Relational Features in Fine-Grained Opinion Analysis

Richard Johansson*
University of Gothenburg

Alessandro Moschitti**
University of Trento

Fine-grained opinion analysis methods often make use of linguistic features but typically do not take the interaction between opinions into account. This article describes a set of experiments that demonstrate that relational features*, mainly derived from dependency-syntactic and semantic role structures, can significantly improve the performance of automatic systems for a number of fine-grained opinion analysis tasks: marking up opinion expressions, finding opinion holders, and determining the polarities of opinion expressions. These features make it possible to model the way opinions expressed in natural-language discourse* interact *in a sentence over arbitrary distances. The use of relations requires us to consider multiple opinions simultaneously, which makes the search for the optimal analysis intractable. However, a reranker can be used as a sufficiently accurate and efficient approximation.*

A number of feature sets and machine learning approaches for the rerankers are evaluated. For the task of opinion expression extraction, the best model shows a 10-point absolute improvement in soft recall on the MPQA corpus over a conventional sequence labeler based on local contextual features, while precision decreases only slightly. Significant improvements are also seen for the extended tasks where holders and polarities are considered: 10 and 7 points in recall, respectively. In addition, the systems outperform previously published results for unlabeled (6 F-measure points) and polarity-labeled (10–15 points) opinion expression extraction. Finally, as an extrinsic evaluation, the extracted MPQA-style opinion expressions are used in practical opinion mining tasks. In all scenarios considered, the machine learning features derived from the opinion expressions lead to statistically significant improvements.

1. Introduction

Automatic methods for the analysis of **opinions** (textual expressions of emotions, beliefs, and evaluations) have attracted considerable attention in the natural language

* Språkbanken, Department of Swedish, University of Gothenburg, Box 100, SE-40530 Gothenburg, Sweden. E-mail: richard.johansson@gu.se. The work described here was mainly carried out at the University of Trento.

** DISI – Department of Information Engineering and Computer Science, University of Trento, Via Sommarive 14, 38123 Trento (TN), Italy. E-mail: moschitti@disi.unitn.it.

Submission received: 11 January 2012; revised version received: 11 May 2012; accepted for publication: 11 June 2012.

doi:10.1162/COLI_a_00141

processing community in recent years (Pang and Lee 2008). Apart from their interest from a linguistic and psychological point of view, the technologies emerging from this research have obvious practical uses, either as stand-alone applications or supporting other tools such as information retrieval or question answering systems.

The research community initially focused on high-level tasks such as retrieving documents or passages expressing opinion, or classifying the polarity of a given text, and these coarse-grained problem formulations naturally led to the application of methods derived from standard retrieval or text categorization techniques. The models underlying these approaches have used very simple feature representations such as purely lexical (Pang, Lee, and Vaithyanathan 2002; Yu and Hatzivassiloglou 2003) or low-level grammatical features such as part-of-speech tags and functional words (Wiebe, Bruce, and O'Hara 1999). This is in line with the general consensus in the information retrieval community that very little can be gained by complex linguistic processing for tasks such as text categorization and search (Moschitti and Basili 2004). There are a few exceptions, such as Karlgren et al. (2010), who showed that construction features added to a bag-of-words representation resulted in improved performance on a number of coarse-grained opinion analysis tasks. Similarly, Greene and Resnik (2009) argued that a speaker's attitude can be predicted from syntactic features such as the selection of a transitive or intransitive verb frame.

In contrast to the early work, recent years have seen a shift towards more detailed problem formulations where the task is not only to find a piece of opinionated text, but also to extract a structured representation of the opinion. For instance, we may determine the person holding the opinion (the **holder**) and towards which entity or fact it is directed (the **topic**), whether it is positive or negative (the **polarity**), and the strength of the opinion (the **intensity**). The increasing complexity of representation leads us from retrieval and categorization deep into natural language processing territory; the methods used here have been inspired by information extraction and semantic role labeling, combinatorial optimization, and structured machine learning. For such tasks, deeper representations of linguistic structure have seen more use than in the coarse-grained case. Syntactic and shallow-semantic relations have repeatedly proven useful for subtasks of opinion analysis that are relational in nature, above all for determining the holder or topic of a given opinion, in which case there is considerable similarity to tasks such as semantic role labeling (Kim and Hovy 2006; Ruppenhofer, Somasundaran, and Wiebe 2008).

There has been no systematic research, however, on the role played by linguistic structure in the relations *between opinions* expressed in text, despite the fact that the opinion expressions in a sentence are not independent but organized rhetorically to achieve a communicative effect intended by the speaker. We therefore expect that the interplay between opinion expressions can be exploited to derive information useful for the analysis of opinions expressed in text. In this article, we start from this intuition and propose several novel features derived from the interdependencies between opinion expressions on the syntactic and shallow-semantic levels.

Based on these features, we devised structured prediction models for (1) extraction of opinion expressions, (2) joint expression extraction and holder extraction, and (3) joint expression extraction and polarity labeling. The models were trained using a number of discriminative machine learning methods. Because the interdependency features required us to consider more than one opinion expression at a time, the inference steps carried out at training and prediction time could not rely on commonly used opinion expression mark-up methods based on Viterbi search, but we show that an approximate search method using **reranking** suffices for this purpose: In a first step a base system

using local features and efficient search generates a small set of hypotheses, and in a second step a classifier using the complex features selects the final output from the hypothesis set. This approach allows us to make use of arbitrary features extracted from the complete set of opinion expressions in a sentence, without having to impose any restriction on the expressivity of the features. An additional advantage is that it is fairly easy to implement as long as the underlying system is able to generate *k*-best output.

The interaction-based reranking systems were evaluated on a test set extracted from the MPQA corpus, and compared to strong baselines consisting of stand-alone systems for opinion expression mark-up, opinion holder extraction, and polarity classification. Our evaluations showed that (1) the best opinion expression mark-up system we evaluated achieved a 10-point absolute improvement in soft recall, and a 5-point improvement in F-measure, over the baseline sequence labeler. Our system outperformed previously described opinion expression mark-up tools by six points in overlap F-measure. (2) The recall was boosted by almost 10 points for the holder extraction task (over three points in F-measure) by modeling the interaction of opinion expressions with respect to holders. (3) We saw an improvement for the extraction of polarity-labeled expression of four F-measure points. Our result for opinion extraction and polarity labeling is especially striking when compared with the best previously published end-to-end system for this task: 10–15 points in F-measure improvement. In addition to the performance evaluations, we studied the impact of features on the subtasks, and the effect of the choice of the machine learning method for training the reranker.

As a final extrinsic evaluation of the system, we evaluated the usefulness of its output in a number of applications. Although there have been several publications detailing the extraction of MPQA-style opinion expressions, as far as we are aware there has been no attempt to use them in an application. In contrast, we show that the opinion expressions as defined by the MPQA corpus may be used to derive machine learning features that are useful in two practical opinion mining tasks; the addition of these features leads to statistically significant improvements in all scenarios we evaluated. First, we develop a system for the extraction of **evaluations of product attributes** from product reviews (Hu and Liu 2004a, 2004b; Popescu and Etzioni 2005; Titov and McDonald 2008), and we show that the features derived from opinion expressions lead to significant improvement. Secondly, we show that fine-grained opinion structural information can even be used to build features that improve a coarse-grained sentiment task: **document polarity classification** of reviews (Pang, Lee, and Vaithyanathan 2002; Pang and Lee 2004).

After the present introduction, Section 2 gives a linguistic motivation and an overview of the related work; Section 3 describes the MPQA opinion corpus and its underlying representation; Section 4 illustrates the baseline systems: a sequence labeler for the extraction of opinion expressions and classifiers for opinion holder extraction and polarity labeling; Section 5 reports on the main contribution: the description of the interaction models and their features; finally, Section 7 presents the experimental results and Section 8 derives the conclusions.

2. Motivation and Related Work

Intuitively, interdependency features could be useful in the process of *locating* and *disambiguating* expressions of opinion. These expressions tend to occur in patterns, and the presence of one opinionated piece of text may influence our interpretation of another as opinionated or not. Consider, for instance, the word *said* in sentences (a) and (b) in Example (1), where the presence or non-presence of emotionally loaded words in the

complement clause provides evidence for the interpretation as a subjective opinion or an objective speech event. (In the example, opinionated expressions are marked S for subjective and the non-opinionated speech event O for objective.)

Example (1)
(a) "We will identify the [**culprits**]$_{S}$ of these clashes and [**punish**]$_{S}$ them," he [**said**]$_{S}$.

(b) On Monday, 80 Libyan soldiers disembarked from an Antonov transport plane carrying military equipment, an African diplomat [**said**]$_{O}$.

Moreover, opinions expressed in a sentence are interdependent when it comes to the resolution of their *holders*—the person or entity having the attitude expressed in the opinion expression. Clearly, the structure of the sentence is also influential for this task because certain linguistic constructions provide evidence for opinion holder correlation. In the most obvious case, shown in the two sentences in Example (2), pejorative words share the opinion holder with the communication and categorization verbs dominating them. (Here, opinions are marked S and holders H.)

Example (2)
(a) [**Domestic observers**]$_{H}$ [**tended to side with**]$_{S}$ the MDC, [**denouncing**]$_{S}$ the election as [**fraud-tainted**]$_{S}$ and [**unfair**]$_{S}$.

(b) [**Bush**]$_{H}$ [**labeled**]$_{S}$ North Korea, Iran and Iraq an "[**axis of evil**]$_{S}$."

In addition, interaction is important when determining opinion *polarity*. Here, relations that influence polarity interpretation include coordination, verb–complement, as well as other types of *discourse relations*. In particular, the presence of a COMPARISON discourse relation, such as contrast or concession (Prasad et al. 2008), may allow us to infer that opinion expressions have different polarities. In Example (3), we see how contrastive discourse connectives (underlined) are used when there are contrasting polarities in the surrounding opinion expressions. (Positive opinions are tagged '+', negative '-'.)

Example (3)
(a) "[**This is no blind violence but rather targeted violence**]$_{-}$," Annemie Neyts [**said**]$_{-}$. "*However*, the movement [**is more than that**]$_{+}$."

(b) "[**Trade alone will not save the world**]$_{-}$," Neyts [**said**]$_{-}$, *but* it constitutes an [**important**]$_{+}$ factor for economic development.

The problems we focus on in this article—extracting opinion expressions with holders and polarity labeling—have naturally been studied previously, especially since the release of the MPQA corpus (Wiebe, Wilson, and Cardie 2005). For the first subtask, because the MPQA corpus uses span-based annotation to represent opinions, it is natural to apply straightforward sequence labeling techniques to extract them. This idea has resulted in a number of publications (Choi, Breck, and Cardie 2006; Breck, Choi, and Cardie 2007). Such systems do not use any features describing the interaction between opinions, and it would not be possible to add interaction features because a Viterbi-based sequence labeler by construction is restricted to using local features in a small contextual window.

Works using syntactic features to extract topics and holders of opinions are numerous (Bethard et al. 2005; Kobayashi, Inui, and Matsumoto 2007; Joshi and Penstein-Rosé

2009; Wu et al. 2009). Semantic role analysis has also proven useful: Kim and Hovy (2006) used a FrameNet-based semantic role labeler to determine holder and topic of opinions. Similarly, Choi, Breck, and Cardie (2006) successfully used a PropBank-based role labeler for opinion holder extraction, and Wiegand and Klakow (2010) recently applied tree kernel learning methods on a combination of syntactic and semantic role trees for extracting holders, but did not consider their relations to the opinion expressions. Ruppenhofer, Somasundaran, and Wiebe (2008) argued that semantic role techniques are useful but not completely sufficient for holder and topic identification, and that other linguistic phenomena must be studied as well. Choi, Breck, and Cardie (2006) built a joint model of opinion expression extraction and holder extraction and applied integer linear programming to carry out the optimization step.

While the tasks of opinion expression detection and polarity classification of opinion expressions (Wilson, Wiebe, and Hoffmann 2009) have mostly been studied in isolation, Choi and Cardie (2010) developed a sequence labeler that simultaneously extracted opinion expressions and assigned them polarity values and this is so far the only published result on joint opinion segmentation and polarity classification. Their experiment, however, lacked the obvious baseline: a standard pipeline consisting of an expression tagger followed by a polarity classifier. In addition, although their model is the first end-to-end system for opinion expression extraction and polarity classification, it is still based on sequence labeling and thus by construction limited in feature expressivity.

