Classification of Inconsistent Sentiment Words Using Syntactic Constructions

Wiltrud KESSLER *Hinrich SCHÜTZE*
Institute for Natural Language Processing
University of Stuttgart
wiltrud.kessler@ims.uni-stuttgart.de

ABSTRACT
An important problem in sentiment analysis are *inconsistent* words. We define an inconsistent word as a sentiment word whose dictionary polarity is reversed by the sentence context in which it occurs. We present a supervised machine learning approach to the problem of *inconsistency classification*, the problem of automatically distinguishing inconsistent from consistent sentiment words in context. Our first contribution to inconsistency classification is that we take into account sentence structure and use syntactic constructions as features – in contrast to previous work that has only used word-level features. Our second contribution is a method for learning polarity reversing constructions from sentences annotated with polarity. We show that when we integrate inconsistency classification results into sentence-level polarity classification, performance is significantly increased.

KEYWORDS: sentiment analysis, polarity modifiers, polarity shifters, polarity reversers, negation.

Proceedings of COLING 2012: Posters, pages 569–578,
COLING 2012, Mumbai, December 2012.

1 Introduction

Sentiment analysis or opinion mining is the computational study of opinions and sentiments expressed in text (Liu, 2010). Sentiment analysis is typically performed based on sentiment words – words that indicate the sentiment polarity of a document or sentence. A challenge for this approach is that the dictionary polarity of a sentiment word may be reversed by sentence context (Polanyi and Zaenen, 2004). We call such words *inconsistent* words[1].

A classical example of an inconsistent word is the sentiment word *"worth"* in the sentence *"this player is not* $\textbf{\textit{worth}}_{pos}$ *any price"* where the negation *"not"* reverses the polarity of *"worth"*, so that the final sentiment expressed in the sentence is not positive, but negative. Such polarity reversing expressions are diverse, e.g., *"lack of* $\textbf{\textit{quality}}_{pos}$*"* or *"*$\textbf{\textit{easy}}_{pos}$ *to hit accidentally"*.

In this work we present a supervised machine learning approach to the problem of inconsistency classification, the problem of automatically distinguishing inconsistent from consistent sentiment words in context. Training examples for inconsistency classification are extracted automatically from sentences annotated with polarity. We make two contributions to the state of the art. First, while previous work has used only features at the word level, we take into account sentence structure and use syntactic constructions as features. Second, we present first steps towards automatically extracting polarity reversing constructions (PRCs) from sentences annotated with polarity. PRCs can be used as features for inconsistency classification as well as for directly identifying inconsistent words. We show that our treatment of inconsistent words improves polarity classification performance on sentence-level compared to a baseline.

This paper is structured as follows. The next section discusses related work. Section 3 describes inconsistency classification, the format of syntactic constructions and the extraction of training examples. We then present experimental results for polarity classification (Section 4). The second part of this paper describes our method for automatically extracting PRCs (Section 5), and evaluates their usefulness (Section 6). Finally, we conclude and outline future work.

2 Related Work

Negations, or, more generally, *polarity reversers*, create inconsistent words which are a major source of errors for polarity classification. Polarity reversers are diverse and do not include only negation function words (Choi and Cardie, 2008). Thus, some treatment of inconsistent words in polarity classification is common; for a survey see Wiegand et al. (2010).

Most approaches for polarity classification work on word-level and simply consider a word w as inconsistent if it is preceded by a word out of a fixed list of polarity reversers, this includes rule-based (Polanyi and Zaenen, 2004; Hu and Liu, 2004) as well as statistical approaches (Pang et al., 2002). Unlike these approaches, we use syntactic information.

Some approaches go beyond word-level, e.g., Wilson et al. (2005) use special features to model the existence of polarity modifiers in the syntactic context of a sentiment word, Choi and Cardie (2008) use syntactic patterns to treat content negators, and Nakagawa et al. (2010) integrate polarity reversing words into a dependency tree based method. While these works include some syntactic information, they still use a manually defined list of polarity reversing words. In contrast, we use machine learning to identify polarity reversing constructions (PRCs).

[1]Note that our terminology differs from that used by (Dragut et al., 2012) who use the term "inconsistent" to refer to a word that has conflicting polarity information in a sentiment dictionary or across dictionaries.

