lexicon from (Wilson et al. 2009) containing 2,718 positive and 4,910 negative entries.

[1] http://www.rateitall.com.

[2] Stemming may also negatively affect polar expressions (i.e., words containing a prior polarity, such as *great* and *awful*) by conflating expressions with different polarity types to the same stem, such as *hopeful* and *hopeless* to *hope*$. To estimate the impact of that problem, we stemmed the entries of the polarity lexicon we use in this work (i.e., a list of polar expressions along their respective polarity type) and counted the cases of those erroneously conflated expressions. Less than 1 % of the entries were affected; most critical suffixes, such as *-less*, were preserved by our stemmer (Porter 1980). On average, we measured only some slight improvement by using stemming (<1 % point).

Table 1 Properties of the different domain corpora

Domain	Source	Positive (4 and 5 stars)	Mixed (3 stars)	Negative (1 and 2 stars)	Vocabulary	Average no. of sentences per document
Computer	*Rate-It-All*	952	428	1,253	11,319	6.58
Products	*Rate-It-All*	2,292	554	1,342	16,615	6.63
Sports	*Rate-It-All*	4,975	725	1,348	19,096	4.38
Travel	*Rate-It-All*	9,397	1,772	3,289	29,685	4.65
Movies	*IMDb*	1,000	0	1,000	37,374	32.36

A rule-based polarity classifier assigns scores to the polar expressions (it identifies by using the polarity lexicon) in a test document. Positive polar expressions are assigned the positive score +1, while negative polar expressions are assigned the negative score −1. In order to classify a data instance, that is, in our case a test document, the scores assigned to the individual polar expressions are summed. If the sum is positive, then the instance is classified as positive. It is classified as negative, if the sum is negative. We assign to all cases in which the sum is 0 the polarity type that gives best performance on that individual data set (which is usually negative polarity). The polarity sum is 0 if the amount of detected positive information equals the amount of negative information or, in the rare case, when not a single polar expression could be identified within the document. By including this default label for instances with a score of 0, we have a stronger baseline that is to be beaten by self-training.

For the following experiments—with the exception of those presented in Sect. 7—we use a balanced subset (randomly generated) for each domain. The *Rate-It-All* data set consists of 1,800 data instances per domain, whereas the *IMDb* data set consists of 2,000 data instances. We just consider (definite) positive and (definite) negative reviews. All words are normalized by applying *Porter stemming* (Porter 1980).

3.1 Different versions of classifiers

We define four different types of rule-based classifiers. They differ in complexity. The simplest classifier, that is, RB_{Plain}, is basically the algorithm described above. 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 classifier, RB_{Weight}, is like RB_{Neg} but it also employs some heuristic weighting. Table 2

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				✔

summarizes the different classifiers with their respective properties. In the following subsections, we will describe in detail each of these different properties.

3.2 Basic word sense disambiguation with part-of-speech tags

There are several ambiguous words that only contain a polar meaning in some of their senses. For some of these words the sense can be determined depending on the part of speech of the word in its particular context. For example, the word *novel* has a meaning similar to *new* or *original* if it is an adjective (5) and refers to a particular type of prose when it is a noun (6). Only the adjective should be considered a polar expression.

(5) L.A. County officials offer a $\underline{\textit{novel}}_{Adj}$ idea to save millions.
(6) Papillon is a 1973 film based on a $\textit{novel}_{Noun}$ by French ex-convict Henri Charrière.

In a similar fashion, one can establish a rule for the word *plot*. It contains a polar meaning as a verb when it describes the act of secretly, most often illegally, planning something (7), while the noun may refer to a story told in a play, novel, or film (8).

(7) They $\underline{\textit{plot}}_{Verb}$ to instigate unrest by sending messages via the Internet, telephones, and mobile phones.
(8) The $\textit{plot}_{Noun}$ of the novel is based upon a true story.

Unfortunately, these rules are in many cases a simplification. For instance, the word *plot* has actually several senses even with a specific part of speech. The noun (when it is a deverbal noun from *to plot*) may also refer to the act of secretly planning something. However, automatic methods to distinguish such senses—in spite of the fact that they are highly relevant to sentiment analysis (Wiebe and Mihalcea 2006)—are still in their infancy (Balamurali et al. 2011) as the necessary labeled resources are extremely sparse and difficult to produce (Akkaya et al. 2009, 2011). Consequently, this type of disambiguation is beyond the scope of this work and we limit our disambiguation to the one based on part-of-speech information. We obtain these disambiguation rules from our polarity lexicon (Wilson et al. 2009). For part-of-speech tagging we use the *C&C* tagger.[3]

3.3 Negation modeling

Negation is one of the most prominent contextual phenomena that affects polarity. Even though there exists a plethora of different approaches to take this into account, it is fairly difficult to judge their general impact as the methods are often evaluated in different contexts (Wiegand et al. 2010). We, therefore, only address the issues that are most frequently dealt with.

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

3.3.1 Plain negation

The most commonly accepted type of negation modeling is the following: If a polar expression, such as *nice*, occurs within the scope of a negation, its polarity is reversed (9).

(9) Overall, [*not* a nice^{+}]$^{-}$ place to take the family!

The views differ, however, as to what should be considered a negation and how its scope should be determined.

3.3.2 Scope of negation

Before discussing in more detail which lexical units we consider as negation expressions, we briefly describe our scope modeling. We take a simple approach that considers a polar expression to be negated if it follows a negation marker within a window of *n* words. Figure 1 shows the performance of this negation model using different window sizes. The plot shows the averaged results over all domains. We use the negation markers from (Wilson et al. 2009). The figure shows that negation modeling is important (as the window size $n = 0$ performs worst). A maximum is reached at $n = 5$, however some larger windows only marginally degrade performance. This observation is consistent with other approaches that consider the scope as anything following a negation until the next punctuation mark (Pang et al. 2002). We will use this optimized window size (of five words) as a scope in the following experiments, i.e., the scope of a negation expression are the five words following the mention of that expression.

In sentiment analysis, one often resorts to very shallow notions of scope (e.g., on the basis of window size) (Pang et al. 2002; Wilson et al. 2009). There are only few works which establish the scope of negation on the basis of syntactic rules (Jia et al. 2009; Council et al. 2010) while in other areas, such as the biomedical domain, this is much more common, e.g., (Huang and Lowe 2007; Morante 2010). We will examine such a potential of syntactic information in Sect. 5 when we discuss compositional polarity classification but we are aware that the parsing quality is severely affected by the heavy noise in our user generated data (e.g., misspellings, missing punctuation etc.).

3.3.3 Polarity shifters

In addition to common negation expressions, such as *not*, there are also other lexical units that may similarly express negation. These expressions are commonly referred to as *polarity shifters*. (10) and (11) differ only in the type of negation marker that is used. While the former uses the common negation word *no*, the latter employs the polarity shifter *little*. These two sentences show that polarity shifters convey a weaker degree of negation than common negation markers.

(10) I have [*no* faith^{+}]$^{-}$ in that country.
(11) I have [*little* faith^{+}]$^{-}$ in that country.

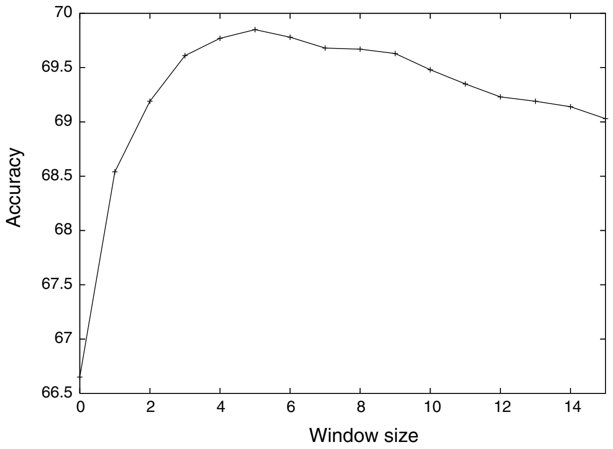

Fig. 1 Optimizing window size for negation

Moreover, there are several shifters that only reverse a particular polarity type. For example, the shifter *lack* only modifies positive polar expressions (12), while the shifter *abate* only modifies negative polar expressions (13).

(12) The movie *The Edge* has intelligence and smart characters, but the [*lack* of originality$^+$]$^-$ is its downfall.
(13) Financial support may [*abate* these problems$^-$]$^+$.

In this article, we treat polarity shifters in the same manner as negation markers, that is, we strongly assume that for a document-level polarity classification (10) and (11) should be treated as synonymous. We consider the slight semantic differences as irrelevant to the (coarse) binary classification we want to carry out. The main advantage of taking polarity shifters into account thus should lie in the increased coverage of negation detection. Table 3 compares a rule-based classifier with plain negation modeling just using conventional negation markers ($\mathrm{RB}_{NegPlain}$) and a classifier also incorporating polarity shifters ($\mathrm{RB}_{NegShifter}$). The list of polarity shifters is taken from (Wilson et al. 2009).[4] The table shows that there is only a marginal impact of incorporating polarity shifters (in some domains the inclusion is even slightly detrimental). Since the incorporation does not harm the overall performance, however, we include polarity shifters in our subsequent experiments.

3.3.4 Disambiguation of negation

Some negation markers are ambiguous and do not express negations when appearing in certain phrases, such as *not* in (14) that is part of the phrase *not only … but also*. We ran experiments with the disambiguation rules from (Wilson et al. 2009)[5] but could not find any improvement for polarity classification. A closer

[4] By polarity shifters, we refer to all entries marked as *genshifter, shiftneg*, or *shiftpos* from that lexicon.

[5] Those rules are encoded by entries marked as *notshifter*.

Table 3 Accuracies of rule-based classifier with plain negation modeling ($\mathrm{RB}_{NegPlain}$) and negation modeling with polarity shifters ($\mathrm{RB}_{NegShifter}$)

Domain	$\mathrm{RB}_{NegPlain}$	$\mathrm{RB}_{NegShifter}$
Computer	73.33	73.56
Products	70.50	71.06
Sports	67.61	67.50
Travel	70.83	70.72
Movies	67.00	67.85
Average	69.85	70.14

inspection of occurrences of those ambiguous markers in our data set revealed that the sentences in which they appear within such phrases usually enumerate either several positive or negative items. It is, therefore, usually irrelevant for document-level polarity classification to carry out this kind of disambiguation as the misinterpretation of one polar phrase will not affect the overall result since the overall amount of polar expressions will still be correctly interpreted. In (14), for example, without the disambiguation of negation markers we would erroneously identify one negative polar expression, that is, *not … fascinating*.[6] But given that we correctly identify the other remaining positive polar expressions *greatest, treasure, charm*, and *beauty*, this single misclassification will not affect the overall result.

(14) Spain is *not only* one of Europe's most fascinating countries, *but* is *also* home to some of the world's greatest treasures of history, culture, charm, and beauty.

3.4 Heuristic weighting

So far, all polar expressions contained in the polarity lexicon are assigned the same absolute weight, that is, +1 for positive polar expressions and −1 for negative polar expressions, respectively. This does not reflect 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. In the following, we will describe particular (intrinsic or contextual) properties of polar expressions and suggest a (very simple) ad-hoc weight that should reflect that particular property. As we do not have any development data and our classifier should be domain-independent, we chose a very coarse-grained weighting scheme.

