Discovering fine-grained sentiment with latent variable structured prediction models

Oscar Täckström*[1,2] and Ryan McDonald[3]

Swedish Institute of Computer Science[1]
Dept. of Linguistics and Philology, Uppsala University[2]
oscar@sics.se

Google, Inc.[3]
ryanmcd@google.com

SICS Technical Report T2011:02
ISSN 1100-3154

January 6, 2011

Abstract. In this paper we investigate the use of latent variable structured prediction models for fine-grained sentiment analysis in the common situation where only coarse-grained supervision is available. Specifically, we show how sentence-level sentiment labels can be effectively learned from document-level supervision using hidden conditional random fields (HCRFs) [25]. Experiments show that this technique reduces sentence classification errors by 22% relative to using a lexicon and by 13% relative to machine-learning baselines.
We provide a comprehensible description of the proposed probabilistic model and the features employed. Further, we describe the construction of a manually annotated test set, which was used in a thorough empirical investigation of the performance of the proposed model.[1]

1 Introduction

Determining the sentiment of a fragment of text is a central task in the field of opinion classification and retrieval [22]. Most research in this area can be categorized into one of two categories: lexicon or machine-learning centric. In the former, large lists of phrases are constructed manually or automatically indicating the polarity of each phrase in the list. This is typically done by exploiting common patterns in language [12, 27, 14], lexical resources such as WordNet or thesauri [15, 3, 26, 19], or via distributional similarity [33, 31, 32]. The latter approach – machine-learning centric – builds statistical text classification models based on labeled data, often obtained via consumer reviews that have been tagged with an associated star-rating [23, 21, 10, 11, 17, 4, 28].

Both approaches have their strengths and weaknesses. Systems that rely on lexicons can analyze text at all levels, including the clausal and phrasal level, which is fundamental

* Part of this work was performed while the author was an intern at Google, Inc.
[1] This technical report is an expanded version of the shorter conference paper [29].

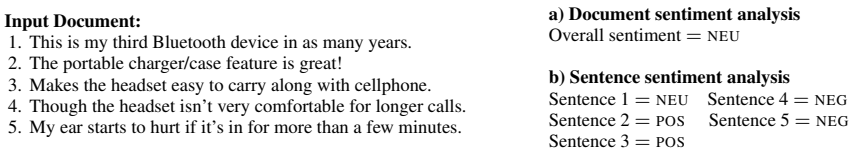

Fig. 1. Sentiment analysis at different levels. a) Standard document level analysis. b) A simple example of fine-grained sentiment analysis epitomized through sentence predictions.

to building user-facing technologies such as faceted opinion search and summarization [1, 13, 10, 24, 5, 3, 30, 38]. However, lexicons are typically deployed independent of the context in which mentions occur, often making them brittle, especially in the face of domain shift and complex syntactic constructions [35, 7]. The machine-learning approach, on the other hand, can be trained on the millions of labeled consumer reviews that exist on review aggregation websites, often covering multiple domains of interest [23, 21, 4]. The downside is that the supervised learning signal is often at a coarse level, i.e., the document level.

Attempts have been made to bridge this gap. The most common approach is to obtain a labeled corpus at the granularity of interest in order to train classifiers that take into account the analysis returned by a lexicon and its context [35, 3]. This approach combines the best of both worlds – knowledge from broad-coverage lexical resources in concert with highly tuned machine-learning classifiers that take into account context. The primary downside of such models is that they are often trained on small sets of data, since fine-grained sentiment annotations rarely exist naturally and instead require significant annotation effort per domain [34].

To circumvent laborous annotation efforts, we propose a model that can learn to analyze fine-grained sentiment strictly from coarse annotations. Such a model can leverage the plethora of labeled documents from multiple domains available on the web. The model we present is based on hidden conditional random fields (HCRFs) [25], a well-studied latent variable structured learning model that has been used previously in speech and vision. We show that this model naturally fits the task and can reduce fine-grained classification errors by up to 20%.

