# Hierarchical Sequential Learning for Extracting Opinions and their Attributes

**Yejin Choi and Claire Cardie**
Department of Computer Science
Cornell University
Ithaca, NY 14853
{ychoi,cardie}@cs.cornell.edu

## Abstract

Automatic opinion recognition involves a number of related tasks, such as identifying the boundaries of opinion expression, determining their polarity, and determining their intensity. Although much progress has been made in this area, existing research typically treats each of the above tasks in isolation. In this paper, we apply a hierarchical parameter sharing technique using Conditional Random Fields for fine-grained opinion analysis, jointly detecting the boundaries of opinion expressions as well as determining two of their key attributes — polarity and intensity. Our experimental results show that our proposed approach improves the performance over a baseline that does not exploit hierarchical structure among the classes. In addition, we find that the joint approach outperforms a baseline that is based on cascading two separate components.

## 1 Introduction

Automatic opinion recognition involves a number of related tasks, such as identifying expressions of opinion (e.g. Kim and Hovy (2005), Popescu and Etzioni (2005), Breck et al. (2007)), determining their polarity (e.g. Hu and Liu (2004), Kim and Hovy (2004), Wilson et al. (2005)), and determining their strength, or intensity (e.g. Popescu and Etzioni (2005), Wilson et al. (2006)). Most previous work treats each subtask in isolation: opinion expression extraction (i.e. detecting the boundaries of opinion expressions) and opinion attribute classification (e.g. determining values for polarity and intensity) are tackled as separate steps in opinion recognition systems. Unfortunately, errors from individual components will propagate in systems with cascaded component architectures, causing performance degradation in the end-to-end system (e.g. Finkel et al. (2006)) — in our case, in the end-to-end opinion recognition system.

In this paper, we apply a *hierarchical parameter sharing* technique (e.g., Cai and Hofmann (2004), Zhao et al. (2008)) using Conditional Random Fields (CRFs) (Lafferty et al., 2001) to fine-grained opinion analysis. In particular, we aim to jointly identify the boundaries of opinion expressions as well as to determine two of their key attributes — polarity and intensity.

Experimental results show that our proposed approach improves the performance over the baseline that does not exploit the hierarchical structure among the classes. In addition, we find that the joint approach outperforms a baseline that is based on cascading two separate systems.

## 2 Hierarchical Sequential Learning

We define the problem of joint extraction of opinion expressions and their attributes as a sequence tagging task as follows. Given a sequence of tokens, $x = x_1 \ ... \ x_n$, we predict a sequence of labels, $y = y_1 \ ... \ y_n$, where $y_i \in \{0, ..., 9\}$ are defined as conjunctive values of polarity labels and intensity labels, as shown in Table 1. Then the conditional probability $p(y|x)$ for linear-chain CRFs is given as (Lafferty et al., 2001)

$$P(y|x) = \frac{1}{Z_x} \exp \sum_i \Big( \lambda \, f(y_i, x, i) + \lambda' \, f'(y_{i-1}, y_i, x, i) \Big)$$

where $Z_x$ is the normalization factor.

In order to apply a hierarchical parameter sharing technique (e.g., Cai and Hofmann (2004), Zhao et al. (2008)), we extend parameters as follows.



*Proceedings of the ACL 2010 Conference Short Papers*, pages 269–274,
Uppsala, Sweden, 11-16 July 2010. 

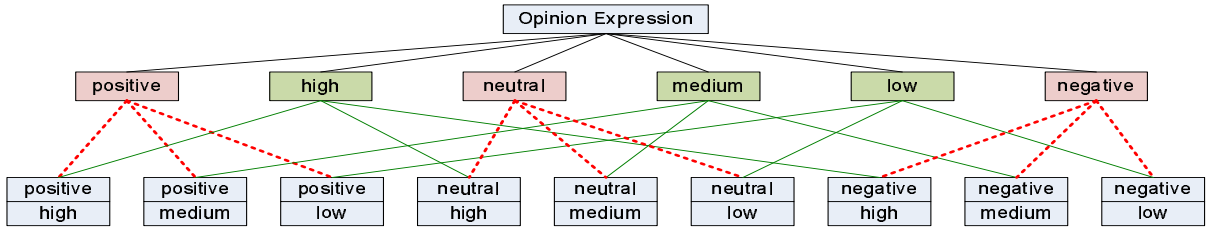


Figure 1: The hierarchical structure of classes for opinion expressions with polarity (positive, neutral, negative) and intensity (high, medium, low)

| LABEL | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| POLARITY | none | positive | positive | positive | neutral | neutral | neutral | negative | negative | negative |
| INTENSITY | none | high | medium | low | high | medium | low | high | medium | low |

