Joint Modeling of Opinion Expression Extraction and Attribute Classification

Bishan Yang
Department of Computer Science
Cornell University
bishan@cs.cornell.edu

Claire Cardie
Department of Computer Science
Cornell University
cardie@cs.cornell.edu

Abstract

In this paper, we study the problems of opinion expression extraction and expression-level polarity and intensity classification. Traditional fine-grained opinion analysis systems address these problems in isolation and thus cannot capture interactions among the textual spans of opinion expressions and their opinion-related properties. We present two types of joint approaches that can account for such interactions during 1) both learning and inference or 2) only during inference. Extensive experiments on a standard dataset demonstrate that our approaches provide substantial improvements over previously published results. By analyzing the results, we gain some insight into the advantages of different joint models.

1 Introduction

Automatic extraction of opinions from text has attracted considerable attention in recent years. In particular, significant research has focused on extracting detailed information for opinions at the fine-grained level, e.g. identifying opinion expressions within a sentence and predicting phrase-level polarity and intensity. The ability to extract fine-grained opinion information is crucial in supporting many opinion-mining applications such as opinion summarization, opinion-oriented question answering and opinion retrieval.

In this paper, we focus on the problem of identifying opinion expressions and classifying their attributes. We consider as an opinion expression any subjective expression that explicitly or implicitly conveys emotions, sentiment, beliefs, opinions (i.e. private states) (Wiebe et al., 2005), and consider two key attributes — polarity and intensity — for characterizing the opinions. Consider the sentence in Figure 1, for example. The phrases "a bias in favor of" and "being severely criticized" are opinion expressions containing positive sentiment with medium intensity and negative sentiment with high intensity, respectively.

Most existing approaches tackle the tasks of opinion expression extraction and attribute classification in isolation. The first task is typically formulated as a sequence labeling problem, where the goal is to label the boundaries of text spans that correspond to opinion expressions (Breck et al., 2007; Yang and Cardie, 2012). The second task is usually treated as a binary or multi-class classification problem (Wilson et al., 2005; Choi and Cardie, 2008; Yessenalina and Cardie, 2011), where the goal is to assign a class label to a text fragment (e.g. a phrase or a sentence). Solutions to the two tasks can be applied in a pipeline architecture to extract opinion expressions and their attributes. However, pipeline systems suffer from error propagation: opinion expression errors propagate and lead to unrecoverable errors in attribute classification.

Limited work has been done on the joint modeling of opinion expression extraction and attribute classification. Choi and Cardie (2010) first proposed a joint sequence labeling approach to extract opinion expressions and label them with polarity and intensity. Their approach treats both expression extraction and attribute classification as token-level se-

Transactions of the Association for Computational Linguistics, vol. 2, pp. 505–516, 2014. Action Editor: Janyce Wiebe.
Submitted 4/2014; Revised 8/2014; Published November 1, 2014.

He demonstrated a bias in favor of $_{medium}$ the rebels despite being severely criticized $_{high}$.

Figure 1: An example sentence annotated with opinion expressions and their polarity and intensity. We use colored boxes to mark the textual spans of opinion expressions where green (red) denotes positive (negative) polarity, and use subscripts to denote intensity.

quence labeling tasks, and thus cannot model the label distribution over expressions even though the annotations are given at the expression level. Johansson and Moschitti (2011) considered a pipeline of opinion extraction followed by polarity classification and propose re-ranking its k-best outputs using global features. One key issue, however, is that the approach enumerates the k-best output in a pipeline manner and thus they do not necessarily correspond to the k-best global decisions. Moreover, as the number of opinion attributes grows, it is not clear how to identify the best k for each attribute.

In contrast to existing approaches, we formulate opinion expression extraction as a segmentation problem and attribute classification as segment-level attribute labeling. To capture their interactions, we present two types of joint approaches: (1) *joint learning* approaches, which combine opinion segment detection and attribute labeling into a single probabilistic model, and estimate parameters for this joint model; and (2) *joint inference* approaches, which build separate models for opinion segment detection and attribute labeling at training time, and jointly apply these (via a single objective function) only at test time to identify the best "combined" decision of the two models.

To investigate the effectiveness of our approaches, we conducted extensive experiments on a standard corpus for fine-grained opinion analysis (the MPQA corpus (Wiebe et al., 2005)). We found that all of our proposed approaches provide substantial improvements over the previously published results. We also compared our approaches to a strong pipeline baseline and observed that joint learning results in a significant boost in precision while joint inference, with an appropriate objective, can significantly boost both precision and recall and obtain the best overall performance. Error analysis provides additional understanding of the differences between the joint learning and joint inference approaches, and suggests that joint inference can be more effective and more efficient for the task in practice.

2 Related Work

Significant research effort has been invested in the task of fine-grained opinion analysis in recent years (Wiebe et al., 2005; Wilson et al., 2009). Wilson et al. (2005) first motivated and studied phrase-level polarity classification on an open-domain corpus. Choi and Cardie (2008) developed inference rules to capture compositional effects at the lexical level on phrase-level polarity classification. Yessenalina and Cardie (2011) and Socher et al. (2013) learn continuous-valued phrase representations by combining the representations of words within an opinion expression and using them as features for classifying polarity and intensity. All of these approaches assume the opinion expressions are available before training the classifiers. However, in real-world settings, the spans of opinion expressions within the sentence are not available. In fact, Choi and Cardie (2008) demonstrated that the performance of expression-level polarity classification degrades as more surrounding (but irrelevant) context is considered. This motivates the additional task of identifying the spans of opinion expressions.