On a conceptual level, discourse-oriented approaches (Asher, Benamara, and Mathieu 2009; Somasundaran et al. 2009; Zirn et al. 2011) applying interaction features for polarity classification are arguably the most related because they are driven by a vision similar to ours: Individual opinion expressions interplay in discourse and thus provide information about each other. On a practical level there are obvious differences, since our features are extracted from syntactic and shallow-semantic linguistic representations, which we argue are *reflections* of discourse structure, while they extract features directly from a discourse representation. It is doubtful whether automatic discourse representation extraction in text is currently mature enough to provide informative features, whereas syntactic parsing and shallow-semantic analysis are today fairly reliable. Another related line of work is represented by Choi and Cardie (2008), where interaction features based on compositional semantics were used in a joint model for the assignment of polarity values to pre-segmented opinion expressions in a sentence.

3. The MPQA Corpus and its Annotation of Opinion Expressions

The most detailed linguistic resource useful for developing automatic systems for opinion analysis is the MPQA corpus (Wiebe, Wilson, and Cardie 2005). In this article, we use the word **opinion** in its broadest sense, equivalent to the word **private state** used by the MPQA annotators (page 2): "opinions, emotions, sentiments, speculations, evaluations, and other private states (Quirk et al. 1985), i.e., internal states that cannot be directly observed by others."

The central building block in the MPQA annotation is the **opinion expression** (or subjective expression): A text piece that expresses a private state, allowing us to draw the conclusion that someone has a particular emotion or belief about some topic. Identifying these units allows us to carry out further analysis, such as the determination of opinion holder and the polarity of the opinion. The annotation scheme defines two types of opinion expressions: **direct subjective expressions** (DSEs), which are explicit

mentions of attitude or evaluation, and **expressive subjective elements** (ESEs), which signal the attitude of the speaker by the choice of words. The prototypical example of a DSE would be a verb of statement, feeling, or categorization such as *praise* or *disgust*. ESEs, on the other hand, are less easy to categorize syntactically; prototypical examples would include simple value-expressing adjectives such as *beautiful* and strongly charged words like *appeasement* or *big government*. In addition to DSEs and ESEs, the corpus also contains annotation for non-subjective statements, which are referred to as **objective speech events** (OSEs). Some words such as *say* may appear as DSEs or OSEs depending on the context, so for an automatic system there is a need for disambiguation.

Example (4) shows a number of sentences from the MPQA corpus where DSEs and ESEs have been manually annotated.

Example (4)
(a) He [**made such charges**]$_{\text{DSE}}$ [**despite the fact**]$_{\text{ESE}}$ that women's political, social, and cultural participation is [**not less than that**]$_{\text{ESE}}$ of men.

(b) [**However**]$_{\text{ESE}}$, it is becoming [**rather fashionable**]$_{\text{ESE}}$ to [**exchange harsh words**]$_{\text{DSE}}$ with each other [**like kids**]$_{\text{ESE}}$.

(c) For instance, he [**denounced**]$_{\text{DSE}}$ as a [**human rights violation**]$_{\text{ESE}}$ the banning and seizure of satellite dishes in Iran.

(d) This [**is viewed**]$_{\text{DSE}}$ as the [**main impediment**]$_{\text{ESE}}$ to the establishment of political order in the country.

Every expression in the corpus is connected to an **opinion holder**,[1] an entity that experiences the sentiment or utters the evaluation that appears textually in the opinion expression. For DSEs, it is often fairly straightforward to find the opinion holders because they tend to be realized as direct semantic arguments filling semantic roles such as SPEAKER or EXPERIENCER—and the DSEs tend to be verbs or nominalizations. For ESEs, the connection between the expression and the opinion holder is typically less clear-cut than for DSEs; the holder is more frequently implicit or located outside the sentence for ESEs than for DSEs.

The MPQA corpus does not annotate links directly from opinion expressions to particular mentions of a holder entity. Instead, the opinions are connected to **holder coreference chains** that may span the whole text. Some opinion expressions are linked to entities that are not explicitly mentioned in the text. If this entity is the author of the text, it is called the **writer**, otherwise **implicit**. Example (5) shows sentences annotated with expressions and holders.

Example (5)
(a) For instance, [**he**]$_{\text{H1}}$ [**denounced**]$_{\text{DSE/H1}}$ as a [**human rights violation**]$_{\text{ESE/H1}}$ the banning and seizure of satellite dishes in Iran.

(b) [**(writer)**]$_{\text{H1}}$: [**He**]$_{\text{H2}}$ [**made such charges**]$_{\text{DSE/H2}}$ [**despite the fact**]$_{\text{ESE/H1}}$ that women's political, social, and cultural participation is [**not less than that**]$_{\text{ESE/H1}}$ of men.

(c) [**(implicit)**]$_{\text{H1}}$: This [**is viewed**]$_{\text{DSE/H1}}$ as the [**main impediment**]$_{\text{ESE/H1}}$ to the establishment of political order in the country.

1 The MPQA uses the term *source* but we prefer the term *holder* because it seems to be more common.

Finally, MPQA associates opinion expressions (DSEs and ESEs, but not OSEs) with a **polarity** feature taking the values POSITIVE, NEGATIVE, NEUTRAL, and BOTH, and with an **intensity** feature taking the values LOW, MEDIUM, HIGH, and EXTREME. The two sentences in Example (6) from the MPQA corpus show opinion expressions with polarities. Positive polarity is represented with a '+' and negative with a '-'.

Example (6)
(a) We foresaw electoral [**fraud**]$_-$ but not [**daylight robbery**]$_-$.

(b) Join in this [**wonderful**]$_+$ event and help Jameson Camp continue to provide the year-round support that gives kids a [**chance to create dreams**]$_+$.

The corpus does not currently contain annotation of **topics** (evaluees) of opinions, although there have been efforts to add this separately (Stoyanov and Cardie 2008).

4. Baseline Systems for Fine-Grained Opinion Analysis

The assessment of our reranking-based systems requires us to compare against strong baselines. We developed (1) a sequence labeler for opinion expression extraction similar to that by Breck, Choi, and Cardie (2007), (2) a set of classifiers to determine the opinion holder, and (3) a multiclass classifier that assigns polarity to a given opinion expression similar to that described by Wilson, Wiebe, and Hoffmann (2009). These tools were also used to generate the hypothesis sets for the rerankers described in Section 5.

4.1 Sequence Labeler for Opinion Expression Mark-up

To extract opinion expressions, we implemented a standard sequence labeler for subjective expression mark-up similar to the approach by Breck, Choi, and Cardie (2007). The sequence labeler extracted basic grammatical and lexical features (word, lemma, and POS tag), as well as prior polarity and intensity features derived from the lexicon created by Wilson, Wiebe, and Hoffmann (2005), which we refer to as **subjectivity clues**. It is important to note that prior subjectivity does not always imply subjectivity in a particular context; this is why contextual features are essential for this task. The grammatical features and subjectivity clues were extracted in a window of size 3 around the word in focus. We encoded the opinionated expression brackets by means of the IOB2 scheme (Tjong Kim Sang and Veenstra 1999). When using this representation, we are unable to handle overlapping opinion expressions, but they are fortunately rare.

To exemplify, Figure 1 shows an example of a sentence and how it is processed by the sequence labeler. The ESE *defenseless situation* is encoded in IOB2 as two tags `B-ESE` and `I-ESE`. There are four input columns (words, lemmas, POS tags, subjectivity clues) and one output column (opinion expression tags in IOB2 encoding). The figure also shows the sliding window from which the feature extraction function can extract features when predicting an output tag (at the arrow).

We trained the model using the method by Collins (2002), with a Viterbi decoder and the online Passive–Aggressive algorithm (Crammer et al. 2006) to estimate the model weights. The learning algorithm parameters were tuned on a development set. When searching for the best value of the C parameter, we varied it along a log scale from

HRW	HRW	NNP	–	**O**
has	have	VBZ	–	**O**
denounced	denounce	VBN	str/neg	**B–DSE**
the	the	DT	–	**O**
defenseless	defenseless	JJ	–	**B–ESE**
situation	situation	NN	–	**I–ESE**
of	of	IN	–	←
these	this	DT	–	
prisoners	prisoner	NNS	weak/neg	

Figure 1
Sequence labeling example.

0.001 to 100, and the best value was 0.1. We used the max-loss version of the algorithm and ten iterations through the training set.

4.2 Classifiers for Opinion Holder Extraction

The problem of extracting opinion holders for a given opinion expression is in many ways similar to argument detection in semantic role labeling (Choi, Breck, and Cardie 2006; Ruppenhofer, Somasundaran, and Wiebe 2008). For instance, in the simplest case when the opinion expression is a verb of evaluation or categorization, finding the holder would entail finding a semantic argument such as an EXPERIENCER or COMMUNICATOR. We therefore approached this problem using methods inspired by semantic role labeling: Given an opinion expression in a sentence, we define binary classifiers that decide whether each noun phrase of the sentence is its holder or not. Separate classifiers were trained to extract holders for DSEs, ESEs, and OSEs.

Hereafter, we describe the feature set used by the classifiers. Our walkthrough example is given by the sentence in Figure 1. Some features are derived from the syntactic and shallow semantic analysis of the sentence, shown in Figure 2 (Section 6.1 gives more details on this).

SYNTACTIC PATH. Similarly to the path feature widely used in semantic role labeling (SRL), we extract a feature representing the path in the dependency tree between the expression and the holder (Johansson and Nugues 2008). For instance, the path from *denounced* to *HRW* in the example is `VC`↑`SBJ`↓.

SHALLOW-SEMANTIC RELATION. If there is a direct shallow-semantic relation between the expression and the holder, we use a feature representing its semantic role, such as `A0` between *denounced* and *HRW*.

EXPRESSION HEAD WORD, POS, AND LEMMA. `denounced`, `VBD`, `denounce` for *denounced*; `situation`, `NN`, `situation` for *defenseless situation*.

HEAD WORD AND POS OF HOLDER CANDIDATE. `HRW`, `NNP` for *HRW*.

DOMINATING EXPRESSION TYPE. When locating the holder for the ESE *defenseless situation*, we extract a feature representing the fact that it is syntactically dominated by a DSE. At test time, the expression and its type were extracted automatically.

CONTEXT WORDS AND POS FOR HOLDER CANDIDATE. Words adjacent to the left and right; for *HRW* we extract `Right:has`, `Right:VBZ`.

EXPRESSION VERB VOICE. Similar to the common voice feature used in SRL. Takes the values `Active`, `Passive`, and `None` (for non-verbal opinion expressions). In the example, we get `Active` for *denounced* and `None` for *defenseless situation*.

EXPRESSION DEPENDENCY RELATION TO PARENT. `VC` and `OBJ`, respectively.

There are also differences compared with typical argument extraction in SRL, however. First, as outlined in Section 3, it is important to note that the MPQA corpus does not annotate direct links from opinions to holders, but from opinions to *holder coreference chains*. To handle this issue, we used the following approach when training the classifier: We created a positive training instance for each member of the coreference chain occurring in the same sentence as the opinion, and negative training instances for all other noun phrases in the sentence. We do not use coreference information at test time, in order for the system not to rely on automatic coreference resolution.

A second complication is that in addition to the explicit noun phrases in the same sentence, an opinion may be linked to an *implicit* holder; a special case of implicit holder is the *writer* of the text. To detect implicit and writer links, we trained two separate classifiers that did not use the features requiring a holder phrase. We did not try to link opinion expressions to explicitly expressed holders *outside* the sentence; this is an interesting open problem with some connections to inter-sentential semantic role labeling, a problem whose study is in its infancy (Gerber and Chai 2010).

We implemented the classifiers as linear support vector machines (SVMs; Boser, Guyon, and Vapnik 1992) using the LIBLINEAR software (Fan et al. 2008). To handle the restriction that every expression can have at most one holder, the classifier selects only the highest-scoring opinion holder candidate at test time. We tuned the learning parameters on a development set, and the best results were obtained with an L2-regularized L2-loss SVM and a C value of 1.

4.3 Polarity Classifier

Given an expression, we use a classifier to assign a polarity value: POSITIVE, NEUTRAL, or NEGATIVE. Following Choi and Cardie (2010), the polarity value BOTH was mapped to NEUTRAL—the expressions having this value were in any case very few. In the cases where the polarity value was empty or missing, we set the polarity to NEUTRAL. In addition, the annotators of the MPQA corpus may use special uncertainty labels in the case where the annotator was unsure of which polarity to assign, such as UNCERTAIN-POSITIVE. In these cases, we just removed the uncertainty label.