Table 1: Labels for Opinion Extraction with Polarity and Intensity

$$\begin{aligned} \lambda\, f(y_i, x, i) &= \lambda_\alpha\, g_{\text{O}}(\alpha, x, i) \\ &+ \lambda_\beta\, g_{\text{P}}(\beta, x, i) \\ &+ \lambda_\gamma\, g_{\text{S}}(\gamma, x, i) \end{aligned} \quad (1)$$

$$\begin{aligned} \lambda'\, f'(y_{i-1}, y_i, x, i) &= \lambda'_{\alpha,\hat{\alpha}}\, g'_{\text{O}}(\alpha, \hat{\alpha}, x, i) \\ &+ \lambda'_{\beta,\hat{\beta}}\, g'_{\text{P}}(\beta, \hat{\beta}, x, i) \\ &+ \lambda'_{\gamma,\hat{\gamma}}\, g'_{\text{S}}(\gamma, \hat{\gamma}, x, i) \end{aligned}$$

where $g_{\text{O}}$ and $g'_{\text{O}}$ are feature vectors defined for **O**pinion extraction, $g_{\text{P}}$ and $g'_{\text{P}}$ are feature vectors defined for **P**olarity extraction, and $g_{\text{S}}$ and $g'_{\text{S}}$ are feature vectors defined for **S**trength extraction, and

$$\begin{aligned} \alpha, \hat{\alpha} &\in \{\text{OPINION, NO-OPINION}\} \\ \beta, \hat{\beta} &\in \{\text{POSITIVE, NEGATIVE, NEUTRAL, NO-POLARITY}\} \\ \gamma, \hat{\gamma} &\in \{\text{HIGH, MEDIUM, LOW, NO-INTENSITY}\} \end{aligned}$$

For instance, if $y_i = 1$, then

$$\begin{aligned} \lambda\, f(1, x, i) &= \lambda_{\text{OPINION}}\, g_{\text{O}}(\text{OPINION}, x, i) \\ &+ \lambda_{\text{POSITIVE}}\, g_{\text{P}}(\text{POSITVE}, x, i) \\ &+ \lambda_{\text{HIGH}}\, g_{\text{S}}(\text{HIGH}, x, i) \end{aligned}$$

If $y_{i-1} = 0, y_i = 4$, then

$$\begin{aligned} &\lambda'\, f'(0, 4, x, i) \\ &= \lambda'_{\text{NO-OPINION,OPINION}}\, g'_{\text{O}}(\text{NO-OPINION, OPINION}, x, i) \\ &+ \lambda'_{\text{NO-POLARITY, NEUTRAL}}\, g'_{\text{P}}(\text{NO-POLARITY, NEUTRAL}, x, i) \\ &+ \lambda'_{\text{NO-INTENSITY, HIGH}}\, g'_{\text{S}}(\text{NO-INTENSITY, HIGH}, x, i) \end{aligned}$$

This hierarchical construction of feature and weight vectors allows similar labels to share the same subcomponents of feature and weight vectors. For instance, all $\lambda\ f(y_i, x, i)$ such that $y_i \in \{1, 2, 3\}$ will share the same component $\lambda_{\text{POSITIVE}}\ g_{\text{P}}(\text{POSITVE}, x, i)$. Note that there can be other variations of hierarchical construction. For instance, one can add $\lambda_\delta\ g_{\text{I}}(\delta, x, i)$ and $\lambda'_{\delta,\hat{\delta}}\ g'_{\text{I}}(\delta, \hat{\delta}, x, i)$ to Equation (1) for $\delta \in \{0, 1, ..., 9\}$, in order to allow more individualized learning for each label.

Notice also that the number of sets of parameters constructed by Equation (1) is significantly smaller than the number of sets of parameters that are needed without the hierarchy. The former requires $(2+4+4)+(2\times 2+4\times 4+4\times 4) = 46$ sets of parameters, but the latter requires $(10) + (10 \times 10) = 110$ sets of parameters. Because a combination of a polarity component and an intensity component can distinguish each label, it is not necessary to define a separate set of parameters for each label.

