Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis

Theresa Wilson*
University of Edinburgh

Janyce Wiebe**
University of Pittsburgh

Paul Hoffmann**
University of Pittsburgh

Many approaches to automatic sentiment analysis begin with a large lexicon of words marked with their prior polarity (also called semantic orientation). However, the contextual polarity of the phrase in which a particular instance of a word appears may be quite different from the word's prior polarity. Positive words are used in phrases expressing negative sentiments, or vice versa. Also, quite often words that are positive or negative out of context are neutral in context, meaning they are not even being used to express a sentiment. The goal of this work is to automatically distinguish between prior and contextual polarity, with a focus on understanding which features are important for this task. Because an important aspect of the problem is identifying when polar terms are being used in neutral contexts, features for distinguishing between neutral and polar instances are evaluated, as well as features for distinguishing between positive and negative contextual polarity. The evaluation includes assessing the performance of features across multiple machine learning algorithms. For all learning algorithms except one, the combination of all features together gives the best performance. Another facet of the evaluation considers how the presence of neutral instances affects the performance of features for distinguishing between positive and negative polarity. These experiments show that the presence of neutral instances greatly degrades the performance of these features, and that perhaps the best way to improve performance across all polarity classes is to improve the system's ability to identify when an instance is neutral.

1. Introduction

Sentiment analysis is a type of subjectivity analysis (Wiebe 1994) that focuses on identifying positive and negative opinions, emotions, and evaluations expressed in natural language. It has been a central component in applications ranging from recognizing

* School of Informatics, Edinburgh EH8 9LW, U.K. E-mail: twilson@inf.ed.ac.uk.
** Department of Computer Science, Pittsburgh, PA 15260, USA. E-mail: {wiebe,hoffmanp}@cs.pitt.edu.

Submission received: 14 November 2006; revised submission received: 8 March 2008; accepted for publication: 16 April 2008.

inflammatory messages (Spertus 1997), to tracking sentiments over time in online discussions (Tong 2001), to classifying positive and negative reviews (Pang, Lee, and Vaithyanathan 2002; Turney 2002). Although a great deal of work in sentiment analysis has targeted documents, applications such as opinion question answering (Yu and Hatzivassiloglou 2003; Maybury 2004; Stoyanov, Cardie, and Wiebe 2005) and review mining to extract opinions about companies and products (Morinaga et al. 2002; Nasukawa and Yi 2003) require sentence-level or even phrase-level analysis. For example, if a question answering system is to successfully answer questions about people's opinions, it must be able not only to pinpoint expressions of positive and negative sentiments, such as we find in sentence (1), but also to determine when an opinion is *not* being expressed by a word or phrase that typically does evoke one, such as *condemned* in sentence (2).

(1) African observers generally approved (**positive**) of his victory while Western governments denounced (**negative**) it.

(2) Gavin Elementary School was condemned in April 2004.

A common approach to sentiment analysis is to use a lexicon with information about which words and phrases are positive and which are negative. This lexicon may be manually compiled, as is the case with the General Inquirer (Stone et al. 1966), a resource often used in sentiment analysis. Alternatively, the information in the lexicon may be acquired automatically. Acquiring the polarity of words and phrases is itself an active line of research in the sentiment analysis community, pioneered by the work of Hatzivassiloglou and McKeown (1997) on predicting the polarity or semantic orientation of adjectives. Various techniques have been proposed for learning the polarity of words. They include corpus-based techniques, such as using constraints on the co-occurrence in conjunctions of words with similar or opposite polarity (Hatzivassiloglou and McKeown 1997) and statistical measures of word association (Turney and Littman 2003), as well as techniques that exploit information about lexical relationships (Kamps and Marx 2002; Kim and Hovy 2004) and glosses (Esuli and Sebastiani 2005; Andreevskaia and Bergler 2006) in resources such as WordNet.

Acquiring the polarity of words and phrases is undeniably important, and there are still open research challenges, such as addressing the sentiments of different senses of words (Esuli and Sebastiani 2006b; Wiebe and Mihalcea 2006), and so on. However, what the polarity of a given word or phrase is when it is used in a particular context is another problem entirely. Consider, for example, the underlined positive and negative words in the following sentence.

(3) 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."

The first underlined word is *Trust*. Although many senses of the word *trust* express a positive sentiment, in this case, the word is not being used to express a sentiment at all. It is simply part of an expression referring to an organization that has taken on the charge of caring for the environment. The adjective *well* is considered positive, and indeed it is positive in this context. However, the same is not true for the words *reason*

and *reasonable.* Out of context, we would consider both of these words to be positive.[1] In context, the word *reason* is being negated, changing its polarity from positive to negative. The phrase *no reason at all to believe* changes the polarity of the proposition that follows; because *reasonable* falls within this proposition, its polarity becomes negative. The word *polluters* has a negative connotation, but here in the context of the discussion of the article and its position in the sentence, *polluters* is being used less to express a sentiment and more to objectively refer to companies that pollute. To clarify how the polarity of *polluters* is affected by its subject role, consider the purely negative sentiment that emerges when it is used as an object: *They are polluters*.

We call the polarity that would be listed for a word in a lexicon the word's **prior polarity**, and we call the polarity of the expression in which a word appears, considering the context of the sentence and document, the word's **contextual polarity**. Although words often do have the same prior and contextual polarity, many times a word's prior and contextual polarities differ. Words with a positive prior polarity may have a negative contextual polarity, or vice versa. Quite often words that are positive or negative out of context are *neutral* in context, meaning that they are not even being used to express a sentiment. Similarly, words that are neutral out of context, neither positive or negative, may combine to create a positive or negative expression in context.

The focus of this work is on the recognition of contextual polarity—in particular, disambiguating the contextual polarity of words with positive or negative prior polarity. We begin by presenting an annotation scheme for marking sentiment expressions and their contextual polarity in the Multi-perspective Question Answering (MPQA) opinion corpus. We show that, given a set of subjective expressions (identified from the existing annotations in the MPQA corpus), contextual polarity can be annotated reliably.

Using the contextual polarity annotations, we conduct experiments in automatically distinguishing between prior and contextual polarity. Beginning with a large lexicon of clues tagged with prior polarity, we identify the contextual polarity of the instances of those clues in the corpus. The process that we use has two steps, first classifying each clue as being in a neutral or polar phrase, and then disambiguating the contextual polarity of the clues marked as polar. For each step in the process, we experiment with a variety of features and evaluate the performance of the features using several different machine learning algorithms.

Our experiments reveal a number of interesting findings. First, being able to accurately identify neutral contextual polarity—when a positive or negative clue is *not* being used to express a sentiment—is an important aspect of the problem. The importance of neutral examples has previously been noted for classifying the sentiment of documents (Koppel and Schler 2006), but ours is the first work to explore how neutral instances affect classifying the contextual polarity of words and phrases. In particular, we found that the performance of features for distinguishing between positive and negative polarity greatly degrades when neutral instances are included in the experiments.

We also found that achieving the best performance for recognizing contextual polarity requires a wide variety of features. This is particularly true for distinguishing

1 It is open to question whether *reason* should be listed as positive in a sentiment lexicon, because the more frequent senses of *reason* involve intention, not sentiment. However, any existing sentiment lexicon one would start with will have some noise and errors. The task in this article is to disambiguate instances of the entries in a given sentiment lexicon.

between neutral and polar instances. Although some features help to increase polar or neutral recall or precision, it is only the combination of features together that achieves significant improvements in accuracy over the baselines. Our experiments show that for distinguishing between positive and negative instances, features capturing negation are clearly the most important. However, there is more to the story than simple negation. Features that capture relationships between instances of clues also perform well, indicating that identifying features that represent more complex interdependencies between sentiment clues may be an important avenue for future research.

The remainder of this article is organized as follows. Section 2 gives an overview of some of the things that can influence contextual polarity. In Section 3, we describe our corpus and present our annotation scheme and inter-annotator agreement study for marking contextual polarity. Sections 4 and 5 describe the lexicon used in our experiments and how the contextual polarity annotations are used to determine the gold-standard tags for instances from the lexicon. In Section 6, we consider what kind of performance can be expected from a simple, prior-polarity classifier. Section 7 describes the features that we use for recognizing contextual polarity, and our experiments and results are presented in Section 8. In Section 9 we discuss related work, and we conclude in Section 10.

2. Polarity Influencers

Phrase-level sentiment analysis is not a simple problem. Many things besides negation can influence contextual polarity, and even negation is not always straightforward. 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., *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: 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* vs. *He has won the people's trust*); the syntactic role of a word in the sentence: whether the word is the subject or object of a copular verb (consider *polluters are* versus *they are polluters*); and diminishers such as *little* (e.g., *little truth*, *little threat*). Polanyi and Zaenen (2004) give a detailed discussion of many of these types of polarity influencers. Many of these contextual polarity influencers are represented as features in our experiments.

Contextual polarity may also be influenced by the domain or topic. For example, the word *cool* is positive if used to describe a car, but it is negative if it is used to describe someone's demeanor. Similarly, a word such as *fever* is unlikely to be expressing a sentiment when used in a medical context. We use one feature in our experiments to represent the topic of the document.

Another important aspect of contextual polarity is the perspective of the person who is expressing the sentiment. For example, consider the phrase *failed to defeat* in the sentence *Israel failed to defeat Hezbollah*. From the perspective of Israel, *failed to defeat* is negative. From the perspective of Hezbollah, *failed to defeat* is positive. Therefore, the contextual polarity of this phrase ultimately depends on the perspective of who is expressing the sentiment. Although automatically detecting this kind of pragmatic influence on polarity is beyond the scope of this work, this as well as the other types of polarity influencers all are considered when annotating contextual polarity.

3. Data and Annotations

For the experiments in this work, we need a corpus that is annotated comprehensively for sentiment expressions and their contextual polarity. Rather than building a corpus from scratch, we chose to add contextual polarity annotations to the existing annotations in the Multi-perspective Question Answering (MPQA) opinion corpus[2] (Wiebe, Wilson, and Cardie 2005).

The MPQA corpus is a collection of English-language versions of news documents from the world press. The documents contain detailed, expression-level annotations of attributions and **private states** (Quirk et al. 1985). Private states are mental and emotional states; they include beliefs, speculations, intentions, and sentiments, among others. Although sentiments are not distinguished from other types of private states in the existing annotations, they are a subset of what already is annotated. This makes the annotations in the MPQA corpus a good starting point for annotating sentiment expressions and their contextual polarity.

3.1 Annotation Scheme

When developing our annotation scheme for sentiment expressions and contextual polarity, there were three main questions to address. First, which of the existing annotations in the MPQA corpus have the possibility of being sentiment expressions? Second, which of the possible sentiment expressions actually are expressing sentiments? Third, what coding scheme should be used for marking contextual polarity?

The MPQA annotation scheme has four types of annotations: objective speech event frames, two types of private state frames, and agent frames that are used for marking speakers of speech events and experiencers of private states. A full description of the MPQA annotation scheme and an agreement study evaluating key aspects of the scheme are found in Wiebe, Wilson, and Cardie (2005).