We again trained SVMs to carry out this classification. The problem of polarity classification has been studied in detail by Wilson, Wiebe, and Hoffmann (2009), who used a set of carefully devised linguistic features. Our classifier is simpler and is based on fairly shallow features. Like the sequence labeler for opinion expressions, this classifier made use of the lexicon of subjectivity clues.

The feature set used by the polarity classifier consisted of the following features. The examples come from the opinion expression *defenseless situation* in Figure 1.

WORDS IN EXPRESSION: `defenseless`, `situation`.

POS TAGS IN EXPRESSION: `JJ`, `NN`

SUBJECTIVITY CLUES OF WORDS IN EXPRESSION: None.

WORD BIGRAMS IN EXPRESSION: `defenseless_situation`.

WORDS BEFORE AND AFTER EXPRESSION: `B:the`, `A:of`.

POS TAGS BEFORE AND AFTER EXPRESSION: `B:DT`, `A:IN`.

To train the support vector classifiers, we again used LIBLINEAR with the same parameters. The three-class classification problem was binarized using the one-versus-all method.

5. Fine-Grained Opinion Analysis with Interaction Features

Because there is a combinatorial number of ways to segment a sentence into opinion expressions, and label the opinion expressions with type labels (DSE, ESE, OSE) as well as polarities and opinion holders, the tractability of the opinion expression segmentation task will obviously depend on whether we impose restrictions on the problem in a way that allows for efficient inference. Most previous work (Choi, Breck, and Cardie 2006; Breck, Choi, and Cardie 2007; Choi and Cardie 2010) used Markov factorizations and could thus apply standard sequence labeling techniques where the arg max step was carried out using the Viterbi algorithm (as described in Section 4.1). As we argued in the introduction, however, it makes linguistic sense that opinion expression segmentation and other tasks could be improved if *relations* between possible expressions were considered; these relations can be syntactic or semantic in nature, for instance.

We will show that adding relational features causes the Markov assumption to break down and the problem of finding the best analysis to become computationally intractable. We thus have to turn to approximate inference methods based on reranking, which can be trained efficiently.

5.1 Formalization of the Model

We formulate the problem of extracting the opinion structure (the set of opinion expressions, and possibly also their holders or polarities) from a given sentence as a structured prediction problem

$$\hat{y} = \arg\max_{y} \; w \cdot \Phi(x, y) \tag{1}$$

where w is a weight vector and $\Phi(x, y)$ a feature extraction function representing a sentence x and an opinion structure y as a high-dimensional feature vector. We now further decompose the feature representation Φ into a local part Φ_L and a nonlocal part Φ_{NL}:

$$\Phi = \Phi_L + \Phi_{NL} \tag{2}$$

where Φ_L is a standard first-order Markov factorization, and Φ_{NL} represents the nonlocal interactions between *pairs* of opinion expressions:

$$\Phi_{NL}(x, y) = \sum_{e_i, e_j \in y, e_i \neq e_j} \phi_p(e_i, e_j, x) \tag{3}$$

The feature function ϕ_p represents a pair of opinion expressions e_i and e_j and their interaction in the sentence x, such as the syntactic and semantic relations connecting them.

5.2 Approximate Inference with Interaction Features

It is easy to see that the inference step $\arg\max_y \; w \cdot \Phi(x, y)$ is NP-hard for unrestricted pairwise interaction feature representations ϕ: This class of models includes simpler ones such as loopy Markov random fields, where inference is known to be NP-hard and require the use of approximate approaches such as belief propagation. Although it is possible that search algorithms for approximate inference in our model could be devised along similar lines, we sidestepped this issue by using a *reranking* decomposition of the problem:

- Apply a simple model based on local context features Φ_L but no structural interaction features. Generate a small hypothesis set of size k.
- Apply a complex model using interaction features Φ_{NL} to pick the top hypothesis from the hypothesis set.

The advantages of a reranking approach compared with more complex approaches requiring advanced search techniques are mainly simplicity and efficiency: This approach is conceptually simple and fairly easy to implement provided that k-best output can be generated efficiently, which is certainly true for the Viterbi-based sequence labeler described in Section 4.1. The features can then be arbitrarily complex because we do not have to think about how the problem structure affects the algorithmic complexity of the inference step. Reranking has been used in a wide range of applications, starting in speech recognition (Schwartz and Austin 1991) and very commonly in syntactic parsing of natural language (Collins 2000).

The hypothesis generation procedure becomes slightly more complex when polarity values and opinion holders of the opinion expressions enter the picture. In that case, we not only need hypotheses generated by a sequence labeler, but also the outputs of a secondary classifier: the holder extractor (Section 4.2) or the polarity classifier (Section 4.3). The hypothesis set generation thus proceeds as follows:

- For a given sentence, let the base sequence labeler generate up to k_1 sequences of unlabeled opinion expressions;
- for every sequence, apply the secondary classifier to generate up to k_2 outputs.

The hypothesis set size is thus at most $k_1 \cdot k_2$.

To illustrate this process we give a hypothetical example, assuming $k_1 = k_2 = 2$ and the sentence *The appeasement emboldened the terrorists*. We first apply the opinion expression extractor to generate a set of k_1 possible segmentations of the sentence:

The [appeasement] emboldened the [terrorists]

The [appeasement] [emboldened] the [terrorists]

In the second step, we add polarity values, up to k_2 labelings for every segmentation candidate:

The [appeasement]$_-$ emboldened the [terrorists]$_-$

The [appeasement]$_-$ [emboldened]$_+$ the [terrorists]$_-$

The [appeasement]$_0$ emboldened the [terrorists]$_-$

The [appeasement]$_-$ [emboldened]$_0$ the [terrorists]$_-$

5.3 Training the Rerankers

In addition to the approximate inference method to carry out the maximization of Equation (1), we still need a machine learning method to assign weights to the vector w by estimating on a training set. We investigated a number of machine learning approaches to train the rerankers.

5.3.1 Structured SVM Learning. We first applied the method of **large-margin estimation** for structured output spaces, also known as **structured support vector machines**. In this method, we use quadratic optimization to find the smallest weight vector w that satisfies the constraint that the difference in output score between the correct output y and an incorrect output $\hat{y}$ should be at least $\Delta(y, \hat{y})$, where Δ is a **loss function** based on the degree of error in the output $\hat{y}$ with respect to the gold standard y. This is a generalization of the well-known support vector machine from binary classification to prediction of structured objects.

Formally, for a given training set $\mathcal{T} = \{\langle x_i, y_i \rangle\}$ where the output space for the input x_i is $\mathcal{Y}_i$, we state the learning problem as a constrained quadratic optimization program:

$$\begin{array}{ll} \text{minimize}_w & \|w\|^2 \\ \text{subject to} & w \cdot (\Phi(x_i, y_i) - \Phi(x_i, y_{ij})) \geq \Delta(y_i, y_{ij}), \\ & \forall \langle x_i, y_i \rangle \in \mathcal{T}, y_{ij} \in \mathcal{Y}_i \end{array} \tag{4}$$

Because real-world data tend to be noisy, this optimization problem is usually also regularized to reduce overfitting, which leads to the introduction of a parameter C as in regular support vector machines (see Taskar, Guestrin, and Koller [2004] inter alia for details).

The optimization problem (4) is usually not solved directly because the number of constraints makes a direct solution of the optimization program intractable for most realistic types of problems. Instead, an approximation has to be used in practice, and we used the SVM^{struct} software package (Tsochantaridis et al. 2005; Joachims, Finley, and Yu 2009), which finds a solution to the quadratic program by successively finding its most violated constraints and adding them to a working set. We used the default values for the learning parameters, except for the parameter C, which was determined by optimizing on a development set. This resulted in a C value of 500.

We defined the loss function Δ as 1 minus the intersection F-measure defined in Section 7.1. When training rerankers for the complex extraction tasks (expressions + holders or expressions + polarities), we used a weighted combination of F-measures for the primary task (expressions) and the secondary task (holders or polarities, see Sections 7.1.1 and 7.1.2, respectively).

5.3.2 On-line Learning. In addition to the structured SVM learning method, we trained models using two variants of on-line learning. Such learning methods are a feasible alternative for performance reasons. We investigated two on-line learning algorithms: the popular **structured perceptron** (Collins 2002) and the **Passive–Aggressive** (PA) algorithm (Crammer et al. 2006). To increase robustness, we used an averaged implementation (Freund and Schapire 1999) of both algorithms.

The difference between the two algorithms is the way the weight vector is incremented in each step. In the perceptron, for a given input x, we compute an update to the current weight vector by computing the difference between the correct output y and the predicted output $\hat{y}$. Pseudocode is given in Algorithm 1.

Algorithm 1 The structured perceptron algorithm.

function PERCEPTRON($\mathcal{T}, N$)
input Training set $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{T}$
Number of iterations N
$w_0 \leftarrow (0, \ldots, 0)$
repeat N times
for (x, y) in $\mathcal{T}$
$\hat{y} \leftarrow \arg\max_h w \cdot \Phi(x, h)$
$w_{i+1} \leftarrow w_i + \Phi(x, y) - \Phi(x, \hat{y})$
$i \leftarrow i + 1$
return $\frac{1}{NT} \sum_{i=1}^{NT} w_i$

The PA algorithm, with pseudocode in Algorithm 2, is based on the theory of large-margin learning similar to the structured SVM. Here we instead base the update step on the $\hat{y}$ that violates the margin constraints maximally, also taking the loss function Δ into account. The update step length τ is computed based on the margin; this update is bounded by a regularization constant C, which we set to 0.005 after tuning on a development set. The number N of iterations through the training set was 8 for both on-line algorithms.

Algorithm 2 The on-line passive–aggressive algorithm.

function PASSIVE–AGGRESSIVE($\mathcal{T}, N, C$)
input Training set $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{T}$
Number of iterations N
Regularization parameter C
$w_0 \leftarrow (0, \ldots, 0)$
repeat N times
for (x, y) in $\mathcal{T}$
$\hat{y} \leftarrow \arg\max_h w \cdot (\Phi(x, h) - \Phi(x, y)) + \sqrt{\Delta(y, h)}$
$\tau \leftarrow \min\left(C, \frac{w \cdot (\Phi(x,\hat{y}) - \Phi(x,y)) + \sqrt{\Delta(y,\hat{y})}}{\|\Phi(x,\hat{y}) - \Phi(x,y)\|^2}\right)$
$w_{i+1} \leftarrow w_i + \tau(\Phi(x, y) - \Phi(x, \hat{y}))$
$i \leftarrow i + 1$
return $\frac{1}{NT} \sum_{i=1}^{NT} w_i$

6. Overview of the Interaction Features

The feature extraction function Φ_{NL} extracts three groups of interaction features: (1) features considering the opinion expressions only; (2) features considering opinion holders; and (3) features considering polarity values.

In addition to the interaction features Φ_{NL}, the rerankers used features representing the scores output by the base models (opinion expression sequence labeler and secondary classifiers); they did not directly use the local features Φ_L. We normalized the scores over the k candidates so that their exponentials summed to 1.

6.1 Syntactic and Shallow Semantic Analysis

The features used by the rerankers, as well as the opinion holder extractor in Section 4.2, are to a large extent derived from syntactic and semantic role structures. To extract them, we used the syntactic–semantic parser by Johansson and Nugues (2008), which annotates the sentences with dependency syntax (Mel'čuk 1988) and shallow semantics in the PropBank (Palmer, Gildea, and Kingsbury 2005) and NomBank (Meyers et al. 2004) frameworks, using the format of the CoNLL-2008 Shared Task (Surdeanu et al. 2008). The system includes a sense disambiguator that assigns PropBank or NomBank senses to the predicate words.

Figure 2 shows an example of the structure of the annotation: The sentence *HRW denounced the defenseless situation of these prisoners*, where *denounced* is a DSE and *defenseless situation* is an ESE, has been annotated with dependency syntax (above the text) and semantic role structure (below the text). The predicate *denounced*, which is an instance of the PropBank frame `denounce.01`, has two semantic arguments: the Speaker (A0, or Agent in VerbNet terminology) and the Subject (A1, or Theme), which are realized on the surface-syntactic level as a subject and a direct object, respectively. Similarly, *situation* has the NomBank frame `situation.01` and an EXPERIENCER semantic argument (A0).