## 3 Features

We first introduce definitions of key terms that will be used to describe features.

- **PRIOR-POLARITY & PRIOR-INTENSITY**: We obtain these prior-attributes from the *polarity lexicon* populated by Wilson et al. (2005).
- **EXP-POLARITY, EXP-INTENSITY & EXP-SPAN**: Words in a given opinion expression often do not share the same prior-attributes. Such discontinuous distribution of features can make it harder to learn the desired opinion expression boundaries. Therefore, we try to obtain expression-level attributes (EXP-POLARITY and EXP-INTENSITY) using simple heuristics. In order to derive EXP-POLARITY, we perform simple

voting. If there is a word with a negation effect, such as "never", "not", "hardly", "against", then we flip the polarity. For EXP-INTENSITY, we use the highest PRIOR-INTENSITY in the span. The text span with the same expression-level attributes are referred to as EXP-SPAN.

### 3.1 Per-Token Features

Per-token features are defined in the form of $g_{\text{O}}(\alpha, x, i), g_{\text{P}}(\beta, x, i)$ and $g_{\text{S}}(\gamma, x, i)$. The domains of $\alpha, \beta, \gamma$ are as given in Section 3.

#### Common Per-Token Features

Following features are common for all class labels. The notation $\otimes$ indicates conjunctive operation of two values.

- PART-OF-SPEECH($x_i$):
  based on GATE (Cunningham et al., 2002).
- WORD($x_i$), WORD($x_{i-1}$), WORD($x_{i+1}$)
- WORDNET-HYPERNYM($x_i$):
  based on WordNet (Miller, 1995).
- OPINION-LEXICON($x_i$):
  based on *opinion lexicon* (Wiebe et al., 2002).
- SHALLOW-PARSER($x_i$):
  based on CASS partial parser (Abney, 1996).
- PRIOR-POLARITY($x_i$) $\otimes$ PRIOR-INTENSITY($x_i$)
- EXP-POLARITY($x_i$) $\otimes$ EXP-INTENSITY($x_i$)
- EXP-POLARITY($x_i$) $\otimes$ EXP-INTENSITY($x_i$) $\otimes$ STEM($x_i$)
- EXP-SPAN($x_i$):
  boolean to indicate whether $x_i$ is in an EXP-SPAN.
- DISTANCE-TO-EXP-SPAN($x_i$): $0, 1, 2, 3+$.
- EXP-POLARITY($x_i$) $\otimes$ EXP-INTENSITY($x_i$) $\otimes$ EXP-SPAN($x_i$)

#### Polarity Per-Token Features

These features are included only for $g_{\text{O}}(\alpha, x, i)$ and $g_{\text{P}}(\beta, x, i)$, which are the feature functions corresponding to the polarity-based classes.

- PRIOR-POLARITY($x_i$), EXP-POLARITY(($x_i$)
- STEM($x_i$) $\otimes$ EXP-POLARITY($x_i$)
- COUNT-OF-$Polarity$:
  where $Polarity \in$ {positive, neutral, negative}. This feature encodes the number of positive, neutral, and negative EXP-POLARITY words respectively, in the current sentence.
- STEM($x_i$) $\otimes$ COUNT-OF-$Polarity$
- EXP-POLARITY($x_i$) $\otimes$ COUNT-OF-$Polarity$
- EXP-SPAN($x_i$) and EXP-POLARITY($x_i$)
- DISTANCE-TO-EXP-SPAN($x_i$) $\otimes$ EXP-POLARITY($x_p$)

#### Intensity Per-Token Features

These features are included only for $g_{\text{O}}(\alpha, x, i)$ and $g_{\text{S}}(\gamma, x, i)$, which are the feature functions corresponding to the intensity-based classes.

