# Reranking Models in Fine-grained Opinion Analysis

**Richard Johansson** and **Alessandro Moschitti**
University of Trento
{johansson, moschitti}@disi.unitn.it

## Abstract

We describe the implementation of reranking models for fine-grained opinion analysis – marking up opinion expressions and extracting opinion holders. The reranking approach makes it possible to model complex relations between multiple opinions in a sentence, allowing us to represent how opinions *interact* through the syntactic and semantic structure. We carried out evaluations on the MPQA corpus, and the experiments showed significant improvements over a conventional system that only uses local information: for both tasks, our system saw recall boosts of over 10 points.

## 1 Introduction

Recent years have seen a surge of interest in the automatic processing of *subjective language*. The technologies emerging from this research have obvious practical uses, either as stand-alone applications or supporting other NLP tools such as information retrieval or question answering systems. While early efforts in subjectivity analysis focused on coarse-grained tasks such as retrieving the subjective documents from a collection, most recent work on this topic has focused on fine-grained tasks such as determining the attitude of a particular person on a particular topic. The development and evaluation of such systems has been made possible by the release of manually annotated resources using fairly fine-grained representations to describe the structure of subjectivity in language, for instance the MPQA corpus (Wiebe et al., 2005).

A central task in the automatic analysis of subjective language is the indentification of *subjective expressions*: the text pieces that allow us to draw the conclusion that someone has a particular feeling about something. This is necessary for further analysis, such as the determination of opinion holder and the polarity of the opinion. The MPQA corpus defines two types of subjective expressions: *direct subjective expressions* (DSEs), which are explicit mentions of attitude, 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 or categorization such as *praise* or *disgust*, and the opinion holder would typically be a direct semantic argument of this verb. ESEs, on the other hand, are less easy to categorize syntactically; prototypical examples would include value-expressing adjectives such as *beautiful* and strongly charged words like *appeasement*, while the relation between the expression and the opinion holder is typically less clear-cut than for DSEs. 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)** He [made such charges]$_{DSE}$ [despite the fact]$_{ESE}$ that women's political, social and cultural participation is [not less than that]$_{ESE}$ of men.
**(2)** [However]$_{ESE}$, it is becoming [rather fashionable]$_{ESE}$ to [exchange harsh words]$_{DSE}$ with each other [like kids]$_{ESE}$.

The task of marking up these expressions has usually been approached using straightforward sequence labeling techniques using 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.

**(3)** "We will identify the $[\text{culprits}]_{ESE}$ of these clashes and $[\text{punish}]_{ESE}$ them," he $[\text{said}]_{DSE}$.

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

In addition, the various opinions expressed in a sentence are very interdependent when it comes to the resolution of their *holders*, i.e. determining the entity that harbors the sentiment manifested textually in the opinion expression. Clearly, the structure of the sentence is influential also for this task: an ESE will be quite likely to be linked to the same opinion holder as a DSE directly above it in the syntactic tree.

In this paper, we demonstrate how syntactic and semantic structural information can be used to improve the detection of opinion expressions and the extraction of opinion holders. 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. In an evaluation on the MPQA corpus, the best system we evaluated, a 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. Similarly, the recall is boosted by almost 11 points for the holder extraction (3 points in F-measure) by modeling the interaction of opinion expressions with respect to holders.

## 2 Related Work

Since the most significant body of work in subjectivity analysis has been dedicated to coarse-grained tasks such as document polarity classification, 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 not unexpected since these problems have typically been formulated as text categorization problems, and it has long been agreed 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).

As the field moves towards increasingly sophisticated tasks requiring a detailed analysis of the text, the benefit of syntactic and semantic analysis becomes more clear. For the task of subjective expression detection, Choi et al. (2006) and Breck et al. (2007) used syntactic features in a sequence model. In addition, 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. Choi et al. (2006) is notable for the use of a global model based on hand-crafted constraints and an integer linear programming optimization step to ensure a globally consistent set of opinions and holders.

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. 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 subjectivity analysis community (Somasundaran et al., 2009).

## 3 Modeling Interaction over Syntactic and Semantic Structure

Previous systems for opinion 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 opinion 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:

- Apply a standard Viterbi-based sequence labeler based on local context features but no structural interaction features. Generate a small candidate set of size $k$.
- Generate opinion holders for every proposed opinion expression.
- Apply a complex model using interaction features to pick the top candidate from the candidate set.

