Context-aware Sentiment Detection From Ratings

Yichao Lu, Ruihai Dong, Barry Smyth

Abstract The explosion of user-generated content, especially tweets, customer reviews, makes it possible to build sentiment lexicons automatically by harnessing the consistency between the content and its accompanying emotional signal, either explicitly or implicitly. In this work we describe novel techniques for automatically producing domain specific sentiment lexicons that are optimised for the language patterns and idioms of a given domain. We describe how we use review ratings as sentiment signals. We also describe an approach to recognising contextual variations in sentiment and show how these variations can be exploited in practice. We evaluate these ideas in a number of different product domains.

1 Introduction

Sentiment analysis and opinion mining techniques aim to identify, analyse, and understand the subjective information in textual material provided by users. They have become increasingly popular when it comes to harness the explosion of user-generated content from short tweets and status updates to more detailed customer reviews. For example, [7] mine positive and negative sentiment from news articles about stocks and uses this to visualise market sentiment and pricing information for the user. Likewise Tumasjan et al. [19] demonstrated the potential to predict election outcomes by extracting the political sentiment contained within tweets. Elsewhere [15] describes how to help users compare different products by summarizing sentiment information across different product features from customer reviews. And Dong et al. propose an opinionated product recommendation technique that com-

Yichao Lu
Fudan University, e-mail: luyc13@fudan.edu.cn

Ruihai Dong, Barry Smyth
Insight Centre for Data Analytics, University College Dublin
e-mail: [Ruihai.Dong, Barry.Smyth]@insight-centre.org

bines similarity and sentiment to estimate review ratings and generate more effective product recommendations[5, 4].

The requirement to handle huge amounts of informal, opinionated, multi-domain, user-generated content has proven to be a challenge for traditional sentiment analysis techniques, which are often based on static dictionaries [10], fixed rules [9] or labeled data [14, 6]. In particular, such static approaches seldom deal with the common idioms that emerge across different domains. And even within a single domain different terms can signal very different sentiment extremes depending on context. For instance, in tablet reviews, the term *high* has positive polarity when modifying the aspect *quality*, but has negative polarity when modifying the aspect *price*.

These challenges speak to the need for a more flexible and dynamic approach to sentiment analysis, perhaps one that can be based on the common language patterns associated with a target domain. One way forward is to harness the sentiment signals (implicit or explicit) that are often associated with user-generated content. For example, users tend to use emoticon symbols such as *:), :(,* or hashtags (#cool, #fail etc.), to express emotional intent. These signals can be used to enhance sentiment learning algorithms; see[2, 16, 8, 17]. For instance, Davidov et al. [2] proposed a supervised classification techniques by using 50 Twitter tags and 15 'smileys' as sentiment labels to avoid intensive manual annotation. Liu et al. [16] present a model (ESLAM) by utilizing both manually labeled data and emoticons to train and smooth for twitter sentiment analysis.

For shorter forms of user-generated content a guiding assumption is that positive sentiment words commonly have a closer relationship with positive emotional signals than with negative emotional signals, and vice versa; see for example [11]. Kiritchenko et al. introduced an weakly surpvised approach for assessing the sentiment of short informal textual messages on Twitter [13]. Hashtags are used as an indicator of the sentiment polarity of the message, and they use a pointwise mutual information (PMI) technique to measure the relationship between the candidate sentiment words and the polarity of the tweets.

Longer forms of user-generated content, such as blogs or customer reviews, are less amenable to this use of local emotional signals: they are longer and more diverse than others, and typically containing multiple topics and expressing complex and varied sentiment. For example, "*The Thai red curry was delicious, ... , but price is $12, quite expensive*" expresses positive sentiment about the *Thai red curry*, but negative sentiment about *price*. Nevertheless, implicit indications of intent may be available. For example, Bross and Ehrig [1] exploit the information contained in pros and cons summaries as sentiment signals; the assumption is that authors choose positive expressions when describing a product aspect in the pros, whereas negative expressions are common in the cons. Wu and Wen [22] predict the sentiment of nouns by calculating their statistical associations with positive and negative search engine hits, which are obtained by using the search engine Baidu with positive and negative query-patterns.