An important challenge that most approaches ignore is the detection of the scope of negation. Councill et al. (2010) use dependency parses to predict the scope of polarity reversing words. Our approach goes the opposite way: given a sentiment word, we determine if it is in the scope of any PRC. Our definition of syntactic constructions explicitly includes scope.

The work most closely related to our approach is (Ikeda et al., 2008) who also address the task of inconsistency classification. Their inconsistency classifier uses the local context of three words to the left and right of the target sentiment word as features. Li et al. (2010) extend that method to document level by stacking two classifiers trained on reversed and nonreversed sentences. Both works use only word-level information in their classifiers. We go beyond word-level and use syntactic constructions. We also attempt to explicitly identify and extract the syntactic constructions that are responsible for making a sentiment word inconsistent.

3 Approach

The main component of our approach is the inconsistency classifier, that assigns a score $s_{\text{incons}}(w)$ to each sentiment word token w in context, and classifies w as being `inconsistent` ($s_{\text{incons}}(w) > 0$) or `consistent` ($s_{\text{incons}}(w) \leq 0$) with its dictionary polarity.

The final task we want to improve is sentence-level polarity classification. To determine the polarity of a sentence, we calculate a positivity score $s_{\text{pos}}(S)$ for the sentence S using a dictionary of positive and negative sentiment words (p and n). The sentence is labeled `positive` iff $s_{\text{pos}}(S) \geq 0$, else `negative`. We integrate inconsistency classification by counting a word with its score $s_{\text{incons}}(w)$. Thus, we define $s_{\text{pos}}(S)$ as follows (cf. (Ikeda et al., 2008)):

$$s_{\text{pos}}(S) = \sum_{w \in p} -s_{\text{incons}}(w) + \sum_{w \in n} s_{\text{incons}}(w) \quad (1)$$

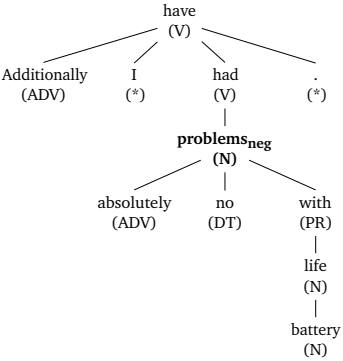
for all $w \in S$. In our proposed **consistency voting** we use a statistical classifier to determine $s_{\text{incons}}(w)$ and use its classification confidence as score. Our first contribution is to include syntactic constructions as defined below as features for inconsistency classification.

We use two baselines with simpler ways of determining $s_{\text{incons}}(w)$: **Standard voting** assumes every word to be consistent, so we set $s_{\text{incons}}(w) = -1$ for all words and Equation 1 is simplified to $s_{\text{pos}}(S) = |\{w \in p\}| - |\{w \in n\}|$. A common way of treating inconsistent words is **negation voting**, which sets $s_{\text{incons}}(w) = 1$ (`inconsistent`) iff an odd number of negation cues occurs in the context of w, else $s_{\text{incons}}(w) = -1$ (`consistent`).

3.1 Syntactic constructions

Polarity modifiers are a syntactic phenomenon and word-level approaches fail to take into account the scope of a polarity reverser (cf. Wiegand et al. (2010)). To integrate syntactic information, we parse all training examples with a dependency parser. The parts of speech (POS) produced by the parser are generalized to the categories N (noun), V (verb), ADJ (adjective), ADV (adverb), PR (preposition), DT (determiner), and * (everything else).

We extract *syntactic constructions* from the parses that describe the syntactic context of a sentiment word. We define a syntactic construction as any path that starts at a sentiment word, ends at another word in the sentence, and contains the POS categories of all nodes that are traversed on the path. An example, the syntactic construction `N<V<V>additionally_ADV`, is given in Figure 1. The sentiment word *"problems"* is represented by POS category (`N`), but is not



$k = 0$	`N`

$k = 1$	`N>absolutely_ADV`
	`N>no_DT`
	`N>with_PR`
	`N<have_V`

$k = 2$	`N>PR>life_N` `N<V<have_V`

$k = 3$	`N>PR>N>battery_N`
	`N<V<V>additionally_ADV`
	`N<V<V>i_*`

Figure 1: Formalization of syntactic constructions. Left: The basis for extracting constructions is a dependency parse, in this case for the sentence *"Additionally I have had absolutely no problems with battery life."* Right: Extracted constructions for the sentiment word *"problems"*.

included, as we are interested in constructions that are independent of specific sentiment words. The path contains the direction in the parse tree (up < or down >), the nodes that are traversed on the way – represented by POS category (`V` for *"had"* and *"have"*) – and lemma/POS of the final word on the path (*"additionally"*, `ADV`).

