Learning with Compositional Semantics as Structural Inference for Subsentential Sentiment Analysis

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

Abstract

Determining the polarity of a sentiment-bearing expression requires more than a simple bag-of-words approach. In particular, words or constituents within the expression can interact with each other to yield a particular overall polarity. In this paper, we view such subsentential interactions in light of *compositional semantics*, and present a novel learning-based approach that incorporates structural inference motivated by compositional semantics into the learning procedure. Our experiments show that (1) simple heuristics based on compositional semantics can perform better than learning-based methods that do not incorporate compositional semantics (accuracy of 89.7% vs. 89.1%), but (2) a method that integrates compositional semantics into learning performs better than all other alternatives (90.7%). We also find that "content-word negators", not widely employed in previous work, play an important role in determining expression-level polarity. Finally, in contrast to conventional wisdom, we find that expression-level classification accuracy uniformly *decreases* as additional, potentially disambiguating, context is considered.

1 Introduction

Determining the polarity of sentiment-bearing expressions at or below the sentence level requires more than a simple bag-of-words approach. One of the difficulties is that words or constituents within the expression can interact with each other to yield a particular overall polarity. To facilitate our discussion, consider the following examples:

1: [I did [*not*]$^\neg$ have any [*doubt*]$^-$ about it.]$^+$
2: [The report [*eliminated*]$^\neg$ my [*doubt*]$^-$.]$^+$
3: [They could [*not*]$^\neg$ [*eliminate*]$^\neg$ my [*doubt*]$^-$.]$^-$

In the first example, "doubt" in isolation carries a negative sentiment, but the overall polarity of the sentence is positive because there is a *negator* "not", which flips the polarity. In the second example, both "eliminated" and "doubt" carry negative sentiment in isolation, but the overall polarity of the sentence is positive because "eliminated" acts as a negator for its argument "doubt". In the last example, there are effectively two negators – "not" and "eliminated" – which reverse the polarity of "doubt" twice, resulting in the negative polarity for the overall sentence.

These examples demonstrate that words or constituents interact with each other to yield the expression-level polarity. And a system that simply takes the majority vote of the polarity of individual words will not work well on the above examples. Indeed, much of the previous learning-based research on this topic tries to incorporate salient interactions by encoding them as features. One approach includes features based on *contextual valence shifters*[1] (Polanyi and Zaenen, 2004), which are words that affect the polarity or intensity of sentiment over neighboring text spans (e.g., Kennedy and Inkpen (2005), Wilson et al. (2005), Shaikh et al. (2007)). Another approach encodes frequent subsentential patterns (e.g., McDonald et al. (2007)) as features; these might indirectly capture some of the subsentential interactions that affect polarity. How-

[1] For instance, "never", "nowhere", "little", "most", "lack", "scarcely", "deeply".

ever, both types of approach are based on learning models with a flat bag-of-features: some structural information can be encoded as higher order features, but the final representation of the input is still a flat feature vector that is inherently too limited to adequately reflect the complex structural nature of the underlying subsentential interactions. (Liang et al., 2008)

Moilanen and Pulman (2007), on the other hand, handle the structural nature of the interactions more directly using the ideas from *compositional semantics* (e.g., Montague (1974), Dowty et al. (1981)). In short, *the Principle of Compositionality* states that the meaning of a compound expression is a function of the meaning of its parts and of the syntactic rules by which they are combined (e.g., Montague (1974), Dowty et al. (1981)). And Moilanen and Pulman (2007) develop a collection of composition rules to assign a sentiment value to individual expressions, clauses, or sentences. Their approach can be viewed as a type of structural inference, but their hand-written rules have not been empirically compared to learning-based alternatives, which one might expect to be more effective in handling some aspects of the polarity classification task.

In this paper, we begin to close the gap between learning-based approaches to expression-level polarity classification and those founded on compositional semantics: we present a novel learning-based approach that incorporates structural inference motivated by compositional semantics into the learning procedure.