Opinion expression extraction has been successfully tackled via sequence tagging methods. Breck et al. (2007) applied conditional random fields to assign each token a label indicating whether it belongs to an opinion expression or not. Yang and Cardie (2012) employed a segment-level sequence labeler based on semi-CRFs with rich phrase-level syntactic features. In this work, we also utilize semi-CRFs to model opinion expression extraction.

There has been limited work on the joint modeling of opinion expression extraction and attribute classification. Choi and Cardie (2010) first developed a joint sequence labeler that jointly tags opinions, polarity and intensity by training CRFs with hierarchical features (Zhao et al., 2008). One major drawback of their approach is that it models both opinion extraction and attribute labeling as tasks in token-level sequence labeling, and thus cannot model their inter-

actions at the expression-level. Johansson and Moschitti (2011) and Johansson and Moschitti (2013) propose a joint approach to opinion expression extraction and polarity classification by re-ranking its k-best output using global features. One major issue with their approach is that the k-best candidates were obtained without global reasoning about the relative uncertainty in the individual stages. As the number of considered attributes grows, it also becomes harder to decide how many predictions to select from each attribute classifier.

Compared to the existing approaches, our joint models have the advantage of modeling opinion expression extraction and attribute classification at the segment-level, and more importantly, they provide a principled way of combining the segmentation and classification components.

Our work follows a long line of joint modeling research that has demonstrated great success for various NLP tasks (Roth and Yih, 2004; Punyakanok et al., 2004; Finkel and Manning, 2010; Rush et al., 2010; Choi et al., 2006; Yang and Cardie, 2013). Methods tend to fall into one of two joint modeling frameworks: the first learns a joint model that captures global dependencies; the other uses independently-learned models and considers global dependencies only during inference. In this work, we study both types of joint approaches for opinion expression extraction and opinion attribute classification.

3 Approach

In this section, we present our approaches for the joint modeling of opinion expression extraction and attribute classification. Specifically, given a sentence, our goal is to identify the spans of opinion expressions, and simultaneously assign their polarity and intensity. Training data consists of a collection of sentences with manually annotated opinion expression spans, each associated with a polarity label that takes values from {*positive, negative, neutral*}, and an intensity label, taking values from {*high, medium, low*}.

In the following, we first describe how we model opinion expression extraction as a segment-level sequence labeling problem and model attribute prediction as a classification problem. Then we propose our joint models for combining opinion segmentation and attribute classification.

3.1 Opinion Expression Extraction

The problem of opinion expression extraction assumes tokenized sentences as input and outputs the spans of the opinion expressions in each sentence. Previous work has tackled this problem using token-based sequence labeling methods such as CRFs (e.g. Breck et al. (2007), Yang and Cardie (2012)). However, semi-Markov CRFs (Sarawagi and Cohen, 2004) (henceforth *semi-CRF*) have been shown more appropriate for the task than CRFs since they allow contiguous spans in the input sequence (e.g. a noun phrase) to be treated as a group rather than as distinct tokens. Thus, they can easily capture segment-level information like syntactic constituent structure (Yang and Cardie, 2012). Therefore we adopt the semi-CRF model for opinion expression extraction here.

Given a sentence $\mathbf{x}$, denote an opinion segmentation as $\mathbf{y_s} = \langle (s_0, b_0), ..., (s_k, b_k) \rangle$, where the $s_{0:k}$ are consecutive segments that form a segmentation of $\mathbf{x}$; each segment $s_i = (t_i, u_i)$ consists of the positions of the start token t_i and an end token u_i; and each s_i is associated with a binary variable $b_i \in \{I, O\}$, which indicates whether it is an opinion expression (I) or not (O). Take the sentence in Figure 1, for example. The corresponding opinion segmentation is $\mathbf{y_s} = \langle ((0,0), O), ((1,1), O), ((2,6), I), ((7,8), O), ((9,9), O), ((10,12), I), ((13,13), O) \rangle$, where each segment corresponds to an opinion expression or to a phrase unit that does not express any opinion.

Using a semi-Markov CRF, we model the conditional distribution over all possible opinion segmentations given the input $\mathbf{x}$:

$$P(\mathbf{y_s}|\mathbf{x}) = \frac{\exp\{\sum_{i=1}^{|\mathbf{y_s}|} \theta \cdot f(y_{s_i}, y_{s_{i-1}}, \mathbf{x})\}}{\sum_{\mathbf{y'_s} \in \mathcal{Y}} \exp\{\sum_{i=1}^{|\mathbf{y'_s}|} \theta \cdot f(y'_{s_i}, y'_{s_{i-1}}, \mathbf{x})\}} \quad (1)$$

where θ denotes the model parameters, $y_{s_i} = (s_i, b_i)$ and f denotes a feature function that encodes the potentials of the boundaries for opinion segments and the potentials of transitions between two consecutive labeled segments.

Note that the probability is normalized over all possible opinion segmentations. To reduce the training complexity, we adopted the method described in Yang and Cardie (2012), which only normalizes over segment candidates that are plausible according to the parsing structure of the sentence. Figure 2 shows some candidate segmentations generated for an example sentence. Such a technique results in a large reduction in training time and was shown to be effective for identifying opinion expressions.

