Building Domain Specific Sentiment Lexicons Combining Information from Many Sentiment Lexicons and a Domain Specific Corpus

Hugo Hammer, Anis Yazidi, Aleksander Bai, Paal Engelstad

Building domain specific sentiment lexicons combining information from many sentiment lexicons and a domain specific corpus

Hugo Hammer, Anis Yazidi, Aleksander Bai and Paal Engelstad

Oslo and Akershus University College of Applied Sciences
Department of Computer Science
hugo.hammer|anis.yazidi|aleksander.bai|paal.engelstad@hioa.no

Abstract. Most approaches to sentiment analysis requires a sentiment lexicon in order to automatically predict sentiment or opinion in a text. The lexicon is generated by selecting words and assigning scores to the words, and the performance the sentiment analysis depends on the quality of the assigned scores. This paper addresses an aspect of sentiment lexicon generation that has been overlooked so far; namely that the most appropriate score assigned to a word in the lexicon is dependent on the domain. The common practice, on the contrary, is that the same lexicon is used *without adjustments* across different domains ignoring the fact that the scores are normally highly sensitive to the domain. Consequently, the same lexicon might perform well on a single domain while performing poorly on another domain, unless some score adjustment is performed. In this paper, we advocate that a sentiment lexicon needs some further adjustments in order to perform well in a specific domain. In order to cope with these domain specific adjustments, we adopt a stochastic formulation of the sentiment score assignment problem instead of the classical deterministic formulation. Thus, viewing a sentiment score as a stochastic variable permits us to accommodate to the domain specific adjustments. Experimental results demonstrate the feasibility of our approach and its superiority to generic lexicons without domain adjustments.

Keywords : *bayesian decision theory, cross-domain, sentiment classification, sentiment lexicon*

1 Introduction

With the increasing amount of unstructured textual information available on the Internet, sentiment analysis and opinion mining have recently gained a groundswell of interest from the research community as well as among practitioners. In general terms, sentiment analysis attempts to automate the classification of text materials as either expressing positive sentiment or negative sentiment. Such classification is particularity interesting for making sense of huge amount

of text information and extracting the "word of mouth" from different domains like product reviews, movie reviews, political discussions etc.

There are two main approaches to sentiment classification

- *Sentiment lexicon:* A sentiment lexicon is merely composed of sentiment words and sentiment phrases (idioms) characterized by sentiment polarity, positive or negative, and by sentimental strength. For example, the word 'excellent' has positive polarity and high strength whereas the word 'good' is also positive but has a lower strength. Once a lexicon is built and in place, a range of different approaches can be deployed to classify the sentiment in a text as positive or negative. These approaches range from simply computing the difference between the sum of the scores for the positive lexicon and the sum of the scores for the negative lexicon, and subsequently classifying the sentiment in the text according to the sign of the difference.
- *Supervised learning:* Given a set of documents with known sentiment class, the material can be used to train a model to classify the sentiment class of new documents.

A major challenge in sentiment classification is that the classification method normally is highly sensitive to the domain. A method that performs well in one domain, may not perform well in a different domain. It is worth mentioning that the later problem is common and well studied in the field of Machine Learning, since supervised learning is especially sensitive to the domain, and typically it performs well only in the domain of the annotated documents. The later problem is referred to in the literature as cross-domain classification.

Several methods have been suggested to overcome this challenge in the field of sentiment analysis. However, they are merely inspired by the legacy research on cross-domain classification in the field of machine learning. These methods are often referred to as cross-domain sentiment classification [1]. The premises of these methods is to adjust a supervised classifier to the domain of interest. The approaches consist of either using a small annotated corpus or, alternatively, a large non-annotated corpus from the domain of interest [2–4].

In this paper we study another problem, which is very common in practice, but to the best of our knowledge has not been studied in the literature. For many languages several different sentiment lexicons are available, and it is often difficult to know which sentiment lexicon is preferable. Ideally one would like to use the information from all the lexicons, but this is often challenging since the scores of a sentiment word varies between the lexicons and may also be contradictory. In addition there is usually also a large amount of text from the domain of interest, e.g. a large set of product reviews that we want to classify with respect to sentiment. We present a method that builds domain specific sentiment lexicons using information from the sentiment lexicons and the corpus from the domain of interest in an advantageous way. The suggested method is based on Bayesian decision theory.