The two types of private state frames, **direct subjective frames** and **expressive subjective element frames**, are where we will find sentiment expressions. Direct subjective frames are used to mark direct references to private states as well as speech events in which private states are being expressed. For example, in the following sentences, *fears*, *praised*, and *said* are all marked as direct subjective annotations.

(4) The U.S. **fears** a spill-over of the anti-terrorist campaign.

(5) Italian senator Renzo Gubert **praised** the Chinese government's efforts.

(6) "The report is full of absurdities," he **said**.

The word *fears* directly refers to a private state; *praised* refers to a speech event in which a private state is being expressed; and *said* is marked as direct subjective because a private state is being expressed within the speech event referred to by *said*. Expressive subjective elements indirectly express private states through the way something is described or through a particular wording. In example (6), the phrase *full of absurdities* is an expressive subjective element. **Subjectivity** (Banfield 1982; Wiebe

2 Available at `http://nrrc.mitre.org/NRRC/publications.htm`.

1994) refers to the linguistic expression of private states, hence the names for the two types of private state annotations.

All expressive subjective elements are included in the set of annotations that have the possibility of being sentiment expressions, but the direct subjective frames to include in this set can be pared down further. Direct subjective frames have an attribute, **expression intensity**, that captures the contribution of the annotated word or phrase to the overall intensity of the private state being expressed. Expression intensity ranges from *neutral* to *high*. In the given sentences, *fears* and *praised* have an expression intensity of medium, and *said* has an expression intensity of neutral. A neutral expression intensity indicates that the direct subjective phrase itself is not contributing to the expression of the private state. If this is the case, then the direct subjective phrase cannot be a sentiment expression. Thus, only direct subjective annotations with a *non-neutral* expression intensity are included in the set of annotations that have the possibility of being sentiment expressions. We call this set of annotations, the union of the expressive subjective elements and the direct subjective frames with a non-neutral intensity, the **subjective expressions** in the corpus; these are the annotations we will mark for contextual polarity.

Table 1 gives a sample of subjective expressions marked in the MPQA corpus. Although many of the words and phrases express what we typically think of as sentiments, others do not, for example, *believes, very definitely,* and *unconditionally and without delay.*

Now that we have identified which annotations have the possibility of being sentiment expressions, the next question is which of these annotated words and phrases are actually expressing sentiments. We define a sentiment as a positive or negative emotion, evaluation, or stance. On the left of Table 2 are examples of positive sentiments; examples of negative sentiments are on the right.

Table 1
Sample of subjective expressions from the MPQA corpus.

victory of justice and freedom	such a disadvantageous situation
grown tremendously	must
such animosity	not true at all
throttling the voice	imperative for harmonious society
disdain and wrath	glorious
so exciting	disastrous consequences
could not have wished for a better situation	believes
freak show	the embodiment of two-sided justice
if you're not with us, you're against us	appalling
vehemently denied	very definitely
everything good and nice	once and for all
under no circumstances	shameful mum
most fraudulent, terrorist and extremist	enthusiastically asked
number one democracy	hate
seems to think	gross misstatement
indulging in blood-shed and their lunaticism	surprised, to put it mildly
take justice to pre-historic times	unconditionally and without delay

so conservative that it makes Pat Buchanan look vegetarian
those digging graves for others, get engraved themselves
lost the reputation of commitment to principles of human justice
ultimately the demon they have reared will eat up their own vitals

Table 2
Examples of positive and negative sentiments.

	Positive sentiments	Negative sentiments
Emotion	I'm happy	I'm sad
Evaluation	Great idea!	Bad idea!
Stance	She supports the bill	She's against the bill

The final issue to address is the actual annotation scheme for marking contextual polarity. The scheme we developed has four tags: *positive*, *negative*, *both*, and *neutral*. The *positive* tag is used to mark positive sentiments. The *negative* tag is used to mark negative sentiments. The *both* tag is applied to expressions in which both a positive and negative sentiment are being expressed. Subjective expressions with *positive*, *negative*, or *both* tags are our sentiment expressions. The *neutral* tag is used for all other subjective expressions, including emotions, evaluations, and stances that are neither positive or negative. Instructions for the contextual-polarity annotation scheme are available at `http://www.cs.pitt.edu/mpqa/databaserelease/polarityCodingInstructions.txt`.

Following are examples from the corpus of each of the different contextual-polarity annotations. Each underlined word or phrase is a subjective expression that was marked in the original MPQA annotations.[3] In bold following each subjective expression is the contextual polarity with which it was annotated.

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

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

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

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

As a final note on the annotation scheme, annotators are asked to judge the contextual polarity of the sentiment that is ultimately being conveyed by the subjective expression, that is, once the sentence has been fully interpreted. Thus, the subjective expression, *they have not succeeded, and will never succeed,* is marked as positive in the following sentence:

(11) They have not succeeded, and will never succeed (**positive**), in breaking the will of this valiant people.

The reasoning is that breaking the will of a valiant people is negative, so to not succeed in breaking their will is positive.

3 Some sentences contain additional subjective expressions that are not underlined as examples.

Table 3
Contingency table for contextual polarity agreement.

	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 4
Contingency table for contextual polarity agreement, borderline cases removed.

	Neutral	Positive	Negative	Both	Total
Neutral	**113**	7	8	0	128
Positive	9	**59**	3	0	71
Negative	5	2	**156**	1	164
Both	0	2	0	**2**	4
Total	127	70	167	3	**367**

3.2 Agreement Study

To measure the reliability of the polarity annotation scheme, we conducted an agreement study with two annotators[4] using 10 documents from the MPQA corpus. The 10 documents contain 447 subjective expressions. Table 3 shows the contingency table for the two annotators' judgments. Overall agreement is 82%, with a kappa value of 0.72.

As part of the annotation scheme, annotators are asked to judge how certain they are in their polarity tags. For 18% of the subjective expressions, at least one annotator used the *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. Table 4 shows the revised contingency table with the uncertain cases removed. This shows that annotator agreement is especially high when both annotators are certain, and that annotators are certain for over 80% of their tags.

Note that all annotations are included in the experiments.

3.3 Contextual Polarity Annotations

In total, all 19,962 subjective expressions in the 535 documents (11,112 sentences) of the MPQA corpus were annotated with their contextual polarity as just described.[5] Three annotators carried out the task: the two who participated in the annotation study and a third who was trained later.[6] Table 5 gives the distribution of the contextual polarity tags. Looking at this table, we see that a small majority of subjective expressions (54.6%)

4 Both annotators are authors of this article.
5 The revised version of the MPQA corpus with the contextual polarity annotations is available at `http://www.cs.pitt.edu/mpqa`.
6 The third annotator received training until her reliability of performance on the task was comparable to that of the first two annotators who participated in the study.

Table 5
Distribution of contextual polarity tags.

Neutral	Positive	Negative	Both	Total
9,057	3,311	7,294	299	19,961
45.4%	16.6%	36.5%	1.5%	100%

are expressing a *positive*, *negative*, or *both* (positive and negative) sentiment. We refer to these expressions as **polar in context**. Many of the subjective expressions are neutral and do not express a sentiment. This suggests that, although sentiment is a major type of subjectivity, distinguishing other prominent types of subjectivity will be important for future work in subjectivity analysis.

As many NLP applications operate at the sentence level, one important issue to consider is the distribution of sentences with respect to the subjective expressions they contain. In the 11,112 sentences in the MPQA corpus, 28% contain no subjective expressions, 24% contain only one, and 48% contain two or more. Of the 5,304 sentences containing two or more subjective expressions, 17% contain mixtures of positive and negative expressions, and 61% contain mixtures of polar (positive/negative/both) and neutral subjective expressions.

4. Prior-Polarity Subjectivity Lexicon

For the experiments in this article, we use a lexicon of over 8,000 **subjectivity clues**. Subjectivity clues are words and phrases that may be used to express private states. In other words, subjectivity clues 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 the list of subjectivity clues from Riloff and Wiebe (2003), which includes the positive and negative adjectives from Hatzivassiloglou and McKeown (1997). The words in this list were grouped in previous work according to their reliability as subjectivity clues. Words that are subjective in most contexts are considered **strong subjective clues**, indicated by the *strongsubj* tag. Words that may only have certain subjective usages are considered **weak subjective clues**, indicated by the *weaksubj* tag.

We expanded the list using a dictionary and a thesaurus, and added words from the General Inquirer positive and negative word lists (Stone et al. 1966) that we judged to be potentially subjective.[7] We also gave the new words *strongsubj* and *weaksubj* reliability tags. The final lexicon has a coverage of 67% of subjective expressions in the MPQA corpus, where coverage is the percentage of subjective expressions containing one or more instances of clues from the lexicon. The coverage of just sentiment expressions is even higher: 75%.

The next step was to tag the clues in the lexicon with their prior polarity: *positive, negative, both,* or *neutral*. A word in the lexicon is tagged as *positive* if out of context it seems to evoke something positive, and *negative* if it seems to evoke something negative. If a word has both positive and negative meanings, it is tagged with the polarity that seems the most common. A word is tagged as *both* if it is at the same time

7 In the end, about 70% of the words from the General Inquirer positive word list and 80% of the words from the negative word list were included in the subjectivity lexicon.

both positive and negative. For example, the word *bittersweet* evokes something both positive and negative. Words like *brag* are also tagged as *both*, because the one who is bragging is expressing something positive, yet at the same time describing someone as bragging is expressing a negative evaluation of that person. A word is tagged as *neutral* if it does not evoke anything positive or negative.

For words that came from positive and negative word lists (Stone et al. 1966; Hatzivassiloglou and McKeown 1997), we largely retained their original polarity. However, we did change the polarity of a word if we strongly disagreed with its original class.[8] For example, the word *apocalypse* is listed as positive in the General Inquirer; we changed its prior polarity to negative for our lexicon.

By far, the majority of clues in the lexicon (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. We refer to the set of clues marked as *positive*, *negative*, or *both* as **sentiment clues**. A total of 6.9% of the clues in the lexicon are marked as *neutral*. Examples of neutral clues are verbs such as *feel*, *look*, and *think*, and intensifiers such as *deeply*, *entirely*, and *practically*. Although the neutral clues make up a small proportion of the total words in the lexicon, we retain them for our later experiments in recognizing contextual polarity because many of them are good clues that a sentiment is being expressed (e.g., *feels slighted*, *feels satisfied*, *look kindly on*, *look forward to*). Including them increases the coverage of the system.

At the end of the previous section, we considered the distribution of sentences in the MPQA corpus with respect to the subjective expressions they contain. It is interesting to compare that distribution with the distribution of sentences with respect to the instances they contain of clues from the lexicon. We find that there are more sentences with two or more clue instances (62%) than sentences with two or more subjective expressions (48%). More importantly, many more sentences have mixtures of positive and negative clue instances than actually have mixtures of positive and negative subjective expressions. Only 880 sentences have a mixture of both positive and negative subjective expressions, whereas 3,234 sentences have a mixture of positive and negative clue instances. Thus, a large number of positive and negative instances are either neutral in context, or they are combining to form more complex polarity expressions. Either way, this provides strong evidence of the need to be able to disambiguate the contextual polarity of subjectivity and sentiment clues.

