Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis

Theresa Wilson
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA 15260
twilson@cs.pitt.edu

Janyce Wiebe
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260
wiebe@cs.pitt.edu

Paul Hoffmann
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA 15260
hoffmanp@cs.pitt.edu

Abstract

This paper presents a new approach to phrase-level sentiment analysis that first determines whether an expression is neutral or polar and then disambiguates the polarity of the polar expressions. With this approach, the system is able to automatically identify the *contextual polarity* for a large subset of sentiment expressions, achieving results that are significantly better than baseline.

1 Introduction

Sentiment analysis is the task of identifying positive and negative opinions, emotions, and evaluations. Most work on sentiment analysis has been done at the document level, for example distinguishing positive from negative reviews. However, tasks such as multi-perspective question answering and summarization, opinion-oriented information extraction, and mining product reviews require sentence-level or even phrase-level sentiment analysis. For example, if a question answering system is to successfully answer questions about people's opinions, it must be able to pinpoint expressions of positive and negative sentiments, such as we find in the sentences below:

(1) African observers $generally\ approved^{+}$ of his victory while Western governments $denounced^{-}$ it.

(2) A succession of officers filled the TV screen to say they $supported^{+}$ the people and that the killings were "$not\ tolerable^{-}$."

(3) "We $don't\ \ hate^{+}$ the sinner," he says, "but we $hate^{-}$ the sin."

A typical approach to sentiment analysis is to start with a lexicon of positive and negative words and phrases. In these lexicons, entries are tagged with their a priori *prior polarity*: out of context, does the word seem to evoke something positive or something negative. For example, *beautiful* has a positive prior polarity, and *horrid* has a negative prior polarity. However, the *contextual polarity* of the phrase in which a word appears may be different from the word's prior polarity. Consider the underlined polarity words in the sentence below:

(4) Philip Clapp, president of the National Environment Trust, sums up well the general thrust of the reaction of environmental movements: "There is no reason at all to believe that the polluters are suddenly going to become reasonable."

Of these words, "Trust," "well," "reason," and "reasonable" have positive prior polarity, but they are not all being used to express positive sentiments. The word "reason" is negated, making the contextual polarity negative. The phrase "no reason at all to believe" changes the polarity of the proposition that follows; because "reasonable" falls within this proposition, its contextual polarity becomes negative. The word "Trust" is simply part of a referring expression and is not being used to express a sentiment; thus, its contextual polarity is neutral. Similarly for "polluters": in the context of the article, it simply refers to companies that pollute. Only "well" has the same prior and contextual polarity.

Many things must be considered in phrase-level sentiment analysis. Negation may be local (e.g., **not** *good*), or involve longer-distance dependencies such as the negation of the proposition (e.g., **does not** *look very good*) or the negation of the subject (e.g.,

Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 347–354, Vancouver, October 2005.

no one *thinks that it's good*). In addition, certain phrases that contain negation words intensify rather than change polarity (e.g., **not only** *good but amazing*). Contextual polarity may also be influenced by modality (e.g., whether the proposition is asserted to be real (*realis*) or not real (*irrealis*) – *no reason at all to believe* is irrealis, for example); word sense (e.g., *Environmental* **Trust** versus *He has won the people's* **trust**); the syntactic role of a word in the sentence (e.g., **polluters** *are* versus *they are* **polluters**); and diminishers such as *little* (e.g., **little** *truth*, **little** *threat*). (See (Polanya and Zaenen, 2004) for a more detailed discussion of contextual polarity influencers.)

This paper presents new experiments in automatically distinguishing prior and contextual polarity. Beginning with a large stable of clues marked with prior polarity, we identify the contextual polarity of the phrases that contain instances of those clues in the corpus. We use a two-step process that employs machine learning and a variety of features. The first step classifies each phrase containing a clue as neutral or polar. The second step takes all phrases marked in step one as polar and disambiguates their contextual polarity (*positive*, *negative*, *both*, or *neutral*). With this approach, the system is able to automatically identify the contextual polarity for a large subset of sentiment expressions, achieving results that are significantly better than baseline. In addition, we describe new manual annotations of contextual polarity and a successful inter-annotator agreement study.