Before, we proceed to presenting our solution and our experimental results, we shall present a brief review of the related work. Most of the research within

cross domain sentiment classification focuses on devising approaches to join information from labelled and/or unlabelled corpuses from different domains and the domain of interest to improve sentiment classification.

Bollegale et al. [5] argue that a major challenge of applying a classifier trained on one domain to another is that features may be quite different in different domains. The authors suggest to develop a sentiment sensitive thesaurus to expand the number of features in both the training and test sets.

Pan et al. [6] consider the case with unlabelled data in the domain of interest and labelled data from an other domain. To bridge the gap between the domains, the authors propose a spectral feature alignment algorithm to align domain-specific words from different domains into unified clusters, with the help of domain independent words as bridges.

Chetviorkin and Loukachevitch [7] propose a statistical features based approach in order to discriminate sentiment words in different domains do develop domain specific sentiment lexicons. The method requires labeled corpuses from both the domain of interest and the other domains.

Contextual sentiment lexicons takes the context of the sentiment words into account. Such lexicons are usually even more sensitive to the domain than ordinary sentiment lexicons are. Gindl et al. [8] suggest a method that identifies unstable contextualizations and refines the contextualized sentiment dictionaries accordingly, eliminating the need for specific training data for each individual domain.

In [9], the authors identified words that exhibit dis-ambiguity based on cross-domain evaluations. In simple terms, if a word gets a positive score in a domain with high confidence and a negative score in another domains, then this terms is considered dis-ambiguous. The next step was to create a domain-independent lexicon by simply excluding the words which are dis-ambiguous across domains. In [10], a taxonomy is used to determine the domain such as movies, politics, sports, then the different lexicons are learned on a domain basis. However, the authors did not discuss adjusting the scores across domains.

2 Joining information from sentiment lexicons and domain specific corpus

Our method consists of two parts. First we join the information from the sentiment lexicons, and second we adjust this information using the domain specific corpus.

2.1 Posterior expected sentiment score

We assume that we have a total of n_L sentiment lexicons consisting of a total of n_W sentiment words occurring in at least one of the sentiment lexicons. We denote the sentiment words $w_1, w_2, \ldots, w_{n_W}$. Let $s_{i,i(j)},\ i = 1, \ldots, n_W,\ j = 1, \ldots, |s_i|$ denote the sentiment score for sentiment word w_i in sentiment lexicon $i(j) \in \{1, 2, \ldots n_L\}$. $|s_i|$ denotes the number of lexicons that word i occurs in, while $i(1), i(2), \ldots, i(|s_i|)$ are references to these lexicons. Naturally

$|s_i| \leq n_L,\ i = 1, 2, \ldots, n_W$. We assume that $s_{i,i(j)}, j \in 1, \ldots, |s_i|$ are independent outcomes from $N(\mu_i, \sigma)$ denoting a normal distribution with expectation μ_i and standard deviation σ. Further we assume that outcomes from different sentiment words are independent. We associate prior distributions to the unknown parameters $\mu_i \sim N(0, \tau)$ and $\sigma^2 \sim \text{InvGamma}(\alpha, \beta)$. From the regression model we can estimate the posterior distributions $P(\mu_i | s_{i,i(1)}, \ldots, s_{i,i(|s_i|)}), i = 1, \ldots, n_W$ which will be used in the next Section.

2.2 Bayesian decision theory

In the traditional decision theory we assume that we have a set of stochastic variables $X_1, X_2, \ldots, X_n$ where $X_i \sim f(x|\theta)$ and in the Bayesian framework we assume a prior distribution $\theta \sim p(\theta)$. We want to decide a value for the unknown parameter θ and denote this decision (action) a. In Bayesian decision theory we chose a value a minimizing the posterior expected loss

$$\begin{aligned}\widehat{a} &= \underset{a}{\text{argmin}} \left\{ E_\theta(L(a;\theta)|x_1, x_2, \ldots, x_n) \right\} \\ &= \underset{a}{\text{argmin}} \left\{ \int_\theta L(a;\theta)\, p(\theta|x_1, x_2, \ldots, x_n)\, \mathrm{d}\theta \right\}\end{aligned}$$

where $p(\theta|x_1, x_2, \ldots, x_n)$ is the posterior distribution and $L(a;\theta)$ the loss function that returns the loss of the decision $\theta = a$. The most common loss function is the quadratic loss $L(a;\theta) = (a - \theta)^2$ which results in the action $\widehat{a} = E_\theta(\theta|x_1, x_2, \ldots, x_n)$, the posterior expectation.