Explicit ratings are also an important signal for reviews. A review's rating reflects the overall sentiment of a review without necessarily reflecting the sentiment of each individual review topic. For example, Kiritchenko et al [12] create an domain-

specific sentiment lexicon for restaurants by utilizing one-, two-star rating as negative signals and four-, five-star ratings as positive signals. But a more interesting challenge is if the ratings can be useful signals to build context-aware sentiment lexicons in these longer forms of user-generated content.

In this paper, we describe a set of novel techniques for automatically building context-sensitive, domain-specific sentiment lexicons. We also use review ratings as emotional signals but describe how context can also be leveraged. We evaluate these ideas in a number of different product domains. We compare our lexicons to conventional one-size-fits-all lexicons using data from TripAdvisor and Amazon and we evaluate the quality of our lexicons by comparing their performance to a number of baselines and state-of-the-art alternatives on a sentiment polarity prediction task.

2 Creating a Domain-Specific Sentiment Lexicon

We describe an approach to automatically constructing a domain specific sentiment lexicon given a set of user-generated reviews which can be reliably separated into positive and negative groupings. For the purpose of this paper we will focus on the domain on hotel reviews from TripAdvisor — and thus creating a travel-related lexicon — but many other domains are amenable to the approach described. To separate our reviews into positive and negative groups we use the corresponding review rating, focusing on 5-star (positive) and 1-star (negative) reviews in this instance. As mentioned in the introduction the basic principle is that words that are frequent in positive reviews but infrequent in negative reviews are more likely to be positive, and vice versa; by comparing a word's positive and negative review frequency we can compute a fine-grained sentiment score.

2.1 Word-Level Sentiment Scoring

We begin with a collection of n positive (5-star) and negative (1-star) reviews and for each word w we calculate the number of times it occurs in positive ($pos(w)$) and negative ($neg(w)$) reviews. On the face of it the relative difference between $pos(w)$ and $neg(w)$ is an indication of the polarity of w. However, we observe that in many online review settings there can be significant differences between the length of positive and negative reviews thus leading to the need for some sort of normalisation approach.

For example, the average length of negative reviews on TripAdvisor is 191 words, significantly higher than the 120 words for the average positive review. Given that we ensure an equal number of positive and negative reviews in our training set we can apply a simple normalisation procedure by dividing word counts by the total review lengths (POS and NEG for positive and negative reviews, respectively) as in Equations 1 and 2.

$$freq^{+}(w) = \frac{pos(w)}{POS} \tag{1}$$

$$freq^{-}(w) = \frac{neg(w)}{NEG} \tag{2}$$

Next we calculate the *polarity difference* (or *PD*) of a word as the relative difference between its positive and negative frequency; see Equation 3. Thus, a word that is more likely to appear in positive reviews than in negative reviews will have a positive *PD* value, whereas a word that is more likely to be in negative reviews will have a negative *PD* value. *PD* values close to 0 indicate neutral polarity or sentiment whereas PD values at the extremes of -1 or +1 indicate strong negative or positive polarity, respectively.

$$PD(w) = \frac{freq^{+}(w) - freq^{-}(w)}{freq^{+}(w) + freq^{-}(w)} \tag{3}$$

Finally, we compute the sentiment score of a word ($sent(w)$) as the signed square of its PD value; see Equation 4 where $sign(x)$ returns -1 if $x < 0$ or +1 otherwise. The effect of this transformation is to relatively amplify differences between PD values that are non-neutral.

$$sent(w) = PD(w)^2 * sign(PD(w)) \tag{4}$$

2.2 Dealing with Negation

The approach discussed so far does not consider the issue of negation when it comes to evaluating sentiment. For example the polarity of the sentiment of a word can completely change when it is used in a negative context; for instance the sentiment of a highly positive words like "terrific" is almost completely reversed when used in a negative context. For example, "*the checkin experience was far from terrific*". However dealing with negation is far from straightforward and considerable attention has been paid to this aspect of sentiment analysis; see [20].

In this work, we deal with negation by using a dictionary of common negative terms (e.g. "not") and phrases (e.g. "far from") and look for these terms within a *k_word* distance of a given sentiment word *w*, within the same sentence. When a negative term is associated in this way with a sentiment word *w* then we assume *w* is in a negated context. In order to deal with this negative context we maintain two forms of *w*. When *w* is found in a regular (non negated) context it is dealt with in the normal way as described above. However when *w* is found in a negated context we maintain a separate count for its negated form, which we label and record as *not_w*.