3.4.1 Strength of polar expressions (StrongPol)

The polarity lexicon we use (Wilson et al. 2009) includes a binary feature expressing the strength of a polar expression. It distinguishes between *strong* and *weak* polar expressions. Strong polarity in this context does not primarily refer to a high prior polar intensity but the tendency to appear as a polar expression in most contexts. An example for a typical strong polar expression is *hate*. Weak polar

[6] This classification of course requires a correct identification of the scope of the negation.

expressions, such as *dream*, on the other hand, are more ambiguous. They, too, may appear in polar contexts (15) but the likelihood to occur in contexts in which they do not contain a polar meaning, such as (16), is much higher than for strong polar expressions.

(15) Not only is it a thing of beauty, but it runs like a *dream*!
(16) No suspense occurs in the *dream* sequences either.

Intuitively, strong polar expressions should obtain a higher weight than weak polar expressions. That is why we assign them the weight of 2.

3.4.2 Intensifiers (Intens) and detensifiers (Detens)

When a polar expression is modified by a so-called *intensifier*, such as *definitely* or *extremely*, its polar intensity is increased (17). On the other hand, if a polar expression is modified by a so-called *detensifier* or *diminisher*, such as *kind of* or *slightly*, its polar intensity is decreased (18).

(17) She *definitely* deserved her gold.
(18) It's *kind of* expensive, but well worth the investment.

For our experiments, we use the intensifiers from (Wilson et al. 2009) and the list of detensifiers from (Jason 1988). We propose to double the polarity score of intensified polarity expressions and to halve the score of detensified polar expressions, respectively.

For the detection of scope, we use the same method (i.e., word-based window size) we applied to negation modeling (see Sect. 3.3.2). We also use the same window size.

3.4.3 Polar adjectives (PolAdj)

The part of speech of a polar expression may also shed some 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 (Hatzivassiloglou and McKeown 1997; Hatzivassiloglou and Wiebe 2000; Pang et al. 2002; Wiegand and Klakow 2009a). Therefore, polar adjectives should obtain a larger weight than polar expressions with other parts of speech. That is why we assign them the weight of 2.

3.4.4 Modal embedding (Modal)

If a proposition is embedded in an epistemic modal context, that is, a context in which the speaker expresses some certainty about the factuality of the proposition, the proposition itself cannot be considered factual (19).

(19) While this *may* sound reasonable, it isn't.

We identify those contexts by the occurrence of a modal verb. Unlike the previous linguistic phenomena, it does not make sense to just decrease the weight of a polar expression that occurs within the scope of such a verb. Instead, we totally discard its value, that is, we set the score to 0. This feature is a simplication of the *modal operators* proposed in (Neviarouskaya et al. 2009). In that work, each modal verb was assigned an individual score rather assigning all modal verbs the same score. We make use of a more coarse-grained feature design as we considered an out-of-context annotation of individual modal verbs too difficult.

For the detection of scope, we again use the same method (i.e., word-based window size) we applied to negation modeling (see Sect. 3.3.2). Again, we also use the same window size.

3.4.5 The importance of the last sentence (LastSent)

Usually, judgmental texts, such as reviews, end with a conclusion summarising the author's point of view. There is even psycholinguistic and psychophysical evidence for the special significance of that sentence with respect to polarity classification (Becker and Aharonson 2010). A polarity classifier should therefore take this into account and give special emphasis to polar expressions occuring in that sentence. That is why we assign them the weight of 2.

3.4.6 Comparison of different features

Table 4 compares the performance of the individual features on rule-based polarity classification using the ad-hoc weights that we have previously suggested. For polar expressions for which several properties apply, we *multiply* the corresponding weights. For instance, an intensified adjective is assigned the value of 2·2 since both the feature Intens and the feature PolAdj fire. As the differences between the resulting accuracies produced by the different feature sets are often marginal, we display the tendencies of those features, rather than the actual accuracies of the different classifiers. Thus, we hope to improve legibility. Increases and decreases (in terms of accuracy) as compared to a classifier without heuristic weighting (i.e., the baseline RB_{Neg}) are indicated by + or −, respectively. + + or − − indicates the change is significant (chi-square test) at the $p < 0.1$ level, whereas + + + or − − − indicates the significance at the $p < 0.05$ level. Finally, ○ indicates no change.

The table shows that PolAdj is the best feature to use. On the basis of the union of all domain corpora, the improvement over the baseline is even statistically significant. The second best feature is Modal which also makes a positive contribution on all domains except one. StrongPol and Intens only have a positive effect on some domains. The low impact of Intens and Detens suggest that for polarity classification the polar intensity is less relevant. That is, for the classifier, it primarily matters whether some polar expression is either positive or negative. LastSent has only a positive impact on the *Movies* domain. As this data set originates from another Web site than the other data sets, the average document

Table 4 Comparison of different features employed for heuristic weighting

Domain	PolAdj	Modal	StrongPol	Intens	Detens	LastSent	Combination
Weight	2.0	0.0	2.0	2.0	0.5	2.0	*N/A*
Computer	+	+	−	+	−	−	+
Products	+	+	−	−	○	−	+
Sports	+	○	+	−	○	−	+
Travel	+	+	+	−	−	−	++
Movies	+	+	+	+	−	+	+++
All	+++	+	+	○	−	−−	+++

Increases and decreases (in terms of accuracy) as compared to a classifier without heuristic weighting (i.e., the baseline RB$_{Neg}$) are indicated by + or −, respectively; ++ or − − indicates the change is significant (chi-square test) at the $p < 0.1$ level; +++ or − − − indicates the significance at the $p < 0.05$ level; ○ indicates no change

size[7] between that domain and the remaining ones hugely differs, that is, 32.36 sentences compared to 5.56 sentences (see also Table 1 in Sect. 2). We assume that a discourse feature, such as LastSent, only makes sense for large documents, as they are more likely to follow a certain discourse structure that finishes with a summary or conclusion.

Finally, we also assess the contribution of a combination of those features. For that, we chose all features that have a positive impact on at least two domains, that is, PolAdj, Modal, StrongPol, and Intens. This is also the configuration we use in the subsequent experiments for RB$_{Weight}$. For all domains, we observe some improvement over the baseline. On the *Travel* and *Movies* domain, the combination even reaches weak significance. This shows that the addition of other features to the best individual feature, that is, PolAdj (which as such does not reach a significant improvement over the baseline) is effective. In absolute numbers (i.e., accuracy), the performance on the *Computer* and *Products* domain actually also improves. Unfortunately, this cannot be captured by the notation we chose for presenting this comparison.

3.5 Comparison of different rule-based classifiers

Figure 2 summarizes all steps of the most complex rule-based classifier. For the less complex classifiers, certain steps within that program are skipped.

Table 5 shows the results of the different rule-based classifiers across the different domains. On average, the more complex the rule-based classifier becomes, the better it performs. The only notable exception is the *Sports* domain (from RB$_{Plain}$ to RB$_{bWSD}$). By visual inspection, we noticed some heavy noise on that particular data set, that is, a large number of words are misspelt. This may have severely affected text processing, especially part-of-speech tagging, which is vital for RB$_{bWSD}$. On several domains, the improvement from one classifier to the next more complex classifier is significant. On average (i.e., considering the union of all domain data sets), the improvements are always significant.

[7] We measure this by the average number of sentences within a document.

1. Lexicon loading, i.e., polar expressions, negation words, and intensifiers.
2. Preprocessing:
   - (i) Stem words within test instance.
   - (ii) Apply part-of-speech tagging to test instance.
3. Polar expression marking:
   - (i) Identify potential polar expressions (with polarity lexicon).
   - (ii) Discard expressions whose part-of-speech tag does not match with that stated within the polarity lexicon (*basic word sense disambiguation*).
4. Negation modeling:
   - (i) Identify potential negation words (including polarity shifters).
   - (ii) Reverse polarity of polar expression in scope of negation.
5. Heuristic weighting: assign special weight in case polar expression is:
   - (i) a polar adjective (weight: 2.0)
   - (ii) a strong polar expression (weight: 2.0)
   - (iii) an intensified polar expression (weight: 2.0)
   - (iv) a polar expression within the scope of a modal (weight: 0.0).
6. Classification: assign the polarity type to test instance with the largest sum of scores.

Fig. 2 Algorithm of the rule-based polarity classifier (most complex classifier: RB_{Weight})

4 Self-training a polarity classifier using the output of a rule-based classifier

The idea of this bootstrapping method is that a domain-independent rule-based classifier is used to label an unlabeled data set. Unlike in semi-supervised learning, no labeled training data are used. The only available knowledge is encoded in the rule-based classifier. The data instances labeled by the rule-based classifier serve as labeled training data for a supervised machine-learning classifier. Usually, only instances that have been assigned a label with a high confidence are used. (We will show below how we translate *confidence* to our task.) Ideally, the resulting supervised classifier is more robust on the domain on which it was trained than the rule-based classifier. The improvement can be explained by the fact that the rule-based classifier only comprises domain-independent knowledge. The supervised classifier, however, makes use of domain-specific features, that is, words such as *crunchy*$^{+}$ (food domain) or *buggy*$^{-}$ (computer domain), that are not part of the rule-based classifier. It may also learn to correct polar expressions that are specified in the polarity lexicon but have a wrong polarity 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 films, 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 that 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. Figure 3 illustrates the self-training method that we are going to examine in this article.

Table 5 Comparison of different rule-based classifiers (RB) (evaluation measure: accuracy)

Domain	RB_{Plain}	RB_{bWSD}	RB_{Neg}	RB_{Weight}
Computer	64.11	70.61*	73.56*	**75.11**
Products	60.78	66.06*	71.06*	**71.72**
Sports	64.33	64.39	67.50	**69.17**
Travel	64.61	67.39	70.72*	**73.56**
Movies	61.75	64.80*	67.85*	**72.10***
Average	63.12	66.65*	70.14*	**72.33***

* Significantly better than *all* less complex rule-based classifiers on the basis of a chi-square test using $p < 0.05$; for *Average* the significance is tested on the union of all domain data sets

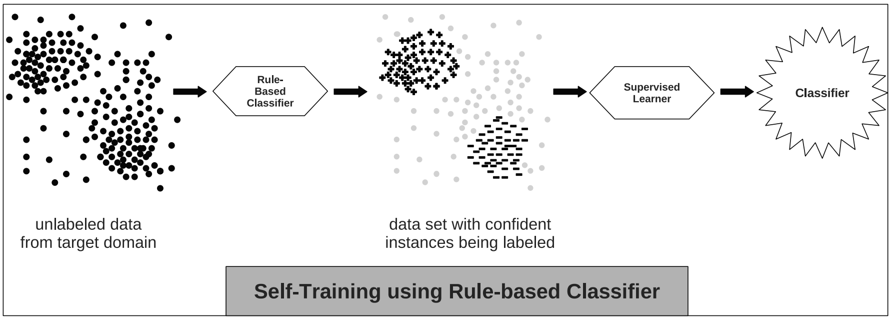
Rule-Based Classifier
Supervised Learner
Classifier
unlabeled data from target domain
data set with confident instances being labeled
Self-Training using Rule-based Classifier

Fig. 3 Illustration of self-training using a rule-based classifier for bootstrapping

4.1 Feature sets

Table 6 lists the different feature sets we examine for the supervised classifier (within self-training). We list the feature sets along their abbreviation with which they will henceforth be addressed. We removed the stopwords for the frequently occurring unigrams in Top2000 using the list by the *Glasgow Information Retrieval Group*.[8] The features can be divided into two groups. Top2000, Adj600, and MPQA have been found effective for semi-supervised learning (Wiegand and Klakow 2009a). These feature sets contain less noise than the overall vocabulary of a domain corpus. In particular, Adj600 or MPQA contain highly relevant features (i.e., many polar expressions) and very few or, in case of MPQA, even no irrelevant features. Uni and Uni+Bi, on the other hand, contain those features that have been found effective for supervised learning (Ng et al. 2006). 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 potentially 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.