The standard training objective of a semi-CRF, is to minimize the log loss

$$L(\theta) = \arg\min_{\theta} - \sum_{i=1}^{N} \log P(\mathbf{y_s^{(i)}} | \mathbf{x}^{(i)}) \quad (2)$$

It penalizes any predicted opinion expression whose boundaries do not exactly align with the boundaries of the correct opinion expressions using 0-1 loss. Unfortunately, exact boundary matching is often not used as an evaluation metric for opinion expression extraction since it is hard for human annotators to agree on the exact boundaries of opinion expressions.[1] Most previous work used *proportional matching* (Johansson and Moschitti, 2013) as it takes into account the overlapping proportion of the predicted and the correct opinion expressions to compute precision and recall. To incorporate this evaluation metric into training, we use softmax-margin (Gimpel and Smith, 2010) that replace $P(\mathbf{y_s^{(i)}}|\mathbf{x}^{(i)})$ in (2) with $P_{cost}(\mathbf{y_s^{(i)}}|\mathbf{x}^{(i)})$, which equals

$$\frac{\exp\{\sum_{i=1}^{|\mathbf{y_s}|} \theta \cdot f(y_{s_i}, y_{s_{i-1}}, \mathbf{x})\}}{\sum_{\mathbf{y'_s} \in \mathcal{Y}} \exp\{\sum_{i=1}^{|\mathbf{y'_s}|} \theta \cdot f(y'_{s_i}, y'_{s_{i-1}}, \mathbf{x}) + l(\mathbf{y'_s}, \mathbf{y_s})\}}$$

and we define the loss function $l(\mathbf{y'_s}, \mathbf{y_s})$ as

$$\sum_{i=1}^{|\mathbf{y'_s}|} \sum_{j=1}^{|\mathbf{y_s}|} (\mathbb{1}\{b'_i \neq b_j \wedge b'_i \neq O\} \frac{|s_j \cap s'_i|}{|s'_i|} + \mathbb{1}\{b'_i \neq b_j \wedge b_j \neq O\} \frac{|s_j \cap s'_i|}{|s_j|})$$

which is the sum of the precision and recall errors of segment labeling using proportional matching. The loss-augmented probability is only computed during training. The more the proposed labeled segmentation overlaps with the true labeled segmentation for $\mathbf{x}$, the less it will be penalized.

[1] The inter-annotator agreement on boundaries of opinion expressions is not stressed in MPQA (Wiebe et al., 2005).

```
We hope to eradicate the eternal scourge of corruption .
[  ][   ][ ][        ][              ][ ][        ][ ]
[  ][   ][ ][        ][              ][           ][ ]
[  ][   ][ ][                        ][           ][ ]
[  ][   ][ ][                                     ][ ]
```

Figure 2: Examples of Segmentation Candidates

During inference, we can obtain the best labeled segmentation by solving

$$\underset{\mathbf{y_s}}{\text{argmax}} P(\mathbf{y_s}|\mathbf{x}) = \underset{\mathbf{y_s}}{\text{argmax}} \sum_{i=1}^{|\mathbf{y_s}|} \theta \cdot f(y_{s_i}, y_{s_{i-1}}, \mathbf{x})$$

This can be done efficiently via dynamic programming:

$$V(t) = \underset{s=(u,t)\in s_{:t}, y=(s,b), y'}{\text{argmax}} G(y, y') + V(u-1) \quad (3)$$

where $s_{:t}$ denotes all candidate segments ending at position t and $G(y, y') = \theta \cdot f(y, y', \mathbf{x})$. The optimal $\mathbf{y_s}^*$ can be obtained by computing $V(n)$, where n is the length of the sentence.

3.2 Opinion Attribute Classification

We consider two types of opinion attributes: *polarity* and *intensity*. For each attribute, we model the multinomial distribution of an attribute class given a text segment Denoting the class variable for each attribute as a^j, we have

$$P(a^j|\mathbf{x}_s) = \frac{\exp\{\phi_j \cdot g_j(a^j, \mathbf{x}_s)\}}{\sum_{a' \in \mathcal{A}_j} \exp\{\phi_j \cdot g_j(a', \mathbf{x}_s)\}} \quad (4)$$

where x_s denotes a text segment, ϕ_j is a parameter vector and g_j denotes feature functions for attribute a^j. The label space for polarity classification is $\{positive, negative, neutral, \varnothing\}$ and the label space for intensity classification is $\{high, medium, low, \varnothing\}$. We include an empty value $\varnothing$ to denote assigning no attribute value to those text segments that are not opinion expressions.

In the following description of our joint models, we omit the superscript on the attribute variable and derive our models with one single opinion attribute for simplicity. The derivations can be carried through with more than one opinion attribute by assuming the independence of different attributes.

3.3 The Joint Models

We propose two types of joint models for opinion segmentation and attribute classification: (1) *joint learning* models, which train a single sequence labeling model that maximizes a joint probability distribution over segmentation and attribute labeling, and infers the most probable labeled segmentations according to the joint probability; and (2) *joint inference* models, which train a sequence labeling model for opinion segmentation and separately train classification models for attribute labeling, and combine the segmentation and classification models during inference to make global decisions. In the following, we first present the joint learning models and then introduce the joint inference models.

3.3.1 Joint Sequence Labeling

We can formulate joint opinion segmentation and classification as a sequence labeling problem on the label space $\mathcal{Y} = \{\mathbf{y}|\mathbf{y} = \langle(s_0, \tilde{b}_0), ..., (s_k, \tilde{b}_k)\rangle\}$ where $\tilde{b}_i = (b_i, a_i) \in \{I, O\} \times \mathcal{A}$, where b_i is a binary variable as described before and a_i is an attribute class variable associated with segment s_i. Since only opinion expressions should be assigned opinion attributes, we consider the following labeling constraints: $a_i = \varnothing$ if and only if $b_i = O$.