6.2 Opinion Expression Interaction Features

The rerankers use two types of structural features: syntactic features extracted from the dependency tree, and semantic features extracted from the predicate–argument (semantic role) graph.

The syntactic features are based on paths through the dependency tree. This leads to a minor complication for multiword opinion expressions; we select the shortest possible path in such cases. For instance, in the sentence in Figure 2, the path will be computed between *denounced* and *situation*.

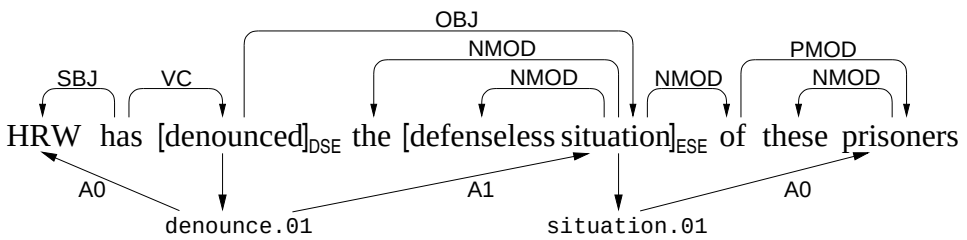

Figure 2
Example sentence and its syntactic and shallow-semantic analysis.

We used the following syntactic interaction features. All examples refer to Figure 2.

SYNTACTIC PATH. Given a pair of opinion expressions, we use a feature representing the labels of the two expressions and the path between them through the syntactic tree, following standard practice from dependency-based semantic role labeling (Johansson and Nugues 2008). For instance, for the DSE *denounced* and the ESE *defenseless situation* in Figure 2, we represent the syntactic configuration using the feature `DSE:OBJ↓:ESE`, meaning that the syntactic relation between the DSE and the ESE consists of a single link representing a grammatical object.

LEXICALIZED PATH. Same as above, but with lexical information attached: `DSE/denounced:OBJ↓:ESE/situation`.

DOMINANCE. In addition to the features based on syntactic paths, we created a more generic feature template describing dominance relations between expressions. For instance, from the graph in Figure 2, we extract the feature `DSE/denounced→ESE/situation`, meaning that a DSE with the word *denounced* dominates an ESE with the word *situation*.

The features based on semantic roles were the following:

PREDICATE SENSE LABEL. For every predicate found inside an opinion expression, we add a feature consisting of the expression label and the predicate sense identifier. For instance, the verb *denounced*, which is also a DSE, is represented with the feature `DSE/denounce.01`.

PREDICATE AND ARGUMENT LABEL. For every argument of a predicate inside an opinion expression, we also create a feature representing the predicate–argument pair: `DSE/denounced.01:A0`.

CONNECTING ARGUMENT LABEL. When a predicate inside some opinion expression is connected to some argument inside another opinion expression, we use a feature consisting of the two expression labels and the argument label. For instance, the ESE *defenseless situation* is connected to the DSE *denounced* via an A1 label, and we represent this using a feature `DSE:A1:ESE`.

6.3 Opinion Holder Interaction Features

In addition, we modeled the interaction between different opinions with respect to their holders. We used the following two features to represent this interaction:

SHARED HOLDERS. A feature representing whether or not two opinion expressions have the same holder. For instance, if a DSE dominates an ESE and they have the same holder as in Figure 2, where the holder is *HRW*, we represent this by the feature `DSE:ESE:true`.

HOLDER TYPES + PATH. A feature representing the types of the holders, combined with the syntactic path between the expressions. The types take the following possible values: explicit, implicit, writer. In Figure 2, we would thus extract the feature `DSE/Expl:OBJ↓:ESE/Expl`.

6.4 Polarity Interaction Features

The model used the following features that take the polarities of the expressions into account. These features are extracted from DSEs and ESEs only, because the OSEs have no polarity values. The examples of extracted features are given with respect to the two opinion expressions (*denounced* and *defenseless situation*) in Figure 2, both of which have a negative polarity value.

POLARITY PAIR. For every pair of opinion expressions in the sentence, we create a feature consisting of the pair of polarity values, such as NEGATIVE:NEGATIVE.

POLARITY PAIR AND SYNTACTIC PATH. NEGATIVE:OBJ↓:NEGATIVE.

POLARITY PAIR AND SYNTACTIC DOMINANCE. NEGATIVE→NEGATIVE.

POLARITY PAIR AND WORD PAIR. NEGATIVE-denounced:NEGATIVE-situation.

POLARITY PAIR AND EXPRESSION TYPES. Adds the expression types (ESE or DSE) to the polarity pair: DSE-NEGATIVE:ESE-NEGATIVE.

POLARITY PAIR AND TYPES AND SYNTACTIC PATH. Adds syntactic information to the type and polarity combination: DSE-NEGATIVE:OBJ↓:ESE-NEGATIVE.

POLARITY PAIR AND SHALLOW-SEMANTIC RELATION. When two opinion expressions are directly connected through a link in the shallow-semantic structure, we create a feature based on the semantic role label of the connecting link: NEGATIVE:A1:NEGATIVE.

POLARITY PAIR AND WORDS ALONG SYNTACTIC PATH. We follow the syntactic path between the two expressions and create a feature for every word we pass on the way. In the example, no such feature is extracted because the expressions are directly connected.

7. Experiments

We trained and evaluated the rerankers on version 2.0 of the MPQA corpus,[2] which contains 692 documents. We discarded one document whose annotation was garbled and we split the remaining 691 into a training set (541 documents) and a test set (150 documents). We also set aside a development set of 90 documents from the training set that we used when developing features and tuning learning algorithm parameters; all experiments described in this article, however, used models that were trained on the full training set. Table 1 shows some statistics about the training and test sets: the number of documents and sentences; the number of DSEs, ESEs, and OSEs; and the number of expressions marked with the various polarity labels.

We considered three experimental settings: (1) opinion expression extraction; (2) joint opinion expression and holder extraction; and (3) joint opinion expression and polarity classification. Finally, the polarity-based opinion extraction system was used in an extrinsic evaluation: document polarity classification of movie reviews.

To generate the training data for the rerankers, we carried out a 5-fold hold-out procedure: We split the training set into five pieces, trained a sequence labeler and secondary classifiers on pieces 1–4, applied them to piece 5, and so on.

2 http://www.cs.pitt.edu/mpqa/databaserelease/.

Table 1
Statistics for the training and test splits of the MPQA collection.

	Training	Test
Documents	541	150
Sentences	12,010	3,743
DSE	8,389	2,442
ESE	10,279	3,370
OSE	3,048	704
POSITIVE	3,192	1,049
NEGATIVE	6,093	1,675
NEUTRAL	9,105	3,007
BOTH	278	81

7.1 Evaluation Metrics

Because expression boundaries are hard to define rigorously (Wiebe, Wilson, and Cardie 2005), our evaluations mainly used **intersection-based precision and recall** measures to score the quality of the system output. The idea is to assign values between 0 and 1, as opposed to traditional precision and recall where a span is counted as either correctly or incorrectly detected. We thus define the **span coverage** c of a span s (a set of token indices) with respect to another span s', which measures how well s' is covered by s:

$$c(s, s') = \frac{|s \cap s'|}{|s'|}$$

In this formula, $|s|$ means the length of the span s, and the intersection $\cap$ gives the set of token indices that two spans have in common. Because our evaluation takes span labels (DSE, ESE, OSE) into account, we set $c(s, s')$ to zero if the labels associated with s and s' are different.

Using the span coverage, we define the **span set coverage** C of a set of spans $\boldsymbol{S}$ with respect to a set $\boldsymbol{S'}$:

$$C(\boldsymbol{S}, \boldsymbol{S'}) = \sum_{s_j \in \boldsymbol{S}} \sum_{s'_k \in \boldsymbol{S'}} c(s_j, s'_k)$$

We now define the intersection-based precision P and recall R of a proposed set of spans $\hat{\boldsymbol{S}}$ with respect to a gold standard set $\boldsymbol{S}$ as follows:

$$P(\boldsymbol{S}, \hat{\boldsymbol{S}}) = \frac{C(\boldsymbol{S}, \hat{\boldsymbol{S}})}{|\hat{\boldsymbol{S}}|} \qquad R(\boldsymbol{S}, \hat{\boldsymbol{S}}) = \frac{C(\hat{\boldsymbol{S}}, \boldsymbol{S})}{|\boldsymbol{S}|}$$

Note that in this formula, $|S|$ means the number of spans in a set S.

Conventionally, when measuring the quality of a system for an information extraction task, a predicted entity is counted as correct if it exactly matches the boundaries of a corresponding entity in the gold standard; there is thus no reward for close matches. Because the boundaries of the spans annotated in the MPQA corpus are not strictly

defined in the annotation guidelines (Wiebe, Wilson, and Cardie 2005), however, measuring precision and recall using exact boundary scoring will result in figures that are too low to be indicative of the usefulness of the system. Therefore, most work using this corpus instead use overlap-based precision and recall measures, where a span is counted as correctly detected if it *overlaps* with a span in the gold standard (Choi, Breck, and Cardie 2006; Breck, Choi, and Cardie 2007). As pointed out by Breck, Choi, and Cardie (2007), this is problematic because it will tend to reward long spans—for instance, a span covering the whole sentence will always be counted as correct if the gold standard contains any span for that sentence. Conversely, the overlap metric does not give higher credit to a span that is perfectly detected than to one that has a very low overlap with the gold standard.

The precision and recall measures proposed here correct the problem with overlap-based measures: If the system proposes a span covering the whole sentence, the span coverage will be low and result in a low soft precision, and a low soft recall will be assigned if only a small part of a gold standard span is covered. Note that our measures are bounded below by the exact measures and above by the overlap-based measures.

7.1.1 Opinion Holders. To score the extraction of opinion holders, we started from the same basic idea: Assign a score based on intersection. The evaluation of this task is more complex, however, because (1) we only want to give credit for holders for correctly extracted opinion expressions; (2) the gold standard links opinion expressions to coreference chains rather than individual mentions of holders; and (3) the holder entity may be the writer or implicit (see Section 4.2).

We therefore used the following method: If the system has proposed an opinion expression e and its holder h, we first located the expression e' in the gold standard that most closely corresponds to e, that is $e' = \arg\max_x c(x, e)$, regardless of the span labels of e and e'. To assign a score to the proposed holder entity, we then selected the most closely corresponding gold standard holder entity h' in the coreference chain H' linked to e': $h' = \arg\max_{x \in H'} c(x, h)$. Finally, we computed the precision and recall scores using $c(h', h)$ and $c(h, h')$. We stress again that the gold standard coreference chains were used for evaluation purposes only, and that our system did not make use of them at test time.

If the system guesses that the holder of some opinion is the writer entity, we score it as perfectly detected (coverage 1) if the coreference chain H annotated in the gold standard contains the writer, and a full error (coverage 0) otherwise, and similar if h is implicit.

7.1.2 Polarity. In our experiments involving opinion expressions with polarities, we report precision and recall values for polarity-labeled opinion expression segmentation: In order to be assigned an intersection score above zero, a segment must be labeled with the correct polarity. In the gold standard, all polarity labels were changed as described in Section 4.3. In these evaluations, OSEs were ignored and DSEs and ESEs were not distinguished.

7.2 Experiments in Opinion Expression Extraction

The first task we considered was the extraction of opinion expression (labeled with expression types). We first studied the impact of the machine learning method and hypothesis set size on the reranker performance. Then, we carried out an analysis of the effectiveness of the features used by the reranker. We finally compared the performance of the expression extraction system with previous work (Breck, Choi, and Cardie 2007).

Table 2
Evaluation of reranking learning methods.

Learning method	*P*	*R*	*F*
Baseline	63.4 ± 1.5	46.8 ± 1.2	53.8 ± 1.1
Structured SVM	61.8 ± 1.5	52.5 ± 1.3	56.8 ± 1.1
Perceptron	62.8 ± 1.5	48.1 ± 1.3	54.5 ± 1.2
Passive–Aggressive	63.5 ± 1.5	51.8 ± 1.3	57.0 ± 1.1

Table 3
Oracle and reranker performance as a function of candidate set size.