[8] http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words.

Table 6 Description of the different feature sets

Feature set	Abbreviation
The 2,000 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

The usage of MPQA, that is, the feature set just comprising the polar expressions from the polarity lexicon that are also used for the rule-based classifier, may seem contradictory at first sight. One motivation for self-training is that the supervised classifier should be trained with a different (and hopefully more expressive) feature set than the feature set that is used for the rule-based classifier. Admittedly, the feature set derived from the polarity lexicon cannot accomplish this. However, by keeping the feature set between rule-based classifier and supervised learner for self-training fixed, we can examine the impact of the domain-specific weighting. Recall from the description of the rule-based polarity classifier in Sect. 3 that one major downside of this classifier is that it does not distinguish between the different polar expressions. We, partially, try to rectify this by applying some heuristic weighting (Sect. 3.4) but this weighting scheme is still extremely coarse in comparison to the weighting that can potentially be achieved by supervised learning. As the supervised learner is only given the lexical units representing polar expressions but not the corresponding polarity types (these have to be inferred to from the data in the form of feature weights), the learner may also find the correct orientation for polar expressions whose type is incorrect in the polarity lexicon according to the domain that is to be classified (see also the remarks concerning the ambiguous polar expression *cheap* above).

4.2 Evaluation of self-training

As in Sect. 3, the rule-based classifiers and the self-trained classifiers (bootstrapped with the help of rule-based classification) are evaluated on the entire (balanced) domain data set. For the supervised classifier, we chose Support Vector Machines (SVMs) as they are considered one of the most robust state-of-the-art learning algorithm (Joachims 1999). As a toolkit, we use *SVMLight*[9] with its standard configuration (i.e., linear kernel). Feature vectors are normalized to unit length and additionally weighted with *tf-idf* scores. All words are stemmed.

4.2.1 Optimizing the size of pseudo-labeled data

One important parameter of our self-training framework is the size of labeled training data. We assume that it is more effective to use only those labeled data instances for supervised learning that have been predicted with a high confidence

[9] http://svmlight.joachims.org.

Table 7 Accuracy of different amounts of pseudo-labeled data

Domain	250 per class	500 per class	750 per class	Entire data set
Computer	61.94 (79.44)	**64.17 (80.22)**	64.00 (75.67)	63.50 (72.67)
Products	55.39 **(71.83)**	59.44 (70.78)	61.06 (69.22)	**61.56** (66.28)
Sports	65.94 **(66.66)**	**65.72** (66.44)	**65.72** (65.56)	64.94 (63.89)
Travel	65.39 (66.06)	**68.44 (69.56)**	67.39 (67.28)	66.72 (66.00)
Movies	67.70 (72.15)	**70.70 (72.70)**	69.75 (70.35)	66.65 (66.70)
Average	63.27 (71.23)	**65.69 (71.94)**	65.58 (69.62)	64.67 (67.11)

Numbers in brackets denote performance with best normalization

score rather than considering the entire data set. In order to substantiate this claim, we examine different amounts of ranked documents for our labeled training set. The ranking is derived from the contextual scores of the rule-based classifier (see Sect. 3). For positive instances we consider the *n* instances with the highest scores and for negative instances we take the *n* instances with the lowest scores, respectively. By including only highly ranked instances, documents with a score of (or close to) 0 should be excluded. Recall from Sect. 3 that these documents contain either an equal amount of positive and negative content or no polar content (according to the rule-based classifier).

Table 7 compares the performance of self-training using 250, 500, 750, and all labeled documents. For this experiment, we took the simplest rule-based classifier (i.e., RB_{Plain}) and a completely unrestricted feature set (i.e., Uni+Bi). The numbers in brackets denote the performance of the best normalization (see also Sect. 4.2.2). Overall, 500 documents per class provide best performance. This is true for both the unnormalized and normalized scores. This is why we will use this configuration in our subsequent experiments using this data set.

Using normalized scores does not only result in a systematic improvement of performance but it also occasionally means that fewer labeled data are required (this is another side-effect of an improved ranking). This is most obvious in the *Products* domain.

4.2.2 The impact of normalization

The output of the rule-based classifier as described in Sect. 3 is the plain sum of contextual scores of the polar expressions. The previous section established that for finding highly ranked data instances some kind of normalization is useful. In this section, we examine different kinds of normalization. As Table 8 shows, we compare the plain score without normalization (NoNorm) with a score that normalizes by the overall number of polar expressions detected in a particular document (NormByPol), a score that normalizes by the number of words in the document (NormByWord), and a score that normalizes by the number of sentences (NormBySent).

Table 9 compares the performance of self-training using the different normalization methods. As in the previous section, we evaluate the simplest rule-based

Table 8 Description of the different normalization methods

NoNorm	No normalization (i.e., just contextual polarity score)
NormByPol	Contextual polarity score divided by the number of polar expressions in document
NormByWord	Contextual polarity score divided by the number of words in document
NormBySent	Contextual polarity score divided by the number of sentences in document

classifier (i.e., RB_{Plain}) and a completely unrestricted feature set (i.e., Uni+Bi). The table shows that normalization as such is important, that is, for all normalization methods the improvement over NoNorm is statistically significant if the union of all data sets is evaluated. However, it is less clear which type of normalization performs best since the performance of the different measures varies throughout the different domains. However, none of the differences among these normalization methods is statistically significant. For the subsequent experiments, we will always apply NormByWord as, on average, it performs slightly (but not significantly) better than the other normalization measures.

All normalization scores have in common that they reflect the length of a test document.[10] Thus, a document being assigned a label with a high confidence score can be translated as a document with a high density of polar expressions combined with a clear majority of one particular polarity type.

4.2.3 Comparing the different feature sets for supervised learning within self-training

Table 10 compares the different feature sets used within the embedded supervised classifier within self-training (SelfTr). As in previous sections, we bootstrap with the standard rule-based classifier (i.e., RB_{Plain}). We also include as a baseline the performance of that rule-based classifier.

The table shows that—with the exception of the *Sports* domain—no matter which feature set is used, we obtain an improvement in performance over the plain rule-based classifier that is statistically significant. On the *Sports* domain no single feature set reaches significantly better results than the rule-based classifier. This particular domain already displayed some problematic behavior on the comparison of the different rule-based classifiers in Sect. 3.5. The reasons given for that (e.g., many misspelt words) may also be responsible for the deviation in performance between this domain and the remaining ones on the experiments discussed in this section.

Self-training exceeds the performance of the rule-based classifier using any of the feature sets including MPQA (i.e., the feature set that is used in both the rule-based and the supervised classifier). As already described in Sect. 4.1, this means that a notable increase of performance is obtained by (just) learning domain-specific weights for the features that are already used within the rule-based classifier. As MPQA is, however, usually worse than the other feature sets we have evidence that it is also important for supervised learning to consider other features (i.e., domain-

[10] Even NormByPol reflects the length of the document as the longer a document is the more polar expressions it will (potentially) contain.

Table 9 Accuracy of different normalization methods in self-training

Domain	NoNorm	NormByPol	NormBySent	NormByWord
Computer	64.17	78.61*	78.39*	**80.22***
Products	59.44	69.17*	68.89*	**70.78***
Sports	65.72	66.33	**67.78**	66.44
Travel	68.44	**70.50**	69.67	69.56
Movies	70.70	72.75	**73.25**	72.70
Average	65.69	71.47*	71.60*	**71.94***

Statistical significance is based on a chi-square test using $p < 0.05$; for *Average* the significance is tested on the union of all domain data sets

* Significantly better than NoNorm

Table 10 Accuracy of self-trained classifiers with different feature sets

Domain	RB (baseline)	SelfTr				
		Top2000	Adj600	MPQA	Uni	Uni+Bi
Computer	64.11	77.67*	74.67*	73.06*	78.56*	**80.22***
Products	60.78	69.00*	68.33*	66.72*	69.06*	**70.78***
Sports	64.33	65.83	63.89	65.11	64.22	**66.44**
Travel	64.61	69.33*	70.83*	68.44*	69.17*	**9.56***
Movies	61.75	70.80*	69.50*	69.40*	71.55*	**72.70***
Average	63.12	70.53*	69.44*	68.55*	70.51*	**71.94***,†

Statistical significance is based on a chi-square test using $p < 0.05$; for *Average* the significance is tested on the union of all domain data sets

* Significantly better than RB

† Significantly better than any other feature set

specific features) than those contained in the domain-independent polarity lexicon. It is also worth noting that Adj600 performs on average slightly better than MPQA (with its 7,600 features) even though this feature set only comprises 600 words. This finding, however, is consistent with previous work on semi-supervised learning where this feature set displayed good performance throughout the different domains (Wiegand and Klakow 2009a). The advantage of this feature set is that it contains domain-specific features of which a very high proportion are predictive words, that is, polar expressions.

The feature set producing the best results is Uni+Bi. Though on some domains the differences to other feature sets is comparatively small, there is no domain in which another feature set outperforms this feature set. Top2000 and Uni are very similar to each other and usually only slightly worse than Uni+Bi. Considering the union of all domain data sets, however, the improvement of Uni+Bi (over Top2000 and Uni) is even statistically significant. This means that, as far as feature design is concerned, the supervised classifier within self-training behaves similar to an ordinary supervised classifier (Ng et al. 2006). Unlike in semi-supervised learning (Wiegand and Klakow 2009a), a noiseless feature set is not necessary.

Qiu et al. (2009) report best performance of SelfTr using a large set of polar expressions. The feature set comprises an open-domain polarity lexicon and is automatically extended by domain-specific expressions. Our results suggest that a less complex alternative has a similar effect. Using SelfTr with unigrams and bigrams (i.e., SelfTr$_{Uni+Bi}$) already provides better classifiers than SelfTr with a polarity lexicon (i.e., SelfTr$_{MPQA}$). The increase is by approximately 3 % points.

4.2.4 Comparing the different rule-based classifiers with self-training

Table 11 compares the different rule-based classifiers and self-training. As a feature set for the supervised classifier within self-training, we chose the best performing feature set from our previous experiments, that is Uni+Bi. The table shows that improving a rule-based classifier also results in an improvement of the self-trained classifier. If the union of all domain data sets is considered, this is even significant with the exception of SelfTr(RB$_{Plain}$) to SelfTr(RB$_{bWSD}$).

Self-training does not work with some rule-based classifiers on particular domains. This is most evident in the *Sports* domain using self-training with RB$_{bWSD}$. Apparently, the better the rule-based classifier is, the more likely a notable improvement by self-training can be obtained. Note that in the *Sports* domain the self-trained classifier using the most complex rule-based classifier, that is, SelfTr(RB$_{Weight}$), achieves the largest improvement compared to the rule-based classifier. We also checked the other feature sets for this particular case and could confirm similar tendencies.