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.

### 3.1 Syntactic and Semantic Structures

We used the syntactic–semantic parser by Johansson and Nugues (2008) 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.

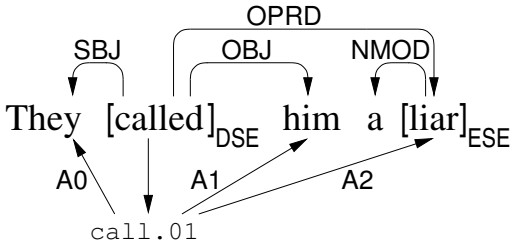


Figure 1: Syntactic and shallow semantic structure.

### 3.2 Base Sequence Labeling Model

To solve the first subtask, we implemented a standard sequence labeler for subjective expression markup, similar to the approach by Breck et al. (2007). We encoded the opinionated expression brackets using the IOB2 encoding scheme (Tjong Kim Sang and Veenstra, 1999) and trained the model using the metod by Collins (2002).

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). 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.

This sequence labeler was used to generate the candidate set for the reranker. To generate reranking training data, we carried out a 5-fold hold-out 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 Base Opinion Holder Extractor

For every opinion expression, we extracted *opinion holders*, i.e. mentions of the entity holding the opinion denoted by the opinion expression. Since the problem of holder extraction is in many ways similar to semantic argument detection – when the opinion expression is a verb, finding the holder typically entails finding a SPEAKER argument – we approached this problem using methods inspired by semantic role labeling. We thus trained support vector machines using the LIBLINEAR software (Fan et al., 2008), and applied them to the noun phrases in the same sentence as the holder. Separate classifiers were trained to extract holders for DSEs, ESEs, and OSEs. The classifiers used the following feature set:

**SYNTACTIC PATH**. Similarly to the path feature widely used in 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 the DSE *called* to the holder *They* is `SBJ`↓.

**SHALLOW-SEMANTIC RELATION**. If there is a direct shallow-semantic relation between the expression and the holder, use a feature representing its semantic role, such as `A0` for *They* with respect to *called*.

**EXPRESSION HEAD WORD AND POS**.

**HOLDER HEAD WORD AND POS**.

**DOMINATING EXPRESSION TYPE**.

**CONTEXT WORDS AND POS FOR HOLDER**.

**EXPRESSION VERB VOICE**.

However, there are also differences compared to typical argument extraction in SRL. First, it is important to note that the MPQA corpus does not annotate direct links from opinions to a holders, but from opinions to *holder coreference chains*. To handle this issue, we created positive training instances for *all* members of the coreference chain in the same sentence as the opinion, and negative instances for the other noun phrases.

Secondly, an opinion may be linked not to an overt noun phrase in a sentence, but to an *implicit* holder; a special case of implicit holder is the *writer* of the text. We trained separate classifiers to detect these situations. These classifiers did not use the features requiring a holder phrase.

Finally, there is a restriction that every expression may have at most one holder, so at test time we select only the highest-scoring opinion holder candidate.

### 3.4 Opinion Expression 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 opinion expressions; we select the shortest possible path in such cases. For instance, in (1) above, the path will be computed between *denounced* and *violation*, and in (2) between *viewed* and *impediment*.

We used the following syntactic interaction features:

**SYNTACTIC PATH**. Given a pair 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 follows an OPRD link downward.

**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 also 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.

### 3.5 Opinion Holder Reranker 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 1 where the holder is *They*, 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 1, we would thus extract the feature `DSE/Expl:OPRD↓:ESE/Expl`.

Similar to base model feature for the expression detection, we also used a feature for the output score from the holder extraction classifier.

### 3.6 Training the Reranker

We trained the reranker using the method employed by many rerankers following Collins (2002), which learns a scoring function that is trained to maximize performance on the reranking task. While there are batch learning algorithms that work in this setting (Tsochantaridis et al., 2005), online learning methods have been more popular for performance reasons. We investigated two online 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 update based on the difference between the correct output $y$ and the predicted output $\hat{y}$, where $\Phi$ 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 $\tau$ 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 $\rho$. 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 h ow 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 tha t 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 $\boldsymbol{S'}$:

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

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: replacing $c(s, s')$ with $\lfloor c(s, s') \rfloor$ gives the exact measures and replacing $c(s, s')$ with $\lceil c(s, s') \rceil$ the overlap-based measures.