5. Definition of the Gold Standard

In the experiments described in the following sections, the goal is to classify the contextual polarity of the expressions that contain instances of the subjectivity clues in our lexicon. However, determining which clue instances are part of the same expression and identifying expression boundaries are not the focus of this work. Thus, instead of trying to identify and label each expression, in the following experiments, each clue instance is labeled individually as to its contextual polarity.

We define the gold-standard contextual polarity of a clue instance in terms of the manual annotations (Section 3) as follows. If a clue instance is not in a subjective expression (and therefore not in a sentiment expression), its gold class is *neutral*. If a clue instance appears in just one subjective expression or in multiple subjective

8 We decided on a different polarity for about 80 of the words in our lexicon that appeared on other positive and negative word lists.

expressions with the same contextual polarity, its gold class is the contextual polarity of the subjective expression(s). If a clue instance appears in a mixture of negative and neutral subjective expressions, its gold class is *negative*; if it is in a mixture of positive and neutral subjective expressions, its gold class is *positive*. Finally, if a clue instance appears in at least one positive and one negative subjective expression (or in a subjective expression marked as both), then its gold class is *both*. A clue instance can appear in more than one subjective expression because in the MPQA annotation scheme, it is possible for direct subjective frames and expressive subjective elements frames to overlap.

6. A Prior-Polarity Classifier

Before delving into the task of recognizing contextual polarity, an important question to address is how useful prior polarity alone is for identifying contextual polarity. To answer this question, we create a classifier that simply assumes the contextual polarity of a clue instance is the same as the clue's prior polarity. We explore this classifier's performance on a small amount of development data, which is not part of the data used in the subsequent experiments.

This simple classifier has an accuracy of 48%. From the confusion matrix given in Table 6, we see that 76% of the errors result from words with non-neutral prior polarity appearing in phrases with neutral contextual polarity. Only 12% of the errors result from words with neutral prior polarity appearing in expressions with non-neutral contextual polarity, and only 11% of the errors come from words with a positive or negative prior polarity appearing in expressions with the opposite contextual polarity. Table 6 also shows that positive clues tend to be used in negative expressions far more often than negative clues tend to be used in positive expressions.

Given that by far the largest number of errors come from clues with *positive*, *negative*, or *both* prior polarity appearing in neutral contexts, we were motivated to try a two-step approach to the problem of sentiment classification. The first step, *Neutral–Polar Classification*, tries to determine if an instance is neutral or polar in context. The second step, *Polarity Classification*, takes all instances that step one classified as polar, and tries to disambiguate their contextual polarity. This two-step approach is illustrated in Figure 1.

7. Features

The features used in our experiments were motivated both by the literature and by exploration of the contextual-polarity annotations in our development data. A number

Table 6
Confusion matrix for the prior-polarity classifier on the development set.

		Prior-Polarity Classifier				
		Neutral	Positive	Negative	Both	Total
	Neutral	**798**	784	698	4	2284
Gold	Positive	81	**371**	40	0	492
Class	Negative	149	181	**622**	0	952
	Both	4	11	13	**5**	33
	Total	1032	1347	1373	9	**3761**

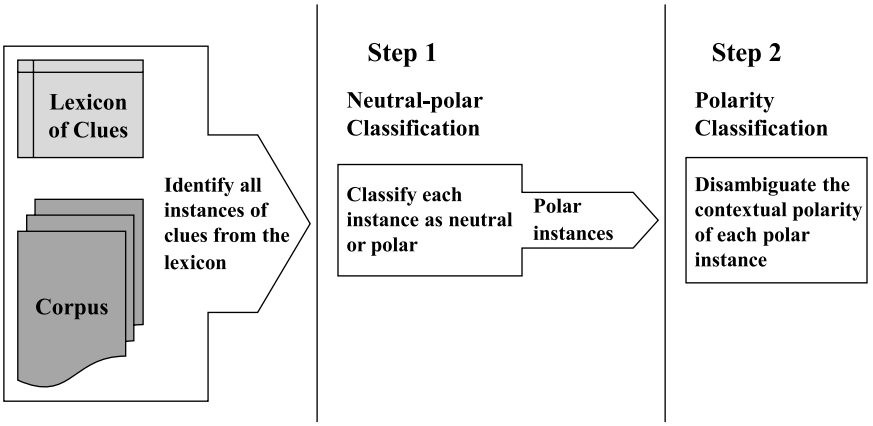

Figure 1
Two-step approach to recognizing contextual polarity.

of features were inspired by the paper on contextual-polarity influencers by Polanyi and Zaenan (2004). Other features are those that have been found useful in the past for recognizing subjective sentences (Wiebe, Bruce, and O'Hara 1999; Wiebe and Riloff 2005).

7.1 Features for Neutral–Polar Classification

For distinguishing between neutral and polar instances, we use the features listed in Table 7. For ease of description, we group the features into six sets: word features, general modification features, polarity modification features, structure features, sentence features, and one document feature.

Word Features In addition to the word token (the token of the clue instance being classified), the word features include the parts of speech of the previous word, the word itself, and the next word. The *prior polarity* and *reliability class* features represent those pieces of information about the clue which are taken from the lexicon.

General Modification Features These are binary features that capture different types of relationships involving the clue instance.

The first four features involve relationships with the word immediately before or after the clue instance. The *preceded by adjective* feature is true if the clue instance is a noun preceded by an adjective. The *preceded by adverb* feature is true if the preceding word is an adverb other than *not*. The *preceded by intensifier* feature is true if the preceding word is an intensifier, and the *self intensifier* feature is true if the clue instance itself is an intensifier. A word is considered to be 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 list of intensifiers is a compilation of those listed in Quirk et al. (1985), intensifiers identified from existing entries in the subjectivity lexicon, and intensifiers identified during explorations of the development data.

The *modifies/modifed by* features involve the dependency parse tree of 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 parent word is called the **head**, and its children are its **modifiers**. The

Table 7
Features for neutral–polar classification.

Word Features
word token
word part of speech
previous word part of speech
next word part of speech
prior polarity: *positive, negative, both, neutral*
reliability class: *strongsubj* or *weaksubj*
General Modification Features
preceded by adjective: *binary*
preceded by adverb (other than *not*): *binary*
preceded by intensifier: *binary*
self intensifier: *binary*
modifies strongsubj: *binary*
modifies weaksubj: *binary*
modified by strongsubj: *binary*
modified by weaksubj: *binary*
Polarity Modification Features
modifies polarity: *positive, negative, neutral, both, notmod*
modified by polarity: *positive, negative, neutral, both, notmod*
conjunction polarity: *positive, negative, neutral, both, notmod*
Structure Features
in subject: *binary*
in copular: *binary*
in passive: *binary*
Sentence Features
strongsubj clues in current sentence: *0, 1, 2, 3 (or more)*
strongsubj clues in previous sentence: *0, 1, 2, 3 (or more)*
strongsubj clues in next sentence: *0, 1, 2, 3 (or more)*
weaksubj clues in current sentence: *0, 1, 2, 3 (or more)*
weaksubj clues in previous sentence: *0, 1, 2, 3 (or more)*
weaksubj clues in next sentence: *0, 1, 2, 3 (or more)*
adjectives in sentence: *0, 1, 2, 3 (or more)*
adverbs in sentence (other than *not*): *0, 1, 2, 3 (or more)*
cardinal number in sentence: *binary*
pronoun in sentence: *binary*
modal in sentence (other than *will*): *binary*
Document Feature
document topic/domain

edge between a parent and a child specifies the grammatical relationship between the two words. Figure 2 shows an example of a dependency parse tree. Instances of clues in the tree are marked with the clue's prior polarity and reliability class from the lexicon.

For each clue instance, the *modifies/modifed by* features capture whether there are *adj*, *mod*, or *vmod* relationships between the clue instance and any other instances from the lexicon. Specifically, the *modifies strongsubj* feature is true if the clue instance and its parent share an *adj*, *mod*, or *vmod* relationship, and if its parent is an instance of a *strongsubj* clue from the lexicon. The *modifies weaksubj* feature is the same, except that it looks in the parent for an instance of a *weaksubj* clue. The *modified by strongsubj*

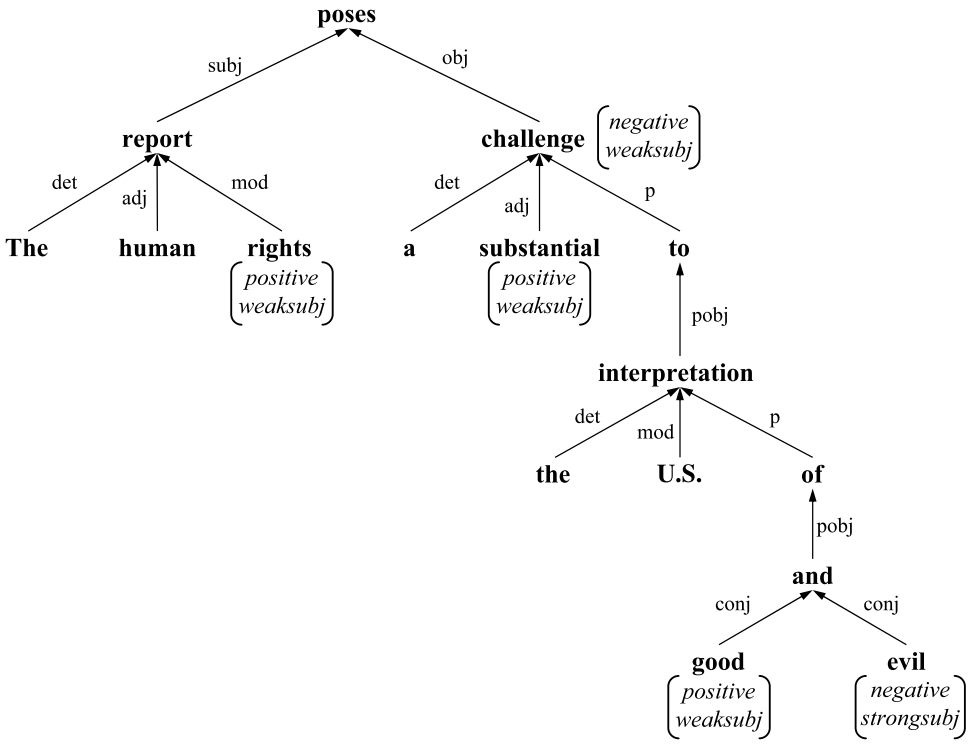

Figure 2
The dependency tree for the sentence *The human rights report poses a substantial challenge to the U.S. interpretation of good and evil.* Prior polarity and reliability class are marked in parentheses for words that match clues from the lexicon.

feature is true for a clue instance if one of its children is an instance of a *strongsubj* clue, and if the clue instance and its child share an *adj*, *mod*, or *vmod* relationship. The *modified by weaksubj* feature is the same, except that it looks for instances of *weaksubj* clues in the children. Although the *adj* and *vmod* relationships are typically local, the *mod* relationship involves longer-distance as well as local dependencies. Figure 2 helps to illustrate these features. The *modifies weaksubj* feature is true for *substantial,* because *substantial* modifies *challenge,* which is an instance of a *weaksubj* clue. For *rights,* the *modifies weaksubj* feature is false, because *rights* modifies *report,* which is not an instance of a *weaksubj* clue. The *modified by weaksubj* feature is false for *substantial,* because it has no modifiers that are instances of *weaksubj* clues. For *challenge,* the *modified by weaksubj* feature is true because it is being modified by *substantial,* which is an instance of a *weaksubj* clue.