We can apply the same training and inference procedure described in Section 3.1 by replacing the label space $\mathbf{y_s}$ with the joint label space $\mathbf{y}$. Note that the feature functions are shared over the joint label space. For the loss function in the loss-augmented objective, the opinion segment label b is also replaced with the augmented label $\tilde{b}$.

3.3.2 Hierarchical Joint Sequence Labeling

The above joint sequence labeling model does not explicitly model the dependencies between opinion segmentation and attribute labeling. The two subtasks share the same set of features and parameters. In the following, we introduce an alternative approach that explicitly models the conditional dependency between opinion segmentation and attribute labeling, and allows segmentation- and attribute-specific parameters to be jointly learned in one single model.

Note that the joint label space naturally forms a hierarchical structure: the probability of choosing a sequence label $\mathbf{y}$ can be interpreted as the probability of first choosing an opinion segmentation $\mathbf{y_s} = \langle(s_0, b_0), ..., (s_k, b_k)\rangle$ given the input $\mathbf{x}$, and then choose a sequence of attribute labels $\mathbf{y_a} = \langle a_0, ..., a_k\rangle$ given the chosen segment sequence. Following this intuition, the joint probability can be decomposed as

$$P(\mathbf{y}|\mathbf{x}) = P(\mathbf{y_s}|\mathbf{x})P(\mathbf{y_a}|\mathbf{y_s}, \mathbf{x})$$

where $P(\mathbf{y_s}|\mathbf{x})$ is modeled as Equation (1) and

$$P(\mathbf{y_a}|\mathbf{y_s}, \mathbf{x}) = \prod_{i=1}^{|\mathbf{y_s}|} P(a_i|y_{s_i}, \mathbf{x}) \propto \exp\{\sum_{i=1}^{|\mathbf{y_s}|} \phi \cdot g(a_i, y_{s_i}, \mathbf{x})\}$$

where g denotes a feature function that encodes attribute-specific information for discriminating different attribute classes for each segment.

For training, we can also apply a softmax-margin by adding a loss function $l(\mathbf{y}', \mathbf{y})$ to the denominator of $P(\mathbf{y}|\mathbf{x})$ (as in the basic joint sequence labeling model described in Section 3.3.1).

With the estimated parameters, we can infer the optimal opinion segmentation and attribute labeling by solving

$$\underset{\mathbf{y_s}, \mathbf{y_a}}{\text{argmax}} P(\mathbf{y_s}|\mathbf{x})P(\mathbf{y_a}|\mathbf{y_s}, \mathbf{x})$$

We can apply a similar dynamic programming procedure by replaceing y in Equation (3) with $y = (s, b, a)$ and $G(y, y')$ with $\theta \cdot f(y, y', \mathbf{x}) + \phi \cdot g(y, \mathbf{x})$.

Our decomposition of labels and features is similar to the hierarchical construction of CRF features in Choi and Cardie (2010). The difference is that our model is based on semi-CRFs and the decomposition is based on a joint probability. We will show that this results in better performance than the methods in Choi and Cardie (2010) in our experiments.

3.3.3 Joint Inference

Modeling the joint probability of opinion segmentation and attribute labeling is arguably elegant. However, training can be expensive as the computation involves normalizing over all possible segmentations and all possible attribute labelings for

each segment. Thus, we also investigate joint inference approaches which combine the separately-trained models during inference without computing the normalization term.

For opinion segmentation, we train a semi-CRF-based model using the approach described in Section 1. For attribute classification, we train a MaxEnt model by maximizing $P(a^j|\mathbf{x}_s)$ in Equation (4). As we only need to estimate the probability of an attribute label given individual text segments, the training data can be constructed by collecting a list of text segments labeled with correct attribute labels. The text segments do not need to form all possible sentence segmentations. To construct such training examples, we collected from each sentence all opinion expressions labeled with their corresponding attributes and use the remaining text segments as examples for the empty attribute value. The training of the MaxEnt model is much more efficient than the training of the segmentation model.

Joint Inference with Probability-based Estimates To combine the separately-trained models at inference time, a natural inference objective is to jointly maximize the probability of opinion segmentation and the probability of attribute labeling given the chosen segmentation

$$\underset{\mathbf{y_s},\mathbf{y_a}}{\operatorname{argmax}} P(\mathbf{y_s}|\mathbf{x})P'(\mathbf{y_a}|\mathbf{y_s},\mathbf{x}) \quad (5)$$

We approximate the conditional probability as

$$P'(\mathbf{y_a}|\mathbf{y_s},\mathbf{x}) = \prod_{i=1}^{|\mathbf{y_s}|} P(a_i|\mathbf{x}_{s_i})^{\alpha} \quad (6)$$

where $\alpha \in (0, 1]$. We found that $\alpha < 1$ provides better performance than $\alpha = 1$ empirically. This is an approximation since the distribution of attribute labeling is estimated independently from the opinion segmentation during training.