	Reranked			Oracle		
k	*P*	*R*	*F*	*P*	*R*	*F*
1	63.36	46.77	53.82	63.36	46.77	53.82
2	63.70	48.17	54.86	72.66	55.18	62.72
4	63.57	49.78	55.84	79.12	62.24	69.68
8	63.50	51.79	57.05	83.72	68.14	75.13
16	63.00	52.94	57.54	86.92	72.79	79.23
32	62.15	54.50	58.07	89.18	76.76	82.51
64	61.03	55.67	58.23	91.09	80.19	85.29
128	60.22	56.45	58.27	92.63	83.00	87.55
256	59.87	57.22	58.51	94.01	85.27	89.43

7.2.1 Evaluation of Machine Learning Methods. We compared the machine learning methods described in Section 5. In these experiments, we used a hypothesis set size k of 8. All features from Section 6.2 were used. Table 2 shows the results of the evaluations using the precision and recall measures described earlier.[3] The baseline is the result of taking the top-scoring labeling from the base sequence labeler.

We note that the margin-based methods—structured SVM and the on-line PA algorithm—outperform the perceptron soundly, which shows the benefit of learning methods that make use of the cost function Δ. Comparing the two best-performing learning methods, we note that the reranker using the structured SVM is more recall-oriented whereas the PA-based reranker more precision-oriented; the difference in F-measure is not statistically significant. In the remainder of this article, all rerankers are trained using the PA learning algorithm (with the same parameters) because its training process is much faster than that of the structured SVM.

7.2.2 Candidate Set Size. In any method based on reranking, it is important to study the influence of the hypothesis set size on the quality of the reranked output. In addition, an interesting question is what the upper bound on reranker performance is—the *oracle* performance. Table 3 shows the result of an experiment that investigates these questions.

3 All confidence intervals in this article are at the 95% level and were estimated using a resampling method (Hjorth 1993). The significance tests for differences were carried out using permutation tests.

Table 4
Investigation of the contribution of syntactic features.

Feature set	*P*	*R*	*F*
Baseline	63.4 ± 1.5	46.8 ± 1.2	53.8 ± 1.1
All syntactic features	62.5 ± 1.4	53.2 ± 1.2	57.5 ± 1.1
Removed SYNTACTIC PATH	64.4 ± 1.5	48.7 ± 1.2	55.5 ± 1.1
Removed LEXICAL PATH	62.6 ± 1.4	53.2 ± 1.2	57.5 ± 1.1
Removed DOMINANCE	62.3 ± 1.5	52.9 ± 1.2	57.2 ± 1.1

As is common in reranking tasks, the reranker can exploit only a fraction of the potential improvement—the reduction of the F-measure error ranges between 10% and 15% of the oracle error reduction for all hypothesis set sizes.

The most visible effect of the reranker is that the recall is greatly improved. This does not seem to have an adverse effect on the precision, however, until the candidate set size goes above eight—in fact, the precision actually improves over the baseline for small candidate set sizes. After the size goes above eight, the recall (and the F-measure) still rises, but at the cost of decreased precision. In the remainder of this article, we used a k value of 64, which we thought gave a good balance between processing time and performance.

7.2.3 Feature Analysis. We studied the impact of syntactic and semantic structural features on the performance of the reranker. Table 4 shows the result of an investigation of the contribution of the syntactic features. Using all the syntactic features (and no semantic features) gives an F-measure roughly four points above the baseline. We then carried out an ablation test and measured the F-measure obtained when each one of the three syntactic features has been removed. It is clear that the unlexicalized syntactic path is the most important syntactic feature; this feature causes a two-point drop in F-measure when removed, which is clearly statistically significant ($p < 0.0001$). The effect of the two lexicalized features is smaller, with only DOMINANCE causing a significant ($p < 0.05$) drop when removed.

A similar result was obtained when studying the semantic features (Table 5). Removing the connecting label feature, which is unlexicalized, has a greater effect than removing the other two semantic features, which are lexicalized. Only the connecting label causes a statistically significant drop when removed ($p < 0.0001$).

Because our most effective structural features combine a pair of opinion expression labels with a tree fragment, it is interesting to study whether the expression labels alone would be enough. If this were the case, we could conclude that the improvement is

Table 5
Investigation of the contribution of semantic features.

Feature set	*P*	*R*	*F*
Baseline	63.4 ± 1.5	46.8 ± 1.2	53.8 ± 1.1
All semantic features	61.3 ± 1.4	53.8 ± 1.3	57.3 ± 1.1
Removed PREDICATE SENSE LABEL	61.3 ± 1.4	53.8 ± 1.3	57.3 ± 1.1
Removed PREDICATE+ARGUMENT LABEL	61.0 ± 1.4	53.6 ± 1.3	57.0 ± 1.1
Removed CONNECTING ARGUMENT LABEL	60.7 ± 1.4	50.5 ± 1.2	55.1 ± 1.1

Table 6
Structural features compared to label pairs.

Feature set	*P*	*R*	*F*
Baseline	63.4 ± 1.5	46.8 ± 1.2	53.8 ± 1.1
Label pairs	62.0 ± 1.5	52.7 ± 1.2	57.0 ± 1.1
All syntactic features	62.5 ± 1.4	53.2 ± 1.2	57.5 ± 1.1
All semantic features	61.3 ± 1.4	53.8 ± 1.3	57.3 ± 1.1
Syntactic + semantic	61.0 ± 1.4	55.7 ± 1.2	58.2 ± 1.1
Syntactic + semantic + label pairs	61.6 ± 1.4	54.8 ± 1.3	58.0 ± 1.1

caused not by the structural features, but just by learning which combinations of labels are common in the training set, such as that DSE+ESE would be more common than OSE+ESE. We thus carried out an experiment comparing a reranker using label pair features against rerankers based on syntactic features only, semantic features only, and the full feature set. Table 6 shows the results. We see that the reranker using label pairs indeed achieves a performance well above the baseline. Its performance is below that of any reranker using structural features, however. In addition, we see no improvement when adding label pair features to the structural feature set; this is to be expected because the label pair information is subsumed by the structural features.

7.2.4 Analysis of the Performance Depending on Expression Type. In order to better understand the performance details of the expression extraction, we analyzed how well it extracted the three different classes of expressions. Table 7 shows the results of this evaluation. The DSE row in the table thus shows the results of the performance on DSEs, without taking ESEs or OSEs into account.

Apart from evaluations of the three different types of expressions, we evaluated the performance for a number of combined classes that we think may be interesting for applications: DSE & ESE, finding all opinionated expressions and ignoring objective speech events; DSE & OSE, finding opinionated and non-opinionated speech and categorization events and ignoring expressive elements; and unlabeled evaluation of all types of MPQA expressions. The same extraction system was used in all experiments, and it was not retrained to maximize the different measures of performance.

Again, the strongest overall tendency is that the reranker boosts the recall. Going into the details, we see that the reranker gives very large improvements for DSEs and OSEs, but a smaller improvement for the combined DSE & OSE class. This shows that

Table 7
Performance depending on the type of expression.

	Baseline			Reranked		
	P	*R*	*F*	*P*	*R*	*F*
DSE	68.5 ± 2.1	57.8 ± 2.0	62.7 ± 1.7	67.0 ± 2.0	64.9 ± 1.9	66.0 ± 1.6
ESE	63.0 ± 2.1	36.9 ± 1.5	46.5 ± 1.4	58.2 ± 1.9	46.2 ± 1.6	51.5 ± 1.3
OSE	53.5 ± 3.5	62.3 ± 3.5	57.5 ± 3.0	57.0 ± 3.3	73.9 ± 3.2	64.3 ± 2.7
DSE & ESE	71.1 ± 1.6	48.1 ± 1.2	57.4 ± 1.1	68.0 ± 1.5	57.6 ± 1.3	62.4 ± 1.0
DSE & OSE	76.6 ± 1.8	70.3 ± 1.6	73.3 ± 1.3	72.6 ± 1.8	75.8 ± 1.5	74.2 ± 1.2
Unlabeled	75.6 ± 1.5	55.3 ± 1.2	63.9 ± 1.0	71.0 ± 1.4	63.7 ± 1.2	67.2 ± 1.0

one of the most clear benefits of the complex features is to help disambiguate these expressions. This also affects the performance for general opinionated expressions (DSE & ESE).

7.2.5 Comparison with Breck, Choi, and Cardie (2007). Comparison of systems in opinion expression detection is often nontrivial because evaluation settings have differed widely. Since our problem setting—marking up and labeling opinion expressions in the MPQA corpus—is most similar to that described by Breck, Choi, and Cardie (2007), we carried out an evaluation using the setting from their experiment.

For compatibility with their experimental set-up, this experiment differed from the ones described in the previous sections in the following ways:

- The results were measured using the overlap-based precision and recall, although this is problematic as pointed out in Section 7.1.
- The system did not need to distinguish DSEs and ESEs and did not have to detect the OSEs.
- Instead of the training/test split used in the previous evaluations, the systems were evaluated using a 10-fold cross-validation over the same set of 400 documents and the same cross-validation split as used in Breck, Choi, and Cardie's experiment. Each of the 10 rerankers was evaluated on one fold and trained on data generated in a cross-validation over the remaining nine folds.

Again, our reranker uses the PA learning method with the full feature set (Section 6.2) and a hypothesis set size k of 64. Table 8 shows the performance of our baseline (Section 4.1) and reranked system, along with the best results reported by Breck, Choi, and Cardie (2007).

We see that the performance of our system is clearly higher—in both precision and recall—than all results reported by Breck, Choi, and Cardie (2007). Note that our system was optimized for the intersection metric rather than the overlap metric and that we did not retrain it for this evaluation.

7.3 Opinion Holder Extraction

Table 9 shows the performance of our holder extraction systems, evaluated using the scoring method described in Section 7.1.1. We compared the performance of the reranker using opinion holder interaction features (Section 6.3) to two baselines: The first of them consisted of the opinion expression sequence labeler (ES, Section 4.1) and the holder extraction classifier (HC, Section 4.2), without modeling any interactions between opinions. The second and more challenging baseline was implemented by

Table 8
Results using the evaluation setting from Breck, Choi, and Cardie (2007).

System	P	R	F
Breck, Choi, and Cardie (2007)	71.64	74.70	73.05
Baseline	86.1 ± 1.0	66.7 ± 0.8	75.1 ± 0.7
Reranked	83.4 ± 1.0	75.0 ± 0.8	79.0 ± 0.6

Table 9
Opinion holder extraction results.

System	*P*	*R*	*F*
ES+HC	57.7 ± 1.7	45.3 ± 1.3	50.8 ± 1.3
ES+ER+HC	53.3 ± 1.5	52.0 ± 1.4	52.6 ± 1.3
ES+HC+EHR	53.2 ± 1.6	55.1 ± 1.5	54.2 ± 1.4

adding the opinion expression reranker (ER) without holder interaction features to the pipeline. This results in a large performance boost simply as a consequence of improved expression detection, because a correct expression is required to get credit for a holder. However, both baselines are outperformed by the reranker using holder interaction features, which we refer to as the expression/holder reranker (EHR); the differences to the strong baseline in recall and F-measure are both statistically significant ($p < 0.0001$).

We carried out an ablation test to gauge the impact of the two holder interaction features; we see in Table 10 that both of them contribute to improving the recall, and the effect on the precision is negligible. The statistical significance for the recall improvement is highest for SHARED HOLDERS ($p < 0.0001$) and lower for HOLDER TYPES + PATH ($p < 0.02$).

We omit a comparison with previous work in holder extraction because our formulation of the opinion holder extraction problem is different from those used in previous publications. Choi, Breck, and Cardie (2006) used the holders of a simplified set of opinion expressions, whereas Wiegand and Klakow (2010) extracted *every* entity tagged as "source" in MPQA regardless of whether it was connected to any opinion expression. Neither of them extracted *implicit* or *writer* holders.

Table 11 shows a detailed breakdown of the holder extraction results based on opinion expression type (DSE, OSE, and ESE), and whether the holder is internally or externally located; that is, whether or not the holder is textually realized in the same sentence as the opinion expression. In addition, Table 12 shows the performance for the two types of externally located holders.

As we noted in previous evaluations, the most obvious change between the baseline system and the reranker is that the recall and F-measure are improved; this is the case in every single evaluation. As previously, a large share of the improvement is explained simply by improved expression detection, which can be seen by comparing the reranked system to the strong baseline (ES+ER+HC). For the most important situations, however, we see improvement when using the reranker with holder interaction features. In those cases it outperforms the strong baseline significantly: DSE internal: $p < 0.001$, ESE internal $p < 0.001$, ESE external $p < 0.05$ (Table 11), writer $p < 0.05$ (Table 12).