2.3 Corpus loss function

In this section we join the information from the sentiment lexicons and the domain corpus minimizing the posterior expected loss. Our loss function consists of two parts. The first part is the quadratic loss function based on the sentiment lexicons

$$L_1(a_i; \mu_i) = (a_i - \mu_i)^2$$

The second part of the loss function incorporates information from a corpus from the domain of interest. We assume that the corpus consist of D document and could for example be a large set of product reviews, movie reviews or news articles that we need to classify with respect to sentiment. We assume that the true sentiment classes of these documents are unknown, but still these documents contain valuable sentiment information by the fact that sentiment words in the same document tend to have similar values [11]. For example a positive review typically consists of more positive than negative sentiment words. In traditional sentiment lexicon based classification this valuable information is not used. In the second part of the loss function we incorporate this information setting that the loss increases if a_i differs more from the expected sentiment value of the

neighboring sentiment words in the same document

$$L_2(a_i;\mu_1,\mu_2,\ldots,\mu_{n_W}) = \sum_{d=1}^{D}\sum_{k=1}^{N_{id}}\sum_{p=1}^{P_{id}} \frac{1}{\delta(w_{ikd},\widetilde{w}_{kdp})+1}\left[a_i - \psi(w_{ikd},\widetilde{w}_{dp})\widetilde{\mu}_{dp}\right]^2$$

where N_{id} is the number of times w_i occurs in document d, w_{ikd} occurrence number k of sentiment word w_i in document d. Further, $\widetilde{w}_{d1},\ldots,\widetilde{w}_{dP_{id}}$ denote the other occurrences of sentiment words in document d except w_{ikd} and $\widetilde{\mu}_{d1},\ldots,\widetilde{\mu}_{dP_{id}}$ is the expected sentiment value of these sentiment words according to the model in Section 2.1.

The word 'good' has a positive sentiment while the phrase 'not good' has a negative sentiment. Thus the word 'not' results in a shift in sentiment. Words like 'not', 'never', 'none', 'nobody' are referred to as sentiment shifters [1] and it is natural to change the sentiment of a sentiment word if it is close to a sentiment shifter. The function $\psi(w_{ikd},\widetilde{w}_{dp})$ includes the sentiment shift in the comparison of w_{ikd} and $\widetilde{w}_{dp}$. If there are no shifters close to either w_{ikd} or $\widetilde{w}_{dp}$ no shift is necessary, and $\psi(w_{ikd},\widetilde{w}_{dp}) = 1$. If there is a sentiment shifter close to w_{ikd} or close to $\widetilde{w}_{dp}$ the sentiment of one of them is shifting, and thus $\psi(w_{ikd},\widetilde{w}_{dp})$ is equal to -1. In some rare cases there is more than one sentiment shifter close to w_{ikd} and $\widetilde{w}_{dp}$. We than use the rule that two shifters outweigh each other. Thus, more generally, we use the rule that if in total there is an odd number of sentiment shifters close to w_{ikd} and $\widetilde{w}_{dp}$, then $\psi(w_{ikd},\widetilde{w}_{dp})$ is equal to -1, or else it is equal to 1.

Finally, the function $\delta(w_{ikd},\widetilde{w}_{dp})$ returns the number of words between w_{ikd} and $\widetilde{w}_{dp}$. The shorter the distance $\delta(w_{ikd},\widetilde{w}_{dp})$, the more likely the sentiment values are expected to be similar [12, 13]. Thus, we set the loss inversely proportional to the distances $\delta(w_{ikd},\widetilde{w}_{dp})+1$.

The overall loss function is a weighted sum of the two loss functions presented above.

$$L(a_i) = \alpha N_i L_1(a_i) + (1-\alpha)L_2(a_i),\ \alpha\in[0,1]$$

where $N_i = \sum_{d=1}^{D} N_{id}$, the number of times w_i occurs in the corpus. With $\alpha = 1$, the loss function only depends on the sentiment lexicons and not on the corpus. The lower the value α, the more the loss function depends on information from corpus ($L_2(a_i)$).