Adopting the view point of compositional semantics, our working assumption is that the polarity of a sentiment-bearing expression can be determined in a two-step process: (1) assess the polarities of the constituents of the expression, and then (2) apply a relatively simple set of inference rules to combine them recursively. Rather than a rigid application of hand-written compositional inference rules, however, we hypothesize that an ideal solution to the expression-level polarity classification task will be a method that can exploit ideas from compositional semantics while providing the flexibility needed to handle the complexities of real-world natural language — exceptions, unknown words, missing semantic features, and inaccurate or missing rules. The learning-based approach proposed in this paper takes a first step in this direction.

In addition to the novel learning approach, this paper presents new insights for *content-word negators*, which we define as content words that can negate the polarity of neighboring words or constituents. (e.g., words such as "eliminated" in the example sentences). Unlike *function-word negators*, such as "not" or "never", content-word negators have been recognized and utilized less actively in previous work. (Notable exceptions include e.g., Niu et al. (2005), Wilson et al. (2005), and Moilanen and Pulman (2007).[2])

In our experiments, we compare learning- and non-learning-based approaches to expression-level polarity classification — with and without compositional semantics — and find that (1) simple heuristics based on compositional semantics outperform (89.7% in accuracy) other reasonable heuristics that do not incorporate compositional semantics (87.7%); they can also perform better than simple learning-based methods that do not incorporate compositional semantics (89.1%), (2) combining learning with the heuristic rules based on compositional semantics further improves the performance (90.7%), (3) content-word negators play an important role in determining the expression-level polarity, and, somewhat surprisingly, we find that (4) expression-level classification accuracy uniformly decreases as additional, potentially disambiguating, context is considered.

In what follows, we first explore heuristic-based approaches in §2, then we present learning-based approaches in §3. Next we present experimental results in §4, followed by related work in §5.

2 Heuristic-Based Methods

This section describes a set of heuristic-based methods for determining the polarity of a sentiment-bearing expression. Each assesses the polarity of the words or constituents using a polarity lexicon that indicates whether a word has positive or negative polarity, and finds negators in the given expression using a negator lexicon. The methods then infer the expression-level polarity using voting-based heuristics (§ 2.1) or heuristics that incorporate compositional semantics (§2.2). The lexicons are described

[2] See §5. Related Work for detailed discussion.

	Vote	Neg(1)	Neg(n)	NegEx(1)	NegEx(n)	Compo
type of negators	none	function-word		function-word & content-word		
maximum # of negations applied	0	1	n	1	n	n
scope of negators	N/A	over the entire expression				compositional

Table 1: Heuristic methods. (n refers to the number of negators found in a given expression.)

	Rules		Examples
1	Polarity(not_[arg1]) =	¬ Polarity(arg1)	not [bad]$_{arg1}$.
2	Polarity([VP]_[NP]) =	Compose([VP], [NP])	[destroyed]$_{VP}$ [the terrorism]$_{NP}$.
3	Polarity([VP1]_to_[VP2]) =	Compose([VP1], [VP2])	[refused]$_{VP1}$ to [deceive]$_{VP2}$ the man.
4	Polarity([adj]_to_[VP]) =	Compose([adj], [VP])	[unlikely]$_{adj}$ to [destroy]$_{VP}$ the planet.
5	Polarity([NP1]_[IN]_[NP2]) =	Compose([NP1], [NP2])	[lack]$_{NP1}$ [of]$_{IN}$ [crime]$_{NP2}$ in rural areas.
6	Polarity([NP]_[VP]) =	Compose([VP], [NP])	[pollution]$_{NP}$ [has decreased]$_{VP}$.
7	Polarity([NP]_be_[adj]) =	Compose([adj], [NP])	[harm]$_{NP}$ is [minimal]$_{adj}$.

Definition of Compose(arg1, arg2)	
	Compose(arg1, arg2) =
For CompoMC:	if (arg1 is a negator) then ¬ Polarity(arg2)
(**COMPO**sition with **M**ajority **C**lass)	else if (Polarity(arg1) == Polarity(arg2)) then Polarity(arg1)
	else the majority polarity of data
	Compose(arg1, arg2) =
For CompoPR:	if (arg1 is a negator) then ¬ Polarity(arg2)
(**COMPO**sition with **PR**iority)	else Polarity(arg1)

Table 2: Compositional inference rules motivated by compositional semantics.

in §2.3.