Joint Inference with Loss-based Estimates Instead of directly using the output probabilities of the attribute classifiers, we explore an alternative that estimates $P'(\mathbf{y_a}|\mathbf{y_s},\mathbf{x})$ based on the prediction uncertainty:

$$P'(\mathbf{y_a}|\mathbf{y_s},\mathbf{x}) \propto \exp(-\alpha \sum_{i=1}^{|\mathbf{y_s}|} U(a_i|\mathbf{x}_{s_i})) \quad (7)$$

where $U(a_i|\mathbf{x}_{s_i})$ is a uncertainty function that measures the classification model's uncertainty in its assignment of attribute class a_i to segment $\mathbf{x}_{s_i}$. Intuitively, we want to penalize attribute assignments that are uncertain or favor attribute assignments with low uncertainty. The prediction uncertainty is measured using the expected loss. The expected loss for a predicted label a' can be written as

$$E_{a|\mathbf{x}_{s_i}}[l(a, a')] = \sum_a P(a|\mathbf{x}_{s_i})l(a, a')$$

where $l(a, a')$ is a loss function over a' and the true label a. We used the standard 0-1 loss function in our experiments[2] and set $U(a_i|\mathbf{x}_{s_i}) = \log(E_{a|\mathbf{x}_{s_i}}[l(a, a_i)])$.

Both joint inference objectives can be solved efficiently via dynamic programming.

4 Features

We consider a set of basic features as well as task-specific features for opinion segmentation and attribute labeling, respectively.

4.1 Basic Features

Unigrams: word unigrams and POS tag unigrams for all tokens in the segment candidate.
Bigrams: word bigrams and POS bigrams within the segment candidate.
Phrase embeddings: for each segment candidate, we associate with it a 300-dimensional phrase embedding as a dense feature representation for the segment. We make use of the recently published word embeddings trained on Google News (Mikolov et al., 2013). For each segment, we compute the average of the word embedding vectors that comprise the phrase. We omit words that are not found in the vocabulary. If no words are found in the text segment, we assign a feature vector of zeros.
Opinion lexicon: For each word in the segment candidate, we include its polarity and intensity as indicated in an existing Subjectivity Lexicon (Wilson et al., 2005).

[2]The loss function can be tuned to better tradeoff precision and recall according to the applications at hand. We did not explore this option in this paper.

4.2 Segmentation-specific Features

Boundary words and POS tags: word-level features (words, POS, lexicon) before and after the segment candidate.
Phrase structure: the syntactic categories of the deepest constituents that cover the segment in the parse tree, e.g. NP, VP, TO_VB.
VP patterns: VP-related syntactic patterns described in Yang and Cardie (2012), e.g. VPsubj, VParg, which have been shown useful for opinion expression extraction.

4.3 Polarity-specific Features

Polarity count: counts of positive, negative and neutral words within the segment candidate according to the opinion lexicon.
Negation: indicator for negators within the segment candidate.

4.4 Intensity-specific Features

Intensity count: counts of words with strong and weak intensity within the segment candidate according to the opinion lexicon.
Intensity dictionary: As suggested in Choi and Cardie (2010), we include features indicating whether the segment contains an intensifier (e.g. highly, really), a diminisher (e.g. little, less), a strong modal verb (e.g. must, will), and a weak modal verb (e.g. may, could).

5 Experiments

All our experiments were conducted on the MPQA corpus (Wiebe et al., 2005), a widely used corpus for fine-grained opinion analysis. We used the same evaluation setting as in Choi and Cardie (2010), where 135 documents were used for development and 10-fold cross-validation was performed on a different set of 400 documents. Each training fold consists of sentences labeled with opinion expression boundaries and each expression is labeled with polarity and intensity. Table 1 shows some statistics of the evaluation data.

We used precision, recall and F1 as evaluation metrics for opinion extraction and computed them using both *proportional matching* and *binary matching* criteria. *Proportional matching* considers the overlapping proportion of a predicted expression s and a gold standard expression s^*, and computes precision as $\sum_{s\in S}\sum_{s^*\in S^*}\frac{|s\cap s^*|}{|s|}/|S|$ and recall as $\sum_{s\in S}\sum_{s^*\in S^*}\frac{|s\cap s^*|}{|s^*|}/|S^*|$, where S and S^* denote the set of predicted opinion expressions and the set of correct opinion expressions, respectively. *Binary matching* is a more relaxed metric that considers a predicted opinion expression to be correct if it overlaps with a correct opinion expression.

We experimented with the following models:

(1) PIPELINE: first extracts the spans of opinion expressions using the semi-CRF model in Section 3.1, and then assigns polarity and intensity to the extracted opinion expressions using MaxEnt models in Section 3.2. Note that the label space of the MaxEnt models does not include $\varnothing$ since they assume that all the opinion expressions extracted by the previous stage are correct.

(2) JSL: the joint sequence labeling method described in Section 3.3.1.

(3) HJSL: the hierarchical joint sequence labeling method described in Section 3.3.2.

(4) JI-PROB: the joint inference method using probability-based estimates (Equation 6).

(5) JI-LOSS: the joint inference method using loss-based estimates (Equation 7).

We also compare our results with previously published results from Choi and Cardie (2010) on the same task.

All our models are log linear models. We use L-BFGS with L2 regularization for training and set the regularization parameter to 1.0. We set the scaling parameter α in JI-PROB and JI-LOSS via grid search over values between 0.1 and 1 with increments of 0.1 using the development set.

We consider the same set of features described in Section 4 in all the models. For the pipeline and joint inference models where the opinion segmentator and attribute classifiers are separately trained, we employ basic features plus segmentation-specific features in the opinion segmentator; and employ basic features plus attribute-specific features in the attribute classifiers.

