Syntactic and Semantic Structure for Opinion Expression Detection

Richard Johansson and **Alessandro Moschitti**
DISI, University of Trento
Via Sommarive 14 Povo, 38123 Trento (TN), Italy
{johansson, moschitti}@disi.unitn.it

Abstract

We demonstrate that *relational features* derived from dependency-syntactic and semantic role structures are useful for the task of detecting opinionated expressions in natural-language text, significantly improving over conventional models based on sequence labeling with local features. These features allow us to model the way opinionated expressions *interact* in a sentence over arbitrary distances.

While the relational features make the prediction task more computationally expensive, we show that it can be tackled effectively by using a reranker. We evaluate a number of machine learning approaches for the reranker, and the best model results in a 10-point absolute improvement in soft recall on the MPQA corpus, while decreasing precision only slightly.

1 Introduction

The automatic detection and analysis of opinionated text – *subjectivity analysis* – is potentially useful for a number of natural language processing tasks. Examples include retrieval systems answering queries about how a particular person feels about a product or political question, and various types of market analysis tools such as review mining systems.

A primary task in subjectivity analysis is to mark up the *opinionated expressions*, i.e. the text snippets signaling the subjective content of the text. This is necessary for further analysis, such as the determination of opinion holder and the polarity of the opinion. The MPQA corpus (Wiebe et al., 2005), a widely used corpus annotated with subjectivity information, defines two types of subjective expressions: *direct subjective expressions* (DSEs), which are explicit mentions of opinion, and *expressive subjective elements* (ESEs), which signal the attitude of the speaker by the choice of words. DSEs are often verbs of statement and categorization, where the opinion and its holder tend to be direct semantic arguments of the verb. ESEs, on the other hand, are less easy to categorize syntactically; prototypical examples would include value-expressing adjectives such as *beautiful*, *biased*, etc. In addition to DSEs and ESEs, the MPQA corpus also contains annotation for non-subjective statements, which are referred to as *objective speech events* (OSEs). Examples (1) and (2) show two sentences from the MPQA corpus where DSEs and ESEs have been manually annotated.

(1) For instance, he [denounced]$_{DSE}$ as a [human rights violation]$_{ESE}$ the banning and seizure of satellite dishes in Iran.
(2) This [is viewed]$_{DSE}$ as the [main impediment]$_{ESE}$ to the establishment of political order in the country .

The task of marking up these expressions has usually been approached using straightforward sequence labeling techniques using simple features in a small contextual window (Choi et al., 2006; Breck et al., 2007). However, due to the simplicity of the feature sets, this approach fails to take into account the fact that the semantic and pragmatic interpretation of sentences is not only determined by words but also by syntactic and shallow-semantic *relations*. Crucially, taking grammatical relations into account allows us to model how expressions *interact* in various ways that influence their interpretation as subjective or not. Consider, for instance, the word *said* in examples (3) and (4) below, where the interpretation as a DSE or an OSE is influenced by the subjective content of the enclosed statement.

Proceedings of the Fourteenth Conference on Computational Natural Language Learning, pages 67–76,
Uppsala, Sweden, 15-16 July 2010.

(3) "We will identify the [culprits]$_{ESE}$ of these clashes and [punish]$_{ESE}$ them," he [said]$_{DSE}$.
(4) On Monday, 80 Libyan soldiers disembarked from an Antonov transport plane carrying military equipment, an African diplomat [said]$_{OSE}$.

In this paper, we demonstrate how syntactic and semantic structural information can be used to improve opinion detection. While this feature model makes it impossible to use the standard sequence labeling method, we show that with a simple strategy based on reranking, incorporating structural features results in a significant improvement. We investigate two different reranking strategies: the Preference Kernel approach (Shen and Joshi, 2003) and an approach based on structure learning (Collins, 2002). In an evaluation on the MPQA corpus, the best system we evaluated, a structure learning-based reranker using the Passive–Aggressive learning algorithm, achieved a 10-point absolute improvement in soft recall, and a 5-point improvement in F-measure, over the baseline sequence labeler .

2 Motivation and Related Work

Most approaches to analysing the sentiment of natural-language text have relied fundamentally on purely lexical information (see (Pang et al., 2002; Yu and Hatzivassiloglou, 2003), *inter alia*) or low-level grammatical information such as part-of-speech tags and functional words (Wiebe et al., 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).