2.1 Voting

We first explore five simple heuristics based on voting. Vote is defined as the majority polarity vote by words in a given expression. That is, we count the number of positive polarity words and negative polarity words in a given expression, and assign the majority polarity to the expression. In the case of a tie, we default to the prevailing polarity of the data.

For Neg(1), we first determine the majority polarity vote as above, and then if the expression contains *any* function-word negator, flip the polarity of the majority vote once. Neg(n) is similar to Neg(1), except we flip the polarity of the majority vote n times after the majority vote, where n is the number of function-word negators in a given expression.

NegEx(1) and NegEx(n) are defined similarly as Neg(1) and Neg(n) above, except both function-word negators and content-word negators are considered as negators when flipping the polarity of the majority vote. See Table 1 for summary. Note that a word can be both a negator and have a negative prior polarity. For the purpose of voting, if a word is defined as a negator per the voting scheme, then that word does not participate in the majority vote.

For brevity, we refer to Neg(1) and Neg(n) collectively as Neg, and NegEx(1) and NegEx(n) collectively as NegEx.

2.2 Compositional semantics

Whereas the heuristics above use voting-based inference, those below employ a set of hand-written rules motivated by compositional semantics. Table 2 shows the definition of the rules along with motivating examples. In order to apply a rule, we first detect a syntactic pattern (e.g., [destroyed]$_{VP}$ [the terrorism]$_{NP}$), then apply the *Compose* function as defined in Table 2 (e.g., Compose([destroyed], [the terrorism]) by rule #2).[3]

[3] Our implementation uses part-of-speech tags and function-words to coarsely determine the patterns. An implementation

Compose first checks whether the first argument is a negator, and if so, flips the polarity of the second argument. Otherwise, *Compose* resolves the polarities of its two arguments. Note that if the second argument is a negator, we do not flip the polarity of the first argument, because the first argument in general is not in the semantic scope of the negation.[4] Instead, we treat the second argument as a constituent with negative polarity.

We experiment with two variations of the *Compose* function depending on how conflicting polarities are resolved: COMPOMC uses a *Compose* function that defaults to the **M**ajority **C**lass of the polarity of the data,[5] while COMPOPR uses a *Compose* function that selects the polarity of the argument that has higher semantic **PR**iority. For brevity, we refer to COMPOPR and COMPOMC collectively as COMPO.

2.3 Lexicons

The polarity lexicon is initialized with the lexicon of Wilson et al. (2005) and then expanded using the General Inquirer dictionary.[6] In particular, a word contained in at least two of the following categories is considered as positive: POSITIV, PSTV, POSAFF, PLEASUR, VIRTUE, INCREAS, and a word contained in at least one of the following categories is considered as negative: NEGATIV, NGTV, NEGAFF, PAIN, VICE, HOSTILE, FAIL, ENLLOSS, WLBLOSS, TRANLOSS.

For the (function- and content-word) negator lexicon, we collect a handful of seed words as well as General Inquirer words that appear in either NOTLW or DECREAS category. Then we expand the list of content-negators using the synonym information of WordNet (Miller, 1995) to take a simple vote among senses.

3 Learning-Based Methods

While we expect that a set of hand-written heuristic rules motivated by compositional semantics can be effective for determining the polarity of a sentiment-bearing expression, we do not expect them to be perfect. Interpreting natural language is such a complex task that writing a perfect set of rules would be extremely challenging. Therefore, a more ideal solution would be a learning-based method that can exploit ideas from compositional semantics while providing the flexibility to the rigid application of the heuristic rules. To this end, we present a novel learning-based approach that incorporates inference rules inspired by compositional semantics into the learning procedure (§3.2). To assess the effect of compositional semantics in the learning-based methods, we also experiment with a simple classification approach that does not incorporate compositional semantics (§3.1). The details of these two approaches are elaborated in the following subsections.