Let $\widehat{a}_i$ denote that value of a_i that minimizes the posterior expected loss

$$\widehat{a}_i = \underset{a_i}{\operatorname{argmin}}\, E\left[L\left(a_i\right)\right]$$

with respect to the posterior distributions of $\mu_i, i = 1,2,\ldots,n_w$. Straight forward computations gives

$$\widehat{a}_i = \frac{\alpha N_i E_i + (1-\alpha)\sum_{d=1}^{D}\sum_{k=1}^{N_{id}}\sum_{p=1}^{P_{id}} \frac{\psi(w_{ikd},\widetilde{w}_{dp})}{\delta(w_{ikd},\widetilde{w}_{dp})+1}\widetilde{E}_{dp}}{\alpha N_i + (1-\alpha)\sum_{d=1}^{D}\sum_{k=1}^{N_{id}}\sum_{p=1}^{P_{id}} \frac{1}{\delta(w_{ikd},\widetilde{w}_{dp})+1}}$$

where E_i denote the posterior expectation $E(\mu_i|s_{i,i(1)},\ldots,s_{i,i(|s_i|)})$ and similarly $\widetilde{E}_{dp}$ is the posterior expectation of $\widetilde{\mu}_{dp}$. In accordance with Section 2.2, with $\alpha = 1$ the sentiment value $\widehat{a}_i$ becomes equal to the posterior expectation, E_i.

3 Preexisting sentiment lexicons

For the method in Section 2 we use three different sentiment lexicons developed for the Norwegian language.

Translation The first sentiment lexicon was generated by translating the well-known English sentiment lexicon AFINN [14] to Norwegian using machine translation (Google translate) and doing further manual improvements. We denote this lexicon AFINN in the rest of the paper.

Synonym antonym word graph To create the second sentiment lexicon we first built a large undirected graph of synonym and antonym relations between words from three Norwegian thesauruses. The words were nodes in the graph and synonym and antonym relations were edges. The full graph consists of a total of 6036 nodes (words), where 109 of the nodes represent the seed words (51 positive and 57 negative), and there are 16475 edges (synonyms and antonyms) in the graph. The seed words were manually selected, picking words that are used frequently in the Norwegian language and that span different dimensions of both positive sentiment ('happy', 'clever', 'intelligent', 'love' etc.) and negative sentiment ('lazy', 'aggressive', 'hopeless', 'chaotic' etc.). The sentiment lexicon was generated using the Label Propagation algorithm [15], which is the most common algorithm for this task. The initial phase of the Label Propagation algorithm consists of giving each positive and negative seed a word score 1 and -1, respectively. All other nodes in the graph are given score 0. The algorithm propagates through each non-seed words updating the score using a weighted average of the scores of all neighbouring nodes (connected with an edge). When computing the weighted average, synonym and antonym edges are given weights 1 and -1, respectively. The algorithm is iterated until changes in scores are below some threshold for all nodes. The resulting score for each node becomes our derived sentiment lexicon. For more details, we refer the reader to our previous work [16]. We denote this sentiment lexicon LABEL in the rest of the paper.

From corpus The third sentiment lexicon was constructed using the corpus based approach [17] on a large Norwegian corpus consisting of about one billion words. We started with 14 seed words, seven with positive and seven with negative sentiment and computed the Pointwise mutual information (PMI) between the seed words and the 5000 most frequent words in the corpus and 8340 adjectives not being part of the 5000 most frequent words. The computed PMI scores lay the foundation for the sentiment lexicon. For more details, see [18]. We denote this lexicon PMI in the rest of the paper.

Based on the sentiment lexicons described above, we generated three sentiment lexicons using the method in Section 2 with $\alpha = 0, \alpha = 0.5$ and $\alpha = 1$. In the rest of the paper we denote these sentiment lexicons W0, W0.5 and W1, respectively. We adjusted the sentiment lexicons towards the domain of product

reviews using the text from 15118 product reviews from the Norwegian online shopping sites `www.komplett.no`, `mpx.no`. For the sentiment shifter function $\psi(w_{ikd}, \widetilde{w}_{dp})$ in the loss function L_2 in Section 2.3 recall that the sentiment of a sentiment word is shifted if a sentiment shifter is close to the sentiment word. In the computations in this paper we decided to shift sentiment if the sentiment shifter was one or two words in front of the sentiment word. We only used the sentiment shifter 'not' ('ikke'), but also considered other sentiment shifters, such as 'never' ('aldri'), and other distances between the sentiment word and the shifter. However, the selected approach presented in this paper seems to be the best for such lexicon approaches in Norwegian [19].