For example, we may find the word "perfect" occurring in many positive reviews but fewer negative reviews and as such it will be assigned a positive $sent(w)$ value according to Equation 4. However, we may also find reviews which mention features

that are “not perfect” or “far from perfect”. In these reviews the sentiment word (“perfect”) occurs in a negated context but the counts of these occurrences are stored with reference to $not\_perfect$ allowing for an independent sentiment calculation for the negated form, $sent(not\_perfect)$.

So as to filter out noise, we set a threshold t of the minimum occurrence. Words that have been found less than t times in the training set will be dropped from the lexicon. The setting of a threshold value guarantees that opinion words included by the lexicon are indeed high-frequency words in the domain and also makes up for the deficiency our sentiment score formula has when dealing with words that only appear a few times in the training set.

3 Creating a Context-Sensitive Sentiment Lexicon

So far we have described an approach to constructing a domain-specific sentiment lexicon; one that captures idiomatic differences between different domains and so provides a more accurate account of sentiment. But this still does not deal with sentiment shifts that can occur at much finer levels of granularity, within a domain. For example describing a level of service as“low” is clearly negative but a “low” price is typically positive. To deal with this we need a context-sensitive lexicon that documents how words assume different sentiment in different contexts and, for example, when applied to different features.

To construct a context-sensitive sentiment lexicon we adopt a similar overall approach to the construction of a domain-specific sentiment lexicon. We begin with two sets of positive and negative reviews, count word occurrences within each of these sets, and deal with negation effects. But this time we document word occurrences with respect to individual *feature words* and keep account of sentiment words that are linked to specific features in positive and negative reviews.

Feature words are assumed to be nouns that occur with above average frequency; as such they are the common aspects that users tend to discuss in a given review domain (e.g. ”restaurant” in a hotel review or ”display” in a laptop review). To identify feature modifying opinion words (sentiment words) we begin by recognising within-sentence part-of-speech patterns; that is, POS tag sequences with a feature word at one end and a sentiment word on the other. We assume there to be a certain number of fixed collocations when sentiment words are used to modify features. Such collocations can be captured by the corresponding POS tag sequences so that frequent part-of-speech patterns can be employed to identify the relationship between features and opinion words. When looking for patterns, we focus on those that indicate sentiment-features relations.

But at the beginning, for a given feature in a sentence, we neither have its corresponding sentiment words nor POS patterns. Thus the domain-specific lexicon is employed to provide a set of candidate sentiment words whose absolute value of polarity difference is no less than 0.3. We assume that these nearby words with strong polarity difference is used to modify the given feature. Then we scan for and count

the sentiment-feature patterns in the training set and mark those with the number of occurrence above the average as valid patterns. Afterwards, we can search for the feature modifying sentiment words by matching the valid patterns. For each sentiment-feature word pairing we count the number of times (w, f) co-occur in positive and negative reviews as $pos(w, f)$ and $neg(w, f)$, respectively. Then we calculate a relative frequency for each (w, f) pairing as in Equations 5 and 6, where $pos(f)$ and $neg(f)$ indicates the number of times the feature f occurs in positive and negative reviews.

$$freq^{+}(w, f) = \frac{pos(w, f)}{pos(f)} \tag{5}$$

$$freq^{-}(w, f) = \frac{neg(w, f)}{neg(f)} \tag{6}$$

Next we can calculate a feature-based polarity difference and corresponding sentiment score as in Equations 7 and 8, respectively.

$$PD(w, f) = \frac{freq^{+}(w, f) - freq^{-}(w, f)}{freq^{+}(w, f) + freq^{-}(w, f)} \tag{7}$$

$$sent(w, f) = PD(w, f)^2 * sign(PD(w, f)) \tag{8}$$

In this way, within a given domain, we create a sentiment lexicon in which each sentiment word w is linked to a co-occurring feature word f and associated with a sentiment score that reflects the relative likelihood of the (w, f) pairing occurring in positive or negative reviews. Once again, positive values for $(sent(w, f)$ indicate that w is more likely to co-occur with f in positive reviews, whereas negative values for $sent(w, f)$ mean that w tends to co-occur with f in negative reviews.

We can deal with negation in context-specific lexicon generation in a manner that is exactly analogous to the approach taken when generating domain-specific lexicons. Briefly, sentiment words that appear in a negated context are recorded with a "not_" prefix and their (feature-based) sentiment score are calculated independently from non-negated versions. The threshold of minimum occurrence is set to be the same as we did in the case of domain-independent lexicons.