3.1 Simple Classification (SC)

Given an expression x consisting of n words x_1, ..., x_n, the task is to determine the polarity $y \in \{positive, negative\}$ of x. In our simple binary classification approach, x is represented as a vector of features $\mathbf{f}(x)$, and the prediction y is given by $\text{argmax}_y \mathbf{w} \cdot \mathbf{f}(x, y)$, where $\mathbf{w}$ is a vector of parameters learned from training data. In our experiment, we use an online SVM algorithm called MIRA (Margin Infused Relaxed Algorithm) (Crammer and Singer, 2003)[7] for training.

For each x, we encode the following features:

- Lexical: We add every word x_i in x, and also add the lemma of x_i produced by the CASS partial parser toolkit (Abney, 1996).
- Dictionary: In order to mitigate the problem of unseen words in the test data, we add features that describe word categories based on the General Inquirer dictionary. We add this feature for each x_i that is not a stop word.
- Vote: We experiment with two variations of voting-related features: for SC-VOTE, we add

based on parse trees might further improve the performance.

[4] Moilanen and Pulman (2007) provide more detailed discussion on the semantic scope of negations and the semantic priorities in resolving polarities.

[5] The majority polarity of the data we use for our experiments is negative.

[6] Available at http://www.wjh.harvard.edu/∼inquirer/. When consulting the General Inquirer dictionary, senses with less than 5% frequency and senses specific to an idiom are dropped.

[7] We use the Java implementation of this algorithm available at http://www.seas.upenn.edu/∼strctlrn/StructLearn/StructLearn.html.

Simple Classification	Classification with Compositional Inference
$y \leftarrow \text{argmax}_y\ \text{score}(y)$ $l \leftarrow \text{loss\_flat}(y^*, y)$ $\mathbf{w} \leftarrow \text{update}(\mathbf{w}, l, y^*, y)$	Find K best $\mathbf{z}$ and denote them as $\mathcal{Z} = \{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(K)}\}$ $\quad s.t. \forall\ i < j,\ \text{score}(\mathbf{z}^{(i)}) > \text{score}(\mathbf{z}^{(j)})$ $\mathbf{z}^{bad} \leftarrow \min_k \mathbf{z}^{(k)}\ s.t.\ \text{loss\_compo}(y^*, \mathbf{z}^{(k)}, x) > 0$ $\quad$(if such $\mathbf{z}^{bad}$ not found in $\mathcal{Z}$, skip parameter update for this.) If $\text{loss\_compo}(y^*, \mathbf{z}^*, x) > 0$ $\quad \mathbf{z}^{good} \leftarrow \min_k \mathbf{z}^{(k)}\ s.t.\ \text{loss\_compo}(y^*, \mathbf{z}^{(k)}, x) = 0$ $\quad z^* \leftarrow \mathbf{z}^{good}$ $\quad$(if such $\mathbf{z}^{good}$ not found in $\mathcal{Z}$, stick to the original z^*.) $l \leftarrow \text{loss\_compo}(y^*, \mathbf{z}^{bad}, x) - \text{loss\_compo}(y^*, \mathbf{z}^*, x)$ $\mathbf{w} \leftarrow \text{update}(\mathbf{w}, l, \mathbf{z}^*, \mathbf{z}^{bad})$
Definitions of score functions and loss functions	
$\text{score}(y) := \mathbf{w} \cdot \mathbf{f}(x, y)$ $\text{loss\_flat}(y^*, y) := \text{if } (y^* = y)\ 0 \text{ else } 1$	$\text{score}(\mathbf{z}) := \sum_i \text{score}(z_i) := \sum_i \mathbf{w} \cdot \mathbf{f}(x, z_i, i)$ $\text{loss\_compo}(y^*, \mathbf{z}, x) := \text{if } (y^* = \mathcal{C}(x, \mathbf{z}))\ 0 \text{ else } 1$

Figure 1: Training procedures. $y^* \in \{positive, negative\}$ denotes the true label for a given expression $x = x_1, ..., x_n$. $\mathbf{z}^*$ denotes the pseudo gold standard for hidden variables $\mathbf{z}$.

a feature that indicates the dominant polarity of words in the given expression, without considering the effect of negators. For SC-NEGEX, we count the number of content-word negators as well as function-word negators to determine whether the final polarity should be flipped. Then we add a conjunctive feature that indicates the dominant polarity together with whether the final polarity should be flipped. For brevity, we refer to SC-VOTE and SC-NEGEX collectively as SC.