2 Manual Annotation Scheme

To create a corpus for the experiments below, we added contextual polarity judgments to existing annotations in the Multi-perspective Question Answering (MPQA) Opinion Corpus[1], namely to the annotations of *subjective expressions*[2]. A subjective expression is any word or phrase used to express an opinion, emotion, evaluation, stance, speculation, etc. A general covering term for such states is *private state* (Quirk et al., 1985). In the MPQA Corpus, subjective expressions of varying lengths are marked, from single words to long phrases.

For this work, our focus is on *sentiment expressions* – positive and negative expressions of emotions, evaluations, and stances. As these are types of subjective expressions, to create the corpus, we just needed to manually annotate the existing subjective expressions with their contextual polarity.

In particular, we developed an annotation scheme[3] for marking the contextual polarity of subjective expressions. Annotators were instructed to tag the polarity of subjective expressions as *positive*, *negative*, *both*, or *neutral*. The *positive* tag is for positive emotions (*I'm happy*), evaluations (*Great idea!*), and stances (*She supports the bill*). The *negative* tag is for negative emotions (*I'm sad*), evaluations (*Bad idea!*), and stances (*She's against the bill*). The *both* tag is applied to sentiment expressions that have both positive and negative polarity. The *neutral* tag is used for all other subjective expressions: those that express a different type of subjectivity such as speculation, and those that do not have positive or negative polarity.

Below are examples of contextual polarity annotations. The tags are in boldface, and the subjective expressions with the given tags are underlined.

> (5) Thousands of coup supporters celebrated (**positive**) overnight, waving flags, blowing whistles ...

> (6) The criteria set by Rice are the following: the three countries in question are repressive (**negative**) and grave human rights violators (**negative**) ...

> (7) Besides, politicians refer to good and evil (**both**) only for purposes of intimidation and exaggeration.

> (8) Jerome says the hospital feels (**neutral**) no different than a hospital in the states.

The annotators were asked to judge the contextual polarity of the sentiment that is ultimately being conveyed by the subjective expression, i.e., once the sentence has been fully interpreted. Thus, the subjective expression, *they have not succeeded, and*

[1]The MPQA Corpus is described in (Wiebe et al., 2005) and available at nrrc.mitre.org/NRRC/publications.htm.

[2]In the MPQA Corpus, subjective expressions are *direct subjective* expressions with *non-neutral expression intensity*, plus all the *expressive subjective elements*. Please see (Wiebe et al., 2005) for more details on the existing annotations in the MPQA Corpus.

[3]The annotation instructions are available at http://www.cs.pitt.edu/~twilson.

will never succeed, was marked as positive in the sentence, *They have not succeeded, and will never succeed, in breaking the will of this valiant people*. The reasoning is that breaking the will of a valiant people is negative; hence, not succeeding in breaking their will is positive.

3 Agreement Study

To measure the reliability of the polarity annotation scheme, we conducted an agreement study with two annotators, using 10 documents from the MPQA Corpus. The 10 documents contain 447 subjective expressions. Table 1 shows the contingency table for the two annotators' judgments. Overall agreement is 82%, with a Kappa (κ) value of 0.72.

	Neutral	Positive	Negative	Both	Total
Neutral	123	14	24	0	161
Positive	16	73	5	2	96
Negative	14	2	167	1	184
Both	0	3	0	3	6
Total	153	92	196	6	447

Table 1: Agreement for Subjective Expressions (Agreement: 82%, κ: 0.72)

For 18% of the subjective expressions, at least one annotator used an *uncertain* tag when marking polarity. If we consider these cases to be borderline and exclude them from the study, percent agreement increases to 90% and Kappa rises to 0.84. Thus, the annotator agreement is especially high when both are certain. (Note that all annotations are included in the experiments described below.)

4 Corpus

In total, 15,991 subjective expressions from 425 documents (8,984 sentences) were annotated with contextual polarity as described above. Of these sentences, 28% contain no subjective expressions, 25% contain only one, and 47% contain two or more. Of the 4,247 sentences containing two or more subjective expressions, 17% contain mixtures of positive and negative expressions, and 62% contain mixtures of polar (positive/negative/both) and neutral subjective expressions.

