Fine-Grained Contextual Predictions for Hard Sentiment Words

Sebastian Ebert and **Hinrich Schütze**
Center for Information and Language Processing
University of Munich, Germany
ebert@cis.lmu.de, inquiries@cislmu.org

Abstract

We put forward the hypothesis that high-accuracy sentiment analysis is only possible if word senses with different polarity are accurately recognized. We provide evidence for this hypothesis in a case study for the adjective "hard" and propose contextually enhanced sentiment lexicons that contain the information necessary for sentiment-relevant sense disambiguation. An experimental evaluation demonstrates that senses with different polarity can be distinguished well using a combination of standard and novel features.

1 Introduction

This paper deals with fine-grained sentiment analysis. We aim to make three contributions. First, based on a detailed linguistic analysis of contexts of the word "hard" (Section 3), we give evidence that highly accurate sentiment analysis is only possible if senses with different polarity are accurately recognized.

Second, based on this analysis, we propose to return to a lexicon-based approach to sentiment analysis that supports identifying sense distinctions relevant to sentiment. Currently available sentiment lexicons give the polarity for each word or each sense, but this is of limited utility if senses cannot be automatically identified in context. We extend the lexicon-based approach by introducing the concept of a *contextually enhanced sentiment lexicon* (CESL). The lexicon entry of a word w in CESL has three components: (i) the senses of w; (ii) a sentiment annotation of each sense; (iii) a data structure that, given a context in which w occurs, allows to identify the sense of w used in that context.

As we will see in Section 3, the CESL sense inventory – (i) above – should be optimized for sentiment analysis: closely related senses with the same sentiment should be merged whereas subtle semantic distinctions that give rise to different polarities should be distinguished.

The data structure in (iii) is a statistical classification model in the simplest case. We will give one other example for (iii) below: it can also be a set of centroids of context vector representations, with a mapping of these centroids to the senses.

If sentiment-relevant sense disambiguation is the first step in sentiment analysis, then powerful contextual features are necessary to support making fine-grained distinctions. Our third contribution is that we experiment with deep learning as a source of such features. We look at two types of deep learning features: word embeddings and neural network language model predictions (Section 4). We show that deep learning features significantly improve the accuracy of context-dependent polarity classification (Section 5).

2 Related work

Initial work on sentiment analysis was either based on sentiment lexicons that listed words as positive or negative sentiment indicators (e.g., Turney (2002), Yu and Hatzivassiloglou (2003)), on statistical classification approaches that represent documents as ngrams (e.g., Pang et al. (2002)) or on a combination of both (e.g., Riloff et al. (2003), Whitelaw et al. (2005)). The underlying assumption of lexicon-based sentiment analysis is that a word always has the same sentiment. This is clearly wrong because words can have senses with different polarity, e.g., "hard wood" (neutral) vs. "hard memory" (negative).

Ngram approaches are also limited because ngram representations are not a good basis for relevant generalizations. For example, the neutral adverbial sense 'intense' of "hard" ("laugh hard", "try hard") vs. the negative adjectival mean-

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1210–1215,
October 25-29, 2014, Doha, Qatar.

		Cobuild	syntax	meaning	example	patterns	sent.	# train	# test
1	FIRM	1	ADJ	firm, stiff	*hard floor*		neu	78	5
2	DIFFICULT	2, 4, 9, 10, 11	ADJ	difficult	*hard question*	*hard for*, *hard on*, *hard to V*	neg	2561	120
3	ADVERB	3a, 5, 6, 7	ADV	intensely	*work hard*		neu	425	19
4	INTENSE	3b	ADJ	intense	*hard look*	*be hard at it*	neu	24	7
5	HARD-MAN	8	ADJ	unkind	*hard man*		neg	15	0
6	HARD-TRUTH	12	attributive ADJ	definitely true	*hard truth*		neu	5	6
7	MUSIC		ADJ	hard-rock-type music	*hard beats*		neu	347	15
8	CONTRAST		ADJ	opposite of soft transition	*hard edge*		neu	3	1
9	NEGATIVE-P	13, 15	phrases				neg	36	2
10	NEUTRAL-P	14, 16	phrases				neu	375	27

Table 1: Sense inventory of "hard".

ing 'difficult' ("hard life", "hard memory") cannot be easily distinguished based on an ngram representation. Moreover, although ngram approaches could learn the polarity of these phrases they do not generalize to new phrases.

More recent compositional approaches to sentiment analysis can outperform lexicon and ngram-based methods (e.g., Socher et al. (2011), Socher et al. (2013)). However, these approaches conflate two different types of contextual effects: differences in sense or lexical meaning ("hard memory" vs. "hard wood") on the one hand and meaning composition like negation on the other hand. From the point of view of linguistic theory, these are different types of contextual effects that should not be conflated. Recognizing that "hard" occurs in the scope of negation is of no use if the basic polarity of the contextually evoked sense of "hard" (e.g., negative in "no hard memories" vs. neutral in "no hard wood") is not recognized.