Notice that in this simple binary classification setting, it is inherently difficult to capture the compositional structure among words in x, because $\mathbf{f}(x, y)$ is merely a flat bag of features, and the prediction is governed simply by the dot product of $\mathbf{f}(x, y)$ and the parameter vector w.

3.2 Classification with Compositional Inference (CCI)

Next, instead of determining y directly from x, we introduce hidden variables $\mathbf{z} = (z_1, ..., z_n)$ as intermediate decision variables, where $z_i \in \{positive, negative,\ negator, none\}$, so that z_i represents whether x_i is a word with positive/negative polarity, or a negator, or none of the above. For simplicity, we let each intermediate decision variable z_i (a) be determined independently from other intermediate decision variables, and (b) depend only on the input x, so that $z_i = \text{argmax}_{z_i} \mathbf{w} \cdot \mathbf{f}(x, z_i, i)$, where $\mathbf{f}(x, z_i, i)$ is the feature vector encoding around the ith word (described on the next page). Once we determine the intermediate decision variables, we apply the heuristic rules motivated by compositional semantics (from Table 2) in order to obtain the final polarity y of x. That is, $y = \mathcal{C}(x, \mathbf{z})$, where $\mathcal{C}$ is the function that applies the compositional inference, either COMPOPR or COMPOMC.

```
For each token $x_i$,
  if $x_i$ is a word in the negator lexicon
    then $z_i^* \leftarrow negator$
  else if $x_i$ is in the polarity lexicon as negative
    then $z_i^* \leftarrow negative$
  else if $x_i$ is in the polarity lexicon as positive
    then $z_i^* \leftarrow positive$
  else
    then $z_i^* \leftarrow none$
```

Figure 2: Constructing Soft Gold Standard $\mathbf{z}^*$

For training, there are two issues we need to handle: the first issue is dealing with the hidden variables z. Because the structure of compositional inference $\mathcal{C}$ does not allow dynamic programming, it is intractable to perform exact expectation-maximization style training that requires enumerating all possible values of the hidden variables $\mathbf{z}$. Instead, we propose a simple and tractable training rule based on the creation of a *soft* gold standard for $\mathbf{z}$. In particular, we exploit the fact that in our task, we can automatically construct a reasonably accurate gold standard for $\mathbf{z}$, denoted as $\mathbf{z}^*$: as shown in Figure 2, we simply rely on the negator and polarity lexicons. Because $\mathbf{z}^*$ is not always correct, we allow the training procedure to replace $\mathbf{z}^*$ with potentially better assignments as learning proceeds: in the event that the soft gold standard $\mathbf{z}^*$ leads to an incorrect prediction, we search for an assignment that leads to a correct prediction to replace $\mathbf{z}^*$. The exact procedure is given in Figure 1, and will be discussed again shortly.

Figure 1 shows how we modify the parameter update rule of MIRA (Crammer and Singer, 2003) to reflect the aspect of compositional inference. In the event that the soft gold standard $\mathbf{z}^*$ leads to an incorrect prediction, we search for $\mathbf{z}^{good}$, the assignment with highest score that leads to a correct prediction, and replace $\mathbf{z}^*$ with $\mathbf{z}^{good}$. In the event of no such $\mathbf{z}^{good}$ being found among the K-best assignments of $\mathbf{z}$, we stick with $\mathbf{z}^*$.

The second issue is finding the assignment of $\mathbf{z}$ with the highest score$(\mathbf{z}) = \sum_i \mathbf{w} \cdot \mathbf{f}(x, z_i, i)$ that leads to an incorrect prediction $y = \mathcal{C}(x, \mathbf{z})$. Because the structure of compositional inference $\mathcal{C}$ does not allow dynamic programming, finding such an assignment is again intractable. We resort to enumerating only over K-best assignments instead. If none of the K-best assignments of $\mathbf{z}$ leads to an incorrect prediction y, then we skip the training instance for parameter update.