4 Evaluation

At the beginning of this paper we highlighted two problems with conventional sentiment lexicons — the domain problem and the context problem — which spoke against a one-size-fits-all approach to sentiment labeling. We then went on to describe ways of coping with these problems by automatically generating domain-specific and context-sensitive lexicons from user-generated reviews. In this section we describe the results of a recent study to evaluate the effectiveness of these approaches on real-world datasets from different content domains.

4.1 Datasets

As the basis for our evaluation we collected a large number of reviews from TripAdvisor and Amazon. A summary description of these data is presented in Table 1. For example the TripAdvisor dataset contains a total of 867,644 reviews for hotels. The Amazon dataset includes 90,138 reviews for electronic products in 6 categories (Digital Cameras, GPSes, Laptops, Phones, Printers, and Tablets). Reviews in both datasets are rated on a scale of 1 to 5.

Table 1 Statistics of the TripAdvisor and Amazon review datasets.

	Tripadvisor		Amazon	
Rating	#reviews	Avg. Len	#reviews	Avg. Len
1	33,646	191	14,708	143
2	43,372	186	6,703	190
3	112,235	161	18.962	204
4	272,077	133	18,962	208
5	406,314	120	42,100	163

We need a source of user-generated reviews as training data to produce our lexicons, and as test data to evaluate the results. For each dataset we construct our *training set* from a random sampling of the rating-1 (negative) and rating-5 (positive) reviews; in the case of TripAdvisor we select 30,000 positive and negative reviews and for Amazon we select 12,000 each. The remaining reviews (regardless of rating) are used as a source of test data and we randomly select 2,000 from each of the five rating levels for the *test set*.

As a baseline against which to evaluate our algorithmically generated lexicons we choose the popular lexicon Bing Liu opinion lexicon [10]; we will refer to this as *BL* in what follows. This is one of the most widely used in the sentiment analysis literature and is a general purpose sentiment lexicon in which each opinion word is labelled as either positive of negative.

As a second baseline we also use the SO-CAL lexicon [18]. While Bing Liu's opinion lexicon only consists of lists of positive and negative words without sentiment scores, the SO-CAL lexicon differs from it in three respects: (1) The lexicon has been split by the POS tags; (2) The lexicon assigns a sentiment score (rather than a binary label) to each opinion word, and (3) The lexicon deals with negation in a similar manner of ours; i.e. with the "not_" prefix. In other words, the SO-CAL lexicon is structurally more similar to our domain-specific lexicons. We refer to the SO-CAL lexicon as *SOCAL* in what follows.

4.2 Domain-Specific Lexicon Coverage

To begin with, we generate domain-specific lexicons for the TripAdvisor and Amazon domains using the technique described Section 2 — we will refer to lexicons generated in this way as *DS* — and compare these lexicons to *BL*. We do this by comparing the *DS* and *BL* lexicons in terms of coverage (their term overlap and differences). Note that for the purpose of this evaluation we only consider terms in *BL* and *DS* that are also present in the test dataset.

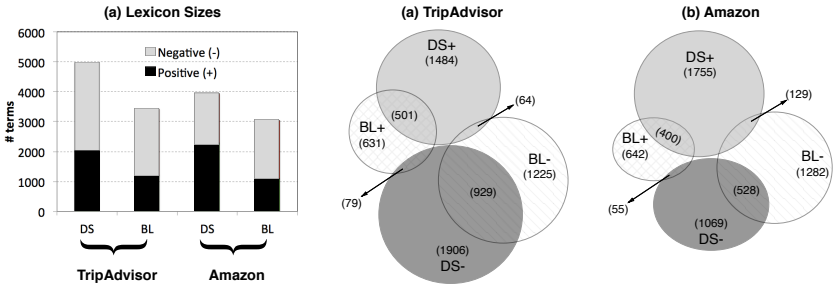


Fig. 1 A coverage analysis of the *DS* and *BL* lexicons for the TripAdvisor and Amazon datasets.

The results are presented in Figure 1 as bar charts of lexicon size and venn diagrams for the TripAdvisor and Amazon data. In each case we present 4 regions as follows:

1. *DS+*: the number of positive terms in the domain-specific lexicon;
2. *DS-*: the number of negative terms in the domain-specific lexicon;
3. *BL+*: the number of positive terms in Bing Liu's lexicon;
4. *BL-*: the number of negative terms in Bing Liu's lexicon.