The annotated documents are divided into two sets. The first (66 documents/1,373 sentences/2,808 subjective expressions) is a development set, used for data exploration and feature development. We use the second set (359 documents/7,611 sentences/13,183 subjective expressions) in 10-fold cross-validation experiments, described below.

5 Prior-Polarity Subjectivity Lexicon

For the experiments in this paper, we use a lexicon of over 8,000 *subjectivity clues*. Subjectivity clues are words and phrases that may be used to express private states, i.e., they have subjective usages (though they may have objective usages as well). For this work, only single-word clues are used.

To compile the lexicon, we began with a list of subjectivity clues from (Riloff and Wiebe, 2003). The words in this list were grouped in previous work according to their reliability as subjectivity clues. Words that are subjective in most contexts were marked strongly subjective (*strongsubj*), and those that may only have certain subjective usages were marked weakly subjective (*weaksubj*).

We expanded the list using a dictionary and a thesaurus, and also added words from the General Inquirer positive and negative word lists (General-Inquirer, 2000) which we judged to be potentially subjective. We also gave the new words reliability tags, either *strongsubj* or *weaksubj*.

The next step was to tag the clues in the lexicon with their prior polarity. For words that came from positive and negative word lists (General-Inquirer, 2000; Hatzivassiloglou and McKeown, 1997), we largely retained their original polarity, either *positive* or *negative*. We assigned the remaining words one of the tags *positive*, *negative*, *both* or *neutral*.

By far, the majority of clues, 92.8%, are marked as having either positive (33.1%) or negative (59.7%) prior polarity. Only a small number of clues (0.3%) are marked as having both positive and negative polarity. 6.9% of the clues in the lexicon are marked as neutral. Examples of these are verbs such as *feel*, *look*, and *think*, and intensifiers such as *deeply*, *entirely*, and *practically*. These words are included because, although their prior polarity is neutral, they are good clues that a sentiment is being expressed (e.g., **feels** *slighted*, **look** *forward to*). Including them increases the coverage of the system.

6 Experiments

The goal of the experiments described below is to classify the contextual polarity of the expressions that contain instances of the subjectivity clues in our lexicon. What the system specifically does is give each clue instance its own label. Note that the system does not try to identify expression boundaries. Doing so might improve performance and is a promising avenue for future research.

6.1 Definition of the Gold Standard

We define the gold standard used to train and test the system in terms of the manual annotations described in Section 2.

The gold standard class of a clue instance that is not in a subjective expression is *neutral*: since the clue is not even in a subjective expression, it is not contained in a sentiment expression.

Otherwise, if a clue instance appears in just one subjective expression (or in multiple subjective expressions with the same contextual polarity), then the class assigned to the clue instance is the class of the subjective expression(s). If a clue appears in at least one positive and one negative subjective expression (or in a subjective expression marked as *both*), then its class is *both*. If it is in a mixture of negative and neutral subjective expressions, its class is *negative*; if it is in a mixture of positive and neutral subjective expressions, its class is *positive*.

6.2 Performance of a Prior-Polarity Classifier

An important question is how useful prior polarity alone is for identifying contextual polarity. To answer this question, we create a classifier that simply assumes that the contextual polarity of a clue instance is the same as the clue's prior polarity, and we explore the classifier's performance on the development set.

This simple classifier has an accuracy of 48%. From the confusion matrix given in Table 2, we see that 76% of the errors result from words with non-neutral prior polarity appearing in phrases with neutral contextual polarity.

		Prior-Polarity Classifier				
		Neut	Pos	Neg	Both	Total
Gold	Neut	798	784	698	4	2284
	Pos	81	371	40	0	492
	Neg	149	181	622	0	952
	Both	4	11	13	5	33
	Total	1032	1347	1373	9	3761

Table 2: Confusion matrix for the prior-polarity classifier on the development set.

