DCU: Aspect-based Polarity Classification for SemEval Task 4

Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman
Dasha Bogdanova, Jennifer Foster and Lamia Tounsi
CNGL Centre for Global Intelligent Content
National Centre for Language Technology
School of Computing
Dublin City University
Dublin, Ireland
{jwagner,parora,scortes,ubarman}@computing.dcu.ie
{dbogdanova,jfoster,ltounsi}@computing.dcu.ie

Abstract

We describe the work carried out by DCU on the Aspect Based Sentiment Analysis task at SemEval 2014. Our team submitted one constrained run for the restaurant domain and one for the laptop domain for sub-task B (aspect term polarity prediction), ranking highest out of 36 systems on the restaurant test set and joint highest out of 32 systems on the laptop test set.

1 Introduction

This paper describes DCU's participation in the Aspect Term Polarity sub-task of the Aspect Based Sentiment Analysis task at SemEval 2014, which focuses on predicting the sentiment polarity of aspect terms for a restaurant and a laptop dataset. Given, for example, the sentence *I have had so many problems with the computer* and the aspect term *the computer*, the task is to predict whether the sentiment expressed towards the aspect term is *positive*, *negative*, *neutral* or *conflict*.

Our polarity classification system uses supervised machine learning with support vector machines (SVM) (Boser et al., 1992) to classify an aspect term into one of the four classes. The features we employ are word n-grams (with n ranging from 1 to 5) in a window around the aspect term, as well as features derived from scores assigned by a sentiment lexicon. Furthermore, to reduce data sparsity, we experiment with replacing sentiment-bearing words in our n-gram feature set with their polarity scores according to the lexicon and/or their part-of-speech tag.

The paper is organised as follows: in Section 2, we describe the sentiment lexicons used in this work and detail the process by which they are combined, filtered and extended; in Section 3, we describe our baseline method, a heuristic approach which makes use of the sentiment lexicon, followed by our machine learning method which incorporates the rule-based method as features in addition to word n-gram features; in Section 4, we present the results of both methods on the training and test data, and perform an error analysis on the test set; in Section 5, we compare our approach to previous research in sentiment classification; Section 6 discusses efficiency of our system and ongoing work to improve its speed; finally, in Section 7, we conclude and provide suggestions as to how this research could be fruitfully extended.

2 Sentiment Lexicons

The following four lexicons are employed:

1. **MPQA**[1] (Wilson et al., 2005) classifies a word or a stem and its part of speech tag into positive, negative, both or neutral with a strong or weak subjectivity.
2. **SentiWordNet**[2] (Baccianella et al., 2010) specifies the positive, negative and objective scores of a synset and its part of speech tag.
3. **General Inquirer**[3] indicates whether a word expresses positive or negative sentiment.
4. **Bing Liu's Opinion Lexicon**[4] (Hu and Liu,

[1] http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/
[2] http://sentiwordnet.isti.cnr.it/
[3] http://www.wjh.harvard.edu/~inquirer/inqtabs.txt
[4] http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 223–229,
Dublin, Ireland, August 23-24, 2014.

2004) indicates whether a word expresses positive or negative sentiment.

2.1 Lexicon Combination

Since the four lexicons differ in their level of detail and in how they present information, it is necessary, when combining them, to consolidate the information and present it in a uniform manner. Our combination strategy assigns a sentiment score to a word as follows:

- **MPQA:** 1 for strong positive subjectivity, -1 for strong negative subjectivity, 0.5 for weak positive subjectivity, -0.5 for weak negative subjectivity, and 0 otherwise
- **SentiWordNet:** The positive score if the positive score is greater than the negative and objective scores, the negative score if the negative score is greater than the positive and the objective scores, and 0 otherwise
- **General Inquirer and Bing Liu's Opinion Lexicon:** 1 for positive and -1 for negative

The above four scores are summed to arrive at a final score between -4 and 4 for a word.[5]

2.2 Lexicon Filtering

Initial experiments with our sentiment lexicon and the training data led us to believe that there were many irrelevant entries that, although capable of conveying sentiment in some other context, were not contributing to the sentiment of aspect terms in the two domains of the task. Therefore, these words are manually filtered from the lexicon. Examples of deleted words are *just*, *clearly*, *indirectly*, *really* and *back*.

