Fine-Grained Contextual Predictions for Hard Sentiment Words

Sebastian Ebert, Hinrich Schütze

Center for Information and Language Processing, University of Munich, Germany

ebert@cis.lmu.de

1. Introduction

- high accuracy sentiment analysis requires sense disambiguation
- flaws of today's systems
  - often words are considered to always have same sentiment
  - ngram approaches lack in ability to generalize
  - compositional approaches conflate differences in lexical meaning ("hard feelings" vs. "hard wood") and meaning composition (e.g., negation)
  - sentiment often for whole documents or sentences

1.1 Contributions

1. detailed **linguistic analysis** of contexts of "hard"
2. introduction of **contextually enhanced sentiment lexicon**, which contains
   (a) senses of word w
   (b) sentiment annotation of each sense
   (c) sense disambiguation structure: statistical classification model or cluster centroids
3. **deep learning features** for sentiment-relevant sense disambiguation

2. Sense Lexicon for "hard"

- basis: 16 Cobuild senses [4] (compiled based on an empirical analysis)
- 4800 contexts of "hard" from Amazon Product Reviews [2]

2.1 Cobuild Senses Refined

- split (3): distinguish the adverbial ("to accelerate hard") and adjectival ("hard acceleration") sense in the meaning 'intense'
- conflated (2, 4, 9, 10, 11): different types of difficulty ("hard question" (2), "hard work" (4), "hard life" (11), "hard on someone" (9), "hard on something" (10)
- conflated (3a, 5, 6, 7): different types of intensity: "to work hard" (3a), "to look hard" (5), "to kick hard" (6), "to laugh hard" (7)
- new non-compositional meanings in addition to (13, 14, 15, 16)
- new: opposites of senses of "soft"
- new: opposite of 'quiet/gentle voice/sound' (7: MUSIC; e.g., "hard beat", "not too hard of a song")
- new: opposite of 'smooth surface/texture' (8: CONTRAST; e.g., "hard line", "hard edge")

2.2 Sentiment Senses of "hard"

i.e., senses of "hard" & sentiment annotation of each sense

sense	Cobuild	syntax	meaning	example	**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*	neg	2561	120
3 ADVERB	3a, 5, 6, 7	ADV	intensely	*work hard*	neu	425	19
4 INTENSE	3b	ADJ	intense	*hard look*	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		*hard drugs*	neg	36	2
10 NEUTRAL-P	14, 16	phrases		*hard drive*	neu	375	27

3. Sense Disambiguation Structure: Classifier

3.1 Features

1. n-gram features for $n \in 1, 2, 3$
2. probability distribution of language model ($P_\theta^c(w)$)
3. **deep learning features**: mean of input and target representations of context learned by language model ($\sum_{i=1}^{n-1} \boldsymbol{r}_{w_i}$ and $\sum_{i=1}^{n-1} \boldsymbol{q}_{w_i}$) (embed)
4. classifier: liblinear [1]

3.2 Language Model

- vectorized log-bilinear language model (vLBL) [3]

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

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

- $\boldsymbol{r}_{w_i}$: **input representation** of word w_i
- $\hat{\boldsymbol{q}}(c)$: predicted target representation given context $c = w_1, \ldots, w_{n-1}$
- $\boldsymbol{q}_w$: correct **target representation** of word w
- $\boldsymbol{d}_i$: position dependent weights, $\odot$: pointwise multiplication, b_w: bias for word w
- s_θ: similarity of predicted target and real target with $\theta = \{R, Q, D, \boldsymbol{b}\}$: model parameters
- trained using noise-contrastive estimation [3], thus no normalization necessary during training
- for prediction: softmax, i.e., **probability distribution**

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

- do not predict last word, but center word in a window of 7 words
- use P_θ^c as context representation
- trained on English Wikipedia

4. Sense Disambiguation Structure: Cluster Centroids

1. cluster P_θ^c of 4000 contexts of "hard"
2. k-means, 100 clusters
3. assign a sense to each cluster
4. new context gets sense of closest cluster centroid

5. Experiments

- task: classify sense of "hard" as positive or negative given its context
- 4800 contexts of "hard"
- 4000+600 training + development examples: pattern based labeling, e.g., "hard drive"
- 200 test examples: manual labeling
- available online: http://www.cis.lmu.de/ebert
- 2 settings
  1. fully manual: manual labels
  2. semi-automatic: manually label 100 k-means cluster, computed using P_θ^c

5.1 Results

			ngram	PCD	embed	acc	prec	rec	F_1
development	bl	1				.62	.62	1.00	.76
	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

(a) Classification results; bl: baseline

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

(b) Significant differences of lines 1–8 in left table;
‡: p=0.01, *: p=0.05, ·: p=0.1

6. Conclusion and Future Work

- sentiment is output of causal chain
- complex linguistic processes
- high-accuracy sentiment analysis needs **meaning of individual words**
- use a contextually enhanced sentiment lexicon for sense disambiguation, i.e., **sense-based lexicon** instead of word-based
- **deep learning features** helpful for sense disambiguation
- future work: show that findings generalize to other words
- future work: use features from WSD community

Acknowledgments

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

References

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

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

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

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