6.3 Contextual Polarity Disambiguation

The fact that words with non-neutral prior polarity so frequently appear in neutral contexts led us to adopt a two-step approach to contextual polarity disambiguation. For the first step, we concentrate on whether clue instances are neutral or polar in context (where *polar in context* refers to having a contextual polarity that is positive, negative or both). For the second step, we take all clue instances marked as polar in step one, and focus on identifying their contextual polarity. For both steps, we develop classifiers using the BoosTexter AdaBoost.HM (Schapire and Singer, 2000) machine learning algorithm with 5000 rounds of boosting. The classifiers are evaluated in 10-fold cross-validation experiments.

6.3.1 Neutral-Polar Classification

The neutral-polar classifier uses 28 features, listed in Table 3.

Word Features: *Word context* is a bag of three word tokens: the previous word, the word itself, and the next word. The *prior polarity* and *reliability class* are indicated in the lexicon.

Modification Features: These are binary relationship features. The first four involve relationships with the word immediately before or after: if the word is a noun preceded by an adjective, if the preceding word is an adverb other than *not*, if the preceding word is an intensifier, and if the word itself is an intensifier. A word is considered an intensifier if it appears in a list of intensifiers and if it precedes a word of the appropriate part-of-speech (e.g., an intensifier adjective must come before a noun).

The *modify* features involve the dependency parse tree for the sentence, obtained by first parsing the sentence (Collins, 1997) and then converting the tree into its dependency representation (Xia and Palmer, 2001). In a dependency representation, every node in the tree structure is a surface word (i.e., there are no abstract nodes such as NP or VP). The edge between a parent and a child specifies the grammatical relationship between the two words. Figure 1 shows

Word Features	Sentence Features	Structure Features
word token	strongsubj clues in current sentence: count	in subject: binary
word part-of-speech	strongsubj clues in previous sentence: count	in copular: binary
word context	strongsubj clues in next sentence: count	in passive: binary
prior polarity: positive, negative, both, neutral	weaksubj clues in current sentence: count	
reliability class: strongsubj or weaksubj	weaksubj clues in previous sentence: count	
Modification Features	weaksubj clues in next sentence: count	**Document Feature**
preceeded by adjective: binary	adjectives in sentence: count	document topic
preceeded by adverb (other than not): binary	adverbs in sentence (other than not): count	
preceeded by intensifier: binary	cardinal number in sentence: binary	
is intensifier: binary	pronoun in sentence: binary	
modifies strongsubj: binary	modal in sentence (other than will): binary	
modifies weaksubj: binary		
modified by strongsubj: binary		
modified by weaksubj: binary		

Table 3: Features for neutral-polar classification

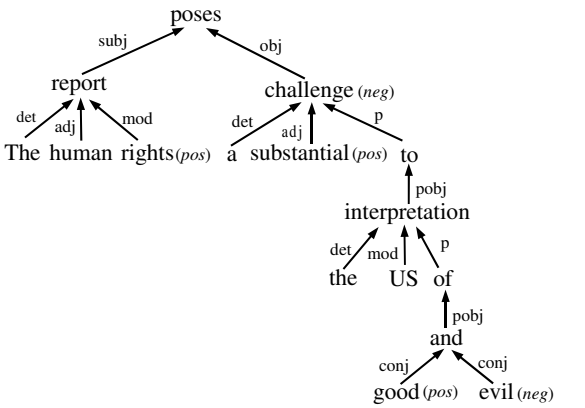

Figure 1: The dependency tree for the sentence *The human rights report poses a substantial challenge to the US interpretation of good and evil.* Prior polarity is marked in parentheses for words that match clues from the lexicon.

an example. The *modifies strongsubj/weaksubj* features are true if the word and its parent share an *adj*, *mod* or *vmod* relationship, and if its parent is an instance of a clue from the lexicon with strongsubj/weaksubj reliability. The *modified by strongsubj/weaksubj* features are similar, but look for relationships and clues in the word's children.

Structure Features: These are binary features that are determined by starting with the word instance and climbing up the dependency parse tree toward the root, looking for particular relationships, words, or patterns. The *in subject* feature is true if we find a *subj* relationship. The *in copular* feature is true if *in subject* is false and if a node along the path is both a main verb and a copular verb. The *in passive* features is true if a passive verb pattern is found on the climb.