Wilson et al. (2009) present an approach to classify contextual polarity building on a two-step process. First, they classify if a sentiment word is polar in a phrase and if so, second, they classify its polarity. Our approach can be seen as an extension of this approach; the main difference is that we will show in our analysis of "hard" that the polarity of phrases depends on the senses of the words that are used. This is evidence that high-accuracy polarity classification depends on sense disambiguation.

There has been previous work on assigning polarity values to senses of words taken from WordNet (e.g., Baccianella et al. (2010), Wiebe and Mihalcea (2006)). However, these approaches are not able to disambiguate the sense of a word given its context.

Previous work on representation learning for sentiment analysis includes (Maas and Ng, 2010) and (Maas et al., 2011). Their models learn word embeddings that capture semantic similarities and word sentiment at the same time. Their approach focuses on sentiment of entire sentences or documents and does not consider each sentiment word instance at a local level.

We present experiments with one supervised and one semisupervised approach to word sense disambiguation (WSD) in this paper. Other WSD approaches, e.g., thesaurus-based WSD (Yarowsky, 1992), could also be used for CESL.

3 Linguistic analysis of sentiment contexts of "hard"

We took a random sample of 5000 contexts of "hard" in the Amazon Product Review Data (Jindal and Liu, 2008). We use 200 as a test set and set aside 200 for future use. We analyzed the remaining 4600 contexts using a tool we designed for this study, which provides functionality for selecting and sorting contexts, including a keyword in context display. If a reliable pattern has been identified (e.g., the phrase "die hard"), then all contexts matching the pattern can be labeled automatically.

Our goal is to identify the different uses of "hard" that are relevant for sentiment. The basis for our inventory is the Cobuild (Sinclair, 1987)

lexicon entry for "hard". We use Cobuild because it was compiled based on an empirical analysis of corpus data and is therefore more likely to satisfy the requirements of NLP applications than a traditional dictionary.

Cobuild lists 16 senses. One of these senses (3) is split into two to distinguish the adverbial ("to accelerate hard") and adjectival ("hard acceleration") uses of "hard" in the meaning 'intense'. We conflated five senses (2, 4, 9, 10, 11) referring to different types of difficulty: "hard question" (2), "hard work" (4), "hard life" (11) and two variants of "hard on": "hard on someone" (9), "hard on something" (10); and four different senses (3a, 5, 6, 7) referring to different types of intensity: "to work hard" (3a), "to look hard" (5), "to kick hard" (6), "to laugh hard" (7). Furthermore, we identified a number of noncompositional meanings or phrases (lists NEGATIVE-P and NEUTRAL-P in the supplementary material[1]) in addition to the four listed by Cobuild (13, 14, 15, 16). In addition, new senses for "hard" are introduced for opposites of senses of "soft": the opposite of 'quiet/gentle voice/sound' (7: MUSIC; e.g., "hard beat", "not too hard of a song") and the opposite of 'smooth surface/texture' (8: CONTRAST; e.g., "hard line", "hard edge").

Table 1 lists the 10 different uses that are the result of our analysis. For each use, we give the corresponding Cobuild sense numbers, syntactic information, meaning, an example, typical patterns, polarity, and number of occurrences in training and test sets.

7 uses are neutral and 3 are negative. As "hard's" polarity in most sentiment lexicons is negative, but only 3 out of 7 senses are negative, "hard" provides evidence for our hypothesis that senses need to be disambiguated to allow for fine-grained and accurate polarity recognition.

We hired two PhD students to label each of the 200 contexts in the test set with one of the 10 labels in Table 1 ($\kappa = .78$). Disagreement was resolved by a third person.

We have published the labeled data set of 4600+200 contexts as supplementary material.

4 Deep learning features

We use two types of deep learning features to be able to make the fine-grained distinctions necessary for sense disambiguation. First, we use word embeddings similar to other recent work (see below). Second, we use a deep learning language model (LM) to predict the distribution of words for the position at which the word of interest occurs. For example, an LM will predict that words like "granite" and "concrete" are likely in the context "a * countertop" and that words like "serious" and "difficult" are likely in the context "a * problem". This is then the basis for distinguishing contexts in which "hard" is neutral (in the meaning 'firm, solid') from contexts in which it is a sentiment indicator (in the meaning 'difficult'). We will use the term *predicted context distribution* or PCD to refer to the distribution predicted by the LM.

We use the vectorized log-bilinear language model (vLBL) (Mnih and Kavukcuoglu, 2013) because it has three appealing features. (i) It learns state of the art word embeddings (Mnih and Kavukcuoglu, 2013). (ii) The model is a language model and can be used to calculate PCDs. (iii) As a linear model, vLBL can be trained much faster than other models (e.g., Bengio et al. (2003)).