Polarity Modification Features The *modifies polarity*, *modified by polarity*, and *conj polarity* features capture specific relationships between the clue instance and other sentiment clues it may be related to. If the clue instance 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 parent. If the parent is not in the prior-polarity lexicon, its prior polarity is considered *neutral*. If the clue instance is at the root of the tree and has no parent, the value of the feature is *notmod*. The *modified by polarity* feature is similar, looking for *adj*, *mod*, and *vmod* relationships and other sentiment clues in the children of the clue instance. The *conj polarity* feature determines if the clue instance is in a conjunction. If so, the value of this feature is its sibling's prior polarity. As before, if the sibling is not in the

lexicon, its prior polarity is *neutral*. If the clue instance is not in a conjunction, the value for this feature is *notmod*. Figure 2 also helps to illustrate these modification features. The word *substantial* with positive prior polarity modifies the word *challenge* with negative prior polarity. Therefore the *modifies polarity* feature is negative for *substantial*, and the *modified by polarity* feature is positive for *challenge*. The words *good* and *evil* are in a conjunction together; thus the *conj polarity* feature is negative for *good* and positive for *evil*.

Structure Features These are binary features that are determined by starting with the clue 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 on the path to the root. 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* feature is true if a passive verb pattern is found on the climb.

The *in subject* and *in copular* features were motivated by the intuition that the syntactic role of a word may influence whether a word is being used to express a sentiment. For example, consider the word *polluters* in each of the following two sentences.

(12) Under the application shield, **polluters** are allowed to operate if they have a permit.

(13) "The big-city folks are pointing at the farmers and saying you are **polluters** . . ."

In the first sentence, *polluters* is simply being used as a referring expression. In the second sentence, *polluters* is clearly being used to express a negative evaluation of the farmers. The motivation for the *in passive* feature was previous work by Riloff and Wiebe (2003), who found that different words are more or less likely to be subjective depending on whether they are in the active or passive.

Sentence Features These are features that previously were found useful for sentence-level subjectivity classification (Wiebe, Bruce, and O'Hara 1999; Wiebe and Riloff 2005). They include counts of *strongsubj* and *weaksubj* clue instances 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 or domain of the document. The motivation for this feature is that whether or not a word is expressing a sentiment or even a private state in general may depend on the subject of the discourse. For example, the words *fever* and *sufferer* may express a negative sentiment in certain contexts, but probably not in a health or medical context, as is the case in the following sentence.

(14) The disease can be contracted if a person is bitten by a certain tick or if a person comes into contact with the blood of a congo **fever sufferer**.

In the creation of the MPQA corpus, about two-thirds of the documents were selected to be on one of the 10 topics listed in Table 8. The documents for each topic were identified by human searches and by an information retrieval system. The remaining documents were semi-randomly selected from a very large pool of documents from the world press. In the corpus, these documents are listed with the topic *miscellaneous*. Rather than leaving these documents unlabeled, we chose to label them using the

Table 8
Topics in the MPQA corpus.

Topic	Description
argentina	Economic collapse in Argentina
axisofevil	U.S. President's State of the Union Address
guantanamo	Detention of prisoners in Guantanamo Bay
humanrights	U.S. State Department Human Rights Report
kyoto	Kyoto Protocol ratification
settlements	Israeli settlements in Gaza and the West Bank
space	Space missions of various countries
taiwan	Relationship between Taiwan and China
venezuela	Presidential coup in Venezuela
zimbabwe	Presidential election in Zimbabwe

following general domain categories: economics, general politics, health, report events, and war and terrorism.

7.2 Features for Polarity Classification

Table 9 lists the features that we use for step two, polarity classification. *Word token*, *word prior polarity*, and the polarity-modification features are the same as described for neutral–polar classification.

We use two features to capture two different types of negation. The *negated* feature is a binary feature that is used to capture more local negations: Its value is true if a negation word or phrase is found within the four words preceding the clue instance, and if the negation word is not also in a phrase that acts as an intensifier rather than a negator. Examples of phrases that intensify rather than negate are *not only* and *nothing if not*. The *negated subject* feature captures a longer-distance type of negation. This feature

Table 9
Features for polarity classification.

Word Features
word token
word prior polarity: *positive, negative, both, neutral*

Negation Features
negated: *binary*
negated subject: *binary*

Polarity Modification Features
modifies polarity: *positive, negative, neutral, both, notmod*
modified by polarity: *positive, negative, neutral, both, notmod*
conj polarity: *positive, negative, neutral, both, notmod*

Polarity Shifters
general polarity shifter: *binary*
negative polarity shifter: *binary*
positive polarity shifter: *binary*

is true if the subject of the clause containing the clue instance is negated. For example, the *negated subject* feature is true for *support* in the following sentence.

(15) No politically prudent Israeli could **support** either of them.

The last three polarity features look in a window of four words before the clue instance, 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 influencers that we used were identified through explorations of the development data.

8. Experiments in Recognizing Contextual Polarity

We have two primary goals with our experiments in recognizing contextual polarity. The first is to evaluate the features described in Section 7 as to their usefulness for this task. The second is to investigate the importance of recognizing neutral instances—recognizing when a sentiment clue is not being used to express a sentiment—for classifying contextual polarity.

To evaluate features, we investigate their performance, both together and separately, across several different learning algorithms. Varying the learning algorithm allows us to verify that the features are robust and that their performance is not the artifact of a particular algorithm. We experiment with four different types of machine learning: boosting, memory-based learning, rule learning, and support vector learning. For boosting, we use BoosTexter (Schapire and Singer 2000) AdaBoost.MH. For rule learning, we use Ripper (Cohen 1996). For memory-based learning, we use TiMBL (Daelemans et al. 2003b) IB1 (*k*-nearest neighbor). For support vector learning, we use SVM-light and SVM-multiclass (Joachims 1999). SVM-light is used for the experiments involving binary classification (neutral–polar classification), and SVM-multiclass is used for experiments with more than two classes. These machine learning algorithms were chosen because they have been used successfully for a number of natural language processing tasks, and they represent several different types of learning.

For all of the classification algorithms except for SVM, the features for a clue instance are represented as they are presented in Section 7. For SVM, the representations for numeric and discrete-valued features are changed. Numeric features, such as the count of *strongsubj* clue instances in a sentence, are scaled to range between 0 and 1. Discrete-valued features, such as the *reliability class* feature, are converted into multiple binary features. For example, the *reliability class* feature is represented by two binary features: one for whether the clue instance is a *strongsubj* clue and one for whether the clue instance is a *weaksubj* clue.

To investigate the importance of recognizing neutral instances, we perform two sets of polarity classification (step two) experiments. First, we experiment with classifying the polarity of all gold-standard polar instances—the clue instances identified as polar in context by the manual polarity annotations. Second, we experiment with using the polar instances identified automatically by the neutral–polar classifiers. Because the second set of experiments includes the neutral instances misclassified in step one, we can compare results for the two sets of experiments to see how the noise of neutral instances affects the performance of the polarity features.

All experiments are performed using 10-fold cross validation over a test set of 10,287 sentences from 494 MPQA corpus documents. We measure performance in terms of accuracy, recall, precision, and F-measure. Accuracy is simply the total number of instances correctly classified. Recall, precision, and F-measure for a given class *C* are defined as follows. Recall is the percentage of all instances of class *C* correctly identified.

$$Rec(C) = \frac{\mid \text{instances of C correctly identified} \mid}{\mid \text{all instances of C} \mid}$$

Precision is the percentage of instances classified as class *C* that are class *C* in truth.

$$Prec(C) = \frac{\mid \text{instances of C correctly identified} \mid}{\mid \text{all instances identified as C} \mid}$$

F-measure is the harmonic mean of recall and precision.

$$F(C) = \frac{2 \times Rec(C) \times Prec(C)}{Rec(C) + Prec(C)}$$

All results reported are averages over the 10 folds.

8.1 Neutral–Polar Classification

In our two-step process for recognizing contextual polarity, the first step is neutral–polar classification, determining whether each instance of a clue from the lexicon is neutral or polar in context. In our test set, there are 26,729 instances of clues from the lexicon. The features we use for this step were listed above in Table 7 and described in Section 7.1.

In this section, we perform two sets of experiments. In the first, we compare the results of neutral–polar classification using all the neutral–polar features against two baselines. The first baseline uses just the *word token* feature. The second baseline (word+priorpol) uses the *word token* and *prior polarity* features. In the second set of experiments, we explore the performance of different sets of features for neutral–polar classification.

Research has shown that the performance of learning algorithms for NLP tasks can vary widely depending on their parameter settings, and that the optimal parameter settings can also vary depending on the set of features being evaluated (Daelemans et al. 2003a; Hoste 2005). Although the goal of this work is not to identify the optimal configuration for each algorithm and each set of features, we still want to make a reasonable attempt to find a good configuration for each algorithm. To do this, we perform 10-fold cross validation of the more challenging baseline classifier (word+priorpol) on the development data, varying select parameter settings. The results from those experiments are then used to select the parameter settings for each algorithm. For BoosTexter, we vary the number of rounds of boosting. For TiMBL, we vary the value for *k* (the number of neighbors) and the distance metric (overlap or modified value difference metric [MVDM]). For Ripper, we vary whether negative tests are disallowed for nominal (-!n) and set (-!s) valued attributes and how much to simplify the hypothesis (-S). For SVM, we experiment with linear, polynomial, and radial basis function kernels. Table 10 gives the settings selected for the neutral–polar classification experiments for the different learning algorithms.

Table 10
Algorithm settings for neutral–polar classification.

Algorithm	Settings
BoosTexter	2,000 rounds of boosting
TiMBL	*k*=25, MVDM distance metric
Ripper	-!n, -S 0.5
SVM	linear kernel

8.1.1 Classification Results. The results for the first set of experiments are given in Table 11. For each algorithm, we give the results for the two baseline classifiers, followed by the results for the classifier trained using all the neutral–polar features. The results shown in bold are significantly better than both baselines (two-sided t-test, $p \leq 0.05$) for the given algorithm.

Working together, how well do the neutral–polar features perform? For BoosTexter, TiMBL, and Ripper, the classifiers trained using all the features improve significantly over the two baselines in terms of accuracy, polar recall, polar F-measure, and neutral precision. Neutral F-measure is also higher, but not significantly so. These consistent results across three of the four algorithms show that the neutral–polar features are helpful for determining when a sentiment clue is actually being used to express a sentiment.