Table 10
Opinion holder reranker feature ablation test.

Feature set	*P*	*R*	*F*
Both features	53.2 ± 1.6	55.1 ± 1.5	54.2 ± 1.4
Removed HOLDER TYPES + PATH	53.1 ± 1.6	54.6 ± 1.5	53.8 ± 1.3
Removed SHARED HOLDERS	53.1 ± 1.5	53.6 ± 1.5	53.3 ± 1.3

Table 11
Detailed opinion holder extraction results.

DSE	Internal			External		
	P	*R*	*F*	*P*	*R*	*F*
ES+HC	57.4 ± 2.4	48.9 ± 2.2	52.8 ± 1.9	32.3 ± 6.8	25.8 ± 5.8	28.7 ± 5.8
ES+ER+HC	56.7 ± 2.2	54.2 ± 2.2	55.5 ± 1.9	33.3 ± 5.9	34.2 ± 6.1	33.7 ± 5.5
ES+HC+EHR	55.6 ± 2.2	58.8 ± 2.3	57.2 ± 1.9	35.2 ± 6.2	32.1 ± 6.0	33.6 ± 5.6
OSE	Internal			External		
	P	*R*	*F*	*P*	*R*	*F*
ES+HC	46.2 ± 3.6	57.2 ± 3.9	51.1 ± 3.3	39.7 ± 12.0	35.2 ± 11.2	37.3 ± 10.5
ES+ER+HC	48.6 ± 3.4	66.8 ± 3.7	56.2 ± 3.1	36.8 ± 11.0	39.4 ± 11.4	38.1 ± 10.2
ES+HC+EHR	50.4 ± 3.6	65.9 ± 3.9	57.1 ± 3.2	35.9 ± 10.9	39.4 ± 11.4	37.6 ± 10.1
ESE	Internal			External		
	P	*R*	*F*	*P*	*R*	*F*
ES+HC	50.5 ± 4.7	19.2 ± 2.1	27.8 ± 2.7	45.1 ± 3.0	41.2 ± 2.5	43.0 ± 2.4
ES+ER+HC	48.3 ± 3.9	29.3 ± 2.8	36.4 ± 2.9	40.7 ± 2.6	48.4 ± 2.7	44.2 ± 2.3
ES+HC+EHR	40.4 ± 3.4	36.5 ± 3.2	39.8 ± 3.0	43.2 ± 2.8	47.7 ± 2.9	45.3 ± 2.4

The only common case where the improvement is not statistically significant is OSE internal.

The improvements are most notable for internally located holders, and especially for the ESEs. Extracting the opinion holder for ESEs is often complex because the expression and the holder are typically not directly connected on the syntactic or shallow-semantic level, as opposed to the typical situation for DSEs. When we use the reranker, however, the interaction features may help us make use of the holders of other opinion expressions in the same sentence; for instance, the interaction features make it easier to distinguish cases like "*the film was* [***awful***]$_{ESE}$" with an external (writer) holder from cases such as "*I* [***thought***]$_{DSE}$ *the film was* [***awful***]$_{ESE}$" with an internal holder directly connected to a DSE.

Table 12
Opinion holder extraction results for external holders.

Writer	*P*	*R*	*F*
ES+HC	44.8±3.0	42.8±2.6	43.8±2.4
ES+ER+HC	40.6±2.6	50.3±2.7	44.9±2.3
ES+HC+EHR	42.7±2.8	49.7±2.9	45.9±2.4
Implicit	*P*	*R*	*F*
ES+HC	41.2±6.4	28.3±4.8	33.6±4.9
ES+ER+HC	38.7±5.4	34.4±5.1	36.4±4.7
ES+HC+EHR	43.1±5.9	32.9±5.0	37.4±4.8

Table 13
Overall evaluation of polarity-labeled opinion expression extraction.

System	*P*	*R*	*F*
ES+PC	56.5 ± 1.7	38.4 ± 1.2	45.7 ± 1.2
ES+ER+PC	53.8 ± 1.6	44.5 ± 1.3	48.8 ± 1.2
ES+PC+EPR	54.7 ± 1.6	45.6 ± 1.3	49.7 ± 1.2

7.4 Polarity Classification

To evaluate the effect of the polarity-based reranker, we carried out experiments to compare it with two baseline systems similarly to the evaluations of holder extraction performance. Table 13 shows the precision, recall, and F-measures. The evaluation used the polarity-based intersection metric (Section 7.1.2). The first baseline consisted of an expression segmenter and a polarity classifier (ES+PC), and the second also included an expression reranker (ER). The reranker using polarity interaction features is referred to as the expression/polarity reranker (EPR).

The result shows that the polarity-based reranker gives a significant boost in recall, which is in line with our previous results that also mainly improved the recall. The precision shows a slight decrease from the ES+PC baseline but much lower than the recall improvement. The differences between the polarity reranker and the strongest baseline are all statistically significant (precision $p < 0.02$, recall and F-measure $p < 0.005$).

In addition, we evaluated the performance for individual polarity values. The figures are shown in Table 14. We see that the differences in performance when adding the polarity reranker are concentrated to the more frequent polarity values (NEUTRAL and NEGATIVE).

Table 14
Intersection-based evaluation for individual polarity values.

POSITIVE	*P*	*R*	*F*
ES+PC	53.5 ± 3.7	37.3 ± 3.0	43.9 ± 2.8
ES+ER+PC	50.5 ± 3.4	41.8 ± 3.0	45.8 ± 2.6
ES+PC+EPR	51.0 ± 3.5	41.6 ± 3.1	45.8 ± 2.7

NEUTRAL	*P*	*R*	*F*
ES+PC	56.4 ± 2.3	37.8 ± 1.7	45.3 ± 1.7
ES+ER+PC	54.0 ± 2.1	45.2 ± 1.8	49.2 ± 1.6
ES+PC+EPR	55.8 ± 2.1	46.1 ± 1.8	50.5 ± 1.6

NEGATIVE	*P*	*R*	*F*
ES+PC	58.4 ± 2.8	40.1 ± 2.4	47.6 ± 2.2
ES+ER+PC	55.5 ± 2.7	45.0 ± 2.3	49.7 ± 2.0
ES+PC+EPR	54.9 ± 2.7	47.0 ± 2.4	50.6 ± 2.0

Table 15
Overlap-based evaluation for individual polarity values, and comparison with the results reported by Choi and Cardie (2010).

POSITIVE	P	R	F
ES+PC	59.4 ± 2.6	46.1 ± 2.1	51.9 ± 2.0
ES+ER+PC	53.1 ± 2.3	50.9 ± 2.2	52.0 ± 1.9
ES+PC+EPR	58.2 ± 2.5	49.3 ± 2.2	53.4 ± 2.0
ES+PC+EPR$_p$	63.6 ± 2.8	44.9 ± 2.2	52.7 ± 2.1
Choi and Cardie (2010)	67.1	31.8	43.1

NEUTRAL	P	R	F
ES+PC	60.9 ± 1.4	49.2 ± 1.2	54.5 ± 1.0
ES+ER+PC	55.1 ± 1.2	57.7 ± 1.2	56.4 ± 1.0
ES+PC+EPR	60.3 ± 1.3	55.8 ± 1.2	58.0 ± 1.1
ES+PC+EPR$_p$	68.3 ± 1.5	48.2 ± 1.2	56.5 ± 1.2
Choi and Cardie (2010)	66.6	31.9	43.1

NEGATIVE	P	R	F
ES+PC	72.1 ± 1.8	52.0 ± 1.5	60.4 ± 1.4
ES+ER+PC	65.4 ± 1.7	58.2 ± 1.4	61.6 ± 1.3
ES+PC+EPR	67.6 ± 1.7	59.9 ± 1.5	63.5 ± 1.3
ES+PC+EPR$_p$	75.4 ± 2.0	55.0 ± 1.5	63.6 ± 1.4
Choi and Cardie (2010)	76.2	40.4	52.8

Finally, we carried out an evaluation in the setting[4] of Choi and Cardie (2010) and the figures are shown in Table 15. The table shows our baseline and integrated systems along with the figures[5] from Choi and Cardie. Instead of a single value for all polarities, we show the performance for every individual polarity value (POSITIVE, NEUTRAL, NEGATIVE). This evaluation uses the overlap metric instead of the intersection-based one. As we have pointed out, we use the overlap metric for compatibility although it is problematic.

As can be seen from the table, the system by Choi and Cardie (2010) shows a large precision bias despite being optimized with respect to the recall-promoting overlap metric. In recall and F-measure, their system is significantly outperformed for all polarity values by our baseline consisting of a pipeline of opinion expression extraction and polarity classifier. In addition, our joint model clearly outperforms the pipeline. The precision is slightly lower overall, but this is offset by large boosts in recall in all cases.

In order to rule out the hypothesis that our F-measure improvement compared with the Choi and Cardie system could be caused just by rebalancing precision and recall, we additionally trained a precision-biased reranker EPR$_p$ by changing the loss function Δ (see Section 5.3) from $1 - F_i$ to $1 - \frac{1}{3}F_i - \frac{2}{3}P_o$, where F_i is the intersection F-measure and P_o the overlap precision. When we use this reranker, we achieve almost the same levels of precision as reported by Choi and Cardie, even outperforming their precision for the

4 In addition to polarity, their system also assigned opinion intensity, which we do not consider here.
5 Confidence intervals for Choi and Cardie (2010) are omitted because we had no access to their output.

NEUTRAL polarity value, while the recall values are still massively higher. The precision bias causes slight drops in F-measure for the POSITIVE and NEUTRAL polarities.

7.5 First Extrinsic Evaluation: Extraction of Evaluations of Product Attributes

As an extrinsic evaluation of the opinion expression extraction system, we evaluated the impact of the expressions on a practical application: extraction of evaluations of attributes from product reviews. We first describe the collection we used and then the implementation of the extractor.

We used the annotated data set by Hu and Liu (2004a, 2004b)[6] for the experiments in extraction of attribute evaluations from product reviews. The collection contains reviews of five products: one DVD player, two cameras, one MP3 player, and one cellular phone. In this data set, every sentence is associated with a set of attribute evaluations. An evaluation consists of an attribute name and an evaluation value between -3 and +3, where -3 means a strongly negative evaluation and +3 strongly positive. For instance, the sentence *this player boasts a decent size and weight, a relatively-intuitive navigational system that categorizes based on id3 tags, and excellent sound* is tagged with the attribute evaluations `size +2`, `weight +2`, `navigational system +2`, `sound +2`. In this work, we do not make use of the exact value of the evaluation but only its sign. We removed the product attribute mentions in the form of anaphoric pronouns referring to entities mentioned in previous sentences; these cases are directly marked in the data set.

7.5.1 Implementation. We considered two problems: (1) extraction of attribute evaluations without taking the polarity into account, and (2) extraction with polarity (positive or negative). The former is modeled as a binary classifier that tags each word in the review (except the punctuation) as an evaluation or not, and the latter requires the definition of a three-class polarity classifier. For both tasks, we compared three feature sets: a baseline using simple features, a stronger baseline using a lexicon, and finally a system using features derived from opinion expressions.

Similarly to the opinion expression polarity classifier, we implemented the classifiers as SVMs that we trained using LIBLINEAR. For the extraction task without polarities, the best results were obtained using an L2-regularized L2-loss SVM and a C value of 0.1. For the polarity task, we used a multiclass SVM (Crammer and Singer 2001) with the same parameters. To handle the precision/recall tradeoff, we varied the class weighting for the null class.

The baseline classifier used features based on lexical information (word, POS tag, and lemma) in a window of size 3 around the word under consideration (the **focus word**). In addition, it had two features representing the overall sentence polarities. To compute the polarities, we trained bag-of-words classifiers following the implementation by Pang, Lee, and Vaithyanathan (2002). Two separate classifiers were used: one for positive and one for negative polarity. Note that these classifiers detect the *presence* of positive or negative polarity, which may thus occur in the same sentence. The classifiers were trained on the MPQA corpus, where we counted a sentence as positive if it contained a positive opinion expression with an intensity of at least MEDIUM, and conversely for the negative polarity.

6 `http://www.cs.uic.edu/~liub/FBS/CustomerReviewData.zip`.