2.3 Adding Domain-Specific Words

A manual inspection of the training data revealed words missing from the merged sentiment lexicon but which do express sentiment in these domains. Examples are *mouthwatering*, *watery* and *better-configured*. We add these to the lexicon with a score of either 1 or -1 (depending on their polarity in the training data). We also add words (e.g. *zesty, acrid*) from an online list of culinary terms.[6]

[5]We also tried to vote over the four lexicon scores but this did not improve over summing.

[6]http://world-food-and-wine.com/describing-food

2.4 Handling Variation

In order to ensure that all inflected forms of a word are covered, we lemmatise the words in the training data using the IMS TreeTagger (Schmid, 1994) and we construct new possibilities using a suffix list. To correct misspelled words, we consider the corrected form of a misspelled word to be the form with the highest frequency in a reference corpus[7] among all the forms within an edit distance of 1 and 2 from the misspelled word (Norvig, 2012). Multi-word expressions of the form *x-y* are added with the polarity of *xy* or *x*, as in *laid-back/laidback* and *well-shaped/well*. Expressions *x y*, are added with the polarity of *x-y*, as in *so so/so-so*.

3 Methodology

We first build a rule-based system which classifies the polarity of an aspect term based solely on the scores assigned by the sentiment lexicon. We then explore different ways of converting the rule-based system into features which can then be combined with bag-of-n-gram features in a supervised machine learning set-up.

3.1 Rule-Based Approach

In order to predict the polarity of an aspect term, we sum the polarity scores of all the words in the surrounding sentence according to our sentiment lexicon. Since not all the sentiment words occurring in a sentence influence the polarity of the aspect term to the same extent, it is important to weight the score of each sentiment word by its distance to the aspect term. Therefore, for each word in the sentence which is found in our lexicon we take the score from the lexicon and divide it by its distance to the aspect term. The distance is calculated using the sum of the following three distance functions:

- **Token Distance**: This function calculates the difference in the position of the sentiment word and the aspect term by counting the tokens between them.

[7]The reference corpus consists of about a million words retrieved from several public domain books from Project Gutenberg (http://www.gutenberg.org/), lists of most frequent words from Wiktionary (http://en.wiktionary.org/wiki/Wiktionary:Frequency_lists) and the British National Corpus (http://www.kilgarriff.co.uk/bnc-readme.html) and two thousand laptop reviews crawled from CNET (http://www.cnet.com/).

- **Discourse Chunk Distance**: This function counts the discourse chunks that must be crossed in order to get from the sentiment word to the aspect term. If the sentiment word and the aspect term are in the same discourse chunk, then the distance is zero. We use the discourse segmenter described in (Tofiloski et al., 2009).
- **Dependency Path Distance**: This function calculates the shortest path between the sentiment word and the aspect term in a syntactic dependency graph for the sentence, produced by parsing the sentence with a PCFG-LA parser (Attia et al., 2010) trained on consumer review data (Le Roux et al., 2012)[8], and converting the resulting phrase-structure tree into a dependency graph using the Stanford converter (de Marneffe and Manning, 2008) (version 3.3.1).

Since our lexicon also contains multi-word expressions such as *finger licking*, we also look up bigrams and trigrams from the input sentence in our lexicon. Negation is handled by reversing the polarity of sentiment words that appear within a window of three words of the following negators: *not*, *n't*, *no* and *never*.

For each aspect term, we use the distance-weighted sum of the polarity scores to predict one of the three classes *positive*, *negative* and *neutral*.[9] After experimenting with various thresholds we settled on the following simple strategy: if the polarity score for an aspect term is greater than zero then it is classified as positive, if the score is less than zero, then it is classified as negative, otherwise it is classified as neutral.

3.2 Machine Learning Approach

We train a four-way SVM classifier for each domain (laptop and restaurant), using Weka's SMO implementation (Platt, 1998; Hall et al., 2009).[10]

[8]To facilitate parsing, the data was normalised using the process described in (Le Roux et al., 2012) with minor modifications, e. g. treatment of non-breakable space characters, abbreviations and emoticons. The normalised version of the data was used for all experiments.

[9]We also experimented with classifying aspect terms as *conflict* when the individual scores for positive and negative sentiment were both relatively high. However, this proved unsuccessful.

[10]We also experimented with logistic regression, random forests, k-nearest neighbour, naive Bayes and multi-layer perceptron in Weka, but did not match performance of an SVM trained with default parameters.