Interestingly, Ripper is the only algorithm for which the word-token baseline performed better than the word+priorpol baseline. Nevertheless, the *prior polarity* feature is an important component in the performance of the Ripper classifier using all the features. Excluding prior polarity from this classifier results in a significant decrease in

Table 11
Results for neutral–polar classification (step one).

			Polar			Neutral	
	Acc	Rec	Prec	F	Rec	Prec	F
BoosTexter							
word token baseline	74.0	41.9	77.0	54.3	92.7	73.3	81.8
word+priorpol baseline	75.0	55.6	70.2	62.1	86.2	76.9	81.3
neutral–polar features	**76.5**	**58.3**	72.4	**64.6**	87.1	**78.2**	82.4
TiMBL							
word token baseline	74.6	47.9	73.9	58.1	90.1	74.8	81.8
word+priorpol baseline	74.6	48.2	73.7	58.3	90.0	74.9	81.7
neutral–polar features	**76.5**	**59.5**	71.7	**65.0**	86.3	**78.5**	82.3
Ripper							
word token baseline	66.3	11.2	80.6	19.6	98.4	65.6	78.7
word+priorpol baseline	65.5	07.7	84.5	14.1	99.1	64.8	78.4
neutral–polar features	**71.4**	**49.4**	64.6	**56.0**	84.2	**74.1**	78.8
SVM							
word token baseline	74.6	47.9	73.9	58.1	90.1	74.8	81.8
word+priorpol baseline	75.6	54.5	72.5	62.2	88.0	76.8	82.0
neutral–polar features	75.3	52.6	72.7	61.0	88.5	76.2	81.9

performance for every metric. Decreases range from 2.5% for neutral recall to 9.5% for polar recall.

The best SVM classifier is the word+priorpol baseline. In terms of accuracy, this classifier does not perform much worse than the BoosTexter and TiMBL classifiers that use all the neutral–polar features: The SVM word+priorpol baseline classifier has an accuracy of 75.6%, and both the BoosTexter and TiMBL classifiers have an accuracy of 76.5%. However, the BoosTexter and TiMBL classifiers using all the features perform notably better in terms of polar recall and F-measure. The BoosTexter and TiMBL classifiers have polar recalls that are 7% and 9.2% higher than the SVM baseline. Polar F-measures for BoosTexter and TiMBL are 3.9% and 4.5% higher. These increases are significant for $p \leq 0.01$.

8.1.2 Feature Set Evaluation. To evaluate the contribution of the various features for neutral–polar classification, we perform a series of experiments in which different sets of neutral–polar features are added to the word+priorpol baseline and new classifiers are trained. We then compare the performance of these new classifiers to the word+priorpol baseline, with the exception of the Ripper classifiers, which we compare to the higher word baseline. Table 12 lists the sets of features tested in these experiments. The feature sets generally correspond to how the neutral–polar features are presented in Table 7, although some of the groups are broken down into more fine-grained sets that we believe capture meaningful distinctions.

Table 13 gives the results for these experiments. Increases and decreases for a given metric as compared to the word+priorpol baseline (word baseline for Ripper) are indicated by + or –, respectively. Where changes are significant at the $p \leq 0.1$ level, ++ or – – are used, and where changes are significant at the $p \leq 0.05$ level, +++ or – – – are used. An "nc" indicates no change (a change of less than $\pm$ 0.05) compared to the baseline.

What does Table 13 reveal about the performance of various feature sets for neutral–polar classification? Most noticeable is that no individual feature sets stand out as strong performers. The only significant improvements in accuracy come from the PARTS-OF-SPEECH and RELIABILITY-CLASS feature sets for Ripper. These improvements are perhaps not surprising given that the Ripper baseline was much lower to begin with. Very few feature sets show any improvement for SVM. Again, this is not unexpected given that all the features together performed worse than the word+priorpol baseline

Table 12
Neutral–polar feature sets for evaluation.

Experiment	Features
PARTS-OF-SPEECH	parts of speech for clue instance, previous word, and next word
RELIABILITY-CLASS	reliability class of clue instance
PRECEDED-POS	preceded by adjective, preceded by adverb
INTENSIFY	preceded by intensifier, self intensifier
RELCLASS-MOD	modifies strongsubj/weaksubj, modified by strongsubj/weaksubj
POLARITY-MOD	polarity-modification features
STRUCTURE	structure features
CURSENT-COUNTS	strongsubj/weaksubj clue instances in sentence
PNSENT-COUNTS	strongsubj/weaksubj clue instances in previous/next sentence
CURSENT-OTHER	adjectives/adverbs/cardinal number/pronoun/modal in sentence
TOPIC	document topic

Table 13
Results for neutral–polar feature ablation experiments.

		Polar	Neut			Polar	Neut
BoosTexter	Acc	F	F	**Ripper**	Acc	F	F
PARTS-OF-SPEECH	+	–	+	PARTS-OF-SPEECH	+++	+++	– – –
RELIABILITY-CLASS	+	–	+	RELIABILITY-CLASS	+++	+++	+
PRECEDED-POS	nc	–	nc	PRECEDED-POS	–	–	–
INTENSIFY	-	nc	-	INTENSIFY	–	– – –	–
RELCLASS-MOD	+	++	+	RELCLASS-MOD	+	+++	+
POLARITY-MOD	nc	–	+	POLARITY-MOD	–	+++	–
STRUCTURE	–	– – –	+	STRUCTURE	–	+	–
CURSENT-COUNTS	+	– – –	+	CURSENT-COUNTS	– –	+++	– – –
PNSENT-COUNTS	+	– – –	+	PNSENT-COUNTS	– – –	+++	– – –
CURSENT-OTHER	nc	–	+	CURSENT-OTHER	– – –	+++	– – –
TOPIC	+	+	+	TOPIC	–	+++	– – –
		Polar	Neut			Polar	Neut
TiMBL	Acc	F	F	**SVM**	Acc	F	F
PARTS-OF-SPEECH	+	+++	+	PARTS-OF-SPEECH	– –	– – –	–
RELIABILITY-CLASS	+	+	nc	RELIABILITY-CLASS	+	–	+
PRECEDED-POS	nc	+	nc	PRECEDED-POS	nc	nc	nc
INTENSIFY	nc	nc	nc	INTENSIFY	nc	nc	nc
RELCLASS-MOD	+	+	+	RELCLASS-MOD	nc	+	nc
POLARITY-MOD	+	+	+	POLARITY-MOD	– –	– – –	– –
STRUCTURE	nc	+	–	STRUCTURE	–	+	–
CURSENT-COUNTS	–	+	–	CURSENT-COUNTS	–	–	–
PNSENT-COUNTS	+	+++	–	PNSENT-COUNTS	–	–	–
CURSENT-OTHER	+	+++	–	CURSENT-OTHER	–	–	–
TOPIC	–	+	–	TOPIC	–	–	–

Increases and decreases for a given metric as compared to the word+priorpol baseline (word baseline for Ripper) are indicated by + or –, respectively; ++ or – – indicates the change is significant at the $p < 0.1$ level; +++ or – – – indicates significance at the $p < 0.05$ level; nc indicates no change.

for SVM. The performance of the feature sets for BoosTexter and TiMBL are perhaps the most revealing. In the previous experiments using all the features together, these algorithms produced classifiers with the same high performance. In these experiments, six different feature sets for each algorithm show improvements in accuracy over the baseline, yet none of those improvements are significant. This suggests that achieving the highest performance for neutral–polar classification requires a wide variety of features working together in combination.

We further test this result by evaluating the effect of removing the features that produced either no change or a drop in accuracy from the respective all-feature classifiers. For example, we train a TiMBL neutral–polar classifier using all the features except for those in the PRECEDED-POS, INTENSIFY, STRUCTURE, CURSENT-COUNTS, and TOPIC feature sets, and then compare the performance of this new classifier to the TiMBL, all-feature classifier. Although removing the non-performing features has little effect for BoosTexter, performance does drop for both TiMBL and Ripper. The primary source of this performance drop is a decrease in polar recall: 2% for TiMBL and 3.2% for Ripper.

Although no feature sets stand out in Table 13 as far as giving an overall high performance, there are some features that consistently improve performance across the different algorithms. The reliability class of the clue instance (RELIABILITY-CLASS) improves accuracy over the baseline for all four algorithms. It is the only feature that does so. The RELCLASS-MOD features give improvements for all metrics for BoosTexter, Ripper, and TiMBL, as well as improving polar F-measure for SVM. The PARTS-OF-SPEECH features are also fairly consistent, improving performance for all the algorithms except for SVM. There are also a couple of feature sets that consistently do not improve performance for any of the algorithms: the INTENSIFY and PRECEDED-POS features.

8.2 Polarity Classification

For the second step of recognizing contextual polarity, we classify the polarity of all clue instances identified as polar in step one. The features for polarity classification were listed in Table 9 and described in Section 7.2.

We investigate the performance of the polarity features under two conditions: (1) perfect neutral–polar recognition and (2) automatic neutral–polar recognition. For condition 1, we identify the polar instances according to the gold-standard, manual contextual-polarity annotations. In the test data, 9,835 instances of the clues from the lexicon are polar in context according to the manual annotations. Experiments under condition 1 classify these instances as having positive, negative, or both (positive and negative) polarity. For condition 2, we take the best performing neutral–polar classifier for each algorithm and use the output from those algorithms to identify the polar instances. Because polar instances now are being identified automatically, there will be noise in the form of misclassified neutral instances. Therefore, for experiments under condition 2 we include the neutral class and perform four-way classification instead of three-way. Condition 1 allows us to investigate the performance of the different polarity features without the noise of misclassified neutral instances. Also, because the set of polar instances being classified is the same for all the algorithms, condition 1 allows us to compare the performance of the polarity features across the different algorithms. However, condition 2 is the more natural one. It allows us to see how the noise of neutral instances affects the performance of the polarity features.

The following sections describe three sets of experiments. First, we investigate the performance of the polarity features used together for polarity classification under condition 1. As before, the word and word+priorpol classifiers provide our baselines. In the second set of experiments, we explore the performance of different sets of features for polarity classification, again assuming perfect recognition of the polar instances. Finally, we experiment with polarity classification using all the polarity features under condition 2, automatic recognition of the polar instances.

As before, we use the development data to select the parameter settings for each algorithm. The settings for polarity classification are given in Table 14. They were selected based on the performance of the word+priorpol baseline classifier under condition 2.

8.2.1 Classification Results: Condition 1. The results for polarity classification using all the polarity features, assuming perfect neutral–polar recognition for step one, are given in Table 15. For each algorithm, we give the results for the two baseline classifiers, followed by the results for the classifier trained using all the polarity features. For the metrics where the polarity features perform statistically better than both baselines (two-sided t-test, $p \leq 0.05$), the results are given in bold.

Table 14
Algorithm settings for polarity classification.