It is also worth pointing out that, considering the averaged results over all domains the gain in performance that is achieved by improving a basic rule-based classifier (i.e., RB$_{Plain}$) with incorporating the largest amount of context information

Table 11 Comparison of accuracy between different rule-based classifiers (RB) and self-trained classifiers (SelfTr) trained with best feature set (Uni+Bi) on different domains

Domain	RB$_{Plain}$		RB$_{bWSD}$		RB$_{Neg}$		RB$_{Weight}$	
	RB	SelfTr	RB	SelfTr	RB	SelfTr	RB	SelfTr
Computer	64.11	80.22	70.61	81.72	73.56	83.67^{*}	75.11	83.22^{*}
Products	60.78	70.78	66.06	73.89^{*}	71.06	$77.00^{*,\dagger}$	71.72	$77.39^{*,\dagger}$
Sports	64.33	66.44	64.39	64.94	67.50	68.89^{*}	69.17	$72.28^{*,\dagger,\ddagger}$
Travel	64.61	69.56	67.39	69.83	70.72	$73.33^{*,\dagger}$	73.56	$77.61^{*,\dagger,\ddagger}$
Movies	61.75	72.70	64.80	72.45	67.85	73.55	72.10	$77.80^{*,\dagger,\ddagger}$
Average	63.12	71.94	66.65	72.57	70.14	$75.29^{*,\dagger}$	72.33	$77.66^{*,\dagger,\ddagger}$

Statistical significance is based on a chi-square test using $p < 0.05$; for *Average* the significance is tested on the union of all domain data sets

* Significantly better than SelfTr bootstrapped on RB$_{Plain}$

† Significantly better than SelfTr bootstrapped on RB$_{bWSD}$

‡ Significantly better than SelfTr bootstrapped on RB$_{Neg}$

Table 12 Comparison of different evaluation measures between different rule-based classifiers (RB) and self-trained classifiers (SelfTr) trained with best feature set (Uni+Bi) on different domains

Domain	RB_{Plain}		RB_{bWSD}		RB_{Neg}		RB_{Weight}	
	RB	SelfTr	RB	SelfTr	RB	SelfTr	RB	SelfTr
Prec^{+}	59.15	73.28	63.25	74.38	67.19	76.96	68.97	80.71
Rec^{+}	85.36	69.13	79.37	68.67	78.61	72.01	81.23	72.69
F1^{+}	69.81	71.14	70.39	71.41	72.42	74.39	74.58	76.47
Prec^{-}	73.98	70.76	72.50	71.02	74.45	73.84	77.26	75.21
Rec^{-}	40.87	74.75	53.93	76.46	61.66	78.57	63.43	82.63
F1^{-}	52.36	74.69	61.79	73.64	67.40	76.12	69.62	78.73
Acc.	63.12	71.94	66.65	72.57	70.14	75.29	72.33	77.66

(i.e., RB_{Weight}) is very similar to the gain that is achieved by just self-training it (i.e., RB_{Plain}) with the best feature set (i.e., SelfTr_{Uni+Bi}). Fortunately, however, these improvements are complementary which means that if they are combined (i.e., $\text{SelfTr}_{Uni+Bi}(\text{RB}_{Weight})$) this results in a further significant improvement.

4.2.5 Performance on the different classes

Table 12 compares precision, recall, and F(1)-score of the different classes for self-training using 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 significantly better than the score of the negative class. The high F1^{+} results from a high recall and lower precision whereas the low F1^{-} results from a fairly low recall but high precision. This is consistent with previous findings (Andreevskaia and Bergler 2008). The gap of F1 between the two classes, however, varies depending on the complexity of the classifier. In RB_{Plain}, the gap is 17.45 % points, in RB_{bWSD} it is 8.6 % points, whereas it is just approximately 5 % points in RB_{Neg} and RB_{Weight}. These numbers can be interpreted in the following way: People usually explicitly employ positive polar expressions in order to utter a positive opinion. However, they are more reluctant to use negative polar expressions to convey negative opinions. In RB_{Plain}, many negative instances are classified as positive since many negative opinions in a document are not recognized. Moreover, due to the lacking disambiguation of polar expressions many false positive polar expressions are detected. A notable improvement is obtained by applying some disambiguation (i.e., RB_{bWSD}) as thus fewer (false) positive polar expressions are detected. Since we measured another notable improvement on the detection of negative opinions by incorporating negation modeling and the improvement on F1^{-} is much larger than on F1^{+}, we may infer that people often employ negated positive polar expressions to convey negative opinions. The fact, however, that we still measure a performance gap between the detection of positive and negative opinions in spite of all linguistic modeling (i.e., RB_{Weight}) and using a polarity lexicon that contains almost twice as many negative polar expressions as positive expressions (see Sect. 3), shows that detecting negative opinions is really a hard problem.

In SelfTr, $\mathrm{F1}^{-}$ is usually better than $\mathrm{F1}^{+}$. By applying SelfTr, the amount of instances being predicted as positive is reduced (in comparison to RB) which results in a decrease in recall but a notable rise in precision. At the same time, the classifier predicts more negative instances resulting in a boost in recall and a slight drop in precision. This relation between the two classes is typical of supervised polarity classifiers (Andreevskaia and Bergler 2008). However, it should also be pointed out that the gap between $\mathrm{F1}^{+}$ and $\mathrm{F1}^{-}$ is much smaller (approximately 2–3 % points). Moreover, the size of the gap does not bear any relation to the gap in the original RB, that is, although there is a considerable difference in size between the gaps of RB_{Plain} and RB_{Neg} (i.e., the gap in RB_{Plain} is much larger than in RB_{Neg}), the size of the gaps in the self-trained versions is fairly similar. We assume that it lies in the nature of the supervised learner to produce a model that equally well detects positive and negative instances (provided that one uses a data set with an equal class distribution). Since it is not bound to polar expressions and infers negative polarity in a data-driven manner, the supervised learner may be more successful in doing so than the rule-based classifier.

4.2.6 Why the features from rule-based classifier and supervised classifier must be kept apart

We also experimented with a feature set for the supervised classifier (within self-training) combining bag of words and the knowledge encoded in the rule-based classifier. The features we derive from the rule-based classifier are the two basic features, that is, the number of positive and negative polar expressions within a data instance (according to the output of RB_{Neg}) and for each property that we considered for heuristic weighting in RB_{Weight} a feature conjoined with either of those basic features. For instance, for the property StrongPol (see Sect. 3.4.1), there is one feature indicating the number of strong positive polar expressions and another indicating the number of strong negative polar expressions, respectively.

Table 13 compares the performance of self-training without using those features derived from the rule-based classifier ($\mathrm{SelfTr}_{without}$) and a classifier using those features (SelfTr_{with}). That is, $\mathrm{SelfTr}_{without}$ just uses bag-of-words features while SelfTr_{with} uses bag-of-words features and the additional features derived from RB. For self-training, we chose the best configuration from previous experiments (i.e., RB_{Weight} and Uni+Bi for bag of words). Note that for the supervised classifier we omit the tf-idf encoding since it does not make sense to apply it on the features derived from the rule-based classifier.[11] The table shows that the performance of this combination is worse than a classifier trained on bag of words. The correlation between the features derived from the rule-based classifier, in particular the basic

[11] Since those features will occur much more frequently than plain words throughout the documents, the inverted document frequency will always be very low which would consequently heavily downweight those features.

Table 13 Accuracy of self-training *with/without* features from rule-based classifier within supervised learner

Type	Computer	Products	Sports	Travel	Movies	Average
$\text{SelfTr}_{without}$	82.50	75.78	72.39	75.61	75.85	76.43
SelfTr_{with}	78.39	74.44	69.17	73.61	71.65	73.45

features, and the class labels[12] is disproportionately high since these features essentially encode the prediction of the rule-based classifier. (Individual words, on the other hand, correlate much less with those class labels.) Consequently, the supervised classifier develops a strong bias towards these features and inappropriately downweights the bag-of-words features. Therefore, the supervised classifier within self-training should not use any features from the rule-based classifier or more complex features that expand those features from that classifier.

4.2.7 What is learned by self-training

In this section, we want to illustrate that the knowledge learned by self-training is potentially more expressive than the knowledge encoded in a rule-based classifier. For this purpose, we inspect the most highly ranked n-grams in a particular domain data set—we chose the *Computer* domain—according to the point-wise mutual information to the class labels as predicted by self-training. Table 14a illustrates the 50 most highly ranked positive instances while Table 14b illustrates the most highly ranked negative instances.

There are many highly ranked n-grams that do not contain intuitive polar expressions. Several n-grams include product brands, such as *Mac, Intel, Dell Computer*, or *My Yahoo*, or items towards which people usually have a strong sentiment, for instance, *high-speed internet, installation fee*, or *collection agency*. Such entities are domain-specific and are not contained in the polarity lexicon we use, yet they may be helpful for polarity classification (as the previous evaluation has shown). On our data set they correlate with some specific polarity type. Therefore, these expressions can be treated as (traditional) polar expressions as long as one uses this particular data set. However, the general effectiveness of some of these expressions, such as brands, beyond our data set is debatable. The correlation of those brands with a polarity type reflects a strong sentiment of public opinion towards them. This sentiment may be transient. In other words, public opinion towards these items may be different from that five years ago or five years in the future. After all, these expressions should be classified as opinion targets rather than polar expressions. As a consequence, those features could mislead the classification on data sets taken from another point in time rather than improve it. So, at least brands should be used with caution on other data sets. Some opinion targets, however, become fairly reliable polar expressions. For example, in the *Sports* domain we found that *gretzky* is a highly positive cue. It refers to former

[12] We mean the class labels that are predicted by RB_{Weight} and henceforth treated as actual class labels by the supervised learner.

Table 14 Illustration of the 50 most highly ranked features per class from the *Computer* domain

(a) Positive features				
most_stable	intel	just_bought	everything_from	europe
is_superb	prodigy	is_annoying	outstanding_.	mp
program_with	high-speed_internet	great_value	also_has	bell_and
graphics_and	plane_ticket	great_in	an_apple	(_although
is_right	some_really	acrobat	design_,	chile
a_fantastic	mac_,	everyone_.	processing_program	member_.
sheet	joystick_.	a_4	this_joystick	is_outstanding
is_amazing	best_operating	blocker_,	great_feature	the_www
stack	social_network	amiga_was	my_yahoo	to_communicate
,_read	the_blog	apple_is	predecessor	greeting_card
(b) Negative features				
installation_fee	said_we	dell_computer	past_the	rate_to
get_slower	directly_.	stole	we_finally	real_problem
on_ever	hold_and	followed_by	collection_agency	not_cover
but_get	sure_enough	never_signed	it_were	bbb_and
unbelievable_.	last_4	promised_it	sbc_service	since_october
all_have	goes_down	3_of	town_and	isnt
a_favor	you_feel	care_,	it_like	g20
terrible_service	sent_the	month_the	im_not	at&t_worldnet
are_simply	$100_,	sign_of	reps_i	issues_and
andy	never_call	wonder_how	would_send	was_out

professional ice hockey player *Wayne Gretzky* who due to his outstanding success is often used as a (positive) reference point for other opinion targets. In these situations, the former hockey player is not evaluated himself but used as a means of evaluating another sportsperson, usually by applying patterns of the form *X is almost as good as/better than Gretzky*.

Other n-grams not containing polar expressions become plausible if we reconstruct their contexts. For instance, the bigram *also has* is considered positive as it occurs in contexts enumerating a plethora of functions offered by a product (which, as such, can already be interpreted as a positive property).

In general, both tables contain a large portion of bigrams. Our polarity lexicon only contains unigrams, so this is another indication that different features are taken into consideration. Quite many of those bigrams contain polar expressions from the polarity lexicon. A bigram containing a polar expression may be less ambiguous (and hence more expressive) than just the occurrence of a polar expression as the bigram encodes some local context. As we already discussed in Sect. 4.1, such bigrams may encode relevant linguistic phenomena. Indeed, we find cases of these phenomena in the two tables, such as intensification (e.g., *most stable* or *real problem*) or negation (e.g., *not cover*).