Sentence Features: These are features that were found useful for sentence-level subjectivity classification by Wiebe and Riloff (2005). They include counts of strongsubj and weaksubj clues in the current, previous and next sentences, counts of adjectives and adverbs other than *not* in the current sentence, and binary features to indicate whether the sentence contains a pronoun, a cardinal number, and a modal other than *will*.

Document Feature: There is one document feature representing the topic of the document. A document may belong to one of 15 topics ranging from specific (e.g., the 2002 presidential election in Zimbabwe) to more general (e.g., economics) topics.

Table 4 gives neutral-polar classification results for the 28-feature classifier and two simpler classifiers that provide our baselines. The first row in the table lists the results for a classifier that uses just one feature, the word token. The second row shows the results for a classifier that uses both the word token and the word's prior polarity as features. The results for the 28-feature classifier are listed in the last row. The 28-feature classifier performs significantly better (1-tailed t-test, $p \leq .05$) than the two simpler classifiers, as measured by accuracy, polar F-measure, and neutral F-measure ($\beta = 1$). It has an accuracy of 75.9%, with a polar F-measure of 63.4 and a neutral F-measure of 82.1.

Focusing on the metrics for polar expressions, it's interesting to note that using just the word token as a feature produces a classifier with a precision slightly better than the 28-feature classifier, but with a recall that is 20% lower. Adding a feature for the prior

Word Features
word token
word prior polarity: positive, negative, both, neutral
Polarity Features
negated: binary
negated subject: binary
modifies polarity: positive, negative, neutral, both, notmod
modified by polarity: positive, negative, neutral, both, notmod
conj polarity: positive, negative, neutral, both, notmod
general polarity shifter: binary
negative polarity shifter: binary
positive polarity shifter: binary

Table 6: Features for polarity classification

polarity improves recall so that it is only 4.4% lower, but this hurts precision, which drops to 4.2% lower than the 28-feature classifier's precision. It is only with all the features that we get the best result, good precision with the highest recall.

The clues in the prior-polarity lexicon have 19,506 instances in the test set. According to the 28-feature neutral-polar classifier, 5,671 of these instances are polar in context. It is these clue instances that are passed on to the second step in the contextual disambiguation process, polarity classification.

6.3.2 Polarity Classification

Ideally, this second step in the disambiguation process would be a three-way classification task, determining whether the contextual polarity is positive, negative or both. However, although the majority of neutral expressions have been filtered out by the neutral-polar classification in step one, a number still remain. So, for this step, the polarity classification task remains four-way: positive, negative, both, and neutral.

Table 6 lists the features used by the polarity classifier. *Word token* and *word prior polarity* are unchanged from the neutral-polar classifier. *Negated* is a binary feature that captures whether the word is being locally negated: its value is true if a negation word or phrase is found within the four preceeding words or in any of the word's children in the dependency tree, and if the negation word is not in a phrase that intensifies rather than negates (e.g., *not only*). The *negated subject* feature is true if the subject of the clause containing the word is negated.

The *modifies polarity*, *modified by polarity*, and *conj polarity* features capture specific relationships between the word instance and other polarity words it may be related to. If the word and its parent in the dependency tree share an *obj*, *adj*, *mod*, or *vmod* relationship, the *modifies polarity* feature is set to the prior polarity of the word's parent (if the parent is not in our prior-polarity lexicon, its prior polarity is set to neutral). The *modified by polarity* feature is similar, looking for *adj*, *mod*, and *vmod* relationships and polarity clues within the word's children. The *conj polarity* feature determines if the word is in a conjunction. If so, the value of this feature is its sibling's prior polarity (as above, if the sibling is not in the lexicon, its prior polarity is neutral). Figure 1 helps to illustrate these features: *modifies polarity* is negative for the word "substantial," *modified by polarity* is positive for the word "challenge," and *conj polarity* is negative for the word "good."

The last three polarity features look in a window of four words before, searching for the presence of particular types of polarity influencers. *General polarity shifters* reverse polarity (e.g., *little* truth, *little* threat). *Negative polarity shifters* typically make the polarity of an expression negative (e.g., *lack* of understanding). *Positive polarity shifters* typically make the polarity of an expression positive (e.g., *abate* the damage).