4 Evaluating classification performance

For each of the product reviews from `www.komplett.no` and `mpx.no` a rating from 1 to 5 is known and is used to evaluate the classification performance of each of the sentiment lexicon described above.

For each lexicon, we computed the sentiment score of a review by simply adding the score of each sentiment word in a sentiment lexicon together, which is the most common way to do it [20]. Similar as for the sentiment shifter function $\psi(w_{ikd}, \widetilde{w}_{dp})$ in L_2 we shifted the sentiment of a sentiment word if the sentiment shifter 'not' ('ikke') was one or two words in front of the sentiment word. Finally the sum is divided by the number of words in the review, giving us the final sentiment score for the review.

Classification method We divided the reviews in two equal parts, one half being training data and the other half used for testing. We used the training data to estimate the average sentiment score of all reviews related to the different ratings. The computed scores could look like Table 1. We classified a review from

Rating	1	2	3	4	5
Average sentiment score	−0.23	−0.06	0.04	0.13	0.24

Table 1. Average computed sentiment score for reviews with different ratings.

the test set using the sentiment lexicon to compute a sentiment score for the test review and classify to the closest average sentiment score from the training set. E.g. if the computed sentiment score for the test review was -0.05 and estimated averages were as given in Table 1, the review was classified to rating 2. In some rare cases the estimated average sentiment score was not monotonically increasing with the rating. Table 2 shows an example where the average for rating 3, is higher than for the rating 4. For such cases, the average of the two sentiment scores were computed, $(0.10 + 0.18)/2 = 0.14$, and classified to 3 or 4 if the computed sentiment score of the test review was below or above 0.14, respectively.

Rating	1	2	3	4	5
Average sentiment score	−0.23	−0.06	0.18	0.10	0.24

Table 2. Example were sentiment score were not monotonically increasing with rating.

Classification performance We evaluated the classification performance using average difference in absolute value between the true and predicted rating for each review in the test set

$$\text{Average abs. error} = \frac{1}{n}\sum_{i=1}^{n}|p_i - r_i|$$

where n is the number off reviews in the test set and p_i and r_i is the predicted and true rating of review i in the test set. Naturally, a small average absolute error would mean that the sentiment lexicon performs well.

Note that the focus in this paper is not to do a best possible classification performance based on the training material. If that was our goal, other more advanced and sophisticated techniques would be used, such as machine learning based techniques. Our goal is rather to evaluate and compare the performance of sentiment lexicons, and the framework described above is chosen with respect to that.

5 Results

This section presents the results of classification performance on product reviews for the different sentiment lexicons. The results are shown in Table 3. Training

	N	Mean (Stdev)	95% conf.int.
AFINN	2260	1.17 (1.11)	(1.14, 1.19)
W0.5	14987	1.24 (1.17)	(1.22, 1.27)
W1	14987	1.29 (1.17)	(1.26, 1.31)
LABEL	6036	1.38 (1.27)	(1.36, 1.41)
W0	14987	1.52 (1.37)	(1.49, 1.55)
PMI	13340	1.53 (1.34)	(1.50, 1.56)

Table 3. Classification performance for sentiment lexicons on `komplett.no` and `mpx.no` product reviews. The columns from left to right show the sentiment lexicon names, the number of words in the sentiment lexicons, mean absolute error with standard deviation and 95% confidence intervals for mean absolute error.

and test sets were created by randomly adding an equal amount of reviews to both sets. All sentiment lexicons were trained and tested on the same training and test sets, making comparisons easier. This procedure was also repeated several times, and every time the results were in practice identical to the results in

Tables 3, documenting that the results are independent of which reviews that were added to the training and test sets.