Transf.	n	c	n-gram	Freq.
-L—	2	2	cord with	1
AL—	2	2	<aspect> with	56
ALS–	1	4	<negu080>	595
ALSR-	1	4	<negu080>	502
AL—	2	4	and skip	1
ALSR-	2	4	and <negu080>	25
ALSRP	1	4	<negu080>/vb	308

Table 1: 7 of the 2,640 bag-of-n-gram features extracted for the aspect term *cord* from the laptop training sentence *I charge it at night and skip taking the cord with me because of the good battery life.* The last column shows the frequency of the feature in the training data. Transformations: A=aspect, L=lowercase, S=score, R=restricted to certain POS, P=POS annotation

Our system submission uses bag-of-n-gram features and features derived from the rule-based approach. Decisions about parameters are made in 5-fold cross-validation on the training data provided for the task.

3.2.1 Bag-of-N-gram Features

We extract features encoding the presence of specific lower-cased n-grams (L) ($n = 1, ..., 5$) in the context of the aspect term to be classified (c words to the left and c words to the right with $c = 1, ..., 5, \text{inf}$) for 10 combinations of transformations: replacement of the aspect term with <ASPECT> (A), replacement of sentiment words with a discretised score (S), restriction (R) of the sentiment word replacement to certain parts-of-speech, and annotation of the discretised score with the POS (P) of the sentiment word. An example is shown in Table 1.

3.2.2 Adding Rule-Based Score Features

We explore two approaches for incorporating information from the rule-based approach (Section 3.1) into our SVM classifier. The first approach is to encode polarity scores directly as the following four features:

1. distance-weighted sum of scores of positive words in the sentence
2. distance-weighted sum of scores of negative words in the sentence
3. number of positive words in the sentence

4. number of negative words in the sentence

The second approach is less direct: for each domain, we train J48 decision trees with minimum leaf size 60 using the four rule-based features described above. We then use the decision rules and the conjunctions leading from the root node to each leaf node to binarise the above four basic score features, producing 122 features. Furthermore, we add normalised absolute values, rank of values and interval indicators, producing 48 features.

3.2.3 Submitted Runs

We eliminate features that have redundant value columns for the training data, and we apply frequency thresholds (13, 18, 25 and 35) to further reduce the number of features. We perform a grid-search to optimise the parameters C and γ of the SVM RBF kernel. We choose the system to submit based on average cross-validation accuracy. We experiment with combinations of the three feature sets described above. We choose the binarised features over the raw rule-based scores because cross-validation results are inferior for the rule-based scores in initial experiments with feature frequency threshold 35: 70.26 vs. 71.36 for laptop and 72.06 vs. 72.15 for restaurant. Therefore, we decide to focus on systems with binarised score features for lower feature frequency thresholds, which are more CPU-intensive to train. For both domains, the system we end up submitting is a combination of the n-gram features and the binarised features with parameters $C = 3.981$, $\gamma = 0.003311$ for the laptop data, $C = 1.445$, $\gamma = 0.003311$ for the restaurant data, and a frequency threshold of 13.

4 Results and Analysis

Table 2 shows the training and test accuracy of the task baseline system (Pontiki et al., 2014), a majority baseline classifying everything as positive, our rule-based system and our submitted system. The restaurant domain has a higher accuracy than the laptop domain for all systems, the SVM system outperforms the rule-based system on both domains, and the test accuracy is higher than the training accuracy for all systems in the restaurant domain.

We observe that the majority of our systems' errors fall into the following categories:

Dataset	System	Training	Test
Laptop	Baseline	—	51.1%
Laptop	All positive	41.9%	52.1%
Laptop	Rule-based	65.4%	67.7%
Laptop	SVM	72.3%	70.5%
Restaurant	Baseline	—	64.3%
Restaurant	All positive	58.6%	64.2%
Restaurant	Rule-based	69.5%	77.8%
Restaurant	SVM	72.7%	81.0%

Table 2: Accuracy of the task baseline system, a system classifying everything as positive, our rule-based system and our submitted SVM-based system on train (5-fold cross-validation) and test sets

- **Sentiment not expressed explicitly:** The sentiment cannot be inferred from local lexical and syntactic information, e. g. *The sushi is cut in blocks bigger than my cell phone*.
- **Non-obvious expression of negation:** For example, *The Management was less than accomodating [sic]*. The rule-based approach does not capture such cases and there are not enough similar training examples for the SVM to learn to correctly classify them.
- **Conflict cases:** The training data contains too few examples of conflict sentences for the system to learn to detect them.[11]

For the restaurant domain, there are more than fifty cases where the rule-based approach fails to detect sentiment, but the machine learning approach classifies it correctly. Most of these cases contain no sentiment lexicon words, thus the rule-based system marks them as being neutral. However, the machine learning system was able to figure out the correct polarity. Examples of such cases include *Try the rose roll (not on menu)* and *The gnocchi literally melts in your mouth!*. Furthermore, in the laptop domain, a number of the errors made by the rule-based system arise from the ambiguous nature of some lexicon words. For example, the sentence *Only 2 usb ports ... seems kind of ... limited* is misclassified because the word *kind* is considered to be positive.