However, it has been suggested that subjectivity analysis is inherently more subtle than categorization and that *structural* linguistic information should therefore be given more attention in this context. For instance, Karlgren et al. (2010) argued from a Construction Grammar viewpoint (Croft, 2005) that *grammatical constructions* not only connect words, but can also be viewed as lexical items in their own right. Starting from this intuition, they showed that incorporating construction items into a bag-of-words feature representation resulted in improved results on a number of coarse-grained opinion analysis tasks. These constructional features were domain-independent and were manually extracted from dependency parse trees. They found that the most prominent constructional feature for subjectivity analysis was the Tense Shift construction.

While the position by Karlgren et al. (2010) – that constructional features *signal* opinion – originates from a particular theoretical framework and may be controversial, syntactic and shallow-semantic relations have repeatedly proven useful for subtasks of subjectivity analysis that are inherently *relational*, above all for determining the holder or topic of a given opinion. Works using syntactic features to extract topics and holders of opinions are numerous (Bethard et al., 2005; Kobayashi et al., 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 et al. (2006) successfully used a PropBank-based semantic 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 the same task. Ruppenhofer et al. (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. One such linguistic pheonomenon is the *discourse* structure, which has recently attracted some attention in the opinion analysis community (Somasundaran et al., 2009).

3 Opinion Expression Detection Using Syntactic and Semantic Structures

Previous systems for opinionated expression markup have typically used simple feature sets which have allowed the use of efficient off-the-shelf sequence labeling methods based on Viterbi search (Choi et al., 2006; Breck et al., 2007). This is not possible in our case since we would like to extract structural, relational features that involve *pairs* of opinionated expressions and may apply over an arbitrarily long distance in the sentence.

While it is possible that search algorithms for exact or approximate inference can be constructured for the $\arg\max$ problem in this model, we sidestepped this issue by using a *reranking* decomposition of the problem: We first apply a standard Viterbi-based sequence labeler using no structural features and generate a small candidate set of size k. Then, a second and more complex model picks

the top candidate from this set without having to search the whole candidate space.

The advantages of a reranking approach compared to 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, and features can be arbitrarily complex – we don't have to think about how the features affect the algorithmic complexity of the inference step. 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 (Huang, 2008).

3.1 Syntactic and Semantic Structures

We used the syntactic–semantic parser by Johansson and Nugues (2008a) to annnotate the sentences with dependency syntax (Mel'čuk, 1988) and shallow semantic structures in the PropBank (Palmer et al., 2005) and NomBank (Meyers et al., 2004) frameworks. Figure 1 shows an example of the annotation: The sentence *they called him a liar*, where *called* is a DSE and *liar* is an ESE, has been annotated with dependency syntax (above the text) and PropBank-based semantic role structure (below the text). The predicate *called*, which is an instance of the PropBank frame `call.01`, has three semantic arguments: the Agent (A0), the Theme (A1), and the Predicate (A2), which are realized on the surface-syntactic level as a subject, a direct object, and an object predicative complement, respectively.

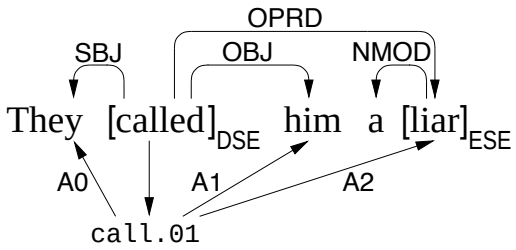

Figure 1: Syntactic and shallow semantic structure.

3.2 Sequence Labeler

We implemented a standard sequence labeler following the approach of Collins (2002), while training the model using the Passive–Aggressive algorithm (Crammer et al., 2006) instead of the perceptron. We encoded the opinionated expression brackets using the IOB2 encoding scheme (Tjong Kim Sang and Veenstra, 1999). Figure 2 shows an example of a sentence with a DSE and an ESE and how they are encoded in the IOB2 encoding.

```
This          O
is            O
viewed        B-DSE
as            O
the           O
main          B-ESE
impediment    I-ESE
```

Figure 2: Sequence labeling example.

The sequence labeler used word, POS tag, and lemma features in a window of size 3. In addition, we used prior polarity and intensity features derived from the lexicon created by Wilson et al. (2005). In the example, *viewed* is listed as having strong prior subjectivity but no polarity, and *impediment* has strong prior subjectivity and negative polarity. Note that prior subjectivity does not always imply subjectivity in a particular context; this is why contextual features are essential for this task.