The vLBL trains one set of word embeddings for the input space (R) and one for the target space (Q). We denote the input representation of word w as $\boldsymbol{r}_w$ and the target representation as $\boldsymbol{q}_w$. For a given context $c = w_1, \ldots, w_n$ the model predicts a target representation $\hat{\boldsymbol{q}}$ by linearly combining the context word representations with position dependent weights:

$$\hat{\boldsymbol{q}}(c) = \sum_{i=1}^{n} \boldsymbol{d}_i \odot \boldsymbol{r}_{w_i}$$

where $\boldsymbol{d}_i \in D$ is the weight vector associated with position i in the context and $\odot$ is pointwise multiplication. Given the model parameters $\theta = \{R, Q, D, \boldsymbol{b}\}$ the similarity between $\hat{\boldsymbol{q}}$ and the correct target word embedding is computed by the similarity function

$$s_\theta(w, c) = \hat{\boldsymbol{q}}(c)^T \boldsymbol{q}_w + b_w$$

where b_w is a bias term.

We train the model with stochastic gradient descent on mini-batches of size 100, following the noise-contrastive estimation training procedure of Mnih and Kavukcuoglu (2013). We use AdaGrad (Duchi et al., 2011) with the initial learning rate set to $\eta = 0.5$. The embeddings size is set to 100.

[1]All supplementary material is available at `http://www.cis.lmu.de/ebert`.

			ngram	PCD	embed	acc	prec	rec	F_1
	bl	1				.62	.62	1.00	.76
development	fully	2	+			.90	.91	.94	.92
		3		+		.90	.91	.92	.92
		4			+	.87	.87	.92	.90
		5	+	+		.92	.92	.94	.93
		6	+		+	.91	.90	.95	.92
		7		+	+	.86	.83	.96	.89
		8	+	+	+	.92	.93	.95	**.94**
	semi	9	+			.85	.87	.89	.88
		10		+		.85	.87	.89	.88
		11			+	.76	.73	.98	.83
		12	+	+		.85	.87	.89	.88
		13	+		+	.85	.87	.89	.88
		14		+	+	.85	.89	.87	.88
		15	+	+	+	.86	.87	.90	**.89**
test	bl	16				.66	.66	1.00	.80
	fully	17	+	+	+	.90	.89	.96	.92
	semi	18	+	+	+	.85	.85	.91	.88

Table 2: Classification results; bl: baseline

During training we do not need to normalize the similarity explicitly, because the normalization is implicitly learned by the model. However, normalization is still necessary for prediction. The normalized PCD for a context c of word w is computed using the softmax function:

$$P_\theta^c(w) = \frac{\exp(s_\theta(w, c))}{\sum_{w'} \exp(s_\theta(w', c))}$$

We use a window size of $ws = 7$ for training the model. We found that the model did not capture enough contextual phenomena for $ws = 3$ and that results for $ws = 11$ did not have better quality than $ws = 7$, but had a negative impact on the training time. Using a vocabulary of the 100,000 most frequent words, we train the vLBL model for 4 epochs on 1.3 billion 7-grams randomly selected from the English Wikipedia.

5 Experiments

The lexicon entry of "hard" in CESL consists of (i) the senses, (ii) the polarity annotations (neutral or negative) and (iii) the sense disambiguation data structure. Components (i) and (ii) are shown in Table 1. In this section, we evaluate two different options for (iii) on the task of sentiment classification.

	1	2	3	4	5	6	7	8
1								
2	‡							
3	‡							
4	‡	‡	·					
5	‡			‡				
6	‡			‡				
7	‡	‡	*		‡	‡		
8	‡	*	*	‡		*	‡	

Table 3: Significant differences of lines 1–8 in Table 2; ‡: p=0.01, *: p=0.05, ·: p=0.1

The first approach is to use a statistical classification model as the sense disambiguation structure. We use liblinear (Fan et al., 2008) with standard parameters for classification based on three different feature types: ngrams, embeddings (embed) and PCDs. Ngram features are all n-grams for $n \in \{1, 2, 3\}$. As embedding features we use (i) the mean of the input space (R) embeddings and (ii) the mean of the target space (Q) embeddings of the words in the context (see Blacoe and Lapata (2012) for justification of using simple mean). As PCD features we use the PCD predicted by vLBL for the sentiment word of interest, in our case "hard".

We split the set of 4600 contexts introduced in Section 3 into a training set of 4000 and a development set of 600.

Table 2 (lines 1–8) shows the classification results on the development set for all feature type combinations. Significant differences between results – computed using the approximate randomization test (Padó, 2006) – are given in Table 3. The majority baseline (bl), which assigns a negative label to all examples, reaches $F_1 = .76$. The classifier is significantly better than the baseline for all feature combinations with F_1 ranging from .89 to .94. We obtain the best classification result (.94) when all three feature types are combined (significantly better than all other feature combinations except for 5).