There are a few cases where the rule-based system outperforms the machine learning one. It happens when a sentence contains a rare word with strong polarity, e. g. the word *heavenly* in *The*

[11] We only classify one test instance as *conflict*.

chocolate raspberry cake is heavenly - not too sweet, but full of flavor.

5 Related Work

The use of supervised machine learning with bag-of-word or bag-of-n-gram feature sets has been a standard approach to the problem of sentiment polarity classification since the seminal work by Pang et al. (2002) on movie review polarity prediction. Heuristic methods which rely on a lexicon of sentiment words have also been widespread and much of the research in this area has been devoted to the unsupervised induction of good quality sentiment indicators (see, for example, Hatzivassiloglou and McKeown (1997) and Turney (2002), and Liu (2010) for an overview). The integration of sentiment lexicon scores as features in supervised machine learning to supplement standard bag-of-n-gram features has also been employed before (see, for example, Bakliwal et al. (2013)). The replacement of training/test words with scores/labels from sentiment lexicons has also been used by Baccianella et al. (2009), who supplement n-grams such as *horrible location* with generalised expressions such as `NEGATIVE` *location*. Linguistic features which capture generalisations at the level of syntax (Matsumoto et al., 2005), semantics (Johansson and Moschitti, 2010) and discourse (Lazaridou et al., 2013) have also been widely applied. In using binarised features derived from the nodes of a decision tree, we are following our recent work which uses the same technique in a different task: quality estimation for machine translation (Rubino et al., 2012; Rubino et al., 2013).

The main novelty in our system lies not in the individual techniques but rather in they way they are combined and integrated. For example, our combination of token/chunk/dependency path distance used to weight the relationship between a sentiment word and the aspect term has – to the best of our knowledge – not been applied before.

6 Efficiency

Building a system for a shared task, we focus solely on the accuracy of the system in all our decisions. For example, we parse all training and test data multiple times using different grammars to increase sentence coverage from 99.87% to 100%.

To offer a more practical system, we work on implementing a simplified, fully automated system that is more efficient. So far, we replaced time-consuming parsing with POS tagging. The system accepts as input and generates as output valid SemEval ABSA XML documents.[12] After extracting the text and the aspect terms from the input, the text is normalised using the process described in Footnote 8. The feature extraction is performed as described in Section 3 with the following modifications:

- The POS information used by the n-gram feature extractor is obtained using the IMS TreeTagger (Schmid, 1994) instead of using the PCFG-LA parser (Attia et al., 2010).
- The distance used by the rule-based approach is the token distance only, instead of a combination of three distance functions.

The sentiment lexicon and the classification models used are described in Sections 2 and 3 respectively.

The test sets containing 800 sentences are POS tagged in less than half a second each. Surprisingly, accuracy of aspect term polarity prediction increases to 71.4% (from 70.5% for the submitted system) on the laptop test set, using the same SVM parameters as for the submitted system. However, we see a degradation to 78.8% (from 81.0% for the submitted system) for the restaurant test set. This is an encouraging result as the SVM parameters are not yet fully optimised for the slightly different information and as the remaining modifications to be implemented should not change accuracy any further.

The next bottleneck that needs to be addressed before the system can be used in applications requiring quick responses is the current implementation of the n-gram feature extractor: It enumerates all n-grams (for all context window sizes and n-gram transformations) only to then intersect these features with the list of selected features. For the shared task, this made sense as we initially need all features to make our selection of features, and as we only need to run the feature extractor a few times. For a practical system that has to process new test sets frequently, however, it will be more efficient to check for each selected feature whether the respective event occurs in the input.

[12] We validate documents using the XML schema definition provided on the shared task website.

7 Conclusion

We have described our aspect term polarity prediction system, which employs supervised machine learning using a combination of n-grams and sentiment lexicon features. Although our submitted system performs very well, it is interesting to note that our rule-based system is not that far behind. This suggests that a state-of-the-art system can be build without machine learning and that careful design of the other system components is important. However, the very good performance of our machine-learning-based system also suggests that word n-gram features do provide useful information that is missed by a sentiment lexicon alone, and that it is always worthwhile to perform careful parameter tuning to eke out as much as possible from such an approach.