This sequence labeler is used to generate the candidate set for the reranker; the Viterbi algorithm is easily modified to give k-best output. To generate training data for the reranker, we carried out a 5-fold cross-validation procedure: We split the training set into 5 pieces, trained a sequence labeler on pieces 1 to 4, applied it to piece 5 and so on.

3.3 Reranker 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 creates a small complication for multiword opinionated expressions; we select the shortest possible path in such cases. For instance, in Example (1), the path will be computed between *denounced* and *violation*, and in Example (2) between *viewed* and *impediment*.

We used the following syntactic features:

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. For instance, for the DSE *called* and the ESE *liar* in Figure 1, we represent the syntactic configuration using the feature `DSE:OPRD↓:ESE`, meaning that the path from the DSE to the ESE consists of a single link, where the dependency edge label is `OPRD` (object predicative complement).

LEXICALIZED PATH. Same as above, but with lexical information attached: `DSE/called:OPRD↓:ESE/liar`.

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 1, we extract the feature `DSE/called→ESE/liar`, meaning that a DSE with the word *called* dominates an ESE with the word *liar*.

The semantic features 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 *call* which is also a DSE is represented with the feature `DSE/call.01`.

PREDICATE AND ARGUMENT LABEL. For every argument of a predicate inside an opinion expression, we create a feature representing the predicate–argument pair: `DSE/call.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 *liar* is connected to the DSE *call* via an A2 label, and we represent this using a feature `DSE:A2:ESE`.

Apart from the syntactic and semantic features, we also used the score output from the base sequence labeler as a feature. We normalized the scores over the k candidates so that their exponentials summed to 1.

3.4 Preference Kernel Approach

The first reranking strategy we investigated was the Preference Kernel approach (Shen and Joshi, 2003). In this method, the reranking problem – learning to select the correct candidate h^1 from a candidate set $\{h^1, \ldots, h^k\}$ – is reduced to a binary classification problem by creating *pairs*: positive training instances $\langle h^1, h^2\rangle, \ldots, \langle h^1, h^k\rangle$ and negative instances $\langle h^2, h^1\rangle, \ldots, \langle h^k, h^1\rangle$. This approach has the advantage that the abundant tools for binary machine learning can be exploited.

It is also easy to show (Shen and Joshi, 2003) that if we have a kernel K over the candidate space T, we can construct a valid kernel P_K over the space of pairs $T \times T$ as follows:

$$\begin{aligned} P_K(h_1, h_2) &= K(h_1^1, h_2^1) + K(h_1^2, h_2^2) \\ &- K(h_1^1, h_2^2) - K(h_1^2, h_2^1), \end{aligned}$$

where h_i are the pairs of hypotheses $\langle h_i^1, h_i^2\rangle$ generated by the base model. This makes it possible to use kernel methods to train the reranker. We tried two types of kernels: linear kernels and tree kernels.

3.4.1 Linear Kernel

We created feature vectors extracted from the candidate sequences using the features described in Section 3.3. We then trained linear SVMs using the LIBLINEAR software (Fan et al., 2008), using L1 loss and L2 regularization.

3.4.2 Tree Kernel

Tree kernels have been successful for a number of structure extraction tasks, such as relation extraction (Zhang et al., 2006; Nguyen et al., 2009) and opinion holder extraction (Wiegand and Klakow, 2010). A tree kernel implicitly represents a large space of fragments extracted from trees and could thus reduce the need for manual feature design. Since the paths that we extract manually (Section 3.3) can be expressed as tree fragments, this method could be an interesting alternative to the manually extracted features used with the linear kernel.

We therefore implemented a reranker using the Partial Tree Kernel (Moschitti, 2006), and we trained it using the SVMLight-TK software[1], which is a modification of SVMLight (Joachims,

[1] Available at `http://dit.unitn.it/∼moschitt`

1999)[2]. It is still an open question how dependency trees should be represented for use with tree kernels (Suzuki et al., 2003; Nguyen et al., 2009); we used the representation shown in Figure 3. Note that we have concatenated the opinion expression labels to the POS tag nodes. We did not use any of the features from Section 3.3 except for the base sequence labeler score.