Recall that we constructed W0, W0.5 and W1 based on the lists AFINN, LABEL and PMI which we call the source lexicons in the rest of this paper. We see that the source lexicons varies quite much in performance, ranging from 1.17 to 1.53, with the AFINN lexicon being the best. This indicates that translation of sentiment lexicons from one language to another can be an efficient way to construct viable sentiment lexicons (at least when the languages are related, such as the two Germanic, Indo-European languages, English and Norwegian.) Both of the sentiment lexicons that solely rely on corpus (PMI and W0) perform poorer than the other sentiment lexicons. Even though the performance of the source lexicons varies quite much, the performance of W0.5 and W1 is very good and almost as well as the best of the source lexicons (AFINN) and much better than the two other source lexicons (LABEL and PMI). Interestingly W0.5 performs significantly better than both W1 (paired T-test p-value = 0.022) and W0 (p-value = $2.3 \cdot 10^{-7}$) showing that the best sentiment lexicon is the one that is constructed by combining the information from both the source sentiment lexicons and the product review corpus.

Tables 4 and 5 show sentiment words that have the largest difference in sentiment score between the two sentiment lexicons W0 and W1 and that occur at least 50 times in the product review corpus. These were the sentiment words that were adjusted the most when the information from the product review corpus were included. Similar to other corpus based methods, noise is introduced, and we observe examples of this noise in the tables. E.g. we see that words like 'fabu-

Norwegian	English	Lexicon W1	Lexicon W0	Difference
skada	damaged	−0.35	1.78	−2.13
gult	yellow	0.19	2.26	−2.07
forklarer	explains	−0.33	1.68	−2.01
rikelig	plenty	−0.45	1.53	−1.98
fabelaktige	fabulous	−0.33	1.61	−1.93
knotete	tricky	0.14	2.07	−1.93
fantastisk	awesome	0.20	2.11	−1.91
dårligt	bad	−0.09	1.82	−1.91
søt	sweet	−0.43	1.31	−1.74
forholdet	relationship	−0.07	1.66	−1.73
jublet	cheered	−0.47	1.26	−1.73
finale	finale	−0.07	1.65	−1.73
anvendelig	applicable	0.29	2.00	−1.71
kontakter	contacts	0.01	1.66	−1.65

Table 4. Sentiment words where the sentiment scores are decreased the most when the information from the corpus is included. Columns from left to right: Sentiment words in Norwegian, in English, sentiment scores in the sentiment lexicons W0 and W1 and the difference between these sentiment scores.

Norwegian	English	Lexicon W1	Lexicon W0	Difference
vinne	win	1.19	−1.20	2.39
nedsatt	reduced	0.37	−1.84	2.22
angitt	specified	0.37	−1.81	2.19
vunnet	won	0.79	−1.37	2.15
sjokkerende	shocking	0.85	−1.29	2.14
reklamerte	advertised	0.18	−1.91	2.09
dust	jerk	0.74	−1.29	2.03
skittent	dirty	0.23	−1.75	1.99
akseptabel	acceptable	0.34	−1.48	1.81
jævlig	damn	0.06	−1.75	1.81
misvisende	misleading	0.22	−1.55	1.76
sensitiv	sensitive	0.05	−1.68	1.73
jenter	girls	0.27	−1.43	1.70
uregelmessig	irregular	0.03	−1.67	1.70
alminnelige	general	0.37	−1.30	1.68

Table 5. Sentiment words where the sentiment scores are increased the most when the information from the corpus is included. Columns from left to right: Sentiment words in Norwegian, in English, sentiment scores in the sentiment lexicons W0 and W1 and the difference between these sentiment scores.

lous' and 'awesome' have been changes from a positive score to negative/neutral and that words like 'jerk', 'dirty' and 'damn' have been changed from a negative score to positive/neutral. On the other hand, we also see several words that seem to have been changed to a more reasonable score. E.g. we see that words like 'damaged', 'tricky', 'bad', and 'contacts' are changed from a positive score to a negative/neutral value. There are also examples of words that seem to be changed in a reasonable way with respect to the domain of product reviews. E.g. the word 'reduced' is in many contexts a word with negative sentiment, but with respect to product reviews the word is mostly used to state that prices are reduced, which is a positive statement. In Table 5, we see that the word is changed from a negative to a positive sentiment score when the corpus is included.