All syntactic constructions extracted from the context of a sentiment word w up to a certain parse tree distance k (defined in number of nodes on the path) are used as features for training the bag-of-constructions inconsistency classifier.

3.2 Finding consistent and inconsistent training examples

For training our inconsistency classifier we need a set of training examples annotated for (in)consistency. We assume that we have a corpus of polarity annotated sentences and a dictionary of positive and negative sentiment words at our disposal.

We follow Ikeda et al. (2008) and extract training examples automatically from the corpus. Given a sentiment word w with dictionary polarity p_w that appears in a sentence s with polarity p_s in the corpus, we label w `consistent` iff $p_w = p_s$, and `inconsistent` otherwise. We ignore words and sentences with any label other than `positive` and `negative` as well as sentiment words occurring with a POS not in the dictionary.

E.g., from the sentence[2] *"The phone isn't* $\textbf{\textit{hard}}_{neg}$ *to use so its* $\textbf{\textit{great}}_{pos}$*"* (labeled `positive`), we extract *"hard"* (resp. *"great"*) as an `inconsistent` (resp. `consistent`) training example.

4 Experiments

4.1 Data

We evaluate on the customer review data set[3] (Hu and Liu, 2004). Statistics about the data set can be found in Table 1. The original data set is annotated at aspect level. To create sentence

[2] All example sentences are from user reviews including all errors in spelling and grammar.
[3] `http://www.cs.uic.edu/~liub/FBS/CustomerReviewData.zip`

		all	positive	negative
1	# sentences	1726	1078	648
2	# available sentences	1446	948	498
3	# sentiment words found	2930	2032	898
4	# inconsistent words	824	465	359

Table 1: Statistics of customer review data set.

		A	P_{pos}	R_{pos}	F_{pos}	P_{neg}	R_{neg}	F_{neg}	F
1	standard voting	76.1	76.5	91.7	83.4	74.5	46.4	57.2	70.3 †
2	negation voting	79.0	78.4	**93.7**	85.4	**80.9**	51.0	62.6	74.0*†
3	consistency voting (BoW)	78.5	81.4	87.0	84.1	71.6	62.2	66.6	75.4 †
4	consistency voting (BoC)	**80.9**	**82.1**	90.6	**86.2**	77.8	**62.4**	**69.3**	**77.7***

Table 2: Results of sentence-level polarity classification on customer review data set: **A**ccuracy, **P**recision, **R**ecall, **F**-measure (positive and negative sentences), macro **F**-measure.

polarity annotations, we take the aspect label as sentence label if there is only one aspect or all aspect labels have the same polarity. If a *"but"* separates two aspects of conflicting polarity, the two parts of the sentence are split and separately annotated. If no splitting is possible or there is no annotated aspect, the sentence is ignored.

From the total number of polarity annotated sentences (line 1) we can only compute a useful polarity score for sentences that contain at least one sentiment word (line 2), all other sentences are ignored for the evaluation.

As a dictionary of sentiment words we use the MPQA subjectivity clues[4] (Wilson et al., 2005) containing 2304 positive and 4152 negative words. A word may have several possible POS tags. Sentences have been parsed with the Bohnet dependency parser (Bohnet, 2010). Sentiment words are extracted as (in)consistent (lines 3 and 4) with the method presented in Section 3.2.

4.2 Results of sentence-level polarity classification

We use two different inconsistency classifiers with consistency voting: The bag-of-words classifier **BoW** determines $s_{\text{incons}}(w)$ with the three context words to the left and right of the sentiment word as features. This is a reimplementation of Ikeda et al.'s (2008) "word-wise" learning. The bag-of-constructions classifier **BoC** uses the syntactic constructions described in Section 3.1 up to parse-tree distance $k = 3$ as features for inconsistency classification. In both cases, we use the Stanford MaxEnt classifier (Manning and Klein, 2003) with default settings and train it in a 5-fold cross-validation setting.