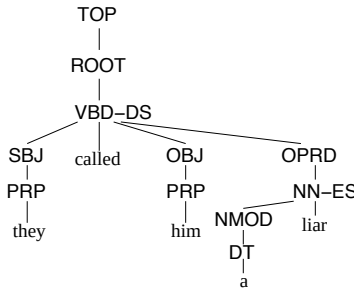

Figure 3: Representation of a dependency tree with opinion expressions for tree kernels.

3.5 Structure Learning Approach

The Preference Kernel approach reduces the reranking problem to a binary classification task on pairs, after which a standard SVM optimizer is used to train the reranker. A problem with this method is that the optimization problem solved by the SVM – maximizing the classification accuracy on a set of independent pairs – is not directly related to the performance of the reranker. Instead, the method employed by many rerankers following Collins and Duffy (2002) directly learn a scoring function that is trained to maximize performance on the reranking task. We will refer to this approach as the *structure learning* method.

While there are batch learning algorithms that work in this setting (Tsochantaridis et al., 2005), online learning methods have been more popular for efficiency reasons. We investigated two online learning algorithms: the popular *structured perceptron* Collins and Duffy (2002) and the Passive–Aggressive (PA) algorithm (Crammer et al., 2006). To increase robustness, we averaged the weight vectors seen during training as in the Voted Perceptron (Freund and Schapire, 1999).

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 update based on the difference between the correct output y and the predicted output $\hat{y}$, where Φ is the feature representation function:

$$\hat{y} \leftarrow \arg\max_h w \cdot \Phi(x, h)$$
$$w \leftarrow w + \Phi(x, y) - \Phi(x, \hat{y})$$

In the PA algorithm, which is based on the theory of large-margin learning, we instead find the $\hat{y}$ that violates the margin constraints maximally. The update step length τ is computed based on the margin; this update is bounded by a regularization constant C:

$$\hat{y} \leftarrow \arg\max_h w \cdot \Phi(x, h) + \sqrt{\rho(y, h)}$$
$$\tau \leftarrow \min\left(C, \frac{w(\Phi(x,\hat{y}) - \Phi(x,y)) + \sqrt{\rho(y,\hat{y})}}{\|\Phi(x,\hat{y}) - \Phi(x,y)\|^2}\right)$$
$$w \leftarrow w + \tau(\Phi(x, y) - \Phi(x, \hat{y}))$$

The algorithm uses a cost function ρ. We used the function $\rho(y, \hat{y}) = 1 - F(y, \hat{y})$, where F is the soft F-measure described in Section 4.1. With this approach, the learning algorithm thus directly optimizes the measure we are interested in, i.e. the F-measure.

4 Experiments

We carried out the experiments on version 2 of the MPQA corpus (Wiebe et al., 2005), which we split into a test set (150 documents, 3,743 sentences) and a training set (541 documents, 12,010 sentences).

4.1 Evaluation Metrics

Since expression boundaries are hard to define exactly in annotation guidelines (Wiebe et al., 2005), we used soft precision and recall measures to score the quality of the system output. To derive the soft precision and recall, we first define the *span coverage* c of a span s 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, the operator $|\cdot|$ counts tokens, and the intersection $\cap$ gives the set of tokens that two spans have in common. Since 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 $\boldsymbol{S'}$:

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

[2] `http://svmlight.joachims.org`

We now define the soft 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:

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

Note that the operator $|\cdot|$ counts spans in this formula.

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. However, since the boundaries of the spans annotated in the MPQA corpus are not strictly defined in the annotation guidelines (Wiebe et al., 2005), 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 et al., 2006; Breck et al., 2007). As pointed out by Breck et al. (2007), this is problematic since 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.

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. Note that our measures are bounded below by the exact measures and above by the overlap-based measures.

4.2 Reranking Approaches

We compared the reranking architectures and the machine learning methods described in Section 3. In these experiments, we used a candidate set size k of 8. Table 1 shows the results of the evaluations using the precision and recall measures described above. The baseline is the result of taking the top-scoring output from the sequence labeler without applying any reranking.