7.5.2 Features Using a Sentiment Lexicon. Many previous implementations for several opinion-related tasks make use of *sentiment lexicons*, so the stronger baseline system used features based on the subjectivity lexicon by Wilson, Wiebe, and Hoffmann (2005), which we previously used for opinion expression segmentation in Section 4.1 and for polarity classification in Section 4.3. We created a classifier using a number of features based on this lexicon.

These features make use of the syntactic and semantic structure of the sentence. In the following examples, we use the sentence *The software itself was not so easy to use*, presented in Figure 3. In this sentence, consider the focus word *software*. One word is listed in the lexicon as associated with positive sentiment: *easy*. The system then extracts the following features:

Sentiment lexicon polarities. For every word in the sentence that is listed in the lexicon, we add a feature. Given the example sentence, we will thus add a feature `lex_pol:positive` because of the word *easy*, which is listed as positive in the sentiment lexicon.

Closest previous and following sentiment word. If there are sentiment words before or after the focus word, we add the closest of them to the feature vector. In this case, there is no previous sentiment word, so we only extract `following_word:easy`.

Syntactic paths to sentiment words. For every sentiment word in the sentence, we extract a syntactic path similar to our previous feature sets. This represents a syntactic pattern describing the relation between the sentiment word and the focus words. For instance, in the example we extract the path `SBJ↑PRD↓`, representing a copula construction: The word *software* is connected to the sentiment word *easy* first up through a subject link and then down through a predicative complement link.

Semantic links to sentiment words. When there is a direct semantic role link between a sentiment word and the focus word, we add a feature for the semantic role label. No such features are extracted in the example sentence. The focus word is an argument but no sentiment word is also a predicate.

7.5.3 Extended Feature Set Based on MPQA Opinion Expressions. We finally created an extended feature set incorporating the following features derived from MPQA-style opinion expressions, which we extracted automatically. The features are similar in construction to those extracted by means of the sentiment lexicon. The following list describes the new features exemplified with the same sentence above, which contains a negative opinion expression *not so easy*.

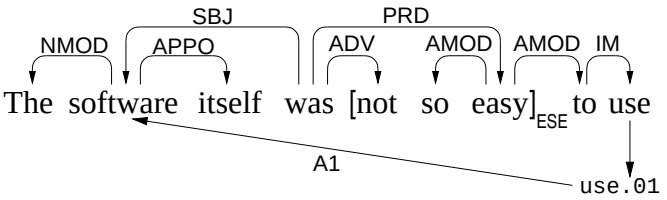

Figure 3
Example sentence for product feature evaluation extraction.

Table 16
Product attribute evaluation extraction performance.

Feature representation	Unlabeled	Polarity-labeled
Baseline	49.8 ± 2.0	39.6 ± 2.0
Lexicon	53.8 ± 2.0	46.2 ± 2.0
Opinion expressions	54.8 ± 2.0	49.0 ± 2.0

OPINION EXPRESSION POLARITIES. For every opinion expression extracted by the automatic system, we add a feature representing the polarity of the expression. In the example, we get `op_expr:negative`.

CLOSEST PREVIOUS AND FOLLOWING OPINION EXPRESSION WORD. We extract features for the closest words before and after the focus word that are contained in some opinion expression. In the example, there is an expression *not so easy* after the focus word *software*, so we get a single feature `following_expr:not`.

SYNTACTIC PATHS TO OPINION EXPRESSIONS. For every opinion expression in the sentence, we extract a path from the expression to the focus word. Because opinion expressions frequently consist of more than one word, we use the *shortest path*. In this case, we will thus again get `SBJ`↑`PRD`↓.

SEMANTIC LINKS TO OPINION EXPRESSIONS. Finally, we extracted features in case there were semantic role links. Again, we get no features based on the semantic role structure in the example since the opinion expression contains no predicate or argument.

7.5.4 Results. We evaluated the performance of the product attribute evaluation extraction using a 10-fold cross-validation procedure on the whole data set. We evaluated three classifiers: a baseline that did not use the lexicon or the opinion expressions, a classifier that adds the lexicon-based features, and finally the classifier that adds the MPQA opinion expressions. The F-measures are shown in Table 16 for the extraction task, and Figure 4 shows the precision/recall plots. There are clear improvements when adding the lexicon features, but the highest performing system is the one that also used the opinion expression features. The difference between the two top-performing

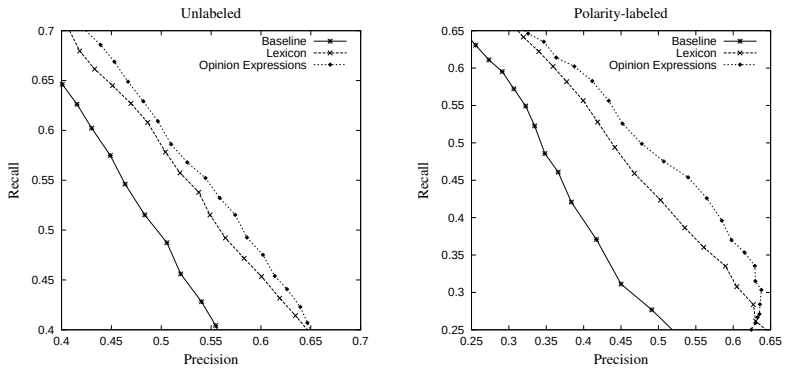

Figure 4
Precision / recall curves for extraction of product attribute evaluations.

classifiers is statistically significant ($p < 0.001$). For the extraction task where we also consider the polarities, the difference is even greater: almost three F-measure points.

7.6 Second Extrinsic Evaluation: Document Polarity Classification Experiment

In a second extrinsic evaluation of the opinion expression extractor, we investigated how expression-based features affect the performance of a document-level polarity classifier of reviews as positive or negative. We followed the same evaluation protocol as in the first extrinsic evaluation, where we compare three classifiers of increasing complexity: (1) a baseline using a pure word-based representation, (2) a stronger baseline adding features derived from a sentiment lexicon, and (3) a classifier with features extracted from opinion expressions.

The task of categorizing a full document as positive or negative can be viewed as a document categorization task, and this has led to the application of standard text categorization techniques (Pang, Lee, and Vaithyanathan 2002). We followed this approach and implemented the document polarity classifier as a binary linear SVM; this learning method has a long tradition of successful application in text categorization (Joachims 2002).

For these experiments, we used six collections. The first one consisted of movie reviews written in English extracted from the Web by Pang and Lee (2004).[7] This data set is an extension of a smaller set (Pang, Lee, and Vaithyanathan 2002) that has been used in a large number of experiments. The remaining five sets consisted of product reviews gathered by Blitzer, Dredze, and Pereira (2007).[8] We used five of the largest subsets: reviews of DVDs, software, books, music, and cameras. In all six collections, 1,000 documents were labeled by humans as positive and 1,000 as negative.

Following Pang and Lee (2004), the documents were represented as bag-of-word feature vectors based on presence features for individual words. No weighting such as IDF was used. The vectors were normalized to unit length. Again, we trained the SVMs using LIBLINEAR, and the best results were obtained using an L2-regularized L2-loss version of the SVM with a C value of 1.

7.6.1 Features Based on the Subjectivity Lexicon. We used features based on the subjectivity lexicon by Wilson, Wiebe, and Hoffmann (2005) that we used for opinion expression segmentation in Section 4.1 and for polarity classification in Section 4.3. For every word whose lemma is listed in the lexicon, we added a feature consisting of the word and its prior polarity and intensity to the bag-of-words feature vector.

The feature examples are taken from the sentence *HRW has denounced the defenseless situation of these prisoners*, where *denounce* is listed in the lexicon as `strong/negative` and *prisoner* as `weak/negative`.

LEXICON POLARITY. `negative`.

LEXICON POLARITY AND INTENSITY. `strong/negative`, `weak/negative`.

LEXICON POLARITY AND WORD. `denounced/negative`, `prisoners/negative`.

7 `http://www.cs.cornell.edu/people/pabo/movie-review-data/`.
8 `http://www.cs.jhu.edu/~mdredze/datasets/sentiment/unprocessed.tar.gz`.

7.6.2 Features Extracted from Opinion Expressions. Finally, we created a feature set based on the opinion expressions with polarities. We give examples from the same sentence; here, *denounced* is a negative DSE and *defenseless situation* is a negative ESE.

EXPRESSION POLARITY. `negative`.

EXPRESSION POLARITY AND WORD. `negative/denounced`, `negative/defenseless`, `negative/situation`.

EXPRESSION TYPE AND WORD. `DSE/denounced`, `ESE/defenseless`, `ESE/situation`.

7.6.3 Evaluation Results. To evaluate the performance of the document polarity classifiers, we carried out a 10-fold cross-validation procedure for every review collection. We evaluated three classifiers: one using only bag-of-words features ("Baseline"); one using features extracted from the subjectivity lexicon ("Lexicon"); and finally one also using the expression-based features ("Expressions").

In order to abstract away from the tuning threshold, the performances were measured using AUC, the area under ROC curve. The AUC values are given in Table 17.

These evaluations show that the classifier adding features extracted from the opinion expressions significantly outperforms the classifier using only a bag-of-words feature representation and also that using the lexicon-based features. This demonstrates that the extraction and disambiguation of opinion expressions in their context is useful for a coarse-grained task such as document polarity classification. The differences in AUC values between the two best configurations are statistically significant ($p < 0.005$ for all six collections). In addition, we show the precision/recall plots in Figure 5; we see that for all six collections, the expression-based set-up outperforms the other two near the precision/recall breakeven point.

The collection where we can see the most significant difference is the movie review set. The main difference of this collection compared with the other collections is that its documents are larger: The average size of a document here is about four times larger than in the other collections. In addition, its reviews often contain large sections that are purely factual in nature, mainly plot descriptions. The opinion expression identification may be seen as a way to process the document to highlight the interesting parts on which the classifier should focus.

8. Conclusion

We have shown that features derived from grammatical and semantic role structure can be used to improve three fundamental tasks in fine-grained opinion analysis: the detection of opinionated expressions, the extraction of opinion holders, and finally the

Table 17
Document polarity classification evaluation (AUC values).

Feature set	Movie	DVD	Software	Books	Music	Cameras
Baseline	93.1 ± 1.0	85.1 ± 1.7	91.0 ± 1.3	85.7 ± 1.6	84.7 ± 1.7	91.9 ± 1.2
Lexicon	93.8 ± 1.0	86.6 ± 1.6	92.3 ± 1.2	87.4 ± 1.5	86.6 ± 1.5	92.9 ± 1.1
Expressions	94.7 ± 0.9	87.2 ± 1.5	92.9 ± 1.1	88.1 ± 1.5	87.5 ± 1.5	93.6 ± 1.1

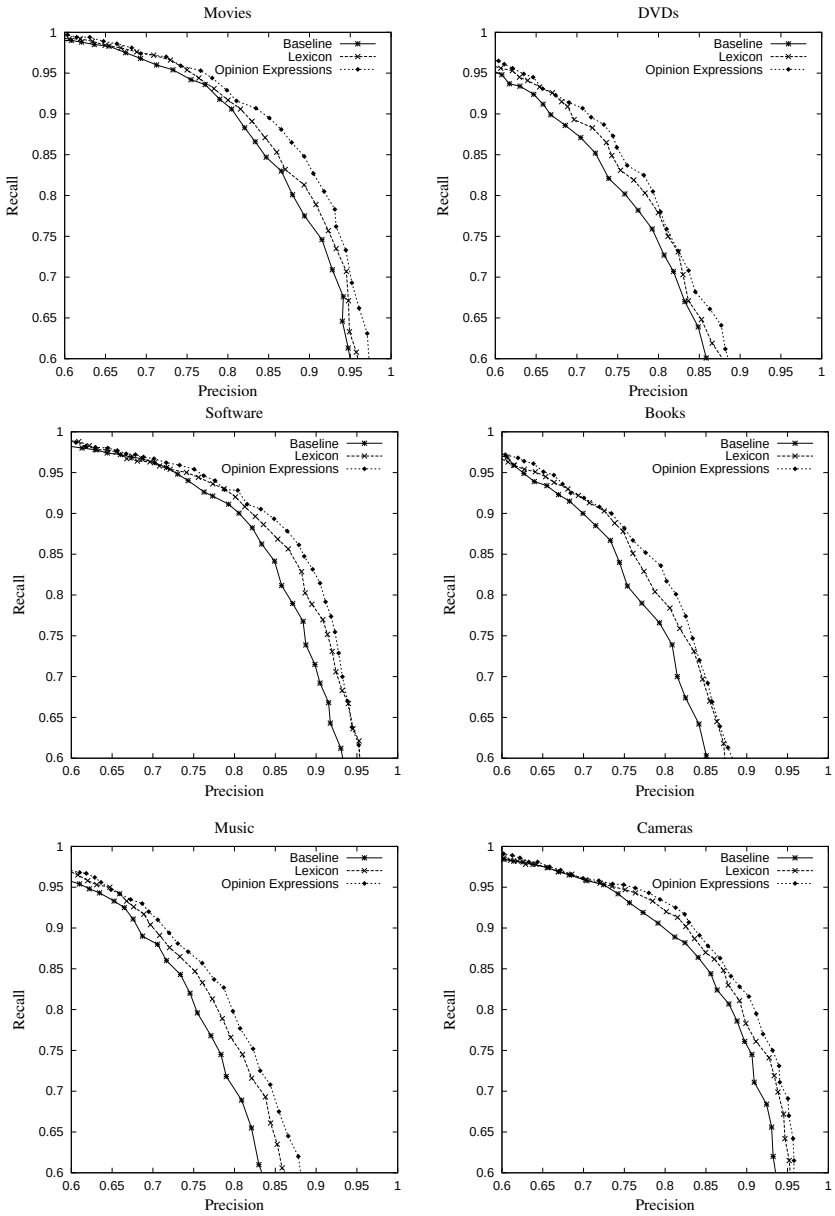

Figure 5
Precision / recall curves for detection of positive reviews.