Features. For each x_i in x, we encode the following features:

- Lexical: We include the current word x_i as well as the lemma of x_i produced by CASS partial parser toolkit (Abney, 1996). We also add a boolean feature to indicate whether the current word is a stop word.
- Dictionary: In order to mitigate the problem with unseen words in the test data, we add features that describe word categories based on the General Inquirer dictionary. We add this feature for each x_i that is not a stop word. We also add a number of boolean features that provide following properties of x_i using the polarity lexicon and the negator lexicon:
  - whether x_i is a function-word negator
  - whether x_i is a content-word negator
  - whether x_i is a negator of any kind
  - the polarity of x_i according to Wilson et al. (2005)'s polarity lexicon
  - the polarity of x_i according to the lexicon derived from the General Inquirer dictionary
  - conjunction of the above two features
- Vote: We encode the same vote feature that we use for SC-NEGEX described in § 3.1.

As in the heuristic-based compositional semantics approach (§ 2.2), we experiment with two variations of this learning-based approach: CCI-COMPOPR and CCI-COMPOMC, whose compositional inference rules are COMPOPR and COMPOMC respectively. For brevity, we refer to both variations collectively as CCI-COMPO.

4 Experiments

The experiments below evaluate our heuristic- and learning-based methods for subsentential sentiment analysis (§ 4.1). In addition, we explore the role of context by expanding the boundaries of the sentiment-bearing expressions (§ 4.2).

4.1 Evaluation with given boundaries

For evaluation, we use the Multi-Perspective Question Answering (MPQA) corpus (Wiebe et al., 2005), which consists of 535 newswire documents manually annotated with phrase-level subjectivity information. We evaluate on all strong (i.e., intensity of expression is 'medium' or higher), sentiment-bearing (i.e., polarity is 'positive' or 'negative') expressions.[8] As a result, we can assume the boundaries of the expressions are given. Performance is reported using 10-fold cross-validation on 400 documents; a separate 135 documents were used as a development set. Based on pilot experiments on the development data, we set parameters for MIRA as follows: slack variable to 0.5, and the number of incorrect labels (constraints) for each parameter update to 1. The number of iterations (epochs) for training is set to 1 for simple classification, and to 4

[8] We discard expressions with confidence marked as 'uncertain'.

Heuristic-Based							Learning-Based			
VOTE	NEG (1)	NEG (N)	NEG EX (1)	NEG EX (N)	COMPO MC	COMPO PR	SC VOTE	SC NEG EX	CCI COMPO MC	CCI COMPO PR
86.5	82.0	82.2	87.7	87.7	89.7	89.4	88.5	89.1	90.6	90.7

Table 3: Performance (in accuracy) on MPQA dataset.

Data	Heuristic-Based							Learning-Based			
	VOTE	NEG (1)	NEG (N)	NEG EX (1)	NEG EX (N)	COMPO MC	COMPO PR	SC VOTE	SC NEG EX	CCI COMPO MC	CCI COMPO PR
[-0,+0]	86.5	82.0	82.2	87.7	87.7	89.7	89.4	88.5	89.1	90.6	90.7
[-1,+1]	86.4	81.0	81.2	87.2	87.2	89.3	89.0	88.3	88.4	89.5	89.4
[-5,+5]	85.9	79.0	79.4	85.7	85.6	88.2	88.0	86.4	87.1	88.7	88.7
[-∞,+∞]	85.3	75.8	76.9	83.9	83.9	87.0	86.9	85.8	85.8	87.3	87.5

Table 4: Performance (in accuracy) on MPQA data set with varying boundaries of expressions.

for classification with compositional inference. We use $K = 20$ for classification with compositional inference.

Results. Performance is reported in Table 3. Interestingly, the heuristic-based methods NEG ($\sim$ 82.2%) that only consider function-word negators perform even worse than VOTE (86.5%), which does not consider negators. On the other hand, the NEGEX methods (87.7%) that do consider content-word negators as well as function-word negators perform better than VOTE. This confirms the importance of content-word negators for determining the polarities of expressions. The heuristic-based methods motivated by compositional semantics COMPO further improve the performance over NEGEX, achieving up to 89.7% accuracy. In fact, these heuristics perform even better than the SC learning-based methods ($\sim$ 89.1%). This shows that heuristics that take into account the compositional structure of the expression can perform better than learning-based methods that do not exploit such structure.