The results show that the rerankers using manual feature extraction outperform the tree-kernel-based reranker, which obtains a score just above the baseline. It should be noted that the massive training time of kernel-based machine learning precluded a detailed tuning of parameters and

System	P	R	F
Baseline	63.36	46.77	53.82
Pref-linear	64.60	50.17	56.48
Pref-TK	63.97	46.94	54.15
Struct-Perc	62.84	48.13	54.51
Struct-PA	63.50	51.79	57.04

Table 1: Evaluation of reranking architectures and learning methods.

representation – on the other hand, we did not need to spend much time on parameter tuning and feature design for the other rerankers.

In addition, we note that the best performance was obtained using the PA algorithm and the structure learning architecture. The PA algorithm is a simple online learning method and still outperforms the SVM used in the preference-kernel reranker. This suggests that the structure learning approach is superior for this task. It is possible that a batch learning method such as SVM$^{\text{struct}}$ (Tsochantaridis et al., 2005) could improve the results even further.

4.3 Candidate Set Size

In any method based on reranking, it is important to study the influence of the candidate 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 2 shows the result of an experiment that investigates these questions. We used the reranker based on the Passive–Aggressive method in this experiment since this reranker gave the best results in the previous experiment.

	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	57.04	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	61.02	55.67	58.22	91.08	80.19	85.28
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

Table 2: Oracle and reranker performance as a function of candidate set size.

As is common in reranking tasks, the reranker can exploit only a fraction of the potential improvement – the reduction of the F-measure error

is between 10 and 15 percent of the oracle error reduction for all candidate set sizes.

The most visible effect of the reranker is that the recall is greatly improved. However, this does not seem to have an adverse effect on the precision until the candidate set size goes above 8 – in fact, the precision actually improves over the baseline for small candidate set sizes. After the size goes above 8, the recall (and the F-measure) still rises, but at the cost of decreased precision.

4.4 Impact of Features

We studied the impact of syntactic and semantic structural features on the performance of the reranker. Table 3 shows the result of the investigation for syntactic features. Using all the syntactic features (and no semantic features) gives an F-measure roughly 4 points above the baseline, using the PA reranker with a k of 64. We then measured the F-measure obtained when each one of the three syntactic features had been removed. It is clear that the unlexicalized syntactic path is the most important syntactic feature; the effect of the two lexicalized features seems to be negligible.

System	P	R	F
Baseline	63.36	46.77	53.82
All syntactic	62.45	53.19	57.45
No SYN PATH	64.40	48.69	55.46
No LEX PATH	62.62	53.19	57.52
No DOMINANCE	62.32	52.92	57.24

Table 3: Effect of syntactic features.

A similar result was obtained when studying the semantic features (Table 4). Removing the CONNECTING ARGUMENT LABEL feature, which is unlexicalized, has a greater effect than removing the other two semantic features, which are lexicalized.

System	P	R	F
Baseline	63.36	46.77	53.82
All semantic	61.26	53.85	57.31
No PREDICATE SL	61.28	53.81	57.30
No PRED+ARGLBL	60.96	53.61	57.05
No CONN ARGLBL	60.73	50.47	55.12

Table 4: Effect of semantic features.

Since 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 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 5 shows the results. We see that the reranker using label pairs indeed achieves a performance well above the baseline. However, its performance is below that of any reranker using structural features. In addition, we see no improvement when adding label pair features to the structural feature set; this is to be expected since the label pair information is subsumed by the structural features.

System	P	R	F
Baseline	63.36	46.77	53.82
Label pairs	62.05	52.68	56.98
All syntactic	62.45	53.19	57.45
All semantic	61.26	53.85	57.31
Syn + sem	61.02	55.67	58.22
Syn + sem + pairs	61.61	54.78	57.99

Table 5: Structural features compared to label pairs.

4.5 Comparison with Breck et al. (2007)

Comparison of systems in opinion expression detection is often nontrivial since 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 et al. (2007), we carried out an evaluation using the setting used in their experiment.

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

- The system did not need to distinguish DSEs and ESEs and did not have to detect the OSEs.
- The results were measured using the overlap-based precision and recall, although this is problematic as pointed out in Section 4.1.

- Instead of the training/test split we used in the previous evaluations, the systems were evaluated using a 10-fold cross-validation over the same set of 400 documents as used in Breck's experiment.

Again, our reranker uses the PA method with a k of 64. Table 6 shows the results.

System	P	R	F
Breck et al. (2007)	71.64	74.70	73.05
Baseline	80.85	64.38	71.68
Reranked	76.40	78.23	77.30

Table 6: Results using the Breck et al. (2007) evaluation setting.