6 Conclusions

In this paper, we have developed a method to construct domain specific sentiment lexicons by combining the information from many pre-existing sentiment lexicons with an unanotated corpus from the domain of interest. Trying to combine this sources of information has not been investigated in the literature earlier.

In order to cope with these domain specific adjustments, we adopt a stochastic formulation of the sentiment score assignment problem instead of the classical deterministic formulation. Our approach is based on minimizing the expected loss of a loss function that punishes deviations from the scores of the source sentiment lexicons and inhomogeneity in sentiment scores for the same review.

Our results show that a lexicon that combines information from both the source sentiment lexicons and the domain specific corpus performs better than

a lexicon that only rely on information from the source lexicons. This lexicon shows an impressive performance that is almost as good as the best of the source lexicons.

References

1. Liu, B.: Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, Toronto, Canada (2012)
2. Aue, A., Gamon, M.: Customizing sentiment classifiers to new domains: A case study. In: Proceedings of Recent Advances in Natural Language Processing (RANLP). (2005)
3. Blitzer, J., Dredze, M., Pereira, F.: Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: Proceedings of the Association for Computational Linguistics (ACL). (2007)
4. Tan, S., Wu, G., Tang, H., Cheng, X.: A novel scheme for domain-transfer problem in the context of sentiment analysis. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management. CIKM '07, New York, NY, USA, ACM (2007) 979–982
5. Bollegala, D., Weir, D., Carroll, J.: Cross-Domain Sentiment Classification Using a Sentiment Sensitive Thesaurus. IEEE Transactions on Knowledge and Data Engineering **25**(8) (2013) 1719–1731
6. Pan, S.J., Ni, X., Sun, J.T., Yang, Q., Chen, Z.: Cross-domain Sentiment Classification via Spectral Feature Alignment. In: Proceedings of the 19th International Conference on World Wide Web. WWW '10, New York, NY, USA, ACM (2010) 751–760
7. Chetviorkin, I., Loukachevitch, N.V.: Extraction of Russian Sentiment Lexicon for Product Meta-Domain. In: COLING. (2012) 593–610
8. Gindl, S., Weichselbraun, A., Scharl, A.: Cross-Domain Contextualization of Sentiment Lexicons. In Coelho, H., Studer, R., Wooldridge, M., eds.: ECAI. Volume 215 of Frontiers in Artificial Intelligence and Applications., IOS Press (2010) 771–776
9. Weichselbraun, A., Gindl, S., Scharl, A.: Extracting and grounding context-aware sentiment lexicons. IEEE Intelligent Systems **28**(2) (2013) 39–46
10. Owsley, S., Sood, S., Hammond, K.J.: Domain specific affective classification of documents. In: AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs. (2006) 181–183
11. Turney, P.: Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the Association for Computational Linguistics (ACL). (2002) 417–424
12. Ding, X., Liu, B., Yu, P.S.: A Holistic Lexicon-based Approach to Opinion Mining. In: Proceedings of the 2008 International Conference on Web Search and Data Mining. WSDM '08, New York, NY, USA, ACM (2008) 231–240
13. Chetviorkin, I., Loukachevitch, N.: Two-Step Model for Sentiment Lexicon Extraction from Twitter Streams. In: Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Association for Computational Linguistics (2014) 90–96
14. Nielsen, F.Å.: A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. CoRR **abs/1103.2903** (2011)
15. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Technical report, Technical Report CMU-CALD-02-107, Carnegie Mellon University (2002)

16. Hammer, H., Bai, A., Yazidi, A., Engelstad, P.: Building sentiment lexicons applying graph theory on information from three Norwegian thesauruses. In: Norweian informatics conference. (2014)
17. Turney, P.D., Littman, M.L.: Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems (TOIS) **21**(4) (2003) 315–346
18. Bai, A., Hammer, H.L., Yazidi, A., Engelstad, P.: Constructing sentiment lexicons in Norwegian from a large text corpus. In: The 17th IEEE International conference on Computational science and Engineering (CSE). (2014) 231–237
19. Hammer, H.L., Solberg, P.E., vrelid, L.O.: Sentiment classification of online political discussions: a comparison of a word-based and dependency-based method. In: Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Association for Computational Linguistics (2014) 90–96
20. Bing, L.: Web Data Mining. Exploring Hyperlinks, Contents, and Usage Data. Springer (2011)