2 Fine-grained sentiment analysis

Figure 1 shows an example where sentence sentiment is contrasted with document sentiment. This is perhaps the simplest form of fine-grained sentiment analysis and one could imagine building an analysis at the clause or phrase level annotating multiple attributes of opinions beyond their polarity [34]. Though all the methods described henceforth could conceivably be applied to finer levels of granularity, in this work, we focus on sentence level sentiment (or polarity) analysis. To be concrete, as input, the system expects a sentence segmented document and outputs the corresponding sentence labels from the set $\{\text{POS}, \text{NEG}, \text{NEU}\}$ as shown in Figure 1 and defined precisely below.

2.1 Data for training and evaluation

There are several freely available data sets annotated with sentiment at various levels of granularity; a comprehensive list of references is given in [22]. For our experiments, described in Section 4, we required a data set annotated at both the sentence and document levels. The data set used in [18] is close in spirit, but it lacks neutral documents, which is an unrealistic over-simplification, since neutral reviews are abundant in most domains. Therefore, we constructed a large corpus of consumer reviews from a range of domains, each review annotated with document sentiment automatically extracted from its star rating, and a small subset of reviews manually annotated at the sentence level.

A training set was created by sampling a total of 150,000 positive, negative and neutral reviews from five different domains: *books*, *dvds*, *electronics*, *music* and *videogames*. We chose to label one and two star reviews as negative (NEG), three star reviews as neutral (NEU), and four and five star reviews as positive (POS). After removing duplicates, a balanced set of 143,580 reviews remained. Each review was split into sentences and each sentence automatically enriched with negation scope information as described in [8] and matches against the polarity lexicon described in [35]. As can be seen from the detailed sentence level statistics in Table 1, the total number of sentences is roughly 1.5 million. Note that the training set only has labels at the document level as reviewers do not typically annotate fine-grained sentiment in consumer reviews.

The same procedure was used to create a smaller separate test set consisting of 300 reviews, again uniformly sampled with respect to the domains and document sentiment categories. After duplicates where removed, 97 positive, 98 neutral and 99 negative reviews remained. Two annotators marked the test set reviews at the sentence level with the following categories: POS, NEG, NEU, MIX, and NR. The category NEU was assigned to sentences that express sentiment, but are neither positive nor negative, e.g., "The image quality is not good, but not bad either.", while the category MIX was assigned to sentences that express both positive and negative sentiment, e.g., "Well, the script stinks, but the acting is great!". The NR category (for 'not relevant') was assigned to sentences that contain no sentiment as well as to sentences that express sentiment about something other than the target of the review. All but the NR category were assigned to sentences that either express sentiment by themselves, or that are part of an expression of sentiment spanning several sentences. This allowed us to annotate, e.g., "Is this good? No." as negative, even though this expression is split into two sentences in the preprocessing step. To simplify our experiments, we considered the MIX and NR categories as belonging to the NEU category. Thus, NEU can be viewed as a type of 'other' category.

The total number of annotated sentences in the test set is close to four thousand. Annotation statistics can be found in Table 3, while Table 2 shows the distribution of sentence level sentiment for each document sentiment category. Clearly, the sentence level sentiment is aligned with the document sentiment, but reviews from all categories contain a substantial fraction of neutral sentences and a non-negligible fraction of both positive and negative sentences. Overall raw inter-annotator agreement was 86% with a Cohen's κ value of 0.79. Class-specific agreements were 83%, 93% and 82% respectively for the POS, NEG and NEU category.[2]

[2] The annotated test set can be freely downloaded from the first author's web site: `http://www.sics.se/people/oscar/datasets`.

	POS	NEG	NEU	Total
Books	56,996	61,099	59,387	177,482
Dvds	121,740	102,207	131,089	355,036
Electronics	73,246	69,149)	84,264	226,659
Music	65,565	55,229	72,430	193,224
Videogames	163,187	125,422	175,405	464,014
Total	480,734	430,307	522,575	1,416,415

Table 1. Number of sentences per document sentiment category for each domain in a large training sample. There are 9,572 documents for each (domain, document sentiment)-pair for a total of 143,580 documents.