Manually labeling all occurrences of a word is expensive. As an alternative we investigate *clustering of the contexts of the word of interest*. Therefore, we represent each of the 4000 contexts of "hard" in the training set as its PCD[2], use

[2]To transform vectors into a format that is more appropriate for the underlying Gaussian model of kmeans, we take the square root of each probability in the PCD vectors.

kmeans clustering with $k = 100$ and then label each cluster. This decreases the cost of labeling by an order of magnitude since only 100 clusters have to be labeled instead of 4000 training set contexts.

Table 2 (lines 9–15) shows results for this semisupervised approach to classification, using the same classifier and the same feature types, but the cluster-based labels instead of manual labels.

For most feature combinations, F_1 drops compared to fully supervised classification. The best performing model for supervised classification (ngram+PCD+embed) loses 5%.

This is not a large drop considering the savings in manual labeling effort. All results are significantly better than the baseline. There are no significant differences between the different feature sets (lines 9–15) with the exception of embed, which is significantly worse than the other 6 sets.

The centroids of the 100 clusters can serve as an alternative sense disambiguation structure for the lexicon entry of "hard" in CESL.[3] Each sense s is associated with the centroids of the clusters whose majority sense is s.

As final experiment (lines 16–18 in Table 2), we evaluate performance for the baseline and for PCD+ngram+embed – the best feature set – on the test set. On the test set, baseline performance is .80 (.04 higher than .76 on line 1, Table 2); F_1 of PCD+ngram+embed is .92 (.02 less than development set) for supervised classification and is .88 (.01 less) for semisupervised classification (comparing to lines 8 and 15 in Table 2). Both results (.92 and .88) are significantly higher than the baseline (.80).

6 Conclusion

The sentiment of a sentence or document is the output of a causal chain that involves complex linguistic processes like contextual modification and negation. Our hypothesis in this paper was that for high-accuracy sentiment analysis, we need to model the root causes of this causal chain: the meanings of individual words. This is in contrast to other work in sentiment analysis that conflates different linguistic phenomena (word sense ambiguity, contextual effects, negation) and attempts to address all of them with a single model.

For sense disambiguation, the first step in the causal chain of generating sentiment, we proposed CESL, a contextually enhanced sentiment lexicon that for each word w holds the inventory of senses of w, polarity annotations of these senses and a data structure for assigning contexts of w to the senses. We introduced new features for sentiment analysis to be able to perform the fine-grained modeling of context needed for CESL. In a case study for the word "hard", we showed that high accuracy in sentiment disambiguation can be achieved using our approach. In future work, we would like to show that our findings generalize from the case of "hard" to the entire sentiment lexicon.

[3] Included in supplementary material.

Acknowledgments

This work was supported by DFG (grant SCHU 2246/10). We thank Lucia Krisnawati and Sascha Rothe for their help with annotation.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *International Conference on Language Resources and Evaluation*, pages 2200–2204.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

William Blacoe and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 546–556. Association for Computational Linguistics.

John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *International Conference on Web Search and Web Data Mining*, pages 219–230.

Andrew L. Maas and Andrew Y. Ng. 2010. A probabilistic model for semantic word vectors. In *Annual Conference on Advances in Neural Information Processing Systems: Deep Learning and Unsupervised Feature Learning Workshop*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Annual Meeting of the Association for Computational Linguistics*, pages 142–150.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Annual Conference on Advances in Neural Information Processing Systems*, pages 2265–2273.

Sebastian Padó, 2006. *User's guide to `sigf`: Significance testing by approximate randomisation*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *Conference on Empirical Methods in Natural Language Processing*, pages 79–86.

Ellen Riloff, Janyce Wiebe, and Theresa Ann Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Conference on Natural Language Learning*, volume 4, pages 25–32.

John Sinclair. 1987. *Looking Up: Account of the Cobuild Project in Lexical Computing*. Collins CoBUILD.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Conference on Empirical Methods in Natural Language Processing*, pages 151–161.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

Peter D. Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Annual Meeting of the Association for Computational Linguistics*, pages 417–424.

Casey Whitelaw, Navendu Garg, and Shlomo Argamon. 2005. Using appraisal groups for sentiment analysis. In *International Conference on Information and Knowledge Management*, pages 625–631. ACM.

Janyce Wiebe and Rada Mihalcea. 2006. Word sense and subjectivity. In *Annual Meeting of the Association for Computational Linguistics*, pages 1065–1072.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(3):399–433.

David Yarowsky. 1992. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *International Conference on Computational Linguistics*, pages 454–460.

Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Conference on Empirical Methods in Natural Language Processing*, pages 129–136.