An interesting case is the highly ranked negative n-gram *wonder how*. The word *wonder* is ambiguous. As a noun its meaning is similar to *marvel* or *miracle* and as a verb it means either *enquire* or *question*. In the former case, the word is definitely positive, whereas in the latter case the word is either not polar or negative (but admittedly with a much weaker polar intensity). Unfortunately, our polarity lexicon does not make this distinction and always classifies *wonder* (irrespective of its part of speech) as positive. Self-training (at least partially) resolves this ambiguity, as it established *wonder how* as a negative n-gram. The word *wonder* followed by *how* usually refers to the verb with the sense of *enquire*. At least for our domain corpus it is appropriate to classify this bigram as negative as a typical context such as (20) taken from our corpus suggests.

(20) I *wonder* how many error reports I've sent to Microsoft in the last hour.

A similar case is *great* which often appears as a modifier of product properties (e.g., *great value* or *great feature*). The word as such is also ambiguous. Apart from being a positive polar expression, it can also function as an intensifier containing no polarity (21).

(21) We are often required to spend a *great* deal of time at each other's homes when there is a *great* deal of work to be done.

There are also several n-grams comprising a tensed auxiliary followed by a positive polar expression, for example, *is right, is superb, is outstanding*. Tense may be informative within this domain, as we observed quite often positive polar expressions in a past tense in negative reviews (22). Present tense, on the other hand, may then be indicative of positive polarity.

(22) I $loved_{Past}$ the Inspiron 8600 … until after 1 week, the hard drive died.

5 Compositional polarity classification and self-training

In this section, we will compare the rule-based classification we presented in previous sections with compositional polarity classification (also with respect to self-training). As a rule-based compositional polarity classifier, we will examine the PolArt system (RB_{PolArt}). This is a multilingual classifier that has already been evaluated on various data sets (Klenner 2009; Klenner et al. 2009a, b).

The main difference between this classifier and the classifiers that have been evaluated in the previous sections is that polar expressions are not considered in isolation from each other but are combined by rules to form larger linguistic units, such as noun phrases (NPs), verbs phrases (VPs), and sentences. The polarity composition is implemented as a cascade of transducers operating on the prior polarities of a polarity lexicon, a chunk parser (TreeTagger, Schmid 1994), and a set of pattern-matching rules. For instance, Table 15 illustrates the rules for NPs. The system employs a total of 60 compositional rules.

Due to the fact that polarity is assigned to general linguistic units rather than just lexical units, PolArt also employs a more dynamic negation scope modeling than the previously proposed method using fixed (word-based) window size (see also Sect. 3.3.2). In general, the rules of PolArt restrict the scope of negation to the

Table 15 Illustration of rules for NP level from PolArt

ADJ	NOUN	→	NP	Example
NEGATIVE	POSITIVE	→	NEGATIVE	a disappointed hope
NEGATIVE	NEGATIVE	→	NEGATIVE	a horrible liar
POSITIVE	POSITIVE	→	POSITIVE	a good friend
POSITIVE	NEGATIVE	→	NEGATIVE	a perfect misery
POSITIVE	NEUTRAL	→	POSITIVE	a perfect meal
NEGATIVE	NEUTRAL	→	NEGATIVE	a horrible meal

following chunk containing content bearing words, that is, typically noun or verb chunks (23). While scope modeling based on window size (recall that the window size we use is 5) may erroneously reverse polar expressions that are not within the actual scope of the negation, such as (24)–(26), the modeling based on chunk parsing is more likely to determine the correct scope.

(23) Locating the human soul and discovering what makes us survive is [*not* [such an $\underline{easy}^{+}$ task$]^{+}_{NounChunk}]^{-}$.

(24) Still earthlink [*cannot* [tell]$_{VerbChunk}$] me what is $\underline{\text{wrong}}^{-}$.

(25) Do [*not* [deal]$_{VerbChunk}$] with these $\underline{\text{morons}}^{-}$!

(26) This is a really good movie. [*No* []], this is a $\underline{\text{great}}^{+}$ movie.

5.1 Evaluation at the document level

We will now compare the performance of compositional polarity classification with that of traditional rule-based classification at the document level. The two types of classifiers will also be evaluated with regard to self-training. We will carry out the experiments on the same data on which the experiments of previous sections have been conducted. As a standard rule-based classifier, we consider the best rule-based classifier from previous sections, that is RB_{Weight}. Since the resources that PolArt uses are different to the ones that have been employed in previous experiments, that is, PolArt uses a different part-of-speech tagger (TreeTagger, Schmid 1994) and different lexical resources, such as negation words and intensifiers, we modify our standard rule-based classifier in that it uses the identical resources as PolArt in order to ensure comparability between those classifiers.[13] In order to indicate the difference between the standard rule-based classifier employed in previous experiments and the one used in the experiments described in this section, we will refer to the (standard) classifier using the resources of PolArt as $\text{RB}_{Weight*}$ (rather than RB_{Weight}). For self-training, we also use the best configuration of previous experiments, that is SelfTr_{Uni+Bi}.

Table 16 displays the results. RB_{PolArt} does not outperform $\text{RB}_{Weight*}$. On most domains, it is actually worse than $\text{RB}_{Weight*}$ though on no domain the drop is statistically significant. Self-training, however, consistently improves a rule-based

[13] We also ensure that both classifiers predict the same default polarity if the rule-based classifier predicts a tie.

Table 16 Comparison of standard rule-based polarity classifier ($\mathrm{RB}_{Weight*}$) and compositional classifier (RB_{PolArt}) with respect to self-training on document-level data sets (evaluation measure: accuracy).

Domain	$\mathrm{RB}_{Weight*}$		RB_{PolArt}	
	RB	SelfTr	RB	SelfTr
Computer	72.50	82.61	72.67	81.56
Products	70.78	77.56	68.33	74.22
Sports	65.14	67.66	62.94	66.49
Travel	70.44	72.22	67.83	69.61
Movies	67.30	73.25	67.30	69.85
Average	69.23	74.66	67.81	72.35

classifier, no matter whether it operates on the output of a standard classifier or a compositional classifier. As in previous experiments, the degree of improvement varies. The low impact of the compositional classifier is reminiscent of the impact of the different (individual) features used for heuristic weighting in the standard classifier (Sect. 3.4.6). Apparently, it is difficult (at least at the document level) to greatly improve polarity classification with straightforward linguistic methods.

5.2 Evaluation at the sentence level

In this section, we evaluate the performance of compositional polarity classification at the sentence level. Sentence-level polarity classification is usually harder than document-level polarity classification since less text (within an instance) for classification is available. The difficulty is also reflected by a lower accuracy achieved by supervised learning with bag-of-words features (Wiegand and Klakow 2009b). Since there is less text and therefore also fewer polar expressions, it may also be more important to disambiguate each individual expression at the sentence level than at the document level.

We evaluate the performance on a standard data set (Hu and Liu 2004) on which PolArt has already been evaluated (Klenner et al. 2009a). We downsample the data set to equal class sizes as class imbalance is a complex issue and will, therefore, be discussed separately in Sect. 7. The resulting data set contains 2,888 sentences (i.e., 1,444 sentences per class). Again we compare the standard rule-based classifier $\mathrm{RB}_{Weight*}$ with the compositional classifier RB_{PolArt}. For self-training, we use the same configuration as in previous experiments (with, of course, the exception that we use unlabeled sentence-level data instead of unlabeled document-level data). Table 17 shows the results. Unlike in the experiments at the document level, the compositional classifier outperforms the standard classifier. The improvement obtained by the former is even statistically significant. This supports our assumption that fine-grained polarity classification requires more linguistically-informed analyses. This insight is also reflected by the fact that other compositional approaches similar to PolArt have not been evaluated at the document level but on expression level (Choi and Cardie 2008) or at the sentence level (Moilanen and Pulman 2007).

Table 17 Comparison of standard rule-based polarity classier (RB_{Weight^*}) and compositional classifier (RB_{PolArt}) with respect to self-training on sentence-level data set (evaluation measure: accuracy)

	RB	SelfTr
RB_{Weight^*}	69.81	70.98
RB_{PolArt}	73.16*	75.55

Statistical significance is based on a chi-square test using $p < 0.05$

* Significantly better than RB_{Weight^*}

Table 17 also shows that self-training consistently improves the rule-based classifier but the general impact is fairly low. This can be explained by the fact that bag-of-words feature sets are much sparser on a sentence-level classification task than on a document-level classification task (as a document usually contains much more unique words than a sentence) (Wiegand and Klakow 2009b).

In summary, compositional polarity classification is much more effective on sentence-level classification tasks than on document-level tasks. On the latter, a standard rule-based classifier in combination with self-training is a more promising alternative.

6 Comparison with statistical domain adaptation

In this section, we will compare self-training with statistical domain adaptation. Again, we will consider document-level classification as this is the typical task on which domain adaptation is evaluated in sentiment analysis (Beineke et al. 2004; Blitzer et al. 2007; Tan et al. 2008, 2009; Melville et al. 2009; Prabowo and Thelwall 2009; Qiu et al. 2009; Titov 2011). By statistical domain adaptation one understands data-driven algorithms that combine labeled out-of-domain training data with unlabeled in-domain training data. This setting mirrors real-life situations as usually the labeled training data that are available for a particular task do not originate from the domain for which one intends to build a classifier. On the other hand, unlabeled in-domain training data are easy to obtain as they require no manual annotation.

Statistical domain adaptation can be considered as a special type of semi-supervised learning, which, in general, incorporates labeled and unlabeled training data but not necessarily labeled training data from a domain that is different to the one from which the test data are sampled. The main differences between statistical domain adaptation and self-training, therefore, are that the former is data-driven and considers little linguistic structure (e.g., most approaches applied on sentiment analysis usually just consider a plain bag-of-words feature representation) while the latter does not consider *any* labeled training data but a polarity lexicon and linguistic rules.

In this work, we only consider (statistical) domain adaptation methods and exclude traditional semi-supervised learning algorithms from our evaluation since a direct comparison of semi-supervised learning and self-training has already been published in (Wiegand and Klakow 2010) showing that self-training is much more effective.

Many state-of-the-art adaptation approaches are based on the idea of *shared feature representation* (Blitzer et al. 2006) in which a new representation of the data instances is induced from the original feature representation (usually bag of words) that makes instances from source and target domain look more similar than in the original representation. This feature representation allows to train more robust classifiers. A popular algorithm that incorporates this idea is *structural corresponding learning* (Blitzer et al. 2007) in which predictive features from a source domain (they are derived from a manually labeled training set) are automatically aligned to a set of predictive features in a target domain (for which only unlabeled data are available) with the help of a set of domain-independent *pivot* features. Designing those pivot features can be considered as an auxiliary task and formulating those features is a non-trivial engineering problem that requires task-specific knowledge. That is why we will make use of a more recent adaptation method that is not dependent on this auxiliary task (even though it still uses shared feature representation). Titov (2011) presents a domain adaptation approach based on *latent variable models*. These latent variables capture regularities on unlabeled data from both domains. In order to damp the influence of latent variables that correspond to clusters of features only specific to the source domain (which would cause classifiers being tested on the target domain to perform poorly) the objective function of the learning algorithm includes a term that regularizes inter-domain differences in marginal distributions of each latent variable. This adaptation method has been shown to be competitive with that of structural corresponding learning despite the omission of pivot features.