5.1 Results

We would like to first investigate how much we can gain from using the loss-augmented training compared to using the standard training objective. Loss-

Number of Opinion Expressions		
Positive	Negative	Neutral
2170	4863	6368
High	Medium	Low
2805	5721	4875

Number of Documents	400
Number of Sentences	8241
Average Length of Opinion Expressions	2.86 words

Table 1: Statistics of the evaluation corpus

augmented training can be applied to the training of the opinion segmentation model used in the pipeline method and the joint inference methods, or be applied to the training of the joint sequence labeling approaches, JSL and HJSL (the loss function takes into account both the span overlap and the matching of attribute values). We evaluate two versions of each method: one uses loss-augmented training and one uses standard log-loss training. Table 2 shows the results of opinion expression detection without evaluating their attributes. Similar trends can be observed in the results of opinion expression detection with respect to each attribute. We can see that incorporating the evaluation-metric-based loss function during training consistently improves the performance for all models in terms of F1 measure. This confirms the effectiveness of loss-augmented training of our sequence models for opinion extraction. As a result, all following results are based on the loss-augmented version of our models.

Comparing the results of different models in Table 2, we can see that PIPELINE provides a strong baseline. In comparison, JSL and HJSL significantly improve precision but fail in recall, which indicates that joint sequence labeling is more conservative and precision-biased for extracting opinion expressions. HJSL significantly outperforms JSL, and this confirms the benefit of modeling the conditional dependency between opinion segmentation and attribute classification. In addition, we see that combining opinion segmentation and attribute classification without joint training (JI-PROB and JI-LOSS) hurt precision but improves recall (vs. JSL and HJSL). JI-LOSS presents the best F1 performance and significantly outperforms the PIPELINE baseline in all evaluation metrics. This suggests that JI-LOSS provides an effective joint inference objective and is able to provide more balanced precision and recall than other joint approaches.

Table 3 shows the performance on opinion extraction with respect to polarity and intensity attributes. Similarly, we can see that JI-LOSS outperforms all other baselines in F1; HJSL outperforms JSL but is slightly worse than PIPELINE in F1; JI-PROB is recall-oriented and less effective than JI-LOSS.

We hypothesize that the worse performance of joint sequence labeling is due to its strong assumption on the dependencies between opinion segmentation and attribute labeling in the training data. For example, the expression "fundamentally unfair and unjust" as a whole is labeled as an opinion expression with negative polarity. However, the subexpression "unjust" can be also viewed as a negative expression but it is not annotated as an opinion expression in this example (as MPQA does not consider nested opinion expressions). As a result, the model would wrongly prefer an empty attribute to the expression "unjust". However, in our joint inference approaches, the attribute classification models are trained independently from the segmentation model, and the training examples for the classifiers only consist of correctly labeled expressions ("unjust" as a nested opinion expression in this example would not be considered in the training data for the attribute classifier). Therefore, the joint inference approaches do not suffer from this issue. Although joint inference does not account for task dependencies during training, the promising performance of JI-LOSS demonstrates that modeling label dependencies during inference can be more effective than the PIPELINE baseline.

In Table 3, we can see that the improvement of JI-LOSS is less significant in the *positive* class and the *high* class. This is due to the lack of training data in these classes. The improvement in the *medium* class is also less significant. This may be because it is inherently harder to disambiguate *medium* from *low*. In general, we observe that extracting opinion expressions with correct intensity is a harder task than extracting opinion expressions with correct polarity.

Table 4 presents the F1 scores (due to space limit only F1 scores are reported) for all subtasks using the binary matching metric. We include the previously published results of Choi and Cardie (2010) for the same task using the same fold split and eval-

	Loss-augmented Training			Standard Training		
	P	R	F1	P	R	F1
PIPELINE	60.96	63.29	62.10	60.05	60.59	60.32
JSL	64.98†	54.60	59.29	67.09†	50.56	57.62
HJSL	**66.16***	56.77	61.05	**67.98**†	50.81	58.11
JI-PROB	50.95	**77.44***	61.32	50.06	**76.98***	60.54
JI-LOSS	63.77†	64.51†	**64.04***	64.97†	61.55†	**63.12***

Table 2: Opinion Expression Extraction (Proportional Matching). In all tables, we use **bold** to indicate the highest score among all the methods; use $*$ to indicate statistically significant improvements ($p < 0.05$) over all the other methods under the paired-t test; use $\dagger$ to denote statistically significance ($p < 0.05$) over the pipeline baseline.

	Positive			Negative			Neutral		
	P	R	F1	P	R	F1	P	R	F1
PIPELINE	45.26	43.07	44.04	50.59	47.91	49.11	40.98	49.30	44.57
JSL	**50.58**†	32.34	39.37	50.22	44.01	46.81	46.83†	39.81	42.85
HJSL	50.34†	37.06	42.59	53.29†	43.98	48.07	**47.29**†	43.27	45.03
JI-PROB	36.47	**47.81***	41.24	40.83	**54.40***	46.51	33.59	**59.22***	42.66
JI-LOSS	46.44†	44.58†	**45.40***	**54.88***	48.50	**51.40***	43.42†	52.02†	**47.09***
	High			Medium			Low		
	P	R	F1	P	R	F1	P	R	F1
PIPELINE	40.98	28.10	33.25	35.44	44.72	39.36	31.19	34.46	32.63
JSL	37.91	30.83†	**33.88**	**39.07**†	37.31	38.05	**40.95**†	26.71	32.24
HJSL	41.05	28.80	33.63	39.06†	39.71	39.17	40.01†	29.88	34.12
JI-PROB	34.82	**30.94**†	32.54	29.16	**50.89***	36.89	25.06	**42.99***	31.53
JI-LOSS	**46.11***	26.36	33.39	37.58†	43.58	**40.15***	33.85†	40.92†	**36.93***

Table 3: Opinion Extraction with Correct Attributes (Proportional Matching)

uation metric. CRF-JSL and CRF-HJSL are both joint sequence labeling methods based on CRFs. Different from JSL and HJSL, they perform sequence labeling at the token level instead of the segment level, and in HJSL, the decomposition of labels are not based on the decomposition of the joint probability of opinion segmentation and attribute labeling. We can see that both the pipeline and joint methods clearly outperform previous results in all evaluation criteria.[3] We can also see that JI-LOSS provides the best performance among all baselines.