Finally, the learning-based methods that incorporate compositional inference CCI-COMPO ($\sim$ 90.7%) perform better than all of the previous methods. The difference between CCI-COMPOPR (90.7%) and SC-NEGEX (89.1%) is statistically significant at the .05 level by paired t-test. The difference between COMPO and any other heuristic that is not based on computational semantics is also statistically significant. In addition, the difference between CCICOMPOPR (learning-based) and COMPOMC (non-learning-based) is statistically significant, as is the difference between NEGEX and VOTE.

4.2 Evaluation with noisy boundaries

One might wonder whether employing additional context outside the annotated expression boundaries could further improve the performance. Indeed, conventional wisdom would say that it is necessary to employ such contextual information (e.g., Wilson et al. (2005)). In any case, it is important to determine whether our results will apply to more real-world settings where human-annotated expression boundaries are not available.

To address these questions, we gradually relax our previous assumption that the exact boundaries of expressions are given: for each annotation boundary, we expand the boundary by x words for each direction, up to sentence boundaries, where $x \in \{1, 5, \infty\}$. We stop expanding the boundary if it will collide with the boundary of an expression with a different polarity, so that we can consistently recover the expression-level gold standard for evaluation. This expansion is applied to both the training and test data, and the performance is reported in Table 4. From this experiment, we make the following observations:

- Expanding the boundaries hurts the performance for any method. This shows that most of relevant context for judging the polarity is contained within the expression boundaries, and motivates the task of finding the boundaries of opinion expressions.

- The NEGEX methods perform better than VOTE only when the expression boundaries are reasonably accurate. When the expression boundaries are expanded up to sentence boundaries, they perform worse than VOTE. We conjecture this is because the scope of negators tends to be limited to inside of expression boundaries.

- The COMPO methods always perform better than any other heuristic-based methods. And their performance does not decrease as steeply as the NEGEX methods as the expression boundaries expand. We conjecture this is because methods based on compositional semantics can handle the scope of negators more adequately.

- Among the learning-based methods, those that involve compositional inference (CCI-COMPO) always perform better than those that do not (SC) for any boundaries. And learning with compositional inference tend to perform better than the rigid application of heuristic rules (COMPO), although the relative performance gain decreases once the boundaries are relaxed.

5 Related Work

The task focused on in this paper is similar to that of Wilson et al. (2005) in that the general goal of the task is to determine the polarity in context at a sub-sentence level. However, Wilson et al. (2005) formulated the task differently by limiting their evaluation to individual words that appear in their polarity lexicon. Also, their approach was based on a flat bag of features, and only a few examples of what we call content-word negators were employed.

Our use of compositional semantics for the task of polarity classification is preceded by Moilanen and Pulman (2007), but our work differs in that we integrate the key idea of compositional semantics into learning-based methods, and that we perform empirical comparisons among reasonable alternative approaches. For comparison, we evaluated our approaches on the polarity classification task from SemEval-07 (Strapparava and Mihalcea, 2007). We achieve 88.6% accuracy with COMPOPR, 90.1% with SCNEGEX, and 87.6% with CCICOMPOMC.[9] There are a number of possible reasons for our lower performance vs. Moilanen and Pulman (2007) on this data set. First, SemEval-07 does not include a training data set for this task, so we use 400 documents from the MPQA corpus instead. In addition, the SemEval-07 data is very different from the MPQA data in that (1) the polarity annotation is given only at the sentence level, (2) the sentences are shorter, with simpler structure, and not as many negators as the MPQA sentences, and (3) there are many more instances with positive polarity than in the MPQA corpus.

Nairn et al. (2006) also employ a “polarity” propagation algorithm in their approach to the semantic interpretation of implicatives. However, their notion of polarity is quite different from that assumed here and in the literature on sentiment analysis. In particular, it refers to the degree of “commitment” of the author to the truth or falsity of a complement clause for a textual entailment task.

McDonald et al. (2007) use a structured model to determine the sentence-level polarity and the document-level polarity simultaneously. But decisions at each sentence level does not consider structural inference within the sentence.