- PRIOR-INTENSITY($x_i$), EXP-INTENSITY($x_i$)
- STEM($x_i$) $\otimes$ EXP-INTENSITY($x_i$)
- COUNT-OF-STRONG, COUNT-OF-WEAK:
  the number of strong and weak EXP-INTENSITY words in the current sentence.
- INTENSIFIER($x_i$): whether $x_i$ is an intensifier, such as "extremely", "highly", "really".
- STRONGMODAL($x_i$): whether $x_i$ is a strong modal verb, such as "must", "can", "will".
- WEAKMODAL($x_i$): whether $x_i$ is a weak modal verb, such as "may", "could", "would".
- DIMINISHER($x_i$): whether $x_i$ is a diminisher, such as "little", "somewhat", "less".
- PRECEDED-BY-$\tau$($x_i$),
  PRECEDED-BY-$\tau$($x_i$) $\otimes$ EXP-INTENSITY($x_i$):
  where $\tau \in$ { INTENSIFIER, STRONGMODAL, WEAKMODAL, DIMINISHER}
- $\tau(x_i) \otimes$ EXP-INTENSITY($x_i$),
  $\tau(x_i) \otimes$ EXP-INTENSITY($x_{i-1}$),
  $\tau(x_{i-1}) \otimes$ EXP-INTENSITY($x_{i+1}$)
- EXP-SPAN($x_i$) $\otimes$ EXP-INTENSITY($x_i$)
- DISTANCE-TO-EXP-SPAN($x_i$) $\otimes$ EXP-INTENSITY($x_p$)

### 3.2 Transition Features

Transition features are employed to help with boundary extraction as follows:

#### Polarity Transition Features

Polarity transition features are features that are used only for $g'_{\text{O}}(\alpha, \hat{\alpha}, x, i)$ and $g'_{\text{P}}(\beta, \hat{\beta}, x, i)$.

- PART-OF-SPEECH($x_i$) $\otimes$ PART-OF-SPEECH($x_{i+1}$) $\otimes$ EXP-POLARITY($x_i$)
- EXP-POLARITY($x_i$) $\otimes$ EXP-POLARITY($x_{i+1}$)

#### Intensity Transition Features

Intensity transition features are features that are used only for $g'_{\text{O}}(\alpha, \hat{\alpha}, x, i)$ and $g'_{\text{S}}(\gamma, \hat{\gamma}, x, i)$.

- PART-OF-SPEECH($x_i$) $\otimes$ PART-OF-SPEECH($x_{i+1}$) $\otimes$ EXP-INTENSITY($x_i$)
- EXP-INTENSITY($x_i$) $\otimes$ EXP-INTENSITY($x_{i+1}$)

## 4 Evaluation

We evaluate our system using the Multi-Perspective Question Answering (MPQA) corpus[1]. Our gold standard opinion expressions cor-

[1]The MPQA corpus can be obtained at http://nrrc.mitre.org/NRRC/publications.htm.

| Method Description | Positive | | | Neutral | | | Negative | | |
|---|---|---|---|---|---|---|---|---|---|
| | r(%) | p(%) | f(%) | r(%) | p(%) | f(%) | r(%) | p(%) | f(%) |
| Polarity-Only ∩ Intensity-Only (BASELINE1) | 29.6 | 65.7 | 40.8 | 26.5 | 69.1 | 38.3 | 35.5 | 77.0 | 48.6 |
| Joint without Hierarchy (BASELINE2) | 30.7 | 65.7 | 41.9 | 29.9 | 66.5 | 41.2 | 37.3 | 77.1 | 50.3 |
| Joint with Hierarchy | 31.8 | 67.1 | **43.1** | 31.9 | 66.6 | **43.1** | 40.4 | 76.2 | **52.8** |

Table 2: Performance of Opinion Extraction with Correct Polarity Attribute

| Method Description | High | | | Medium | | | Low | | |
|---|---|---|---|---|---|---|---|---|---|
| | r(%) | p(%) | f(%) | r(%) | p(%) | f(%) | r(%) | p(%) | f(%) |
| Polarity-Only ∩ Intensity-Only (BASELINE1) | 26.4 | 58.3 | 36.3 | 29.7 | 59.0 | 39.6 | 15.4 | 60.3 | 24.5 |
| Joint without Hierarchy (BASELINE2) | 29.7 | 54.2 | **38.4** | 28.0 | 57.4 | 37.6 | 18.8 | 55.0 | 28.0 |
| Joint with Hierarchy | 27.1 | 55.2 | 36.3 | 32.0 | 56.5 | **40.9** | 21.1 | 56.3 | **30.7** |

Table 3: Performance of Opinion Extraction with Correct Intensity Attribute

| Method Description | r(%) | p(%) | f(%) |
|---|---|---|---|
| Polar-Only ∩ Intensity-Only | 43.3 | 92.0 | 58.9 |
| Joint without Hierarchy | 46.0 | 88.4 | 60.5 |
| Joint with Hierarchy | 48.0 | 87.8 | **62.0** |