	POS	NEG	NEU
POS	0.53	0.08	0.39
NEG	0.05	0.62	0.33
NEU	0.14	0.35	0.51

Table 2. Distribution of sentence labels (columns) in documents by their labels (rows) in the test data.

	Documents per category				Sentences per category			
	POS	NEG	NEU	Total	POS	NEG	NEU	Total
Books	19	20	20	59	160	195	384	739
Dvds	19	20	20	59	164	264	371	799
Electronics	19	19	19	57	161	240	227	628
Music	20	20	19	59	183	179	276	638
Videogames	20	20	20	60	255	442	335	1,032
Total	97	99	98	294	923	1,320	1,593	3,836

Table 3. Number of documents per document sentiment category (left) and number of sentences per sentence sentiment category (right) in the labeled test set for each domain.

2.2 Baselines

Lexicons are a common tool used for fine-grained sentiment analysis. As a first experiment, we examined the polarity lexicon used in [35], which rates a list of phrases on a discrete scale in (-1.0, 1.0), where values less than zero convey negative sentiment and values above zero positive.[3] To classify sentences, we matched elements from this lexicon to each sentence. These matches, and their corresponding polarities, were then fed into the vote-flip algorithm [7], which is a rule-based algorithm that uses the number of positive and negative lexicon matches as well as the existence of negations to classify a sentence. To detect the presence of negation and its scope we used an implementation of the CRF-based negation classifier described in [8]. Results for this system are shown in Table 5 under the row VoteFlip. We can observe that both classification and retrieval statistics are fairly low. This is not surprising. The lexicon is not exhaustive and many potential matches will be missed. Furthermore, sentences like "It would have been good if it had better guitar." will be misclassified as neither context nor syntactic/semantic structure are modeled. We also ran experiments with two machine-learning baselines that can take advantage of the consumer review training corpus (Section 2.1). The first, which we call *Sentence as Document* (SaD), splits the training documents into sentences and assigns each sentence the label of the corresponding document it came from. This new training set is then used to train a logistic regression classifier. Because documents often contain sentences with different sentiment from the overall document sentiment,

[3] Though more broader-coverage lexicons exist in the literature, e.g., [18, 19], we used this lexicon because it is publicly available (`http://www.cs.pitt.edu/mpqa/`).

this is a rather crude approximation. The second baseline, *Document as Sentence* (DaS), trains a logistic regression document classifier on the training data in its natural form. This baseline can be seen as either treating training documents as long sentences (hence the name) or test sentences as short documents. Details of the classifiers and feature sets used to train the baselines are given in Section 4. Results for these baselines are given in Table 5. There is an improvement over using the lexicon alone, but both models make the assumption that the observed document label is a good proxy for all the sentences in the document, which is likely to degrade prediction accuracy.

3 A conditional latent variable model of fine-grained sentiment

The distribution of sentences in documents from our data (Table 2) suggests that documents do contain at least one dominant class, even though they do not have uniform sentiment. Specifically, positive (negative) documents primarily consist of positive (negative) sentences as well as a significant number of neutral sentences and a small amount of negative (positive) sentences. When combined with the problems raised in the previous section, this observation suggests that we would like a model where sentence level classifications are 1) correlated with the observed document label, but 2) have the flexibility to disagree when evidence from the sentence or local context suggests otherwise.

To build such a model, we start with the supervised fine-to-coarse sentiment model described by McDonald et al. [18]. Let d be a document consisting of n sentences, $\boldsymbol{s} = (s_i)_{i=1}^n$. We denote by $\boldsymbol{y}^d = (y^d, \boldsymbol{y}^s)$ random variables that include the document level sentiment, y^d, and the sequence of sentence level sentiment, $\boldsymbol{y}^s = (y_i^s)_{i=1}^n$.[4] Both y^d and all y_i^s belong to $\{\text{POS}, \text{NEG}, \text{NEU}\}$. We hypothesize that there is a sequential relationship over sentence level sentiment and that the document level sentiment is influenced by all sentence level sentiment (and vice versa). Figure 2a shows an undirected graphical model [2] reflecting this idea. A first order Markov property is assumed, according to which each sentence variable y_i^s is independent of all other variables, conditioned on the document variable y^d and its adjacent sentences, y_{i-1}^s/y_{i+1}^s. By making this assumption, [18] was able to reduce this model to standard sequential learning, which has both efficient learning and inference algorithms, such as conditional random fields (CRFs) [16]. The strength of this model is that it allows sentence and document level classifications to influence each other while giving them freedom to disagree when influenced by the input. It was shown that this model can increase both sentence and document level prediction accuracies. However, at training time, it requires labeled data at all levels of analysis.