5.1.1 Error Analysis

Joint vs. Pipeline We found that many errors made by the pipeline system are due to error propagation. Table 5 lists three examples, representing three types of the propagated errors:(1) the attribute classifiers miss the prediction since the opinion expression extractor fails to identify the opinion expression; (2) the attribute classifiers assign attributes to a non-opinionated expression since it was mistakenly extracted; (3) the attribute classifiers misclassify the attributes since the boundaries of opinion expressions are not correctly determined by the opinion expression extractor. Our joint models are able to correct many of these errors, such as the examples in Table 5, due to the modeling of the dependency between opinion expression extraction and attribute classification.

Joint Learning vs. Joint Inference Note that JSL and HJSL both employ joint learning while JI-PROB and JI-LOSS employ joint inference. To investigate the difference between these two types of joint models, we look into the errors made by HJSL and JI-LOSS. In general, we observed that HJSL extracts many fewer opinion expressions compared to JI-LOSS, and as a result, it presents high precision but low recall. The first two examples in Table 6

[3] Significance test was not conducted over the results in Choi and Cardie (2010) as we do not have their 10 fold results.

	Extraction	Positive	Negative	Neutral	High	Medium	Low
PIPELINE	73.30	51.50	58.45	52.45	39.34	47.08	39.05
JSL	69.76	45.24	57.11	50.25	41.48†	45.88	36.49
HJSL	71.43	49.08	58.38	52.25	41.06†	46.82	38.45
JI-PROB	74.37†	50.93	58.20	54.03†	39.80	46.65	40.73†
JI-LOSS	**75.11***	**53.02***	**62.01***	**54.33**†	**41.79**†	**47.38**	**42.53***
Previous work (Choi and Cardie (2010))							
CRF-JSL	60.5	41.9	50.3	41.2	38.4	37.6	28.0
CRF-HJSL	62.0	43.1	52.8	43.1	36.3	40.9	30.7

Table 4: Opinion Extraction Results (Binary Matching)

Example Sentences	Pipeline	Joint Models
It is the victim of an explosive situation $_{high}$ at the economic, ...	No opinions ×	✓
A white farmer who was shot dead Monday was the 10th to be killed.	the 10th to be killed $_{medium}$ ×	✓
They would " fall below minimum standards $_{medium}$ for humane $_{medium}$ treatment".	minimum standards for humane treatment $_{medium}$ ×	✓

Table 5: Examples of mistakes made by the pipeline baseline that are corrected by the joint models

are cases where HJSL gains in precision and loses in recall, respectively. The last example in Table 6 shows an error made by HJSL but corrected by JI-LOSS. Theoretically, joint learning is more powerful than joint inference as it models the task dependencies during training. However, we only observe improvements on precision and see drops in recall. As discussed before, we hypothesize that this is due to the mismatch of dependency assumptions between the model and the jointly annotated data. We found that joint inference can be superior to both pipeline and joint learning, and it is also much more efficient in training. In our experiments on an Amazon EC2 instance with 64-bit processor, 4 CPUs and 15GB memory, training for the joint learning approaches took one hour for each training fold, but only 5 minutes for the joint inference approaches.

5.2 Additional Experiments

5.2.1 Evaluation with Reranking

Previous work (Johansson and Moschitti, 2011) showed that reranking is effective in improving the pipeline of opinion expression extraction and polarity classification. We extended their approach to handle both polarity and intensity and investigated the effect of reranking on both the pipeline and joint models. For the pipeline model, we generated 64-best (distinct) output with 4-best labeling at each pipeline stage; for the joint models, we generated 50-best (distinct) output using Viterbi-like dynamic programming. We trained the reranker using the online PassiveAggressive algorithm (Crammer et al., 2006) as in Johansson and Moschitti (2013) with 100 iterations and a regularization constant $C = 0.01$. For features, we included the probability output by the base models, the polarity and intensity of each pair of extracted opinion expressions, and the word sequence and the POS sequence between the adjacent pairs of extracted opinion expressions.

Table 7 shows the reranking performance (F1) for all subtasks. We can see that after reranking, JI-LOSS still provides the best performance and HJSL achieves comparable performance to PIPELINE. We also found that reranking leads to less performance gain for the joint inference approaches than for the joint learning approaches. This is because the k-best output of JI-PROB and JI-LOSS present less diversity than JSL and HJSL. A similar issue for reranking has also been discussed in Finkel et al. (2006).

5.2.2 Evaluation on Sentence-level Tasks

As an additional experiment, we consider a supervised sentence-level sentiment classification task using features derived from the prediction output of different opinion extraction models. As a stan-

Example Sentences	JointLearn	JointInfer
The expression is undoubtedly strong and well thought out $_{high}$.	✓	well thought out $_{medium}$ ×
But the Sadc Ministerial Task Force said the election was free and fair $_{medium}$.	No opinions ×	✓
The president branded $_{high}$ as the "axis of evil" $_{high}$ in his statement ...	of evil $_{high}$ ×	✓

Table 6: Examples of mistakes that are made by the joint learning model but are corrected by the joint inference model and vice versa. We use the same colored box notation as before, and use yellow color to denote neutral sentiment.