Table 4: Performance of Opinion Extraction

respond to *direct subjective expression* and *expressive subjective element* (Wiebe et al., 2005).[2]

Our implementation of hierarchical sequential learning is based on the Mallet (McCallum, 2002) code for CRFs. In all experiments, we use a Gaussian prior of 1.0 for regularization. We use 135 documents for development, and test on a different set of 400 documents using 10-fold cross-validation. We investigate three options for jointly extracting opinion expressions with their attributes as follows:

**[Baseline-1] Polarity-Only ∩ Intensity-Only:** For this baseline, we train two separate sequence tagging CRFs: one that extracts opinion expressions only with the polarity attribute (using common features and polarity extraction features in Section 3), and another that extracts opinion expressions only with the intensity attribute (using common features and intensity extraction features in Section 3). We then combine the results from two separate CRFs by collecting all opinion entities extracted by both sequence taggers.[3] This baseline effectively represents a cascaded component approach.

**[Baseline-2] Joint without Hierarchy:** Here we use simple linear-chain CRFs without exploiting the class hierarchy for the opinion recognition task. We use the tags shown in Table 1.

**Joint with Hierarchy:** Finally, we test the hierarchical sequential learning approach elaborated in Section 3.

### 4.1 Evaluation Results

We evaluate all experiments at the opinion entity level, i.e. at the level of each opinion expression rather than at the token level. We use three evaluation metrics: recall, precision, and F-measure with equally weighted recall and precision.

Table 4 shows the performance of opinion extraction without matching any attribute. That is, an extracted opinion entity is counted as correct if it overlaps[4] with a gold standard opinion expression, without checking the correctness of its attributes. Table 2 and 3 show the performance of opinion extraction with the correct polarity and intensity respectively.

From all of these evaluation criteria, JOINT WITH

[2]Only 1.5% of the polarity annotations correspond to *both*; hence, we merge *both* into the *neutral*. Similarly, for gold standard intensity, we merge *extremely high* into *high*.

[3]We collect all entities whose portions of text spans are extracted by both models.

[4]Overlap matching is a reasonable choice as the annotator agreement study is also based on overlap matching (Wiebe et al., 2005). One might wonder whether the overlap matching scheme could allow a degenerative case where extracting the entire test dataset as one giant opinion expression would yield 100% recall and precision. Because each sentence corresponds to a different test instance in our model, and because some sentences do not contain any opinion expression in the dataset, such degenerative case is not possible in our experiments.

HIERARCHY performs the best, and the least effective one is BASELINE-1, which cascades two separately trained models. It is interesting that the simple sequential tagging approach even without exploiting the hierarchy (BASELINE-2) performs better than the cascaded approach (BASELINE-1).

When evaluating with respect to the polarity attribute, the performance of the negative class is substantially higher than the that of other classes. This is not surprising as there is approximately twice as much data for the negative class. When evaluating with respect to the intensity attribute, the performance of the LOW class is substantially lower than that of other classes. This result reflects the fact that it is inherently harder to distinguish an opinion expression with low intensity from no opinion. In general, we observe that determining correct intensity attributes is a much harder task than determining correct polarity attributes.

In order to have a sense of upper bound, we also report the individual performance of two separately trained models used for BASELINE-1: for the Polarity-Only model that extracts opinion boundaries only with polarity attribute, the F-scores with respect to the positive, neutral, negative classes are 46.7, 47.5, 57.0, respectively. For the Intensity-Only model, the F-scores with respect to the high, medium, low classes are 37.1, 40.8, 26.6, respectively. Remind that neither of these models alone fully solve the joint task of extracting boundaries as well as determining two attributions simultaneously. As a result, when conjoining the results from the two models (BASELINE-1), the final performance drops substantially.

We conclude from our experiments that the simple joint sequential tagging approach even without exploiting the hierarchy brings a better performance than combining two separately developed systems. In addition, our hierarchical joint sequential learning approach brings a further performance gain over the simple joint sequential tagging method.