As baselines, we include **standard voting** and **negation voting**. For negation voting we define context as the three words to the left and right of the sentiment word and use nine negation cue words from (Ikeda et al., 2008): *no*, *not*, *yet*, *never*, *none*, *nobody*, *nowhere*, *nothing*, *neither*.

Ikeda et al. (2008) report an accuracy of 71.6% for the standard voting baseline on the same data set when using sentiment words from General Inquirer (Stone et al., 1996). Our standard voting baseline with MPQA subjectivity clues yields a much higher accuracy of 76.1%. Accuracy is a less suitable performance measure for this task as the data set is skewed (65.6%

[4] `http://www.cs.pitt.edu/mpqa/subj_lexicon.html`

positive sentences). This is why we have reimplemented their approach and restrict our further discussion to our reimplementation and macro F-measure only.

Table 2 shows the result of our experiments. Bold numbers denote the best result in each column. We mark a macro F-measure result with * if it is significantly higher than the previous line and with † if it is significantly worse than consistency voting with the BoC classifier.[5] Determining inconsistency with the BoC classifier significantly outperforms all other methods.

5 Polarity reversing constructions (PRCs)

We define a *polarity reversing construction* (PRC) as a syntactic construction (see Section 3.1) that reverses the polarity of the sentiment word in its scope. Recall that the sentiment word is the first node of the path represented by the construction.

Our goal is the automatic extraction of PRCs. We work on the assumption that in the syntactic context of inconsistent words there is always a PRC present. Syntactic constructions that appear often in the context of inconsistent words are likely to be PRCs. We use the extracted training examples for consistent and inconsistent words (see Section 3.2). All training examples are parsed with a dependency parser and syntactic constructions are extracted from the context (see Section 3.1). All extracted constructions are candidates for PRCs.

The candidates are scored with Mutual Information (**MI**). **MI** measures how much information the presence or absence of a candidate x contributes to making the correct classification decision for a sentiment word. $MI(x, C)$ between candidate x and the classes $C = \{\texttt{consistent}, \texttt{inconsistent}\}$ is defined as

$$MI(x,C) = \sum_{c \in C} P(x,c) \log_2 \frac{P(x,c)}{P(x) \cdot P(c)} + \sum_{c \in C} P(\bar{x},c) \log_2 \frac{P(\bar{x},c)}{P(\bar{x}) \cdot P(c)} \quad (2)$$

where $P(x)$ is the probability that x occurred, and $P(\bar{x})$ the probability that x didn't occur. The n candidates with the highest scores are taken as PRCs.

MI extracts candidates that serve as a good indicator for *one of* the classes, but not necessarily for the class `inconsistent`. For the **MI+** score, we remove candidates with negative association from the final set of PRCs (Dunning, 1993).

6 Experiments with PRCs

6.1 Results of PRC extraction

For the robust extraction of PRCs we need more annotated sentences than the customer review corpus contains. As there is no such corpus in the domain and to avoid manual annotation effort, we use semistructured reviews in which users provide pros (product aspects the user evaluates as positive) and cons (product aspects the user evaluates as negative) in addition to the written text of the review. We automatically create a corpus annotated with polarity at the sentence level as follows: All pros (resp. cons) longer than 3 tokens are extracted as a sentence with label `positive` (resp. `negative`). Shorter pros (resp. cons) are stripped of sentiment words (using the subjectivity clues dictionary) and if the resulting string is found in the review text, the containing sentence is extracted as `positive` (resp. `negative`). This is a somewhat simplistic method, but we still get enough annotated sentences for our purposes.

[5] Statistically significant at $p < .05$ using the approximate randomization test (Noreen, 1989).

		all	positive	negative
1	# extracted sentences	58 503	34 881	23 622
2	# available sentences	42 943	27 510	15 433
3	# sentiment words found	83 258	57 192	26 066
4	# inconsistent words	24 325	12 502	11 823

Table 3: Statistics of automatically annotated camera/cellphone data set.