6.1 Evaluation

Since we cannot replicate the set-up used in (Titov 2011) for our self-training method as that data set only exists in the form of a bag-of-words feature representation[14] and our rule-based classifier requires some natural language text tagged with parts of speech, we need to re-run the statistical domain adaptation on some different data set on which we can also re-rerun our self-training approach. As a data set, we sampled some data from the original crawl from which Titov (2011) got his preprocessed data.[15] Note that we could not reuse the data set from our previous experiments at the document level (see Sect. 2) as the amount of unlabeled data is insufficient for the statistical domain adaptation method. As labeled training data we use 2,000 documents (i.e., 1,000 positive and 1,000 negative documents each), and as unlabeled training data we use exactly the amount of data that was employed in (Titov 2011) (the size varies throughout the different domains).[16] We always test on 2,000 data instances (again, 1,000 positive and 1,000 negative

[14] Titov (2011) made his experiments on the data set available at: http://www.cs.jhu.edu/~mdredze/datasets/sentiment/processed_acl.tar.gz.

[15] Available at: http://www.cs.jhu.edu/~mdredze/datasets/sentiment/unprocessed.tar.gz.

[16] We even replicated the distribution of positive and negative instances in the unlabeled training data (note that the crawl does not contain any mixed reviews), even though those distributions were always close to uniform class distribution.

documents each). Since we have a varying amount of unlabeled data ranging from 3,586 to 5,945 documents per domain, we need to set the number of unlabeled documents that will be used as (pseudo-)labeled training data within self-training in proportion to the total amount of available documents per domain (rather than employing a fixed number of documents as has been done in Sects. 4, 5). We always use 70 % which provided good performance on all domains. The optimal performance on each individual domain does not necessarily coincide with this configuration. Previous experiments (see Sect. 4.2.1) have shown that the optima of different domains may diverge. We consequently felt that using the specific optimal configuration for each respective domain would be tantamount to overfitting since for an unknown domain the specific configuration would not be known. Therefore, the choice of 70 % is a fairly domain-independent configuration which should also provide reasonable results for a new (unknown) domain.

Table 18 displays the results of this comparison. In addition to the results of statistical domain adaptation and self-training, we also display the results of the best rule-based classifier from previous experiments (i.e., RB_{Weight} from Sect. 3) and both out-of-domain and in-domain supervised learning. For both types of supervised classifiers, we employ SVMs. As a feature set, we use all unigrams and bigrams.[17] For domain adaptation, we only consider the best classifier presented in (Titov 2011), that is, a latent variable model with regularization term combined with an out-of-domain model using product-of-experts.[18] For the out-of-domain classifiers and domain adaptation, we always present three different results (each of them differs in the labeled training set that is used): one considers the source domain that produces worst results on a particular test set (i.e., *Worst Out-of-domain Supervised* and *Worst Domain Adaptation*), one that considers the source domain that produces best results (i.e., *Best Out-of-domain Supervised* and *Best Domain Adaptation*), and the average of all source domains[19] (i.e., *Average Out-of-domain Supervised* and *Average Domain Adaptation*). For self-training, we consider the best model from previous experiments, i.e., $\text{SelfTr}_{Uni+Bi}(\text{RB}_{Weight})$.

The results of Table 18 show that RB is by far the worst classifier. Even *Worst Out-of-domain Supervised* is systematically better. There is always a large gap between the worst and the best source domain for out-of-domain supervised learning (and this is also reflected by domain adaptation). The reason for that is that some domains are very similar, in particular *Electronics* and *Kitchen*.[20] As a consequence, the corresponding out-of-domain classifiers, e.g., a classifier that is trained on *Kitchen* and tested on *Electronics*, produce good results which are extremely hard to beat.

[17] Note that unigrams alone did not produce better results for either in-domain or out-of-domain classification.

[18] In Titov (2011) this model is referred to as *Reg*+.

[19] Of course, we only consider those source domains which are different to the target domain on which is tested.

[20] This is due to the fact that many items in *Kitchen* are electric devices whose reviews cover aspects that are similar to the ones discussed in the reviews from the *Electronics* domain, such as usability or malfunctioning components.

Table 18 Comparison of statistical domain adaptation with other classifiers (evaluation measure: accuracy)

Classifier	Domain				
	Books	Dvd	Electronics	Kitchen	Average
RB	67.15	65.60	64.00	68.25	66.25
Worst Out-of-domain Supervised	70.15	70.55	70.45	73.70	71.21
Best Out-of-domain Supervised	78.65	77.50	81.00	82.60	79.94
Average Out-of-domain Supervised	73.43	73.12	74.12	77.08	74.44
Worst Domain Adaptation	71.55	73.15	74.10	78.50	74.33
Best Domain Adaptation	76.90	79.40	84.20	86.50	81.75
Average Domain Adaptation	74.23	75.23	78.00	81.48	77.24
SelfTr	76.89	74.63	80.88	80.19	78.15
In-domain Supervised	83.20	82.60	86.70	84.25	84.19

Each domain adaptation method outperforms its supervised out-of-domain counterpart with one exception being *Best Domain Adaptation* tested on *Books*. However, the drop in performance compared to *Best Out-of-domain Supervised* is not statistical significant.[21]

On this data set, self-training definitely works as well. It always achieves a notable improvement over our previous baseline RB. However, it is difficult to judge whether self-training or domain adaptation is more robust. The performance of domain adaptation very much depends on the source domain. Self-training, on the other hand, exclusively considers unlabeled in-domain training data. Compared to *Worst Domain Adaptation, SelfTr* is the clear winner. The opposite situation is the case, however, when *SelfTr* is compared with *Best Domain Adaptation*. If we consider the average performance of domain adaptation, however, self-training and domain adaptation are on a par with each other, i.e., on two (test) domains *Average Domain Adaptation* slightly outperforms *SelfTr* and on the other two (test) domains *SelfTr* slightly outperforms *Average Domain Adaptation*.[22] As expected *In-domain Supervised* presents an upper bound. Only *Best Domain Adaptation* slightly outperforms *In-domain Supervised* on *Kitchen*. Both *Average Domain Adaptation* and *SelfTr* are still notably lower than this upper bound. This shows that there is still some considerable room for improvement.

In conclusion, there is no clear winner between statistical domain adaptation and self-training. Considering the average performance of domain adaptation, the performance of these two approaches is in fact very similar. We can only formulate a rule-of-thumb that suggests to consider statistical domain adaptation if the source domain is fairly similar to the target domain, and if a distant source domain is considered, self-training might be a safer option.

[21] Significance is based on a chi-square test using $p < 0.05$.

[22] Unfortunately, we cannot carry out any statistical significance tests on the results of this comparison, as there is no commonly established significance test to compare an averaged result (i.e., *Average Domain Adaptation*) with an individual result (i.e., *SelfTr*).

7 Natural class imbalance and mixed reviews

In this section, we want to investigate what impact natural class imbalance has on bootstrapping polarity classifiers with a rule-based classifier. We want to explore how different class-ratio estimation methods approximating the class distribution on the test set perform. Note that the best classification performance is usually obtained when the class distribution of the training set and test set are identical.

In this section, the unlabeled data set will include mixed reviews (in addition to definite positive and negative reviews), that is, 3 star reviews (see Sect. 2). We refrain from including those reviews in our test data. The reason for this is that (as already stated in Sect. 2) these reviews present a very heterogeneous data set that contain both indefinite polar reviews and definite polar reviews (i.e., positive or negative reviews). Therefore, it is inappropriate to assign all these reviews the same class label. 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.

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

As labeled training data for the embedded supervised classifier within self-training, we—similar to Sect. 6—use 70 % of data instances labeled by the rule-based classifier ranked by confidence of prediction. We consider again the best classifier from previous experiments, that is, SelfTr$_{Uni+Bi}$(RB$_{Weight}$). Figure 4 illustrates the set-up of the experiments in this section.

7.1 Comparison of different class-ratio estimates

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 classifier over the entire data set (Ratio-RB) and estimates gained from a small amount of randomly sampled labeled data instances from the data set. We randomly sample 20 (Ratio-20), 50 (Ratio-50) and 100 (Ratio-100) instances. For each configuration (i.e., 20, 50, and 100), we sample 10 times, run SelfTr for each sample and report the averaged result. Table 19 summarizes the different class-ratio estimation methods.

We compare the self-trained classifier with two baselines, that is, a classifier always assigning a test instance to the majority class (Majority-Cl) and the most robust rule-based classifier from previous experiments (RB$_{Weight}$). Note that these two baselines are complementary. While on a balanced data set, Majority-Cl is usually a weak baseline (i.e., in binary classification this corresponds to an accuracy of 50 %), it is a fairly strong baseline on data sets with a heavily skewed class distribution. The larger the proportion of the majority class is, the more difficult it is for a classifier to

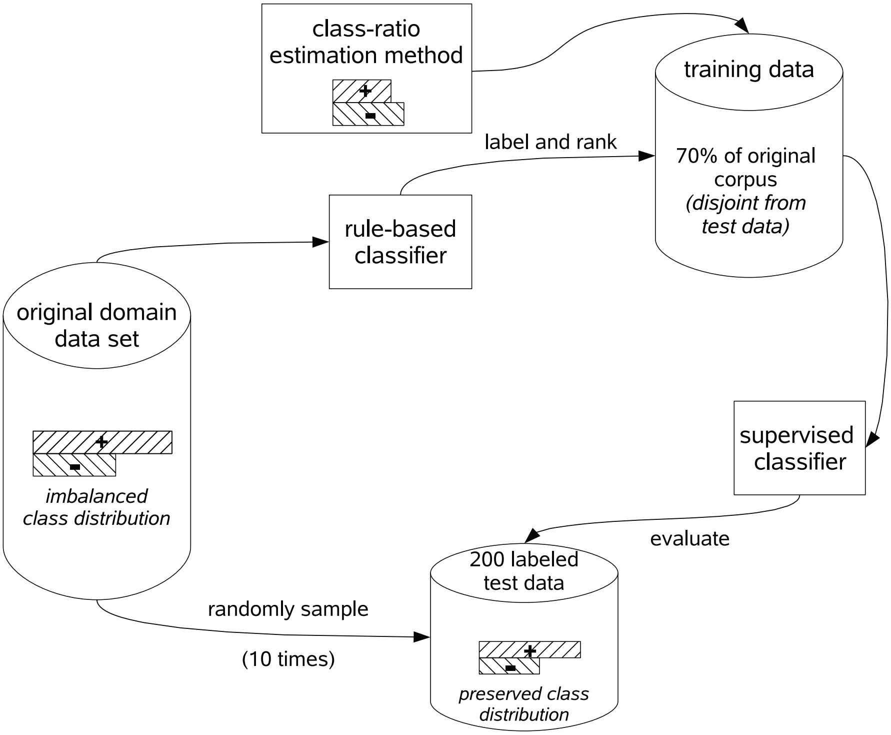

Fig. 4 Set-up of experiments using self-training on data sets with imbalanced class distribution

Table 19 Description of the different class-ratio estimation methods

Ratio-Oracle	Class ratio corresponding to test set (*upper bound*)
Ratio-Balanced	Balanced class ratio (*lower bound*)
Ratio-RB	Class ratio derived from predictions of that data set according to best rule-based classifier (i.e., RB_{Weight})
Ratio-20	Class ratio based on 20 randomly sampled (labeled) documents from the data set
Ratio-50	Class ratio based on 50 randomly sampled (labeled) documents from the data set
Ratio-100	Class ratio based on 100 randomly sampled (labeled) documents from the data set

produce a model that also assigns the label of the minority class to data instances and at the same time makes fewer misclassifications than Majority-Cl. RB_{Weight}, on the other hand, is a much stronger baseline on balanced data sets, while on data sets with a heavily skewed class distribution, it may be worse than Majority-Cl.[23]

Table 20 displays the performance of the different classifiers. (We display the results of the data sets using 3 star reviews within brackets. Note, however, that we

[23] Similar to Sect. 6, we refrain from doing statistical significance tests in this section since Ratio-20, Ratio-50, and Ratio-100 are averaged results over 10 samples whereas the remaining classifiers are single results and there is no commonly accepted way of comparing those different types of data (i.e., averaged results vs. single results).