Among the studies that examined content-word negators, Niu et al. (2005) manually collected a small set of such words (referred as “words that change phases”), but their lexicon was designed mainly for the medical domain and the type of negators was rather limited. Wilson et al. (2005) also manually collected a handful of content-word negators (referred as “general polarity shifters”), but not extensively. Moilanen and Pulman (2007) collected a more extensive set of negators semi-automatically using WordNet 2.1, but the empirical effect of such words was not explicitly investigated.

[9] For lack of space, we only report our performance on instances with strong intensities as defined in Moilanen and Pulman (2007), which amounts to only 208 test instances. The cross-validation set of MPQA contains $4.9k$ instances.

6 Conclusion

In this paper, we consider the task of determining the polarity of a sentiment-bearing expression, considering the effect of interactions among words or constituents in light of compositional semantics. We presented a novel learning-based approach that incorporates structural inference motivated by compositional semantics into the learning procedure. Our approach can be considered as a small step toward bridging the gap between computational semantics and machine learning methods. Our experimental results suggest that this direction of research is promising. Future research includes an approach that learns the compositional inference rules from data.

Acknowledgments

This work was supported in part by National Science Foundation Grants BCS-0624277 and IIS-0535099 and by Department of Homeland Security Grant N0014-07-1-0152. We also thank Eric Breck, Lillian Lee, Mats Rooth, the members of the Cornell NLP reading seminar, and the EMNLP reviewers for insightful comments on the submitted version of the paper.

References

Steven Abney. 1996. Partial parsing via finite-state cascades. *Journal of Natural Language Engineering*, 2(4):337344.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *JMLR* 3:951.

David R. Dowty, Robert E. Wall and Stanley Peters. 1981. *Introduction to Montague Semantics.*

Andrea Esuli and Fabrizio Sebastiani. 2006. SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In *Proceedings of 5th Conference on Language Resources and Evaluation (LREC),.*

Percy Liang, Hal Daumé III and Dan Klein. 2008. Structure Compilation: Trading Structure for Features. In *International Conference on Machine Learning.*

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2004).*

Alistair Kennedy and Diana Inkpen. 2005. Sentiment Classification of Movie and Product Reviews Using Contextual Valence Shifters. In *Proceedings of FINEXIN 2005, Workshop on the Analysis of Informal and Formal Information Exchange during Negotiations.*

Soo-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of COLING.*

Ryan McDonald, Kerry Hannan, Tyler Neylon, Mike Wells and Jeff Reynar. 2007. Structured Models for Fine-to-Coarse Sentiment Analysis. In *Proceedings of Association for Computational Linguistics (ACL)* .

George A. Miller. 1995. WordNet: a lexical database for English. In *Communications of the ACM*, 38(11):3941

Richard Montague. 1974. Formal Philosophy; Selected papers of Richard Montague. Yale University Press.

Karo Moilanen and Stephen Pulman. 2007. Sentiment Composition. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2007).*

Rowan Nairn, Cleo Condoravdi and Lauri Karttunen 2006. Computing relative polarity for textual inference. In *Inference in Computational Semantics (ICoS-5).*

Yun Niu, Xiaodan Zhu, Jianhua Li and Graeme Hirst. 2005. Analysis of polarity information in medical text. In *Proceedings of the American Medical Informatics Association 2005 Annual Symposium (AMIA).*

Livia Polanyi and Annie Zaenen. 2004. Contextual lexical valence shifters. In *Exploring Attitude and Affect in Text: Theories and Applications: Papers from the 2004 Spring Symposium, AAAI.*

Mostafa Shaikh, Helmut Prendinger and Mitsuru Ishizuka. 2007. Assessing sentiment of text by semantic dependency and contextual valence analysis. In *Proc 2nd Int'l Conf on Affective Computing and Intelligent Interaction (ACII'07).*

Carlo Strapparava and Rada Mihalcea. 2007. Semeval-2007 task 14: Affective text. In *Proceedings of SemEval.*

Janyce Wiebe, Theresa Wilson and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. In *Language Resources and Evaluation (formerly Computers and the Humanities), 39(2-3):165210.*

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