There are a number of observations that can be made at this stage. First of all it is clear that the *DS* lexicons tend to be larger (overall and with respect to positive and negative terms) than *BL* lexicon. Figure 1 graphs the number of positive and negative terms in the *DS* and *BL* lexicons for the TripAdvisor and Amazon datasets. For example, the TripAdvisor *DS* lexicon includes 4,963 terms overall (2,049 positive terms and 2,914 negatives) compared to the 3,429 terms in the *BL* lexicon (1,211 positive and 2,218 negatives); similar proportions are evident for the Amazon data. Clearly there are many words in TripAdvisor and Amazon reviews, which can be linked to either positive or negative opinions, but that are absent from the more conventional *BL* lexicon.

A second observation is that there is a relatively small degree of overlap between the *DS* and *BL* lexicons. For TripAdvisor the total overlap is only 1,567 terms and for Amazon it is only 1,112 terms. Perhaps more significantly, there are also large number of terms in each lexicon that are absent from the other. For example, the TripAdvisor *DS* lexicon includes 1906 positive and 1484 negative terms that are

unique to it, that is absent from the *BL* lexicon. These are terms that are commonly associated with positive or negative hotel reviews but are not typically included in conventional lexicons. A good example of such terms includes words like *manager* or *carpets* in the hotel domain. In each case the term is commonly associated with a negative review; travellers rarely talk about a hotel manager in a review unless it is negative and when they mention a hotel's carpets it is almost always to complain about their lack of cleanliness. On the other hand the 631 positive and 1,225 negative terms that are unique to *BL* (and missing from *DS*) simply do not occur as common sentiment words in hotel reviews. We note a very similar pattern for the Amazon data, although obviously many of the terms will be different.

Thirdly, while both lexicons agree on the polarity of the majority of their overlapping terms there are also overlapping terms where there is clear disagreement. For instance, in TripAdvisor we see that 79 terms that are classified as negative by *DS* but that are listed as positive in the *BL* lexicon; words like "portable" usually have a negative connotation in a hotel review (for example, guests often complain about a portable TV) but *BL* classifies it as a positive word. Conversely there are 64 terms that are classified as positive by *DS* but that are considered to be negative by *BL*. For example, words such as "extravagant" or "sunken" are considered to be negative by *BL* but are invariably positive in hotel reviews. Once again we see a similar pattern in the Amazon results.

So far we have demonstrated that our approach to generating domain-specific lexicons has the potential to identify a greater variety of sentiment words within a corpus of positive and negative user-generated reviews; it is worth remembering that these lexicons include the same terms as our context-sensitive lexicons. While some of these words match those in conventional sentiment lexicons, a majority do not. And for those that are found in conventional sentiment lexicons, some are found to have conflicting sentiment polarity according to our domain-specific approach. Of course none of this speaks to the quality of the *DS* lexicons nor have we considered the impact of adding context-sensitivity. Thus the next section we consider a more objective test of lexicon quality and consider these matters more fully.

4.3 Sentiment Polarity Prediction

As an independent test of lexicon quality we propose to use review content and sentiment data to automatically predict the overall polarity of the review. To do this we represent each review as a sentiment vector and train classifiers to predict the overall review polarity. We will evaluate a number of different conditions by varying the lexicons and the classifiers. We will also separately evaluate the influence of dealing with negation and then benefits of our context-sensitive approach.

4.3.1 Setup

To begin with, the testing data from the previous analysis will be used as our overall dataset for this evaluation. This is necessary because our previous training data is used to build the lexicons and we need to ensure that it does not participate in this experiment's classifier training or testing. For the purpose of this classification task we label the 1-star and 2-star rated reviews (4,000 in all in each domain) as negative and the 4-star and 5-star reviews as positive (again 4,000 review in all).

In terms of lexicons we will compare our *DS* approach to *BL* and *SOCAL* as examples of conventional one-size-fits-all lexicons. In addition we will include also include another alternative which we will designate as *PMI*, based on the work of [13]. The significance of this *PMI* lexicon is that it is generated in a similar manner to our *DS* approach, but uses a mutual information score to estimating the sentiment polarity of words. Thus it provides baseline domain-specific lexicon.