	Extraction	Positive	Negative	Neutral	High	Medium	Low
PIPELINE + reranking	73.72	51.45	60.51	53.24	40.07	47.65	40.47
JSL + reranking	72.02	47.52	59.81	52.84	41.04†	46.58	39.40
HJSL + reranking	72.60	50.78	60.85	53.45	41.04†	47.75	40.08
JI-PROB + reranking	74.81†	51.45	59.59	53.98	40.66	46.87	40.80
JI-LOSS + reranking	**75.59**†	**53.29***	**62.50***	**54.94***	**41.79***	**47.67**	**42.66***

Table 7: Opinion Extraction with Reranking (Binary Matching)

Features	Acc	Positive	Negative	Neutral
BOW	65.26	51.90	77.47	36.41
PIPELINE-OP	67.41	55.49	79.42	39.48
JSL-OP	65.86	55.97	77.68	36.46
HJSL-OP	66.79	55.12	79.29	37.56
JI-PROB-OP	67.13	56.49	79.30	38.49
JI-LOSS-OP	**68.23***	**57.32***	**80.12***	**40.45***

Table 8: Sentence-level Sentiment Classification

dard baseline, we train a MaxEnt classifier using unigrams, bigrams and opinion lexicon features extracted from the sentence. Using the prediction output of an opinion extraction model, we construct features by using only words from the extracted opinion expressions, and include the predicted opinion attributes as additional features. We hypothesize that the more informative the extracted opinion expressions are, the more they can contribute to sentence-level sentiment classification as features. Table 8 shows the results in terms of classification accuracy and F1 score in each sentiment category. BOW is the standard MaxEnt baseline. We can see that using features constructed from the opinion expressions *always* improved the performance. This confirms the informativeness of the extracted opinion expressions. In particular, using the opinion expressions extracted by JI-LOSS gives the best performance among all the baselines in all evaluation criteria. This is consistent with its superior performance in our previous experiments.

6 Conclusion

We address the problem of opinion expression extraction and opinion attribute classification by presenting two types of joint models: joint learning, which optimizes the parameters of different subtasks in a joint probabilistic framework; joint inference, which optimizes the separately-trained models jointly during inference time. We show that our models achieve substantially better performance than the previously published results, and demonstrate that joint inference with an appropriate objective can be more effective and efficient than joint learning for the task. We also demonstrate the usefulness of output of our systems for sentence-level sentiment analysis tasks. For future work, we plan to improve joint modeling for the task by capturing semantic relations among different opinion expressions.

Acknowledgement

This work was supported in part by DARPA-BAA-12-47 DEFT grant #12475008 and NSF grant BCS-0904822. We thank the anonymous reviewers, Igor Labutov and the Cornell NLP Group for helpful suggestions.

References

E. Breck, Y. Choi, and C. Cardie. 2007. Identifying expressions of opinion in context. In *Proceedings of the international joint conference on Artifical intelligence*.

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

Yejin Choi and Claire Cardie. 2010. Hierarchical sequential learning for extracting opinions and their attributes. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics - Short Papers*.

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

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

Jenny Rose Finkel and Christopher D Manning. 2010. Hierarchical joint learning: Improving joint parsing and named entity recognition with non-jointly labeled data. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Jenny Rose Finkel, Christopher D Manning, and Andrew Y Ng. 2006. Solving the problem of cascading errors: Approximate bayesian inference for linguistic annotation pipelines. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Kevin Gimpel and Noah A Smith. 2010. Softmax-margin crfs: Training log-linear models with cost functions. In *Human Language Technologies: Conference of the North American Chapter of the Association for Computational Linguistics*.

Richard Johansson and Alessandro Moschitti. 2011. Extracting opinion expressions and their polarities: exploration of pipelines and joint models. In *Proceedings of the Association for Computational Linguistics: Human Language Technologies: short papers*.

Richard Johansson and Alessandro Moschitti. 2013. Relational features in fine-grained opinion analysis. *Computational Linguistics*, 39(3):473–509.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*.

V. Punyakanok, D. Roth, W. Yih, and D. Zimak. 2004. Semantic role labeling via integer linear programming inference. In *Proceedings of the international conference on Computational Linguistics*.

D. Roth and W. Yih. 2004. *A linear programming formulation for global inference in natural language tasks*.

Alexander M Rush, David Sontag, Michael Collins, and Tommi Jaakkola. 2010. On dual decomposition and linear programming relaxations for natural language processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Sunita Sarawagi and William W Cohen. 2004. Semi-markov conditional random fields for information extraction. In *Advances in Neural Information Processing Systems*.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

J. Wiebe, T. Wilson, and C. Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2):165–210.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational linguistics*, 35(3):399–433.

Bishan Yang and Claire Cardie. 2012. Extracting opinion expressions with semi-markov conditional random fields. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Bishan Yang and Claire Cardie. 2013. Joint inference for fine-grained opinion extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Ainur Yessenalina and Claire Cardie. 2011. Compositional matrix-space models for sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Jun Zhao, Kang Liu, and Gen Wang. 2008. Adding redundant features for crfs-based sentence sentiment classification. In *Proceedings of the conference on empirical methods in natural language processing*.