We are interested in the common case where document labels are available (e.g., from star-rated consumer reviews), but sentence labels are not. A modification to the model from Figure 2a is to treat all the sentence labels as unobserved as shown in Figure 2b. When the underlying model from Figure 2a is a conditional random field, the model in Figure 2b is often referred to as a hidden conditional random field (HCRF) [25]. HCRFs are appropriate when there is a strong correlation between the observed coarse label and the unobserved fine-grained variables. We would expect to see positive, negative and

[4] We will abuse notation by using the same symbols to refer to random variables and their particular assignments.

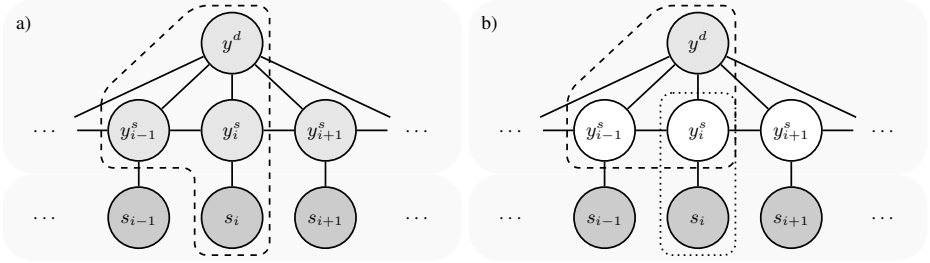

Fig. 2. a) Outline of graphical model from [18]. b) Identical model with latent sentence level states. Dark nodes are observed variables and light nodes are unobserved. The input sentences s_i are always observed. Dashed and dotted regions indicate the maximal cliques at position i. Note that the document and input nodes belong to different cliques in the right model.

neutral sentences in all types of documents, but we are far more likely to see positive sentences than negative sentences in positive documents.

3.1 Probabilistic formulation

In the conditional random field model just outlined, the distribution of the random variables $\boldsymbol{y}^d = (y^d, \boldsymbol{y}^s)$, conditioned on input sentences $\boldsymbol{s}$, belongs to the exponential family and is written

$$p_\theta(y^d, \boldsymbol{y}^s | \boldsymbol{s}) = \exp\left\{\langle \phi(y^d, \boldsymbol{y}^d, \boldsymbol{s}), \theta \rangle - A_\theta(\boldsymbol{s})\right\} ,$$

where θ is a vector of model parameters and $\phi(\cdot)$ is a vector valued feature function (the sufficient statistics), which by the independence assumptions of the graphical models outlined in Figure 2a and Figure 2b, factorizes as

$$\phi(y^d, \boldsymbol{y}^s, \boldsymbol{s}) = \bigoplus_{i=1}^{n} \phi(y^d, y^s_i, y^s_{i-1}, \boldsymbol{s}) ,$$

where $\bigoplus$ indicates vector summation. The $\log$-partition function, $A_\theta(\boldsymbol{s})$, is a normalization constant, which ensures that $p_\theta(y^d, \boldsymbol{y}^s | \boldsymbol{s})$ is a proper probability distribution. This is achieved by summing over the set of all possible variable assignments $\mathcal{Y}_d$

$$A_\theta(\boldsymbol{s}) = \log \sum_{\boldsymbol{y}^{d'} \in \mathcal{Y}_d} \exp\left\{\langle \phi(y^{d'}, \boldsymbol{y}^{s'}, \boldsymbol{s}), \theta \rangle\right\} .$$