		A	P_{pos}	R_{pos}	F_{pos}	P_{neg}	R_{neg}	F_{neg}	F
1	negation vot. (words)	79.0	78.4	93.7	85.4	80.9	51.0	62.6	74.0
2	negation vot. (PRC, gold)	80.2	79.6	**93.8**	86.1	**82.1**	54.2	65.3	75.7*
3	negation vot. (PRC, **MI**)	59.1	68.9	68.6	68.7	40.8	41.2	41.0	54.9
4	negation vot. (PRC, **MI+**)	78.8	78.4	93.2	85.2	79.9	51.2	62.4	73.8
5	consist. vot. (BoC)	80.9	**82.1**	90.6	86.2	77.8	**62.4**	69.3	77.7
6	consist. vot. (BoPRC, gold)	**81.3**	81.9	91.7	86.5	79.5	61.4	69.3	**77.9**
7	consist. vot. (BoPRC, **MI**)	81.2	**82.1**	91.2	86.4	78.8	62.0	**69.4**	**77.9**
8	consist. vot. (BoPRC, **MI+**)	**81.3**	81.6	92.4	**86.6**	80.6	60.2	69.0	77.8

Table 4: Sentence-level polarity classification on customer review data set with PRCs.

We perform the annotation on an existing corpus of 17 442 semistructured camera and cellphone reviews[6] (Branavan et al., 2008) from `epinions.com`. Table 3 contains statistics about the data. We use this corpus only for the automatic extraction of PRCs, not to evaluate polarity classification. To judge the quality of the automatic annotation, we hired a graduate student of computational linguistics to manually annotate a random subset of 1271 sentences. The agreement of the automatic and manual annotation is 0.79; Cohen's κ is 0.61.

To directly evaluate the extracted PRCs, the graduate student also annotated some syntactic constructions as PRCs / non-PRCs. This results in a set of 70 gold PRCs.[7]

Comparing the automatically extracted constructions to our set of gold PRCs, we find that few actual PRCs are found when scoring with **MI** (as we expected). Of the top 70 constructions extracted as PRCs with **MI**, only 15 are correct (21%). Results for **MI+** are better, but still noisy: 20 out of 70 are correct (29%). These results do not look very promising, but as we will see, we can still use noisy PRCs successfully in polarity classification.

6.2 Results of sentence-level polarity classification

We use PRCs in two ways: In **negation voting with PRCs**, we define context as the syntactic context of a sentiment word and use PRCs as negation cues. We also use consistency voting with a bag-of-PRC (**BoPRC**) inconsistency classifier that uses only PRCs as features instead of using all constructions (i.e., a feature-selection on BoC). Our intuition is that as only polarity reversal is marked, PRCs should be all that is needed to identify inconsistent words.

Both methods are tested with PRCs extracted using **MI** and **MI+**. We extract these PRCs from the camera/cellphone data described in Section 6.1. We extract the top 70 constructions to match the number of constructions in our manually annotated PRC set. Additionally, we use the manually annotated PRCs (**gold**) as an upper bound of automatic PRC-based performance.

[6] `http://groups.csail.mit.edu/rbg/code/precis/` (camera and cellphone data sets)
[7] Available at `http://www.ims.uni-stuttgart.de/~kesslewd/data/sentiment.html`

To enable comparison with our previous results, we use the evaluation setup described in Section 4.2. Table 4 shows the results. For easier comparison, we have repeated lines 2 (word-level negation voting) and 4 (consistency voting with BoC) from Table 2 as lines 1 resp. 5 in Table 4. Bold numbers denote the best result in each column.

We compare negation voting with PRCs to the word-level negation voting. The improvement in macro F-measure of negation voting with gold PRCs is significant (marked with $*$).[8] Unsurprisingly, the PRCs extracted with **MI** hurt performance instead of improving it. The noisy PRCs extracted with **MI+** achieve a similar performance than word-level negation voting (the difference is not significant). For such a noisy set (only 29% of the PRCs are correct), this is a promising result.

In consistency voting, telling the BoC inconsistency classifier which features are important by some sort of feature selection either manually or automatically improves performance for all variants of BoPRC. Although no improvement is statistically significant, this is still an interesting result, as it shows that even noisy information about the important features can improve performance of inconsistency classification.