We see that the performance of our system is clearly higher – in both precision and recall – than that reported by Breck et al. (2007). This shows again that the structural features are effective for the task of finding opinionated expressions.

We note that the performance of our baseline sequence labeler is lower than theirs; this is to be expected since they used a more complex batch learning algorithm (conditional random fields) while we used an online learner, and they spent more effort on feature design. This indicates that we should be able to achieve even higher performance using a stronger base model.

5 Conclusion

We have shown that features derived from grammatical and semantic role structure can be used to improve the detection of opinionated expressions in subjectivity analysis. Most significantly, the recall is drastically increased (10 points) while the precision decreases only slightly (3 points). This result compares favorably with previously published results, which have been biased towards precision and scored low on recall.

The long-distance structural features gives us a model that has predictive power as well as being of theoretical interest: this model takes into account the *interactions* between opinion expressions in a sentence. While these structural features give us a powerful model, they come at a computational cost; prediction is more complex than in a standard sequence labeler based on purely local features. However, we have shown that a prediction strategy based on reranking suffices for this task.

We analyzed the impact of the syntactic and semantic features and saw that the best model includes both types of features. The most effective features we have found are purely structural, i.e. based on tree fragments in a syntactic or semantic tree. Features involving words did not seem to have the same impact. We also showed that the improvement is not explainable by mere correlations between opinion expression labels.

We investigated a number of implementation strategies for the reranker and concluded that the structural learning framework seemed to give the best performance. We were not able to achieve the same performance using tree kernels as with manually extracted features. It is possible that this could be improved with a better strategy for representing dependency structure for tree kernels, or if the tree kernels could be incorporated into the structural learning framework.

The flexible architecture we have presented enables interesting future research: (i) a straightforward improvement is the use of lexical similarity to reduce data sparseness, e.g. (Basili et al., 2005; Basili et al., 2006; Bloehdorn et al., 2006). However, the similarity between subjective words, which have multiple senses against other words may negatively impact the system accuracy. Therefore, the use of the syntactic/semantic kernels, i.e. (Bloehdorn and Moschitti, 2007a; Bloehdorn and Moschitti, 2007b), to syntactically contextualize word similarities may improve the reranker accuracy. (ii) The latter can be further boosted by studying complex structural kernels, e.g. (Moschitti, 2008; Nguyen et al., 2009; Dinarelli et al., 2009). (iii) More specific predicate argument structures such those proposed in FrameNet, e.g. (Baker et al., 1998; Giuglea and Moschitti, 2004; Giuglea and Moschitti, 2006; Johansson and Nugues, 2008b) may be useful to characterize the opinion holder and the sentence semantic context.

Finally, while the strategy based on reranking resulted in a significant performance boost, it remains to be seen whether a higher accuracy can be achieved by developing a more sophisticated inference algorithm based on dynamic programming. However, while the development of such an algorithm is an interesting problem, it will not necessarily result in a more usable system – when using a reranker, it is easy to trade accuracy for efficiency.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement 231126: LivingKnowledge – Facts, Opinions and Bias in Time, and from Trustworthy Eternal Systems via Evolving Software, Data and Knowledge (EternalS, project number FP7 247758). In addition, we would like to thank Eric Breck for clarifying his results and experimental setup.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of COLING/ACL-1998*.

Roberto Basili, Marco Cammisa, and Alessandro Moschitti. 2005. Effective use of WordNet semantics via kernel-based learning. In *Proceedings of CoNLL-2005*, pages 1–8, Ann Arbor, Michigan.

Roberto Basili, Marco Cammisa, and Alessandro Moschitti. 2006. A semantic kernel to classify texts with very few training examples. In *in Informatica, an international journal of Computing and Informatics*.

Steven Bethard, 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*.

Stephan Bloehdorn and Alessandro Moschitti. 2007a. Combined syntactic and semantic kernels for text classification. In *Proceedings of ECIR 2007*, Rome, Italy.

Stephan Bloehdorn and Alessandro Moschitti. 2007b. Structure and semantics for expressive text kernels. In *In Proceedings of CIKM '07*.

Stephan Bloehdorn, Roberto Basili, Marco Cammisa, and Alessandro Moschitti. 2006. Semantic kernels for text classification based on topological measures of feature similarity. In *Proceedings of ICDM 06*, Hong Kong, 2006.