assignment of polarity labels to opinion expressions. The main idea is to use relational features describing the interaction of opinion expressions through linguistic structures such as syntax and semantics. This is not only interesting from a practical point of view (improving performance) but also confirms our linguistic intuitions that surface-linguistic structure phenomena such as syntax and shallow semantics are used in the encoding of the rhetorical organization of the sentence, and that we can thus extract useful information from those structures.

Because our feature sets are based on interaction between opinion expressions that can appear anywhere in a sentence, exact inference in this model becomes intractable. To overcome this issue, we used an approximate search strategy based on reranking: In the first step, we used the baseline systems, which use only simple local features, to generate a relatively small hypothesis set; we then applied a classifier using interaction features to pick the final result. A common objection to reranking is that the candidate set may not be diverse enough to allow for much improvement unless it is very large; the candidates may be trivial variations that are all very similar to the top-scoring candidate. Investigating inference methods that take a less brute-force approach than plain reranking is thus another possible future direction. Interesting examples of such inference methods include forest reranking (Huang 2008) and loopy belief propagation (Smith and Eisner 2008). Nevertheless, although the development of such algorithms is a fascinating research problem, it will not necessarily result in a more usable system: Rerankers impose very few restrictions on feature expressivity and make it easy to trade accuracy for efficiency.

We investigated the effect of machine learning features, as well as other design parameters such as the choice of machine learning method and the size of the hypothesis set. For the features, we analyzed the impact of using syntax and semantics and saw that the best models are those making use of both. The most effective features we have found are purely structural, based on tree fragments in a syntactic or semantic tree. Features involving words generally did not seem to have the same impact. Sparsity may certainly be an issue for features defined in terms of tree fragments. Possible future extensions in this area could include bootstrapping methods to mine for meaningful fragments unseen in the training set, or methods to group such features into clusters to reduce the sparsity.

In addition to the core results on fine-grained opinion analysis, we have described experiments demonstrating that features extracted from opinion expressions can be used to improve practical applications: extraction of evaluations of product attributes, and document polarity classification. Although for the first task it may be fairly obvious that it is useful to carry out a fine-grained analysis of the sentence opinion structure, the second result is more unexpected because the document polarity classification task is a high-level and coarse-grained task. For both tasks, we saw statistically significant increases in performance compared not only to simple baselines, but also compared to strong baselines using a lexicon of sentiment words. Although the lexicon leads to clear improvements, the best classifiers also used the features extracted from the opinion expressions.

It is remarkable that the opinion expressions as defined by the MPQA corpus are useful for practical applications on reviews from several domains, because this corpus mainly consists of news documents related to political topics; this shows that the expression identifier has been able to generalize from the specific domains. It would still be relevant, however, to apply domain adaptation techniques (Blitzer, Dredze, and Pereira 2007). It could also be interesting to see how domain-specific opinion word lexicons could improve over the generic lexicon we used here; especially if such a lexicon were automatically constructed (Jijkoun, de Rijke, and Weerkamp 2010).

There are multiple additional opportunities for future work in this area. An important issue that we have left open is the coreference problem for holder extraction, which has been studied by Stoyanov and Cardie (2006). Similarly, recent work has tried to incorporate complex, high-level linguistic structure such as discourse representations (Asher, Benamara, and Mathieu 2009; Somasundaran et al. 2009; Zirn et al. 2011); it is clear that these structures are very relevant for explaining the way humans organize

their expressions of opinions rhetorically. Theoretical depth does not necessarily guarantee practical applicability, however, and the challenge is as usual to find a middle ground that balances our goals: explanatory power in theory, significant performance gains in practice, computational tractability, and robustness in difficult circumstances.

Acknowledgments

We would like to thank Eric Breck and Yejin Choi for clarifying their results and experimental set-up, and for sharing their cross-validation split. In addition, we are grateful to the anonymous reviewers, whose feedback has helped to improve the clarity and readability of this article. The research described here has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant 231126: LivingKnowledge—Facts, Opinions and Bias in Time, and under grant 247758: Trustworthy Eternal Systems via Evolving Software, Data and Knowledge (EternalS).

References

Asher, Nicholas, Farah Benamara, and Yannick Mathieu. 2009. Appraisal of opinion expressions in discourse. *Lingvisticae Investigations*, 31(2):279–292.

Bethard, Steven, Hong Yu, Ashley Thornton, Vasileios Hatzivassiloglou, and Dan Jurafsky. 2005. Extracting opinion propositions and opinion holders using syntactic and lexical cues. In James G. Shanahan, Yan Qu, and Janyce Wiebe, editors, *Computing Attitude and Affect in Text: Theory and Applications*. Springer, New York, chapter 11, pages 125–140.

Blitzer, John, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07)*, pages 440–447, Prague.

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

Breck, Eric, Yejin Choi, and Claire Cardie. 2007. Identifying expressions of opinion in context. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2,683–2,688, Hyderabad.

Choi, Yejin, Eric Breck, and Claire Cardie. 2006. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 431–439, Sydney.

Choi, Yejin and Claire Cardie. 2008. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 793–801, Honolulu, HI.

Choi, Yejin and Claire Cardie. 2010. Hierarchical sequential learning for extracting opinions and their attributes. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 269–274, Uppsala.

Collins, Michael. 2000. Discriminative reranking for natural language parsing. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 175–182, San Francisco, CA.

Collins, Michael. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 1–8, Philadelphia, PA.

Crammer, Koby, Ofer Dekel, Joseph Keshet, Shai Shalev-Schwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 2006(7):551–585.

Crammer, Koby and Yoram Singer. 2001. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2001(2):265–585.

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

Freund, Yoav and Robert E. Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296.

Gerber, Matthew and Joyce Chai. 2010. Beyond NomBank: A study of implicit

arguments for nominal predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1,583–1,592, Uppsala.
Greene, Stephan and Philip Resnik. 2009. More than words: Syntactic packaging and implicit sentiment. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 503–511, Boulder, CO.
Hjorth, J. S. Urban. 1993. *Computer Intensive Statistical Methods*. Chapman and Hall, London.
Hu, Minqing and Bing Liu. 2004a. Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-04)*, pages 168–177, Seattle, WA.
Hu, Minqing and Bing Liu. 2004b. Mining opinion features in customer reviews. In *Proceedings of the Nineteeth National Conference on Artificial Intellgience (AAAI-2004)*, pages 755–760, San Jose, CA.
Huang, Liang. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of ACL-08: HLT*, pages 586–594, Columbus, OH.
Jijkoun, Valentin, Maarten de Rijke, and Wouter Weerkamp. 2010. Generating focused topic-specific sentiment lexicons. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 585–594, Uppsala.
Joachims, Thorsten. 2002. *Learning to Classify Text using Support Vector Machines*. Kluwer/Springer, Boston.
Joachims, Thorsten, Thomas Finley, and Chun-Nam Yu. 2009. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1):27–59.
Johansson, Richard and Pierre Nugues. 2008. Dependency-based syntactic–semantic analysis with PropBank and NomBank. In *CoNLL 2008: Proceedings of the Twelfth Conference on Natural Language Learning*, pages 183–187, Manchester.
Joshi, Mahesh and Carolyn Penstein-Rosé. 2009. Generalizing dependency features for opinion mining. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 313–316, Singapore.
Karlgren, Jussi, Gunnar Eriksson, Magnus Sahlgren, and Oscar Täckström. 2010. Between bags and trees—Constructional patterns in text used for attitude identification. In *Proceedings of ECIR 2010, 32nd European Conference on Information Retrieval*, pages 38–49, Milton Keynes.
Kim, Soo-Min and Eduard Hovy. 2006. Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, pages 1–8, Sydney.
Kobayashi, Nozomi, Kentaro Inui, and Yuji Matsumoto. 2007. Extracting aspect-evaluation and aspect-of relations in opinion mining. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1,065–1,074, Prague.
Mel'čuk, Igor A. 1988. *Dependency Syntax: Theory and Practice*. State University Press of New York, Albany.
Meyers, Adam, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The NomBank project: An interim report. In *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31, Boston, MA.
Moschitti, Alessandro and Roberto Basili. 2004. Complex linguistic features for text classification: A comprehensive study. In *Proceedings of the 26th European Conference on Information Retrieval Research (ECIR 2004)*, pages 181–196, Sunderland.
Palmer, Martha, Dan Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105.
Pang, Bo and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 271–278, Barcelona.
Pang, Bo and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135.
Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 79–86, Philadelphia, PA.
Popescu, Ana-Maria and Oren Etzioni. 2005. Extracting product features and opinions

from reviews. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 339–346, Vancouver.

Prasad, Rashmi, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse Treebank 2.0. In *Proceedings of the 6th International Conference on Languages Resources and Evaluations (LREC 2008)*, pages 2,961–2,968, Marrakech.

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

Ruppenhofer, Josef, Swapna Somasundaran, and Janyce Wiebe. 2008. Finding the sources and targets of subjective expressions. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, pages 2,781–2,788, Marrakech.

Schwartz, Richard and Steve Austin. 1991. A comparison of several approximate algorithms for finding multiple (n-best) sentence hypotheses. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 701–704, Toronto.

Smith, David and Jason Eisner. 2008. Dependency parsing by belief propagation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 145–156, Honolulu, HI.

Somasundaran, Swapna, Galileo Namata, Janyce Wiebe, and Lise Getoor. 2009. Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 170–179, Singapore.

Stoyanov, Veselin and Claire Cardie. 2006. Partially supervised coreference resolution for opinion summarization through structured rule learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 336–344, Sydney.

Stoyanov, Veselin and Claire Cardie. 2008. Annotating topics of opinions. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, pages 3,213–3,217, Marrakech.

Surdeanu, Mihai, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of CoNLL 2008*, pages 159–177, Manchester.

Taskar, Ben, Carlos Guestrin, and Daphne Koller. 2004. Max-margin Markov networks. In *Advances in Neural Information Processing Systems 16*, pages 25–32, Cambridge, MA, MIT Press.

Titov, Ivan and Ryan McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL-08: HLT*, pages 308–316, Columbus, OH.

Tjong Kim Sang, Erik F., and Jorn Veenstra. 1999. Representing text chunks. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 173–179, Bergen.

Tsochantaridis, Ioannis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(Sep):1453–1484.

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*, pages 246–253, College Park, MD.

Wiebe, Janyce, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.

Wiegand, Michael and Dietrich Klakow. 2010. Convolution kernels for opinion holder extraction. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 795–803, Los Angeles, CA.

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

Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing contextual polarity: An exploration of features

for phrase-level sentiment analysis. *Computational Linguistics*, 35(3):399–433.

Wu, Yuanbin, Qi Zhang, Xuangjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1,533–1,541, Singapore.

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.

Zirn, Cäcilia, Mathias Niepert, Heiner Stuckenschmidt, and Michael Strube. 2011. Fine-grained sentiment analysis with structural features. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 336–344, Chiang Mai.