The polarity classification results for this second step in the contextual disambiguation process are given in Table 5. Also listed in the table are results for the two simple classifiers that provide our baselines. The first line in Table 5 lists the results for the classifier that uses just one feature, the word token. The second line shows the results for the classifier that uses both the word token and the word's prior polarity as features. The last line shows the results for the polarity classifier that uses all 10 features from Table 6.

Mirroring the results from step one, the more complex classifier performs significantly better than the simpler classifiers, as measured by accuracy and all of the F-measures. The 10-feature classifier achieves an accuracy of 65.7%, which is 4.3% higher than the more challenging baseline provided by the word + prior polarity classifier. Positive F-measure is 65.1 (5.7% higher); negative F-measure is 77.2 (2.3% higher); and neutral F-measure is 46.2 (13.5% higher).

Focusing on the metrics for positive and negative expressions, we again see that the simpler classifiers

	Acc	Polar Rec	Polar Prec	Polar F	Neut Rec	Neut Prec	Neut F
word token	73.6	45.3	72.2	55.7	89.9	74.0	81.2
word+priorpol	74.2	54.3	68.6	60.6	85.7	76.4	80.7
28 features	75.9	56.8	71.6	63.4	87.0	77.7	82.1

Table 4: Results for Step 1 Neutral-Polar Classification

		Positive			Negative			Both			Neutral		
	Acc	Rec	Prec	F	Rec	Prec	F	Rec	Prec	F	Rec	Prec	F
word token	61.7	59.3	63.4	61.2	83.9	64.7	73.1	9.2	35.2	14.6	30.2	50.1	37.7
word+priorpol	63.0	69.4	55.3	61.6	80.4	71.2	75.5	9.2	35.2	14.6	33.5	51.8	40.7
10 features	65.7	67.1	63.3	65.1	82.1	72.9	77.2	11.2	28.4	16.1	41.4	52.4	46.2

Table 5: Results for Step 2 Polarity Classification.

Experiment	Features Removed
AB1	negated, negated subject
AB2	modifies polarity, modified by polarity
AB3	conj polarity
AB4	general, negative, and positive polarity shifters

Table 7: Features for polarity classification

take turns doing better or worse for precision and recall. Using just the word token, positive precision is slightly higher than for the 10-feature classifier, but positive recall is 11.6% lower. Add the prior polarity, and positive recall improves, but at the expense of precision, which is 12.6% lower than for the 10-feature classifier. The results for negative expressions are similar. The word-token classifier does well on negative recall but poorly on negative precision. When prior polarity is added, negative recall improves but negative precision drops. It is only with the addition of the polarity features that we achieve both higher precisions and higher recalls.

To explore how much the various polarity features contribute to the performance of the polarity classifier, we perform four experiments. In each experiment, a different set of polarity features is excluded, and the polarity classifier is retrained and evaluated. Table 7 lists the features that are removed for each experiment.

The only significant difference in performance in these experiments is neutral F-measure when the modification features (AB2) are removed. These ablation experiments show that the combination of features is needed to achieve significant results over baseline for polarity classification.

7 Related Work

Much work on sentiment analysis classifies documents by their overall sentiment, for example determining whether a review is positive or negative (e.g., (Turney, 2002; Dave et al., 2003; Pang and Lee, 2004; Beineke et al., 2004)). In contrast, our experiments classify individual words and phrases. A number of researchers have explored learning words and phrases with *prior* positive or negative polarity (another term is *semantic orientation*) (e.g., (Hatzivassiloglou and McKeown, 1997; Kamps and Marx, 2002; Turney, 2002)). In contrast, we begin with a lexicon of words with established prior polarities, and identify the *contextual polarity* of phrases in which instances of those words appear in the corpus. To make the relationship between that task and ours clearer, note that some word lists used to evaluate methods for recognizing prior polarity are included in our prior-polarity lexicon (General Inquirer lists (General-Inquirer, 2000) used for evaluation by Turney, and lists of manually identified positive and negative adjectives, used for evaluation by Hatzivassiloglou and McKeown).