In an HCRF, the conditional probability of the observed variables, in our case the document sentiment, is then obtained by marginalizing over the posited hidden variables

$$p_\theta(y^d | \boldsymbol{s}) = \sum_{\boldsymbol{y}^s} p_\theta(y^d, \boldsymbol{y}^s | \boldsymbol{s}) .$$

As indicated in Figure 2b, there are two maximal cliques at each position i in the graphical model: one involving only the sentence s_i and its corresponding latent variable

y_i^s and one involving the consecutive latent variables y_i^s, y_{i-1}^s and the document variable y^d. The assignment of the document variable y^d is thus independent of the input $\boldsymbol{s}$, conditioned on the sequence of latent sentence variables $\boldsymbol{y}^s$. This is in contrast to the original fine-to-coarse model, in which the document variable depends directly on the sentence variables as well as the input [18]. This distinction is important for learning predictive latent variables as it creates a bottleneck between the input sentences and the document label. This forces the model to generate good predictions at the document level only through the predictions at the sentence level. Since the input $\boldsymbol{s}$ is highly informative of the document sentiment, the model may circumvent the latent sentence variables. When we allow the document label to be directly dependent on the input, we observe a substantial drop in sentence level prediction performance.

3.2 Feature functions

The feature function at position i is the sum of the feature functions for each clique at that position, that is $\phi(y^d, y_i^s, y_{i-1}^s, \boldsymbol{s}) = \phi(y^d, y_i^s, y_{i-1}^s) \oplus \phi(y_i^s, \boldsymbol{s})$. The feature function for each clique is in turn defined in terms of binary predicates of the partaking variables. These features are chosen in order to encode the compatibility of the assignments of the variables (and the input) in the clique.

The features of the clique $(y_i^s, \boldsymbol{s})$[5] are defined in terms of predicates encoding the following properties, primarily derived from [32]:

TOKENS(s_i) The set of tokens in s_i.
POSITIVETOKENS(s_i) The set of tokens in s_i matching the positive lexicon.
NEGATIVETOKENS(s_i) The set of tokens in s_i matching the negative lexicon.
NEGATEDTOKENS(s_i) The set of tokens in s_i that are negated according to [8].
#POSITIVE(s_i) The cardinality of POSITIVETOKENS(s_i).
#NEGATIVE(s_i) The cardinality of NEGATIVETOKENS(s_i).
VOTEFLIP(s_i) The output of the vote-flip algorithm [7].

All lexicon matches are against the polarity lexicon described in [35]. Using these predicates, we construct the feature templates listed in Table 4. This table also lists the much simpler set of feature templates for the (y^d, y_i^s, y_{i-1}^s)-clique, which only involves various combinations of the document and sentence sentiment variables. Each instantiation of a feature template is mapped to an element in the feature representation using a hash function.

3.3 Estimation

The parameters of CRFs are generally estimated by maximizing an L_2-regularized conditional log-likelihood function, which corresponds to maximum a posteriori probability (MAP) estimation assuming a Normal prior, $p(\theta) \sim \mathcal{N}(0, \sigma^2)$. Instead of maximizing the joint conditional likelihood of document and sentence sentiment, as would be done

[5] In the present feature model, we ignore all sentences but s_i, so that instead of $(y_i^s, \boldsymbol{s})$, we could have written (y_i^s, s_i). We keep to the more general notation, since we could in principle look at any part of the input $\boldsymbol{s}$.