Table 20 Accuracy of different classifiers tested on naturally imbalanced data: for self-trained classifiers the numbers in brackets state the results on a data set that includes 3 star reviews in the unlabeled (training) data

Domain	Majority-Cl	RB_{Weight}	SelfTr					
			Ratio-Oracle	Ratio-Balanced	Ratio-RB	Ratio-20	Ratio-50	Ratio-100
Computer	56.83	75.05	82.95 (82.80)	**82.55 (82.90)**	77.05 (76.20)	77.41 (76.63)	80.70 (79.85)	81.24 (81.11)
Products	63.07	76.55	81.70 (81.40)	75.85 (76.10)	78.65 (78.55)	77.87 (77.89)	80.26 (80.29)	**81.00 (81.47)**
Sports	78.68	77.35	80.50 (80.80)	61.50 (62.60)	**80.35 (81.15)**	78.97 (79.32)	79.70 (79.80)	80.31 (80.31)
Travel	74.07	79.45	82.25 (82.00)	67.00 (68.15)	**81.80 (81.75)**	79.71 (79.59)	81.26 (81.12)	81.31 (81.44)
Average	68.16	77.10	81.85 (81.75)	71.73 (72.44)	79.46 (79.41)	78.49 (78.36)	80.48 (80.27)	**80.97 (81.08)**

Table 21 Average deviation (in percentage points) of the different class-ratio estimation methods from the actual class distribution along their average accuracy

	Ratio-Balanced	Ratio-RB	Ratio-20	Ratio-50	Ratio-100	(Ratio-Oracle)
Average Deviation	18.16	7.50	8.07	4.28	3.33	(0.00)
Average Accuracy	71.73	79.46	78.49	80.48	80.97	(81.85)

will discuss the impact of mixed reviews in the next section.) SelfTr using Ratio-Balanced produces the worst results among the self-trained classifiers. On average, it outperforms Majority-Cl but it is still worse than RB_{Weight}. On Chinese data, this method (i.e., SelfTr using Ratio-Balanced) has been reported to score rather well (Tan et al. 2008; Qiu et al. 2009). We can only speculate about the reason for these different results, e.g., differences between Chinese and English, differences in the annotation schemes of those data sets, etc. The fact that Ratio-Oracle produces best results comes as no surprise since the class distribution in training and test set is the same. On average, Ratio-100 produces the second best result. Ratio-RB is better than both Ratio-Balanced and the class-ratio estimation method using the smallest labeled sample, that is, Ratio-20.

These results can be best explained by also considering the average deviation (in percentage points) of the individual class-ratio estimation methods towards the actual class distribution on the test set.[24] This information is displayed in Table 21. Ratio-Balanced has the largest deviation and therefore performs worst. Ratio-100 has the smallest deviation and consequently performs better than the other estimation methods. On average, Ratio-RB is slightly better than Ratio-20. As the performance results on Table 20 show, this is mainly due to the fact that Ratio-RB is better on the *Sports* and *Travel* domains. We found that these are domains in which the number of positive opinions largely outweighs the number of negative opinions (see also Table 1). We assume that Ratio-RB works well on these distributions as rule-based classifiers have a general bias towards positive opinions (see also Sect. 4.2.5).

In summary, using (small) samples of labeled data instances is an effective way for class-ratio estimation enabling SelfTr to consistently outperform Majority-Cl and RB_{Weight}.

7.2 Impact of mixed reviews

As Ratio-Oracle, Ratio-RB, Ratio-20, Ratio-50, and Ratio-100 suggest, the presence of mixed polar reviews (see results within brackets in Table 20) does not produce notably different results. The results of Ratio-Balanced even show that using 3 star reviews results in a marginally yet consistently better performance throughout all domains. The reason for these results may be that self-training successfully manages to exclude harmful 3 star reviews and include useful 3 star reviews for the labeled training set. As already stated in Sect. 2, 3 star reviews do not only contain indefinite

[24] Example: if the actual class ratio is 80:20 and the estimated ratio is 90:10, then the deviation will be 10.

polar reviews (harmful reviews) but also positive and negative reviews (potentially helpful reviews). If those reviews with an actually definite polarity were selected for the training collection (and by random selection we identified such cases), this would have the same impact as if a 1, 2, 4, or 5 star review were chosen.

8 Related work

There has been much work on document-level polarity classification using supervised machine learning methods. Various classifiers and feature sets have been explored (Pang et al. 2002; Ng et al. 2006; Salvetti et al. 2006). Support Vector Machines (SVMs) (Joachims 1999) usually provide best results (Pang et al. 2002). Unigram and bigram features outperform complex linguistic features (Ng et al. 2006).

Rule-based polarity classification has attracted similar attention as supervised classification during the last decade. Most rule-based classifiers (that have been empirically validated) share the basic concept of using a polarity lexicon to determine the polarity of a text that is to be classified. These works mainly differ in the way that contextual modification is modeled. Polanyi and Zaenen (2006) propose a framework in which scores are heuristically assigned to polar expressions depending on their individual contexts. Thus, various phenomena such as *negation* and *intensification* are taken into consideration. Implementations inspired by that framework have empirically been proven effective (Kennedy and Inkpen 2006). Further extensions incorporate more complex rules that determine how the polarity of individual expressions or syntactic constituents is combined in order to compute the overall polarity of a phrase, sentence, or even document (Moilanen and Pulman 2007; Shaikh et al. 2007; Choi and Cardie 2008; Klenner et al. 2009b; Min and Park 2011). In addition to these rules, Taboada et al. (2011) propose to assign scores to individual polar expressions rather than giving all polar expressions a uniform (prior-polarity) score. Thus, unlike many other approaches, the individual differences between polar expressions are successfully incorporated into the rule-based classifier.

Semi-supervised learning for polarity classification has been shown to be effective on inducing polarity lexicons from general lexical resources, such as WordNet (Esuli and Sebastiani 2006, 2007; Rao and Ravichandran 2009; Baccianella et al. 2010), or the Web (Turney and Littman 2003; Velikovich et al. 2010) but on text classification, the effectiveness is heavily dependent on the parameter settings. Significant improvement over supervised classification can often only be achieved in the presence of few labeled training data and a predictive feature set, such as in-domain adjectives or polar expressions from a polarity lexicon (Wiegand and Klakow 2009a). A detailed study on cross-domain polarity classification comparing supervised and semi-supervised learning is presented in (Aue and Gamon 2005). Semi-supervised learning (i.e., a derivation of the expectation-maximization algorithm for a naive Bayes classifier) using unlabeled in-domain training data along labeled out-of-domain data outperforms the usage of supervised learning just using labeled out-of-domain data. 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 (Dasgupta and Ng 2009).

Apart from the statistical domain adaptation methods that we already discussed in Sect. 6, there have been other notable methods examined for polarity classification: Tan et al. (2009) propose a semi-supervised version of the naive Bayes classifier, in which the initial classifier using labeled out-of-domain training data is restricted to domain-independent (generalizable) features that are acquired by incorporating a metric based on the Frequently Co-occurring Entropy. During the iterations larger weights are assigned to the contribution of the unlabeled in-domain training data, allowing domain-specific knowledge to be included into the model. There is some conceptual similarity to the self-training algorithm proposed in this article as both approaches make use of an initial classifier with domain-independent knowledge for bootstrapping. Beineke et al. (2004) propose a model in which the knowledge gained from Web-based lexicon induction (Turney and Littman 2003) is incorporated into a Bayes classifier using labeled in-domain training data. Similarly, Andreevskaia and Bergler (2008) present an approach in which a rule-based classifier based on a polarity lexicon and a supervised classifier trained on in-domain data are combined. The combination exploits the complementary precision of the two approaches on positive and negative data instances. Melville et al. (2009) and Prabowo and Thelwall (2009) consider the same types of classifiers as Andreevskaia and Bergler (2008). While in (Melville et al. 2009) they are incorporated into a generative model, in (Prabowo and Thelwall 2009) a sequential order of the classifiers is determined and a prediction of an individual classifier is only considered if the preceding classifier (according to that order) fails to provide a classification. The major difference between Andreevskaia and Bergler (2008), Melville et al. (2009), and Prabowo and Thelwall (2009), on the one hand, and the approach presented in this article, on the other hand, is that our method is the only approach that does not require any labeled training data as we present a (strictly) sequential classifier in which the (unsupervised) rule-based classifier always comes first.

Bootstrapping supervised machine-learning classifiers with the help of rule-based classification has already been effectively applied to subjectivity detection of sentences (Wiebe and Riloff 2005). The method has also been applied to polarity classification, but so far only on Chinese data (Tan et al. 2008; Qiu et al. 2009). While the performance with out-of-domain supervised classifiers is compared in (Tan et al. 2008), this method is embedded into a complex bootstrapping system that also extends the vocabulary (i.e., feature set) of the rule-based classifier in (Qiu et al. 2009). In (Wiegand and Klakow 2010), we already presented further novel contributions, such as examining the impact of the rule-based classifier on the final result, the relation between self-training and semi-supervised learning, and various settings of the self-training algorithm, in particular, different feature sets for the supervised classifier and the impact of imbalanced class distribution. However, this work significantly extends that initial evaluation. In this article, we had a more detailed look at the impact of the different components within a rule-based classifier. For negation modeling, we examined the importance of polarity shifters, negation disambiguation, and scope optimization. For heuristic weighting, we evaluated individual features and also extended the set of features introduced in (Wiegand and Klakow 2010). We found that on a cross-domain evaluation, only very few features (e.g., polar adjectives and modality) systematically help. Moreover, we also examined the impact of compositional rule-based classification showing that these two types of classification are complementary. While self-training works better at the document level than at the sentence level, the reverse case is true for compositional rule-based classification. As far as the self-training *algorithm* is concerned, we looked in more detail at the importance of confidence ranking and normalization and found that the choice of parameters plays a crucial role for the effectiveness of the resulting classifier. In addition, we illustrated for one domain what actual features are learned during self-training and thus proved that these features differ from the knowledge encoded in the rule-based classifier and that they are potentially much more expressive. Last but not least, we compared self-training with state-of-the-art statistical domain adaptation using labeled out-of-domain training data and found that on average self-training produces performance that is competitive to that type of algorithms.

9 Conclusion and future work

In this article, we examined the effectiveness of bootstrapping a supervised polarity classifier using the output of an open-domain rule-based classifier. The resulting self-trained classifier is usually significantly better than the open-domain classifier since the supervised classifier exploits in-domain features. As far as the choice of the feature set is concerned, the supervised classifier within self-training behaves very much like an ordinary supervised classifier. The set of all unigrams and bigrams performs best.

The type of rule-based classifier has an impact on the performance of the final classifier. To some extent, the more accurate the rule-based classifier is, the better the resulting self-trained classifier is. However, not all types of linguistic modeling that can be applied for polarity classification have the same impact. Word disambiguation with the help of part of speech, negation modeling and some ad-hoc heuristic weighting of polar expressions accounting for special contextual properties improve performance. Compositional rule-based polarity classification in which polarity is propagated from lexical units to larger linguistic units, on the other hand, has a more restricted impact. We only measured some improvement on sentence-level data. On such a fine-grained text level, however, the improvement caused by self-training is more limited as text instances are composed of fewer words (i.e., features).