Algorithm	Settings
BoosTexter	2,000 rounds of boosting
TiMBL	*k*=1, MVDM distance metric
Ripper	-!s, -S 0.5
SVM	linear kernel

Table 15
Results for polarity classification (step two) using gold-standard polar instances.

		Positive			Negative			Both		
	Acc	Rec	Prec	F	Rec	Prec	F	Rec	Prec	F
BoosTexter										
word token baseline	78.7	57.7	72.8	64.4	91.5	80.8	85.8	12.9	53.6	20.8
word+priorpol baseline	79.7	70.5	68.8	69.6	87.2	85.1	86.1	13.7	53.7	21.8
polarity features	**83.2**	**76.7**	74.3	**75.5**	89.7	**87.7**	**88.7**	11.8	54.2	19.4
TiMBL										
word token baseline	78.5	63.3	69.2	66.1	88.6	82.5	85.4	14.1	51.0	22.1
word+priorpol baseline	79.4	69.7	68.4	69.1	87.0	84.8	85.9	14.6	53.5	22.9
polarity features	**82.2**	**75.4**	**73.3**	**74.3**	88.5	**87.6**	**88.0**	18.3	34.6	23.9
Ripper										
word token baseline	70.0	14.5	74.5	24.3	98.3	69.7	81.6	09.1	74.4	16.2
word+priorpol baseline	78.9	75.5	65.2	70.0	83.8	86.4	85.1	09.8	75.4	17.4
polarity features	**83.2**	**77.8**	73.5	**75.6**	89.2	87.8	**88.5**	09.8	74.9	17.4
SVM										
word token baseline	69.9	62.4	69.6	65.8	76.0	84.1	79.9	14.1	31.2	19.4
word+priorpol baseline	78.2	76.7	63.7	69.6	82.2	86.7	84.4	09.8	75.4	17.4
polarity features	**81.6**	74.9	71.1	**72.9**	**88.1**	86.6	**87.3**	09.5	77.6	16.9

How well do the polarity features perform working all together? For all algorithms, the polarity classifier using all the features significantly outperforms both baselines in terms of accuracy, positive F-measure, and negative F-measure. These consistent improvements in performance across all four algorithms show that these features are quite useful for polarity classification.

One interesting thing that Table 15 reveals is that negative polarity words are much more straightforward to recognize than positive polarity words, at least in this corpus. For the negative class, precisions and recalls for the word+priorpol baseline range from 82.2 to 87.2. For the positive class, precisions and recalls for the word+priorpol baseline range from 63.7 to 76.7. However, it is with the positive class that polarity features seem to help the most. With the addition of the polarity features, positive F-measure improves by 5 points on average; improvements in negative F-measures average only 2.75 points.

8.2.2 Feature Set Evaluation. To evaluate the performance of the various features for polarity classification, we again perform a series of ablation experiments. As before, we start with the word+priorpol baseline classifier, add different sets of polarity features, train new classifiers, and compare the results of the new classifiers to the baseline.

Table 16
Polarity feature sets for evaluation.

Experiment	Features
NEGATION	negated, negated subject
POLARITY-MOD	modifies polarity, modified by polarity, conjunction polarity
SHIFTERS	general, negative, positive polarity shifters

Table 17
Results for polarity feature ablation experiments.

		Positive			**Negative**		
	Acc	Rec	Prec	F	Rec	Prec	F
BoosTexter							
NEGATION	+++	++	+++	+++	+++	+	+++
POLARITY-MOD	++	+++	+	+++	+	++	+
SHIFTERS	+	+	+	+	+	+	+
TiMBL							
NEGATION	+++	+++	+++	+++	+++	+++	+++
POLARITY-MOD	+	+	+	+	−	+	+
SHIFTERS	+	+	+	+	−	+	+
Ripper							
NEGATION	+++	−−	+++	+++	+++	−	+++
POLARITY-MOD	+	+++	++	+++	+	+	+
SHIFTERS	+	−	+	+	+	−	+
SVM							
NEGATION	+++	−	+++	+++	+++	+	+++
POLARITY-MOD	+	−	+++	+	+	−	+
SHIFTERS	+	−	+	+	+	+	+

Increases and decreases for a given metric as compared to the word+priorpol baseline are indicated by + or –, respectively;
++ or – – indicates the change is significant at the $p < 0.1$ level;
+++ or – – – indicates significance at the $p < 0.05$ level.

Table 16 lists the sets of features tested in each experiment, and Table 17 shows the results of the experiments. Results are reported as they were previously in Section 8.1.2, with increases and decreases compared to the baseline for a given metric indicated by + or –, respectively.

Looking at Table 17, we see that all three sets of polarity features help to increase performance as measured by accuracy and positive and negative F-measures. This is true for all the classification algorithms. As we might expect, including the negation features has the most marked effect on the performance of polarity classification, with statistically significant improvements for most metrics across all the algorithms.[9] The

9 Although the negation features give the best performance improvements of the three feature sets, these classifiers still do not perform as well as the respective all-feature polarity classifiers for each algorithm.

polarity-modification features also seem to be important for polarity classification, in particular for disambiguating the positive instances. For all the algorithms except TiMBL, including the polarity-modification features results in significant improvements for at least one of the positive metrics. The polarity shifters also help classification, but they seem to be the weakest of the features: Including them does not result in significant improvements for any algorithm.

Another question that is interesting to consider is how much the *word token* feature contributes to polarity classification, given all the other polarity features. Is it enough to know the prior polarity of a word, whether it is being negated, and how it is related to other polarity influencers? To answer this question, we train classifiers using all the polarity features except for *word token*. Table 18 gives the results for these classifiers; for comparison, the results for the all-feature polarity classifiers are also given. Interestingly, excluding the *word token* feature produces only small changes in the overall results. The results for BoosTexter and Ripper are slightly lower, and the results for SVM are practically unchanged. TiMBL actually shows a slight improvement, with the exception of the both class. This provides further evidence of the strength of the polarity features. Also, a classifier not tied to actual word tokens may potentially be a more domain-independent classifier.

8.2.3 Classification Results: Condition 2. The experiments in Section 8.2.1 show that the polarity features perform well under the ideal condition of perfect recognition of polar instances. The next question to consider is how well the polarity features perform under the more natural but less-than-perfect condition of automatic recognition of polar instances. To investigate this, the polarity classifiers (including the baselines) for each algorithm in these experiments start with the polar instances identified by the best performing neutral–polar classifier for that algorithm (from Section 8.1.1). The results for these experiments are given in Table 19. As before, statistically significant improvements over both baselines are given in bold.

How well do the polarity features perform in the presence of noise from misclassified neutral instances? Our first observation comes from comparing Table 15 with Table 19: Polarity classification results are much lower for all classifiers with the noise of neutral instances. Yet in spite of this, the polarity features still produce classifiers that

Table 18
Results for polarity classification without and with the *word token* feature.

	Acc	Pos F	Neg F	Both F
BoosTexter				
excluding word token	82.5	74.9	88.0	17.4
all polarity features	83.2	75.5	88.7	19.4
TiMBL				
excluding word token	83.2	75.9	88.4	17.3
all polarity features	82.2	74.3	88.0	23.9
Ripper				
excluding word token	82.9	75.4	88.3	17.4
all polarity features	83.2	75.6	88.5	17.4
SVM				
excluding word token	81.5	72.9	87.3	16.8
all polarity features	81.6	72.9	87.3	16.9

Table 19
Results for polarity classification (step two) using automatically identified polar instances.

		Positive			**Negative**			**Both**			**Neutral**		
	Acc	R	P	F	R	P	F	R	P	F	R	P	F
BoosTexter													
word token	61.5	62.3	62.7	62.5	86.4	64.6	74.0	11.4	49.3	18.5	20.8	44.5	28.3
word+priorpol	63.3	70.0	57.9	63.4	81.3	71.5	76.1	12.5	47.3	19.8	30.9	47.5	37.4
polarity feats	**65.9**	73.6	62.2	**67.4**	84.9	72.3	**78.1**	13.4	40.7	20.2	31.0	50.6	**38.4**
TiMBL													
word token	60.1	68.3	58.9	63.2	81.8	65.0	72.5	11.2	39.6	17.4	21.6	43.1	28.8
word+priorpol	61.0	73.2	53.4	61.8	80.6	69.8	74.8	12.7	41.7	19.5	23.0	44.2	30.3
polarity feats	**64.4**	75.3	58.6	**65.9**	81.1	73.0	**76.9**	16.9	32.7	22.3	**32.1**	**50.0**	**39.1**
Ripper													
word token	54.4	22.2	69.4	33.6	95.1	50.7	66.1	00.0	00.0	00.0	21.7	76.5	33.8
word+priorpol	51.4	24.0	71.7	35.9	97.7	48.9	65.1	00.0	00.0	00.0	09.2	75.8	16.3
polarity feats	54.8	**38.0**	67.2	**48.5**	95.5	52.7	67.9	00.0	00.0	00.0	14.5	66.8	23.8
SVM													
word token	64.5	70.0	60.9	65.1	70.9	74.9	72.9	16.6	41.5	23.7	53.3	51.0	52.1
word+priorpol	62.8	89.0	51.2	65.0	88.4	69.2	77.6	11.1	48.5	18.0	02.4	58.3	04.5
polarity feats	64.1	90.8	53.0	66.9	**90.4**	70.1	79.0	12.7	52.3	20.4	02.2	61.4	04.3

outperform the baselines. For three of the four algorithms, the classifier using all the polarity features has the highest accuracy. For BoosTexter and TiMBL, the improvements in accuracy over both baselines are significant. Also for all algorithms, using the polarity features gives the highest positive and negative F-measures.

Because the set of polarity instances being classified by each algorithm is different, we cannot directly compare the results from one algorithm to the next.

8.3 Two-step versus One-step Recognition of Contextual Polarity

Although the two-step approach to recognizing contextual polarity allows us to focus our investigation on the performance of features for both neutral–polar classification and polarity classification, the question remains: How does the two-step approach compare to recognizing contextual polarity in a single classification step? The results shown in Table 20 help to answer this question. The first row in Table 20 for each algorithm shows the combined result for the two stages of classification. For BoosTexter, TiMBL, and Ripper, this is the combination of results from using all the neutral–polar features for step one, together with the results from using all of the polarity features for step two.[10] For SVM, this is the combination of results from the word+priorpol baseline from step one, together with results for using all the polarity features for step two. Recall that the word+priorpol classifier was the best neutral–polar classifier for SVM (see Table 11). The second rows for BoosTexter, TiMBL, and Ripper show the results of a single classifier trained to recognize contextual polarity using all the neutral–polar and polarity features together. For SVM, the second row shows the results of classifying the contextual polarity using just the word token feature. This classifier outperformed all others for SVM. In the table, the best result for each metric for each algorithm is highlighted in bold.

When comparing the two-step and one-step approaches, contrary to our expectations, we see that the one-step approach performs about as well or better than the two-step approach for recognizing contextual polarity. For SVM, the improvement in accuracy achieved by the two-step approach is significant, but this is not true for the other algorithms. One fairly consistent difference between the two approaches is that the two-step approach scores slightly higher for neutral F-measure, and the one-step approach achieves higher F-measures for the polarity classes. The difference in negative F-measure is significant for BoosTexter, TiMBL, and Ripper. The exception to this is SVM. For SVM, the two-step approach achieves significantly higher positive and negative F-measures.