In terms of classification approach we will consider Bayes Nets (*BN*) [21], a rules-based approach in the form of *JRip* [21], and random forests (*RF*) [21]. We will also compare the classifier performance with and without any negation testing, as per Section 2. The implementation of negation for *BL* is carried our by simply switching the original polarity of sentiment in terms of negation. And finally, we will consider the impact of our context-sensitive approach (Section 3) by applying it to the *DS* and *PMI* lexicons. Applying our approach to *PMI* is analogous to the way in which we describe context-sensitivity in Section 3 as the the mutual information scores can be separately calculated with respect to individual, in-scope feature words. However, the conventional binary sentiment lexicon *BL* is not amenable to this context-sensitivity approach.

4.3.2 Methodology

Our basic classification task begins by representing each review as a simple feature vector. To do this we classify each word in the review by its part-of-speech tag (adjective, adverb, noun, verb) and sum the sentiment scores (-1 or +1 for *BL* versus real-valued scores for the other lexicons). Thus each review is represented by a 4-element sentiment vector and associated with a binary class label (positive or negative). We perform a 10-fold cross-validation, splitting the data into training and testing sets, build the appropriate classifiers with the training data, and evaluate them against the test data. In this experiment, we use WEKA [1] to run the classifications. In what follows we focus on presenting f-measure scores for each condition.

[1] http://www.cs.waikato.ac.nz/ml/weka/

Table 2 Precision, Recall, F-measure for TripAdvisor and Amazon dataset; The differences between domain specific, context sensitive results are significant at the 0.05 level.

		no negation			with negation			negation + context		
	Method	Pre.	Recall	F-measure	Pre.	Recall	F-measure	Pre.	Recall	F-measure
Tripadvisor	DS-BN	0.935	0.935	0.934	0.946	0.946	0.946	**0.954**	**0.954**	**0.954**
	PMI-BN	0.933	0.933	0.933	0.941	0.941	0.941	0.949	0.949	0.949
	BL-BN	0.857	0.856	0.856	0.882	0.882	0.881	-	-	-
	SOCAL-BN	0.837	0.837	0.837	0.858	0.858	0.858	-	-	-
	DS-JRip	0.928	0.928	0.928	0.942	0.942	0.942	0.948	0.948	0.948
	PMI-JRip	0.93	0.93	0.93	0.938	0.938	0.938	0.945	0.945	0.945
	BL-JRip	0.853	0.853	0.853	0.882	0.882	0.881	-	-	-
	SOCAL-JRip	0.842	0.842	0.842	0.862	0.862	0.862	-	-	-
	DS-RF	0.932	0.932	0.932	0.943	0.943	0.943	0.95	0.95	0.95
	PMI-RF	0.927	0.927	0.927	0.941	0.941	0.941	0.949	0.949	0.949
	BL-RF	0.839	0.839	0.838	0.868	0.867	0.867	-	-	-
	SOCAL-RF	0.814	0.814	0.814	0.845	0.845	0.845	-	-	-
Amazon	DS-BN	0.82	0.82	0.82	0.853	0.853	0.853	**0.861**	**0.861**	**0.861**
	PMI-BN	0.819	0.818	0.817	0.846	0.846	0.845	0.857	0.857	0.856
	BL-BN	0.726	0.726	0.726	0.762	0.762	0.762	-	-	-
	SOCAL-BN	0.703	0.703	0.703	0.73	0.73	0.73	-	-	-
	DS-JRip	0.801	0.801	0.801	0.847	0.847	0.847	0.859	0.859	0.859
	PMI-JRip	0.802	0.802	0.801	0.839	0.839	0.839	0.854	0.854	0.854
	BL-JRip	0.738	0.738	0.738	0.76	0.76	0.76	-	-	-
	SOCAL-JRip	0.715	0.715	0.715	0.731	0.731	0.731	-	-	-
	DS-RF	0.802	0.799	0.799	0.848	0.848	0.848	0.859	0.859	0.859
	PMI-RF	0.797	0.793	0.793	0.84	0.84	0.84	0.851	0.85	0.85
	BL-RF	0.709	0.709	0.708	0.743	0.743	0.743	-	-	-
	SOCAL-RF	0.678	0.678	0.678	0.707	0.707	0.707	-	-	-

4.3.3 Results

The results are presented in Table 2 for both the TripAdvisor data and the Amazon data. In the first column we list the basic classification techniques used based on the underlying sentiment lexicon and the classification algorithm; for example, $DS-BN$ refers to our domain-specific lexicon with a Bayes Net classifier. The next three columns present the precision (recall, f-measure) scores classifiers built using *no negation*, *with negation*, and *negation + context*.