Some research classifies the sentiments of sentences. Yu and Hatzivassiloglou (2003), Kim and Hovy (2004), Hu and Liu (2004), and Grefenstette et al. (2001)[4] all begin by first creating prior-polarity lexicons. Yu and Hatzivassiloglou then assign a sentiment to a sentence by averaging the prior semantic orientations of instances of lexicon words in the sentence. Thus, they do not identify the contextual polarity of individual phrases containing clues, as we

[4] In (Grefenstette et al., 2001), the units that are classified are fixed windows around named entities rather than sentences.

do in this paper. Kim and Hovy, Hu and Liu, and Grefenstette et al. multiply or count the prior polarities of clue instances in the sentence. They also consider local negation to reverse polarity. However, they do not use the other types of features in our experiments, and they restrict their tags to *positive* and *negative* (excluding our *both* and *neutral* categories). In addition, their systems assign one sentiment per sentence; our system assigns contextual polarity to individual expressions. As seen above, sentences often contain more than one sentiment expression.

Nasukawa, Yi, and colleagues (Nasukawa and Yi, 2003; Yi et al., 2003) classify the contextual polarity of sentiment expressions, as we do. Thus, their work is probably most closely related to ours. They classify expressions that are about specific items, and use manually developed patterns to classify polarity. These patterns are high-quality, yielding quite high precision, but very low recall. Their system classifies a much smaller proportion of the sentiment expressions in a corpus than ours does.

8 Conclusions

In this paper, we present a new approach to phrase-level sentiment analysis that first determines whether an expression is neutral or polar and then disambiguates the polarity of the polar expressions. With this approach, we are able to automatically identify the *contextual polarity* for a large subset of sentiment expressions, achieving results that are significantly better than baseline.

9 Acknowledgments

This work was supported in part by the NSF under grant IIS-0208798 and by the Advanced Research and Development Activity (ARDA).

References

P. Beineke, T. Hastie, and S. Vaithyanathan. 2004. The sentimental factor: Improving review classification via human-provided information. In *ACL-2004*.

M. Collins. 1997. Three generative, lexicalised models for statistical parsing. In *ACL-1997*.

K. Dave, S. Lawrence, and D. M. Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *WWW-2003*.

The General-Inquirer. 2000. *http://www.wjh.harvard.edu/~inquirer/spreadsheet_guide.htm*.

G. Grefenstette, Y. Qu, J.G. Shanahan, and D.A. Evans. 2001. Coupling niche browsers and affect analysis for an opinion mining application. In *RIAO-2004*.

V. Hatzivassiloglou and K. McKeown. 1997. Predicting the semantic orientation of adjectives. In *ACL-1997*.

M. Hu and B. Liu. 2004. Mining and summarizing customer reviews. In *KDD-2004*.

J. Kamps and M. Marx. 2002. Words with attitude. In *1st International WordNet Conference*.

S-M. Kim and E. Hovy. 2004. Determining the sentiment of opinions. In *Coling 2004*.

T. Nasukawa and J. Yi. 2003. Sentiment analysis: Capturing favorability using natural language processing. In *K-CAP 2003*.

B. Pang and L. Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL-2004*.

L. Polanya and A. Zaenen. 2004. Contextual valence shifters. In *Working Notes — Exploring Attitude and Affect in Text (AAAI Spring Symposium Series)*.

R. Quirk, S. Greenbaum, G. Leech, and J. Svartvik. 1985. *A Comprehensive Grammar of the English Language.* Longman, New York.

E. Riloff and J. Wiebe. 2003. Learning extraction patterns for subjective expressions. In *EMNLP-2003*.

R. E. Schapire and Y. Singer. 2000. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

P. Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *ACL-2002*.

J. Wiebe and E. Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. In *CICLing-2005*.

J. Wiebe, T. Wilson, and C. Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evalution (formerly Computers and the Humanities)*, 1(2).

F. Xia and M. Palmer. 2001. Converting dependency structures to phrase structures. In *HLT-2001*.

J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack. 2003. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *IEEE ICDM-2003*.

H. Yu and V. Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *EMNLP-2003*.