One last question we consider is how much the neutral–polar features contribute to the performance of the one-step classifiers. The third line in Table 20 for BoosTexter, TiMBL, and Ripper gives the results for a one-step classifier trained without the neutral–polar features. Although the differences are not always large, excluding the neutral–polar features consistently degrades performance in terms of accuracy and positive, negative, and neutral F-measures. The drop in negative F-measure is significant for all three algorithms, the drop in neutral F-measure is significant for BoosTexter and TiMBL, and the drop in accuracy is significant for TiMBL and Ripper (and for BoosTexter at the $p \leq 0.1$ level).

10 To clarify, Section 8.2.3 only reported results for instances identified as polar in step one. Here, we report results for all clue instances, including the instances classified as neutral in step one.

Table 20
Results for contextual polarity classification for both two-step and one-step approaches.

	Acc	Pos F	Neg F	Both F	Neutral F
BoosTexter					
two-step	**74.5**	47.1	57.5	12.9	**83.4**
one-step all feats	74.3	**49.1**	**59.8**	14.1	82.9
one-step – neut-pol feats	73.3	48.4	58.7	**16.3**	81.9
TiMBL					
two-step	**74.1**	47.6	56.4	13.8	**83.2**
one-step all feats	73.9	**49.6**	**59.3**	15.2	82.6
one-step – neut-pol feats	72.5	49.5	56.9	**21.6**	81.4
Ripper					
two-step	68.9	26.6	49.0	00.0	**80.1**
one-step all feats	**69.5**	**30.2**	**52.8**	**14.0**	79.4
one-step – neut-pol feats	67.0	28.9	33.0	11.4	78.6
SVM					
two-step	**73.1**	**46.6**	**58.0**	13.0	**82.1**
one-step	71.6	43.4	51.7	**17.0**	81.6

The modest drop in performance that we see when excluding the neutral–polar features in the one-step approach seems to suggest that discriminating between neutral and polar instances is helpful but not necessarily crucial. However, consider Figure 3. In this figure, we show the F-measures for the *positive*, *negative*, and *both* classes for the BoosTexter polarity classifier that uses the gold-standard neutral/polar instances (from Table 15) and for the BoosTexter one-step polarity classifier that uses all features (from Table 20). Plotting the same sets of results for the other three algorithms produces very similar figures. The difference when the classifiers have to contend with the noise from neutral instances is dramatic. Although Table 20 shows that there is room for improvement across all the contextual polarity classes, Figure 3 shows us that perhaps the best way to achieve these improvements is to improve the ability to discriminate the neutral class from the others.

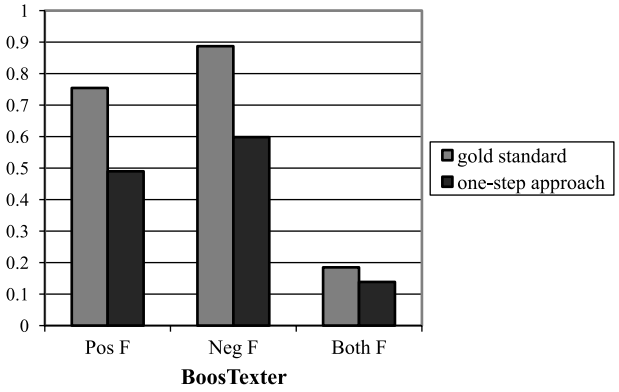

Figure 3
Chart showing the *positive*, *negative*, and *both* class F-measures for the BoosTexter classifier that uses the gold-standard neutral/polar classes and the BoosTexter one-step classifier that uses all the features.

9. Related Work

9.1 Phrase-Level Sentiment Analysis

Other researchers who have worked on classifying the contextual polarity of sentiment expressions are Yi et al. (2003), Popescu and Etzioni (2005), and Suzuki, Takamura, and Okumura (2006). Yi et al. use a lexicon and manually developed patterns to classify contextual polarity. Their patterns are high-quality, yielding quite high precision over the set of expressions that they evaluate. Popescu and Etzioni use an unsupervised classification technique called **relaxation labeling** (Hummel and Zucker 1983) to recognize the contextual polarity of words that are at the heads of select opinion phrases. They take an iterative approach, using relaxation labeling first to determine the contextual polarities of the words, then again to label the polarities of the words with respect to their targets. A third stage of relaxation labeling then is used to assign final polarities to the words, taking into consideration the presence of other polarity terms and negation. As we do, Popescu and Etzioni use features that represent conjunctions and dependency relations between polarity words. Suzuki et al. use a bootstrapping approach to classify the polarity of tuples of adjectives and their target nouns in Japanese blogs. Included in the features that they use are the words that modify the adjectives and the word that the adjective modifies. They consider the effect of a single negation term, the Japanese equivalent of *not*.

Our work in recognizing contextual polarity differs from this research on expression-level sentiment analysis in several ways. First, the set of expressions they evaluate is limited either to those that target specific items of interest, such as products and product features, or to tuples of adjectives and nouns. In contrast, we seek to classify the contextual polarity of all instances of words from a large lexicon of subjectivity clues that appear in the corpus. Included in the lexicon are not only adjectives, but nouns, verbs, adverbs, and even modals.

Our work also differs from other research in the variety of features that we use. As other researchers do, we consider negation and the words that directly modify or are modified by the expression being classified. However, with negation, we have features for both local and longer-distance types of negation, and we take care to count negation terms only when they are actually being used to negate, excluding, for example, negation terms when they are used in phrases that intensify (e.g., *not only*). We also include contextual features to capture the presence of other clue instances in the surrounding sentences, and features that represent the reliability of clues from the lexicon.

Finally, a unique aspect of the work presented in this article is the evaluation of different features for recognizing contextual polarity. We first presented the features explored in this research in Wilson, Wiebe, and Hoffman (2005), but this work significantly extends that initial evaluation. We explore the performance of features across different learning algorithms, and we evaluate not only features for discriminating between positive and negative polarity, but features for determining when a word is or is not expressing a sentiment in the first place (*neutral in context*). This is also the first work to evaluate the effect of neutral instances on the performance of features for discriminating between positive and negative contextual polarity.

9.2 Other Research in Sentiment Analysis

Recognizing contextual polarity is just one facet of the research in automatic sentiment analysis. Research ranges from work on learning the prior polarity (semantic

orientation) of words and phrases (e.g., Hatzivassiloglou and McKeown 1997; Kamps and Marx 2002; Turney and Littman 2003; Hu and Liu 2004; Kim and Hovy 2004; Esuli and Sebastiani 2005; Takamura, Inui, and Okumura 2005; Popescu and Etzioni 2005; Andreevskaia and Bergler 2006; Esuli and Sebastiani 2006a; Kanayama and Nasukawa 2006) to characterizing the sentiment of documents, such as recognizing inflammatory messages (Spertus 1997), tracking sentiment over time in online discussions (Tong 2001), and classifying the sentiment of online messages (e.g., Das and Chen 2001; Koppel and Schler 2006), customer feedback data (Gamon 2004), or product and movie reviews (e.g., Turney 2002; Pang, Lee, and Vaithyanathan 2002; Dave, Lawrence, and Pennock 2003; Beineke, Hastie, and Vaithyanathan 2004; Mullen and Collier 2004; Bai, Padman, and Airoldi 2005; Whitelaw, Garg, and Argamon 2005; Kennedy and Inkpen 2006; Koppel and Schler 2006).

Identifying prior polarity is a different task than recognizing contextual polarity, although the two tasks are complementary. The goal of identifying prior polarity is to automatically acquire the polarity of words or phrases for listing in a lexicon. Our work on recognizing contextual polarity begins with a lexicon of words with established prior polarities and then disambiguates in the corpus the polarity being expressed by the phrases in which instances of those words appear. To make the relationship between that task and ours clearer, some word lists that are used to evaluate methods for recognizing prior polarity (positive and negative word lists from the General Inquirer [Stone et al. 1966] and lists of positive and negative adjectives created for evaluation by Hatzivassiloglou and McKeown [1997]) are included in the prior-polarity lexicon used in our experiments.

For the most part, the features explored in this work differ from the ones used to identify prior polarity with just a few exceptions. Using a feature to capture conjunctions between clue instances was motivated in part by the work of Hatzivassiloglou and McKeown (1997). They use constraints on the co-occurrence in conjunctions of words with similar or opposite polarity to predict the prior polarity of adjectives. Esuli and Sebastiani (2005) consider negation in some of their experiments involving WordNet glosses. Takamura et al. (2005) use negation words and phrases, including phrases such as *lack of* that are members in our lists of polarity shifters, and conjunctive expressions that they collect from corpora.

Esuli and Sebastiani (2006a) is the only work in prior-polarity identification to include a *neutral (objective)* category and to consider a three-way classification between positive, negative, and neutral words. Although identifying prior polarity is a different task, they report a finding similar to ours, namely, that accuracy is lower when neutral words are included.

Some research in sentiment analysis classifies the sentiments of sentences. Morinaga et al. (2002), Yu and Hatzivassiloglou (2003), Kim and Hovy (2004), Hu and Liu (2004), and Grefenstette et al. (2004)[11] 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 clue instances, which is the focus of this work. Morinaga et al. only consider the positive or negative clue instance in each sentence that is closest to some target reference; Kim and Hovy, Hu and Liu, and Grefenstette et al. multiply or count the prior polarities of clue instances in the sentence.

11 In Grefenstette et al. (2004), the units that are classified are fixed windows around named entities rather than sentences.

These researchers also consider local negation to reverse polarity, with Morinaga et al. also taking into account the negating effect of words like *insufficient*. However, they do not use the other types of features that we consider in our experiments. Kaji and Kitsuregawa (2006) take a different approach to recognizing positive and negative sentences. They bootstrap from information easily obtained in "Pro" and "Con" HTML tables and lists, and from one high-precision linguistic pattern, to automatically construct a large corpus of positive and negative sentences. They then use this corpus to train a naive Bayes sentence classifier. In contrast to our work, sentiment classification in all of this research is restricted to identifying only *positive* and *negative* sentences (excluding our *both* and *neutral* categories). In addition, only one sentiment is assigned per sentence; our system assigns contextual polarity to individual expressions, which would allow for a sentence to be assigned to multiple sentiment categories. As we saw when exploring the contextual polarity annotations, it is not uncommon for sentences to contain more than one sentiment expression.