## 5 Related Work

Although there have been much research for fine-grained opinion analysis (e.g., Hu and Liu (2004), Wilson et al. (2005), Wilson et al. (2006), Choi and Claire (2008), Wilson et al. (2009)),[5] none is directly comparable to our results; much of previous work studies only a subset of what we tackle in this paper. However, as shown in Section 4.1, when we train the learning models only for a subset of the tasks, we can achieve a better performance instantly by making the problem simpler. Our work differs from most of previous work in that we investigate how solving multiple related tasks affects performance on sub-tasks.

The hierarchical parameter sharing technique used in this paper has been previously used by Zhao et al. (2008) for opinion analysis. However, Zhao et al. (2008) employs this technique only to classify sentence-level attributes (polarity and intensity), without involving a much harder task of detecting boundaries of sub-sentential entities.

## 6 Conclusion

We applied a hierarchical parameter sharing technique using Conditional Random Fields for fine-grained opinion analysis. Our proposed approach jointly extract opinion expressions from unstructured text and determine their attributes — polarity and intensity. Empirical results indicate that the simple joint sequential tagging approach even without exploiting the hierarchy brings a better performance than combining two separately developed systems. In addition, we found that the hierarchical joint sequential learning approach improves the performance over the simple joint sequential tagging method.

## Acknowledgments

This work was supported in part by National Science Foundation Grants BCS-0904822, BCS-0624277, IIS-0535099 and by the Department of Homeland Security under ONR Grant N0014-07-1-0152. We thank the reviewers and Ainur Yessenalina for many helpful comments.

## References

S. Abney. 1996. Partial parsing via finite-state cascades. In *Journal of Natural Language Engineering*, 2(4).

E. Breck, Y. Choi and C. Cardie. 2007. Identifying Expressions of Opinion in Context. In *IJCAI*.

[5] For instance, the results of Wilson et al. (2005) is not comparable even for our Polarity-Only model used inside BASELINE-1, because Wilson et al. (2005) does not operate on the entire corpus as unstructured input. Instead, Wilson et al. (2005) evaluate only on known words that are in their opinion lexicon. Furthermore, Wilson et al. (2005) simplifies the problem by combining neutral opinions and no opinions into the same class, while our system distinguishes the two.

L. Cai and T. Hofmann. 2004. Hierarchical document categorization with support vector machines. In *CIKM*.

Y. Choi and C. Cardie. 2008. Learning with Compositional Semantics as Structural Inference for Subsentential Sentiment Analysis. In *EMNLP*.

H. Cunningham, D. Maynard, K. Bontcheva and V. Tablan. 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *ACL*.

J. R. Finkel, C. D. Manning and A. Y. Ng. 2006. Solving the Problem of Cascading Errors: Approximate Bayesian Inference for Linguistic Annotation Pipelines. In *EMNLP*.

M. Hu and B. Liu. 2004. Mining and Summarizing Customer Reviews. In *KDD*.

S. Kim and E. Hovy. 2004. Determining the sentiment of opinions. In *COLING*.

S. Kim and E. Hovy. 2005. Automatic Detection of Opinion Bearing Words and Sentences. In Companion Volume to the *Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP-05)*.

J. Lafferty, A. McCallum and F. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML*.

A. McCallum. 2002. MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu.

G. A. Miller. 1995. WordNet: a lexical database for English. In *Communications of the ACM, 38(11)*.

Ana-Maria Popescu and O. Etzioni. 2005. Extracting Product Features and Opinions from Reviews. In *HLT-EMNLP*.

J. Wiebe, E. Breck, C. Buckley, C. Cardie, P. Davis, B. Fraser, D. Litman, D. Pierce, E. Riloff and T. Wilson. 2002. Summer Workshop on Multiple-Perspective Question Answering: Final Report. In *NRRC*.

J. Wiebe and T. Wilson and C. Cardie 2005. Annotating Expressions of Opinions and Emotions in Language. In *Language Resources and Evaluation, volume 39, issue 2-3*.

T. Wilson, J. Wiebe and P. Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *HLT-EMNLP*.

T. Wilson, J. Wiebe and R. Hwa. 2006. Recognizing strong and weak opinion clauses. In *Computational Intelligence*. 22 (2): 73-99.

T. Wilson, J. Wiebe and P. Hoffmann. 2009. Recognizing Contextual Polarity: an exploration of features for phrase-level sentiment analysis. Computational Linguistics 35(3).

J. Zhao, K. Liu and G. Wang. 2008. Adding Redundant Features for CRFs-based Sentence Sentiment Classification. In *EMNLP*.