To score the extraction of opinion holders, we started from the same basic approach. However, the evaluation of this task is more complex because a) we only want to give credit for holders for correctly extracted opinion expressions; b) the gold standard links opinion expressions to coreference chains rather than individual mentions of holders; c) the holder may be the writer or implicit (see 3.3). We therefore used the following method: Given a holder $h$ linked to an expression $e$, 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 labels of $e$ and $e'$. We then located the gold standard holder $h'$ by finding the closest corresponding holder in the coreference chain $H$ linked to $e'$: $h' = \arg\max_{x \in H} c(x, h)$. If $h$ is proposed as the writer, we score it as perfectly detected (coverage 1) if the coreference chain $H$ contains the writer, and a full error (coverage 0) otherwise, and similar if $h$ is implicit.

### 4.2 Machine Learning Methods

We compared 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 labeling

from the base model.

| System | $P$ | $R$ | $F$ |
|---|---|---|---|
| Baseline | 63.36 | 46.77 | 53.82 |
| Perceptron | 62.84 | 48.13 | 54.51 |
| PA | 63.50 | 51.79 | 57.04 |

Table 1: Evaluation of reranking learning methods.

We note that the best performance was obtained using the PA algorithm. While these results are satisfactory, it is possible that they could be improved further if we would use a batch learning method such as SVM$^{\text{struct}}$ (Tsochantaridis et al., 2005) instead of the online learning methods used here.

### 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 16 – in fact, the precision actually improves over the baseline for small candidate set sizes. After the size goes above 16, the recall (and the F-measure) still rises, but at the cost of decreased precision.

### 4.4 Syntactic and Semantic 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 has 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 labels 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.

### 4.5 Opinion Holder Extraction

Table 5 shows the performance of the opinion holder extractor. The baseline applies the holder

classifier (3.3) to the opinions extracted by the base sequence labeler (3.2), without modeling any interactions between opinions. A large performance boost is then achieved simply by applying the opinion expression reranker ($k = 64$); this is simply the consequence of improved expression detection, since a correct expression is required to get credit for a holder).

However, we can improve on this by adding the holder interaction features: both the SHARED HOLDERS and HOLDER TYPES + PATH features contribute to improving the recall even further.

| System | $P$ | $R$ | $F$ |
|---|---|---|---|
| Baseline | 57.66 | 45.14 | 50.64 |
| Reranked expressions | 52.35 | 52.54 | 52.45 |
| SHARED HOLDERS | 52.43 | 55.21 | 53.78 |
| HTYPES + PATH | 52.22 | 54.41 | 53.30 |
| Both | 52.28 | 55.99 | 54.07 |

Table 5: Opinion holder extraction experiments.

## 5 Conclusion

We have shown that features derived from grammatical and semantic role structure can be used to improve two fundamental tasks in fine-grained opinion analysis: the detection of opinionated expressions in subjectivity analysis, and the extraction of opinion holders. Our feature sets are based on interaction between opinions, which makes exact inference intractable. To overcome this issue, we used an implementation based on reranking: we first generated opinion expression sequence candidates using a simple sequence labeler similar to the approach by Breck et al. (2007). We then applied SRL-inspired opinion holder extraction classifiers, and finally a global model applying to all opinions and holders.

Our experiments show that the interaction-based models result in drastic improvements. Significantly, we see significant boosts in recall (10 points for both tasks) while the precision decreases only slightly, resulting in clear F-measure improvements. This result compares favorably with previously published results, which have been precision-oriented and scored quite low on recall.

We analyzed the impact of the syntactic and semantic features and saw that the best model is the one that makes use of 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.

There are multiple 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 (Somasundaran et al., 2009); it is clear that these structures are very relevant for explaining the way humans organize their expressions of opinions rhetorically. However, theoretical depth does not necessarily guarantee practical applicability, 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.

## 6 Acknowledgements

The research described in this paper 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

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

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

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

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.

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.

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.

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, United Kingdom.

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

Kim, Soo-Min 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*.

Kobayashi, Nozomi, 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)*.

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, United States.

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

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

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

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 EMNLP*.

Stoyanov, Veselin and Claire Cardie. 2006. Partially supervised coreference resolution for opinion summarization through structured rule learning. In *Proceedings of EMNLP 2006*.

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

Tsochantaridis, Iannis, 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*.

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.

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

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

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, Japan.