Classifying the sentiment of documents is a very different task than recognizing the contextual polarity of words and phrases. However, some researchers have reported findings about document-level classification that are similar to our findings about phrase-level classification. Bai et al. (2005) argue that dependencies among key sentiment terms are important for classifying document sentiment. Similarly, we show that features for capturing when clue instances modify each other are important for phrase-level classification, in particular, for identifying positive expressions. Gamon (2004) achieves his best results for document classification using a wide variety of features, including rich linguistic features, such as features that capture constituent structure, features that combine part-of-speech and semantic relations (e.g., sentence subject or negated context), and features that capture tense information. We also achieve our best results for phrase-level classification using a wide variety of features, many of which are linguistically rich. Kennedy and Inkpen (2006) report consistently higher results for document sentiment classification when select polarity influencers, including negators and intensifiers, are included.[12] Koppel and Schler (2006) demonstrate the importance of neutral examples for document-level classification. In this work, we show that being able to correctly identify neutral instances is also very important for phrase-level sentiment analysis.

10. Conclusions and Future Work

Being able to determine automatically the contextual polarity of words and phrases is an important problem in sentiment analysis. In the research presented in this article, we tackle this problem and show that it is much more complex than simply determining whether a word or phrase is positive or negative. In our analysis of a corpus with annotations of subjective expressions and their contextual polarity, we find that positive and negative words from a lexicon are used in neutral contexts much more often than they are used in expressions of the opposite polarity. The importance of identifying

12 Das and Chen (2001), Pang, Lee, and Vaithyanathan (2002), and Dave, Lawrence, and Pennock (2003) also represent negation. In their experiments, words which follow a negation term are tagged with a negation marker and then treated as new words. Pang, Lee and Vaithyanathan report that representing negation in this way slightly helps their results, whereas Dave, Lawrence, and Pennock report a slightly detrimental effect. Whitelaw, Garg, and Argamon (2005) also represent negation terms and intensifiers. However, in their experiments, the effect of negation is not separately evaluated, and intensifiers are not found to be beneficial.

when contextual polarity is neutral is further revealed in our classification experiments: When neutral instances are excluded, the performance of features for distinguishing between positive and negative polarity greatly improves.

A focus of this research is on understanding which features are important for recognizing contextual polarity. We experiment with a wide variety of linguistically motivated features, and we evaluate the performance of these features using several different machine learning algorithms. Features for distinguishing between neutral and polar instances are evaluated, as well as features for distinguishing between positive and negative contextual polarity. For classifying neutral and polar instances, we find that, although some features produce significant improvements over the baseline in terms of polar or neutral recall or precision, it is the combination of features together that is needed to achieve significant improvements in accuracy. For classifying positive and negative contextual polarity, features for capturing negation prove to be the most important. However, we find that features that also perform well are those that capture when a word is (or is not) modifying or being modified by other polarity terms. This suggests that identifying features that represent more complex interdependencies between polarity clues will be an important avenue for future research.

Another direction for future work will be to expand our lexicon using existing techniques for acquiring the prior polarity of words and phrases. It follows that a larger lexicon will have a greater coverage of sentiment expressions. However, expanding the lexicon with automatically acquired prior-polarity tags may result in an even greater proportion of neutral instances to contend with. Given the degradation in performance created by the neutral instances, whether expanding the lexicon automatically will result in improved performance for recognizing contextual polarity is an empirical question.

Finally, the overall goal of our research is to use phrase-level sentiment analysis in higher-level NLP tasks, such as opinion question answering and summarization.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported in part by an Andrew Mellow Predoctoral Fellowship, by the NSF under grant IIS-0208798, by the Advanced Research and Development Activity (ARDA), and by the European IST Programme through the AMIDA Integrated Project FP6-0033812.

References

Andreevskaia, Alina and Sabine Bergler. 2006. Mining WordNet for fuzzy sentiment: Sentiment tag extraction from WordNet glosses. In *Proceedings of the 11th Meeting of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, pages 209–216, Trento.

Bai, Xue, Rema Padman, and Edoardo Airoldi. 2005. On learning parsimonious models for extracting consumer opinions. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 3*, page 75.2, Waikoloa, HI.

Banfield, Ann. 1982. *Unspeakable Sentences*. Routledge and Kegan Paul, Boston.

Beineke, Philip, Trevor Hastie, and Shivakumar Vaithyanathan. 2004. The sentimental factor: Improving review classification via human-provided information. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 263–270, Barcelona.

Cohen, William W. 1996. Learning trees and rules with set-valued features. In *Proceedings of the 13th National Conference on Artificial Intelligence*, pages 709–717, Portland, OR.

Collins, Michael. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*, pages 16–23, Madrid.

Daelemans, Walter, Véronique Hoste, Fien De Meulder, and Bart Naudts. 2003a. Combined optimization of feature selection and algorithm parameter

interaction in machine learning of language. In *Proceedings of the 14th European Conference on Machine Learning (ECML-2003)*, pages 84–95, Cavtat-Dubrovnik.

Daelemans, Walter, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2003b. TiMBL: Tilburg Memory Based Learner, version 5.0 Reference Guide. ILK Technical Report 03-10, Induction of Linguistic Knowledge Research Group, Tilburg University. Available at `http://ilk.uvt.nl/downloads/pub/papers/ilk0310.pdf`.

Das, Sanjiv Ranjan and Mike Y. Chen. 2001. Yahoo! for Amazon: Sentiment parsing from small talk on the Web. In *Proceedings of the August 2001 Meeting of the European Finance Association (EFA)*, Barcelona, Spain. Available at `http://ssrn.com/abstract=276189`.

Dave, Kushal, Steve Lawrence, and David M. Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International World Wide Web Conference (WWW2003)*, Budapest. Available at `http://www2003.org`.

Esuli, Andrea and Fabrizio Sebastiani. 2005. Determining the semantic orientation of terms through gloss analysis. In *Proceedings of ACM SIGIR Conference on Information and Knowledge Management (CIKM-05)*, pages 617–624, Bremen.

Esuli, Andrea and Fabrizio Sebastiani. 2006a. Determining term subjectivity and term orientation for opinion mining. In *Proceedings the 11th Meeting of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, pages 193–200, Trento.

Esuli, Andrea and Fabrizio Sebastiani. 2006b. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC-06, the 5th Conference on Language Resources and Evaluation*, pages 417–422, Genoa.

Gamon, Michael. 2004. Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-2004)*, pages 611–617, Geneva.

Grefenstette, Gregory, Yan Qu, James G. Shanahan, and David A. Evans. 2004. Coupling niche browsers and affect analysis for an opinion mining application. In *Proceedings of the Conference Recherche d'Information Assistee par Ordinateur (RIAO-2004)*, pages 186–194, Avignon.

Hatzivassiloglou, Vasileios and Kathy McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*, pages 174–181, Madrid.

Hoste, Véronique. 2005. *Optimization Issues in Machine Learning of Coreference Resolution*. Ph.D. thesis, Language Technology Group, University of Antwerp.

Hu, Minqing and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2004 (KDD-2004)*, pages 168–177, Seattle, WA.

Hummel, Robert A. and Steven W. Zucker. 1983. On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 5(3):167–187.

Joachims, Thorsten. 1999. Making large-scale SVM learning practical. In B. Scholkopf, C. Burgess, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 169–184. MIT Press, Cambridge, MA.

Kaji, Nobuhiro and Masaru Kitsuregawa. 2006. Automatic construction of polarity-tagged corpus from HTML documents. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 452–459, Sydney.

Kamps, Jaap and Maarten Marx. 2002. Words with attitude. In *Proceedings of the 1st International Conference on Global WordNet*, pages 332–341, Mysore.

Kanayama, Hiroshi and Tetsuya Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2006)*, pages 355–363, Sydney.

Kennedy, Alistair and Diana Inkpen. 2006. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22(2):110–125.

Kim, Soo-Min and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-2004)*, pages 1267–1373, Geneva.

Koppel, Moshe and Jonathan Schler. 2006. The importance of neutral examples for learning sentiment. *Computational Intelligence*, 22(2):100–109.

Maybury, Mark T., editor. 2004. *New Directions in Question Answering*. American Association for Artificial Intelligence, Menlo Park, CA.

Morinaga, Satoshi, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. 2002. Mining product reputations on the Web. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002)*, pages 341–349, Edmonton.

Mullen, Tony and Nigel Collier. 2004. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)*, pages 412–418, Barcelona.

Nasukawa, Tetsuya and Jeonghee Yi. 2003. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd International Conference on Knowledge Capture (K-CAP 2003)*, pages 70–77, Sanibel Island, FL.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*, pages 79–86, Philadelphia, PA.

Polanyi, Livia and Annie Zaenen. 2004. Contextual valence shifters. In *Working Notes of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, pages 106–111, The AAAI Press, Menlo Park, CA.

Popescu, Ana-Maria and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of the Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005)*, pages 339–346, Vancouver.

Quirk, Randolph, Sidney Greenbaum, Geoffry Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman, New York.

Riloff, Ellen and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2003)*, pages 105–112, Sapporo.

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

Spertus, Ellen. 1997. Smokey: Automatic recognition of hostile messages. In *Proceedings of the 8th Annual Conference on Innovative Applications of Artificial Intelligence (IAAI-97)*, pages 1058–1065, Providence, RI.

Stone, Philip J., Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, Cambridge, MA.

Stoyanov, Veselin, Claire Cardie, and Janyce Wiebe. 2005. Multi-perspective question answering using the OpQA corpus. In *Proceedings of the Human Language Technologies Conference/ Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005)*, pages 923–930, Vancouver.

Suzuki, Yasuhiro, Hiroya Takamura, and Manabu Okumura. 2006. Application of semi-supervised learning to evaluative expression classification. In *Proceedings of the 7th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2006)*, pages 502–513, Mexico City.

Takamura, Hiroya, Takashi Inui, and Manabu Okumura. 2005. Extracting emotional polarity of words using spin model. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, pages 133–140, Ann Arbor, MI.

Tong, Richard. 2001. An operational system for detecting and tracking opinions in online discussions. In *Working Notes of the SIGIR Workshop on Operational Text Classification*, pages 1–6, New Orleans, LA.

Turney, Peter. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, pages 417–424, Philadelphia, PA.

Turney, Peter and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346.

Whitelaw, Casey, Navendu Garg, and Shlomo Argamon. 2005. Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM International Conference on Information and Knowledge*

Management (CIKM-2005), pages 625–631, Bremen.

Wiebe, Janyce. 1994. Tracking point of view in narrative. *Computational Linguistics*, 20(2):233–287.

Wiebe, Janyce, Rebecca Bruce, and Thomas O'Hara. 1999. Development and use of a gold standard data set for subjectivity classifications. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, pages 246–253, College Park, MD.

Wiebe, Janyce and Rada Mihalcea. 2006. Word sense and subjectivity. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1065–1072, Sydney.

Wiebe, Janyce and Ellen Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of the 6th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2005)*, pages 486–497, Mexico City.

Wiebe, Janyce, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation (formerly Computers and the Humanities)*, 39(2/3):164–210.

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

Xia, Fei and Martha Palmer. 2001. Converting dependency structures to phrase structures. In *Proceedings of the Human Language Technology Conference (HLT-2001)*, pages 1–5, San Diego, CA.

Yi, Jeonghee, Tetsuya Nasukawa, Razvan Bunescu, and Wayne Niblack. 2003. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM'03)*, pages 427–434, Melbourne, FL.

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