Future work should investigate how much each system component contributes to the overall performance, e. g. lexicon combination, lemmatisation, spelling correction, other normalisations, negation handling, distance function and n-gram feature transformations. There is also room for improvements in most of these components, e. g. our handling of complex negations. Detection of conflicts also needs more attention. Features indicating the presence of trigger words for negation and conflicts that are currently used only internally in the rule-based component could be added to the SVM feature set. It would also be interesting to see how the compositional approach described by Socher et al. (2013) handles these difficult cases. The score features could be easily augmented by breaking down scores by the four employed lexicons. This way, the SVM can choose to combine the information from these scores differently than just summing them, allowing it to learn more complex relations. Lexicon filtering and addition of domain-specific entries could be automated to reduce the time needed to adjust to a new domain. Finally, machine learning methods that can efficiently handle large feature sets such as logistic regression should be tried with the full feature set (not applying frequency thresholds).

Acknowledgements

This research is supported by the Science Foundation Ireland (Grant 12/CE/I2267) as part of CNGL (www.cngl.ie) at Dublin City University. The authors wish to acknowledge the DJEI/DES/SFI/HEA Irish Centre for High-End Computing (ICHEC) for the provision of computational facilities and support. We are grateful to Qun Liu and Josef van Genabith for their helpful comments.

References

Mohammed Attia, Jennifer Foster, Deirdre Hogan, Joseph Le Roux, Lamia Tounsi, and Josef van Genabith. 2010. Handling unknown words in statistical latent-variable parsing models for arabic, english and french. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 67–75.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2009. Multi-facet rating of product reviews. In *Proceedings of ECIR*, pages 461–472.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)*.

Akshat Bakliwal, Jennifer Foster, Jennifer van der Puil, Ron O'Brien, Lamia Tounsi, and Mark Hughes. 2013. Sentiment analysis of political tweets: Towards an accurate classifier. In *Proceedings of the NAACL Workshop on Language Analysis in Social Media*, pages 49–58.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The stanford typed dependencies representation. In *COLING 2008 Workshop on Cross-framework and Cross-domain Parser Evaluation.*, pages 1–8.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the ACL and the 8th Conference of the European Chapter of the ACL*, pages 174–181.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177.

Richard Johansson and Alessandro Moschitti. 2010. Syntactic and semantic structure for opinion expression detection. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 67–76.

Angeliki Lazaridou, Ivan Titov, and Caroline Sporleder. 2013. A bayesian model for joint unsupervised induction of sentiment, aspect and discourse representations. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics*, pages 1630–1639.

Joseph Le Roux, Jennifer Foster, Joachim Wagner, Rasul Samad Zadeh Kaljahi, and Anton Bryl. 2012. DCU-Paris13 systems for the SANCL 2012 shared task. Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL).

Bing Liu. 2010. Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing*.

Shotaro Matsumoto, Hiroya Takamura, and Manubu Okumura, 2005. *Advances in Knowledge Discovery and Data Mining*, volume 3518 of *Lecture Notes in Computer Science*, chapter Sentiment Classification Using Word Sub-sequences and Dependency Subtrees, pages 301–311.

Peter Norvig. 2012. How to write a spelling corrector. `http://norvig.com/spell-correct.html`. [Online; accessed 2014-03-19].

Po Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86.

John C. Platt. 1998. Fast training of support vector machines using sequential minimal optimization. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 185–208.

Maria Pontiki, Dimitrios Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*.

Raphael Rubino, Jennifer Foster, Joachim Wagner, Johann Roturier, Rasul Samad Zadeh Kaljahi, and Fred Hollowood. 2012. Dcu-symantec submission for the wmt 2012 quality estimation task. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 138–144.

Raphael Rubino, Joachim Wagner, Jennifer Foster, Johann Roturier, Rasoul Samad Zadeh Kaljahi, and Fred Hollowood. 2013. DCU-Symantec at the WMT 2013 quality estimation shared task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 392–397.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49.

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, pages 1631–1642.

Milan Tofiloski, Julian Brooke, and Maite Taboada. 2009. A syntactic and lexical-based discourse segmenter. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 77–80.

Peter Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classication of reviews. In *Proceedings of the ACL*, pages 417–424.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354.