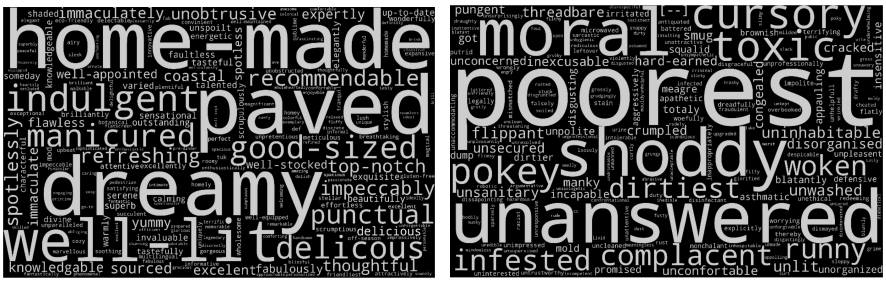
4.3.4 Discussion

We can see that the domain-specific lexicons (*DS* and *PMI*) perform very similarly, with f-measure scores around 0.93, significantly higher than the performance of the one-size-fits-all *BL* and *SOCAL* lexicons which has an average f-measure in the range of 0.81 to 0.85. We can also see that by explicitly dealing with negation during lexicon generation we can improve our classification performance overall. Moreover, by incorporating context into the *DS* and *PMI* lexicons – we cannot eas-

ily apply this approach to the *BL* and *SOCAL* lexicon – we can further improve classification performance, albeit more marginally in the case of TripAdvisor data.

The results for the Amazon domain are broadly similar although this time we see that *DS* performs marginally better than *PMI* across the various conditions. The introduction of negation and context also drives a greater f-measure improvement (compared with TripAdvisor), suggesting that negation and context play a more significant role in Amazon's reviews. That being said, negation has little or no effect when applied to the *BL* lexicon.

In this analysis we have focused on the f-measure results, but similar effects have been noted for precision and recall measures also. Overall it is clear that there is a significant classification performance benefit when we compare the domain specific, context sensitive lexicons to the more traditional one-size-fits-all lexicons. For example, according to Table 2 the best performing TripAdvisor condition is for *DS* using a Bayes Net classifier with negation and context to produce the f-measure score of approximately 0.954. By comparison the corresponding score for *BL* arise only 0.881. Likewise, in the Amazon domain, we see best performance for *DS* using a Bayes Net classifier with f-measure score of about 0.861, as compared with only 0.762 for *BL*. These differences — between domain specific, context sensitive results and *BL* and *SOCAL* results — were were found to be significant at the 0.05 level using a paired t-test [3].



(a) Positive. (b) Negative.

Fig. 2 Domain-specific adjectives and adverbs from the TripAdvisor dataset.

To make more concrete, we also generate word-clouds for the *DS* lexicons for TripAdvisor and Amazon datasets in order to evaluate the quality of the lexicons intuitively. For reasons of space, we only display domain-specific adjectives and adverbs from the TripAdvisor dataset in Figure 2 and context-aware sentiment words in *screen* context from the Amazon dataset in Figure 3. In these figures, size represents the sentiment strength, and brightness represents the frequency. From Figure 2, we can see that *home-made*, *paved*, *dreamy*, *well-lit*, *good-sized*, etc. have strong positive polarity, but *poorest*, *shoddy*, and *unanswered* etc. have strong negative polarity. From Figure 3, we can see that *sharp* is often used to describe *screen* and has the strongest positive polarity under *screen* context, however, it has strong neg-

(a) Positive. (b) Negative.

Fig. 3 Context-aware sentiment words in *screen* context from the Amazon dataset.

ative polarity when describing *corner of the table* in hotel reviews. We also see that *unresponsive*, *blank*, *bad* etc. have strong negative polarity for *screen*.

5 Conclusion

One-size-fits-all sentiment lexicons have proven to be problematic when dealing with idiomatic and contextual word-usage differences that are commonplace in user-generated content. Words that are reliably positive in one setting can have negative sentiment elsewhere, and vice versa. In this paper we have described a novel technique for automatically building context-sensitive, domain-specific sentiment lexicons by employing review ratings as signals. Evaluation results base don TripAdvisor and Amazon review data have demonstrated the effectiveness of these lexicons compared to more conventional baselines. Our domain-specific lexicons capture many new sentiment terms compared to conventional lexicons and by attending to context they are better able to deal with the occasional sentiment shifts that can occur within domains.