Conclusion and perspectives

We have presented a supervised machine learning approach to detect if a sentiment word is consistent or inconsistent with its dictionary polarity in a specific sentence context. We have evaluated our approach on sentence-level polarity classification by integrating the score of such an inconsistency classifier into a majority voting approach. As our first contribution, we have shown that the use of syntactic constructions as features for the inconsistency classifier can improve performance. As a second contribution, we have presented first steps towards automatically extracting polarity reversing constructions from sentences annotated with polarity and demonstrated two possible uses of such constructions in sentence-level polarity classification.

To get sufficient training data for the extraction of polarity reversing constructions, we have automatically annotated sentences from semistructured reviews with polarity. For future work, we plan to improve the quality and coverage of this automatic annotation as a means to get sentence-labeled data from semistructured reviews, which are available in large quantities.

A major problem in sentiment analysis are sentiment words that do not express sentiment in a given context (subjectivity analysis cf. (Wilson et al., 2005)). In a preliminary study, we found that about 50% of words extracted as inconsistent training examples did in fact not express sentiment in the sentence context, e.g., the word *"slow"* in the positive sentence *"easy to hold steady when using slower shutter speeds"*. Identifying and discarding non-subjective phrases like *"slower shutter speeds"* would improve the classification results as well as the quality of the extracted polarity reversing constructions.

Acknowledgments

This research was funded by Deutsche Forschungsgemeinschaft (DFG, SFB 732, D7). We thank Olga Podushko for the annotation. We also thank Andrea Glaser, Charles Jochim, Khalid Al Khatib and Christian Scheible for their suggestions about this work.

[8] Statistically significant at $p < .05$ using the approximate randomization test (Noreen, 1989).

References

Bohnet, B. (2010). Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of COLING '10*, pages 89–97.

Branavan, S. R. K., Chen, H., Eisenstein, J., and Barzilay, R. (2008). Learning document-level semantic properties from free-text annotations. In *Proceedings of ACL '08*, pages 263–271.

Choi, Y. and Cardie, C. (2008). Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of EMNLP '08*, pages 793–801.

Councill, I. G., McDonald, R., and Velikovich, L. (2010). What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. In *Proceedings of NeSp-NLP '10*, pages 51–59.

Dragut, E., Wang, H., Yu, C., Sistla, P., and Meng, W. (2012). Polarity consistency checking for sentiment dictionaries. In *Proceedings of ACL '12*, pages 997–1005.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of KDD '04*, pages 168–177.

Ikeda, D., Takamura, H., Ratinov, L.-A., and Okumura, M. (2008). Learning to shift the polarity of words for sentiment classification. In *Proceedings of IJCNLP '08*, pages 50–57.

Li, S., Lee, S. Y. M., Chen, Y., Huang, C.-R., and Zhou, G. (2010). Sentiment classification and polarity shifting. In *Proceedings of COLING '10*, pages 635–643.

Liu, B. (2010). Sentiment analysis and subjectivity. In Indurkhya, N. and Damerau, F. J., editors, *Handbook of Natural Language Processing, Second Edition*, pages 627–666. Chapman & Hall/CRC.

Manning, C. and Klein, D. (2003). Optimization, maxent models, and conditional estimation without magic. In *Proceedings of NAACL-Tutorials '03*.

Nakagawa, T., Inui, K., and Kurohashi, S. (2010). Dependency tree-based sentiment classification using CRFs with hidden variables. In *Proceedings of HLT '10*, pages 786–794.

Noreen, E. W. (1989). *Computer-intensive methods for testing hypotheses – an introduction*. Wiley & Sons.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of EMNLP '02*, pages 79–86.

Polanyi, L. and Zaenen, A. (2004). Contextual valence shifters. In *Proceedings of AAAI Spring Symposium on Exploring Attitude and Affect in Text*, pages 106–111.

Stone, P. J., Dunphy, D. C., Smith, M. S., and Ogilvie, D. M. (1996). *General Inquirer: A Computer Approach to Content Analysis*. The MIT Press.

Wiegand, M., Balahur, A., Roth, B., and Klakow, D. (2010). A survey on the role of negation in sentiment analysis. In *Proceedings of NeSp-NLP '10*, pages 60–68.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of HLT '05*, pages 347–354.