Eric Breck, Yejin Choi, and Claire Cardie. 2007. Identifying expressions of opinion in context. In *Proceedings of IJCAI-2007*, Hyderabad, India.

Yejin Choi, Eric Breck, and Claire Cardie. 2006. Joint extraction of entities and relations for opinion recognition. In *Proceedings of EMNLP 2006*.

Michael Collins and Nigel Duffy. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of ACL'02*.

Michael Collins. 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.

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

William Croft. 2005. Radical and typological arguments for radical construction grammar. In J.-O. Östman and M. Fried, editors, *Construction Grammars: Cognitive grounding and theoretical extensions*.

Marco Dinarelli, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Re-ranking models based-on small training data for spoken language understanding. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1076–1085, Singapore, August.

Rong-En Fan, 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.

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

Ana-Maria Giuglea and Alessandro Moschitti. 2004. Knowledge Discovering using FrameNet, VerbNet and PropBank. In *In Proceedings of the Workshop on Ontology and Knowledge Discovering at ECML 2004*, Pisa, Italy.

Ana-Maria Giuglea and Alessandro Moschitti. 2006. Semantic role labeling via FrameNet, VerbNet and PropBank. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 929–936, Sydney, Australia, July.

Liang Huang. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of ACL-08: HLT*, pages 586–594, Columbus, United States.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. *Advances in Kernel Methods – Support Vector Learning*, 13.

Richard Johansson and Pierre Nugues. 2008a. 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, United Kingdom.

Richard Johansson and Pierre Nugues. 2008b. The effect of syntactic representation on semantic role labeling. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 393–400, Manchester, UK.

Mahesh Joshi and Carolyn Penstein-Rosé. 2009. Generalizing dependency features for opinion mining. In *Proceedings of ACL/IJCNLP 2009, Short Papers Track*.

Jussi Karlgren, 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*, Milton Keynes, United Kingdom.

Soo-Min Kim and Eduard Hovy. 2006. Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of ACL/COLING Workshop on Sentiment and Subjectivity in Text*.

Nozomi Kobayashi, Kentaro Inui, and Yuji Matsumoto. 2007. Extracting aspect-evaluation and aspect-of relations in opinion mining. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-CoNLL-2007)*.

Igor A. Mel'čuk. 1988. *Dependency Syntax: Theory and Practice*. State University Press of New York, Albany.

Adam Meyers, 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, United States.

Alessandro Moschitti and Roberto Basili. 2004. Complex linguistic features for text classification: A comprehensive study. In *Proceedings of ECIR*.

Alessandro Moschitti. 2006. Making tree kernels practical for natural language learning. In *Proccedings of EACL'06*.

Alessandro Moschitti. 2008. Kernel methods, syntax and semantics for relational text categorization. In *Proceeding of CIKM '08*, NY, USA.

Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *Proceedings of EMNLP*.

Martha Palmer, Dan Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of EMNLP*.

Josef Ruppenhofer, Swapna Somasundaran, and Janyce Wiebe. 2008. Finding the sources and targets of subjective expressions. In *Proceedings of LREC*.

Libin Shen and Aravind Joshi. 2003. An SVM based voting algorithm with application to parse reranking. In *Proceedings of the CoNLL*.

Swapna Somasundaran, Galileo Namata, Janyce Wiebe, and Lise Getoor. 2009. Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In *Proceedings of EMNLP 2009: conference on Empirical Methods in Natural Language Processing*.

Jun Suzuki, Tsutomu Hirao, Yutaka Sasaki, and Eisaku Maeda. 2003. Hierarchical directed acyclic graph kernel: Methods for structured natural language data. In *Proceedings of the 41th Annual Meeting of Association for Computational Linguistics (ACL)*.

Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. In *Proceedings of EACL99*, pages 173–179, Bergen, Norway.

Iannis Tsochantaridis, 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.

Janyce Wiebe, 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*.

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

Michael Wiegand and Dietrich Klakow. 2010. Convolution kernels for opinion holder extraction. In *Proceedings of HLT-NAACL 2010*. To appear.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of HLT/EMNLP 2005*.

Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *Proceedings of EMNLP*.

Hong Yu 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, Japan.

Min Zhang, Jie Zhang, and Jian Su. 2006. Exploring Syntactic Features for Relation Extraction using a Convolution tree kernel. In *Proceedings of NAACL*.