A comparison to statistical domain adaptation revealed that self-training produces performance competitive with that of state-of-the-art domain adaptation.

Self-training also outperforms a rule-based classifier and a majority-class classifier in more difficult settings in which mixed reviews are part of the data set 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 classifier but, on average, using small labeled samples from the data collection produces more reliable results.

In future work, we would like to examine what impact a more fine-grained polarity lexicon assigning individual scores to polar expressions has on self-training. Moreover, we would like to extend the binary classification proposed in this article to a three-way classification in which apart from positive and negative polarity mixed polar reviews are not only part of the unlabeled training data but are also among the test data and consequently have to be explicitly modeled.

Acknowledgments This article is an extension of (Wiegand and Klakow 2010). We refer the reader to the section on related work for the full list of extensions. This work was funded by the German Federal Ministry of Education and Research (Software-Cluster) under grant no. "01IC10S01" and the Cluster of Excellence for Multimodal Computing and Interaction. The authors would like to thank Ivan Titov for running his statistical domain adaptation method on the data used in this article and Sabrina Wilske for insightful comments.

References

Akkaya, C., Wiebe, J., & Mihalcea, R. (2009). Subjectivity word sense disambiguation. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)* (pp. 190–199). Singapore.

Akkaya, C., Wiebe, J., Conrad, A., & Mihalcea, R. (2011). Improving the impact of subjectivity word sense disambiguation on contextual opinion analysis. In *Proceedings of the conference on computational natural language learning* (pp. 87–96). Portland, OR, USA.

Andreevskaia, A., & Bergler, S. (2008). When specialists and generalists work together: Overcoming domain dependence in sentiment tagging. In *Proceedings of the annual meeting of the association for computational linguistics: Human language technologies (ACL/HLT)* (pp. 290–298), Columbus, OH, USA.

Aue, A., & Gamon, M. (2005). Customizing sentiment classifiers to new domains: A case study. In *Proceedings of recent advances in natural language processing (RANLP)*. Borovets, Bulgaria.

Baccianella, S., Esuli, A., & Sebastiani, F. (2010). SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the conference on language resources and evaluation (LREC)* (pp. 2200–2204). Valletta.

Balamurali, A., Joshi, A., & Bhattacharyya, P. (2011). Harnessing wordnet senses for supervised sentiment classification. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)* (pp. 1081–1091). Edinburgh, Scotland, UK.

Becker, I., & Aharonson, V. (2010). Last but definitely not least: On the role of the last sentence in automatic polarity-classification. In *Proceedings of the annual meeting of the association for computational linguistics (ACL)* (pp. 331–335). Uppsala, Sweden.

Beineke, P., Hastie, T., & Vaithyanathan, S. (2004). The sentimental factor: Improving review classification via human-provided information. In *Proceedings of the annual meeting of the association for computational linguistics (ACL)* (pp. 263–270). Barcelona, Spain.

Blitzer, J., McDonald, R., & Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)* (pp. 120–128). Sydney, Australia.

Blitzer, J., Dredze, M., & Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the annual meeting of the association for computational linguistics (ACL)* (pp. 440–447). Prague, Czech Republic.

Choi, Y., & Cardie, C. (2008). Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)* (pp. 793–801). Waikiki, HI, USA.

Council, I., McDonald, R., & Velikovich, L. (2010). What's great and what's not: Learning to classify the scope of negation for improved sentiment analysis. In *Proceedings of the workshop on negation and speculation in natural language processing* (pp. 51–59). Uppsala, Sweden.

Dasgupta, S., & Ng, V. (2009). Mine the easy, classify the hard: A semi-supervised approach to automatic sentiment classification. In *Proceedings of the joint conference of the annual meeting of the association for computational linguistics and the international joint conference on natural language processing of the Asian federation of natural language processing (ACL/IJCNLP)* (pp. 701–709). Suntec, Singapore.

Esuli, A., & Sebastiani, F. (2006). Dermining term subjectivity and term orientation for opinion mining. In *Proceedings of the conference on European chapter of the association for computational linguistics (EACL)* (pp. 193–200). Trento, Italy.

Esuli, A., & Sebastiani, F. (2007). PageRanking wordnet synsets: An application to opinion mining. In *Proceedings of the annual meeting of the association for computational linguistics (ACL)* (pp. 424–431). Prague, Czech Republic.

Hatzivassiloglou, V., & McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of the conference on european chapter of the association for computational linguistics (EACL)* (pp. 174–181). Madrid, Spain.

Hatzivassiloglou, V., & Wiebe, J. (2000). Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the international conference on computational linguistics (COLING), Saarbrücken* (pp. 299–305). Germany.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery & data mining (KDD)* (pp. 168–177). Seattle, WA, USA.

Huang, Y., & Lowe, H. J. (2007). A novel hybrid approach to automated negation detection in clinical radiology reports. *Journal of the American Medical Infomatics Association (JAMIA), 14*, 304–311.

Jason, G. (1988). Hedging as a fallacy of language. *Informal Logic, 10*(3), 169–175.

Jia, L., Yu, C., & Meng, W. (2009). The effect of negation on sentiment analysis and retrieval effectiveness. In *Proceedings of the conference on information and knowledge management (CIKM)* (pp. 1827–1830). Singapore.

Joachims, T. (1999). Making large-scale SVM learning practical. In: Schölkopf, B., Burges, C., Smola, A. (Eds.) *Advances in Kernel methods—Support vector learning*. Cambridge MA: MIT Press.

Kennedy, A., & Inkpen, D. (2006). Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence, 22*(2), 110–125.

Klenner, M. (2009). Süsse Beklommenheit und schmerzvolle Ekstase. Automatische Sentimentanalyse in den Werken von Eduard von Keyserling. In *Proceedings of the biennial GSCL-conference* (pp. 91–97). Potsdam, Germany.

Klenner, M., Fahrni, A., & Petrakis, S. (2009a). PolArt: A robust tool for sentiment analysis. In *Proceedings of the nordic conference on computational linguistics (NoDaLiDa)* (pp. 235–238). Odense, Denmark.

Klenner, M., Petrakis, S., & Fahrni, A. (2009b). Robust Compositional Polarity Classification. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)* (pp. 180–184). Borovets

Melville, P., Gryc, W., & Lawrence, R. D. (2009). Text analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining (KDD)* (pp. 1275–1283). Paris, France.

Min, H. J., & Park, J. C. (2011). Detecting and blocking false sentiment propagation. In *Proceedings of the international joint conference on natural language processing (IJCNLP)* (pp. 354–362). Chian Mai, Thailand.

Moilanen, K., & Pulman, S. (2007). Sentiment construction. In *Proceedings of recent advances in natural language processing (RANLP)*. Borovets, Bulgaria.

Morante, R. (2010). Descriptive analysis of negation cues in biomedical texts. In *Proceedings of the conference on language resources and evaluation (LREC)* (pp. 1429–1436). Valletta, Malta.

Neviarouskaya, A., Prendinger, H., & Ishizuka, M. (2009). Semantically distinct verb classes involved in sentiment analysis. In *Proceedings of the IADIS international conference on applied computing* (pp. 27–34). Rome, Italy.

Ng, V., Dasgupta, S., & Arifin, S. M. N. (2006). Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *Proceedings of the international conference on computational linguistics and annual meeting of the association for computational linguistics (COLING/ACL)* (pp. 611–618). Sydney, Australia.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)* (pp. 79–86). Philadelphia, PA, USA.

Polanyi, L., & Zaenen, A. (2006). Contextual valence shifters. In *Computing attitude and affect in text: Theory and applications* (pp. 1–10). Springer.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130–137.

Prabowo, R., & Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics, 3*(1), 143–157.

Qiu, L., Zhang, W., Hu, C., & Zhao, K. (2009). SELC: A self-supervised model for sentiment classification. In *Proceedings of the conference on information and knowledge management (CIKM)* (pp. 929–936). Hong Kong, China.

Rao, D., & Ravichandran, D. (2009). Semi-supervised polarity lexicon induction. In *Proceedings of the conference on European chapter of the association for computational linguistics (EACL)* (pp. 675–682). Athens, Greece.

Salvetti, F., Reichenbach, C., & Lewis, S. (2006). Opinion Polarity Identification of Movie Reviews. In Shanahan, J. G., Qu, Y., & Wiebe, J. (Eds.) *Computing Attitude and Affect in Text: Theory and Applications* (pp. 303–316). Berlin: Springer.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the international conference on new methods in language processing* (pp. 44–49). Manchester, UK.

Shaikh, M. A. M., Prendinger, H., & Ishizuka, M. (2007). Assessing sentiment of text by semantic dependency and contextual valence analysis. In *Proceedings of the international conference of affective computing and intelligent interface (ACII)*(pp. 191–202). Lisbon, Portugal: Springer.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics, 37*(2), 267–307.

Tan, S., Wang, & Y., Cheng, X. (2008). Combining learn-based and lexicon-based techniques for sentiment detection without using labeled examples. In *Proceedings of the ACM special interest group on information retrieval (SIGIR)* (pp. 744–745). Singapore.

Tan, S., Cheng, X., Wang, Y., & Xu, H. (2009). Adapting naive bayes to domain adaptation for sentiment analysis. In *Proceedings of the European conference in information retrieval (ECIR)* (pp. 337–349). Toulouse, France.

Titov, I. (2011). Domain adaptation by constraining inter-domain variability of latent feature representation. In *Proceedings of the annual meeting of the association for computational linguistics (ACL)* (pp. 62–71). Portland, OR, USA.

Turney, P., & Littman, M. (2003). Measuring praise and criticism: Inference of semantic orientation from association. In *Proceedings of ACM transactions on information systems (TOIS)* (pp. 315–346).

Velikovich, L., Blair-Goldensohn, S., Hannan, K., & McDonald, R. (2010). The viability of web-derived polarity lexicons. In *Proceedings of the human language technology conference of the North American chapter of the ACL (HLT/NAACL)* (pp. 777–785). Los Angeles, CA, USA.

Wiebe, J., & Mihalcea, R. (2006). Word sense and subjectivity. In *Proceedings of the international conference on computational linguistics and annual meeting of the association for computational linguistics (COLING/ACL)* (pp. 1065–1072). Syndney, Australia.

Wiebe, J., & Riloff, E. (2005). Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of the international conference on intelligent text processing and computational linguistics (CICLing)* (pp. 486–497). Mexico City, Mexico.

Wiegand, M., & Klakow, D. (2009a). Predictive features in semi-supervised learning for polarity classification and the role of adjectives. In *Proceedings of the Nordic conference on computational linguistics (NoDaLiDa)* (pp. 198–205). Odense, Denmark.

Wiegand, M., & Klakow, D. (2009b). The role of knowledge-based features in polarity classification at sentence level. In *Proceedings of the international FLAIRS conference (FLAIRS)* (pp. 296–301). Sanibel Island, FL, USA.

Wiegand, M., & Klakow, D. (2010). Bootstrapping supervised machine-learning polarity classifiers with rule-based classification. In *Proceedings of the workshop on computational approaches to subjectivity and sentiment analysis (WASSA)* (pp. 59–66). Lisbon, Portugal.

Wiegand, M., Balahur, A., Roth, B., Klakow, D., & Montoyo, A. (2010). A survey on the role of negation in sentiment analysis. In *Proceedings of the workshop on negation and speculation in natural language processing* (pp. 60–68). Uppsala, Sweden.

Wilson, T., Wiebe, J., & Hoffmann, P. (2009). Recognizing contextual polarity: An exploration for phrase-level analysis. *Computational Linguistics, 35*, 399–433.