Acknowledgements This work is supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289.

References

1. Bross, J., Ehrig, H.: Automatic construction of domain and aspect specific sentiment lexicons for customer review mining. In: Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, pp. 1077–1086. ACM (2013)
2. Davidov, D., Tsur, O., Rappoport, A.: Enhanced sentiment learning using twitter hashtags and smileys. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING '10, pp. 241–249. Association for Computational Linguistics, Stroudsburg, PA, USA (2010). URL http://dl.acm.org/citation.cfm?id=1944566.1944594

3. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural computation **10**(7), 1895–1923 (1998)
4. Dong, R., OMahony, M.P., Smyth, B.: Further experiments in opinionated product recommendation. In: Proceedings of the 22nd International Conference on Case-Based Reasoning, ICCBR '14, pp. 110–124. Springer (2014)
5. Dong, R., Schaal, M., O'Mahony, M.P., Smyth, B.: Topic extraction from online reviews for classification and recommendation. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI '13. AAAI Press, Menlo Park, California (2013)
6. Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: Proceedings of LREC, vol. 6, pp. 417–422. Citeseer (2006)
7. Feldman, R., Rosenfeld, B., Bar-Haim, R., Fresko, M.: The stock sonarsentiment analysis of stocks based on a hybrid approach. In: Twenty-Third IAAI Conference (2011)
8. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford **1**, 12 (2009)
9. Hatzivassiloglou, V., McKeown, K.R.: Predicting the semantic orientation of adjectives. In: Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the european chapter of the association for computational linguistics, pp. 174–181. Association for Computational Linguistics (1997)
10. Hu, M., Liu, B.: Mining opinion features in customer reviews. In: Proceedings of the 19th National Conference on Artifical Intelligence, AAAI'04, pp. 755–760. AAAI Press (2004). URL http://dl.acm.org/citation.cfm?id=1597148.1597269
11. Hu, X., Tang, J., Gao, H., Liu, H.: Unsupervised sentiment analysis with emotional signals. In: Proceedings of the 22nd international conference on World Wide Web, pp. 607–618. International World Wide Web Conferences Steering Committee (2013)
12. Kiritchenko, S., Zhu, X., Cherry, C., Mohammad, S.M.: Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 437–442 (2014)
13. Kiritchenko, S., Zhu, X., Mohammad, S.M.: Sentiment analysis of short informal texts. Journal of Artificial Intelligence Research pp. 723–762 (2014)
14. Liu, B.: Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies **5**(1), 1–167 (2012)
15. Liu, B., Hu, M., Cheng, J.: Opinion observer: Analyzing and comparing opinions on the web. In: Proceedings of the 14th International Conference on World Wide Web, WWW '05, pp. 342–351. ACM, New York, NY, USA (2005). DOI 10.1145/1060745.1060797. URL http://doi.acm.org/10.1145/1060745.1060797
16. Liu, K.L., Li, W.J., Guo, M.: Emoticon smoothed language models for twitter sentiment analysis. In: AAAI (2012)
17. Lu, Y., Castellanos, M., Dayal, U., Zhai, C.: Automatic construction of a context-aware sentiment lexicon: An optimization approach. In: Proceedings of the 20th International Conference on World Wide Web, WWW '11, pp. 347–356. ACM, New York, NY, USA (2011). DOI 10.1145/1963405.1963456. URL http://doi.acm.org/10.1145/1963405.1963456
18. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Computational linguistics **37**(2), 267–307 (2011)
19. Tumasjan, A., Sprenger, T.O., Sandner, P.G., Welpe, I.M.: Predicting elections with twitter: What 140 characters reveal about political sentiment. ICWSM **10**, 178–185 (2010)
20. Wiegand, M., Balahur, A., Roth, B., Klakow, D., Montoyo, A.: 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. Association for Computational Linguistics (2010)
21. Witten, I., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann (2005)
22. Wu, Y., Wen, M.: Disambiguating dynamic sentiment ambiguous adjectives. In: Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10, pp. 1191–1199. Association for Computational Linguistics, Stroudsburg, PA, USA (2010). URL http://dl.acm.org/citation.cfm?id=1873781.1873915