Template	Domain
$[w \in \textsc{Tokens}(s_i) \wedge y^s_i = a]$	$w \in \mathcal{W}, a \in \{\textsc{pos}, \textsc{neg}, \textsc{neu}\}$
$[w \in \textsc{PositiveTokens}(s_i) \wedge y^s_i = a]$	$w \in \mathcal{W}, a \in \{\textsc{pos}, \textsc{neg}, \textsc{neu}\}$
$[w \in \textsc{NegativeTokens}(s_i) \wedge y^s_i = a]$	$w \in \mathcal{W}, a \in \{\textsc{pos}, \textsc{neg}, \textsc{neu}\}$
$[\#\textsc{Positive}(s_i) > \#\textsc{Negative}(s_i) \wedge y^s_i = a]$	$a \in \{\textsc{pos}, \textsc{neg}, \textsc{neu}\}$
$[\#\textsc{Positive}(s_i) > 2 \cdot \#\textsc{Negative}(s_i) \wedge y^s_i = a]$	$a \in \{\textsc{pos}, \textsc{neg}, \textsc{neu}\}$
$[\#\textsc{Negative}(s_i) > \#\textsc{Positive}(s_i) \wedge y^s_i = a]$	$a \in \{\textsc{pos}, \textsc{neg}, \textsc{neu}\}$
$[\#\textsc{Negative}(s_i) > 2 \cdot \#\textsc{Positive}(s_i) \wedge y^s_i = a]$	$a \in \{\textsc{pos}, \textsc{neg}, \textsc{neu}\}$
$[\#\textsc{Positive}(s_i) = \#\textsc{Negative}(s_i) \wedge y^s_i = a]$	$a \in \{\textsc{pos}, \textsc{neg}, \textsc{neu}\}$
$[w \in \textsc{NegationScope}(s_i) \wedge y^s_i = a]$	$w \in \mathcal{W}, a \in \{\textsc{pos}, \textsc{neg}, \textsc{neu}\}$
$[\textsc{VoteFlip}(s_i) = x \wedge y^s_i = a]$	$a, x \in \{\textsc{pos}, \textsc{neg}, \textsc{neu}\}$
$\left[y^d = a\right]$	$a \in \{\textsc{pos}, \textsc{neg}, \textsc{neu}\}$
$[y^s_i = a]$	$a \in \{\textsc{pos}, \textsc{neg}, \textsc{neu}\}$
$\left[y^d = a \wedge y^s_i = b\right]$	$a, b \in \{\textsc{pos}, \textsc{neg}, \textsc{neu}\}$
$\left[y^d = a \wedge y^s_i = b \wedge y^s_{i-1}\right]$	$a, b, c \in \{\textsc{pos}, \textsc{neg}, \textsc{neu}\}$

Table 4. Feature templates and their respective domains. Top: $(y^s_i, \boldsymbol{s})$-clique feature templates. Bottom: (y^d, y^s_i, y^s_{i-1})-clique feature templates. $\mathcal{W}$ represents the set of all tokens.

with a standard CRF, we find the MAP estimate of the parameters with respect to the *marginal* conditional log-likelihood of observed variables. Let $D = \{(\boldsymbol{d}_j, y^d_j)\}_{j=1}^m$ be a training set of document / document-label pairs, where $\boldsymbol{d}_j = (d_j, \boldsymbol{s}_j)$. We find the parameters that maximize the total marginal probability of the observed document labels, while keeping the parameters close to zero, according to the likelihood function

$$L^{\text{soft}}(\theta) = \sum_{j=1}^{|D|} \log \sum_{\boldsymbol{y}^s} p_\theta(y^d_j, \boldsymbol{y}^s | \boldsymbol{s}_j) - \frac{\|\theta\|^2}{2\sigma^2} \,. \tag{1}$$

We use the term *soft* estimation to refer to the maximization of (1). As an alternative to using proper marginalization, we can perform *hard* estimation (also known as *Viterbi* estimation) by instead maximizing

$$L^{\text{hard}}(\theta) = \sum_{j=1}^{|D|} \log p_\theta(y^d_j, \hat{\boldsymbol{y}}^s_j | \boldsymbol{s}_j) - \frac{\|\theta\|^2}{2\sigma^2} \,, \tag{2}$$

$$\text{where} \qquad \hat{\boldsymbol{y}}^s_j = \operatorname*{argmax}_{\boldsymbol{y}^s} p_\theta(y^d_j, \boldsymbol{y}^s | \boldsymbol{s}_j) \,. \tag{3}$$

In the hard estimation case, we only move probability mass to the most probable latent variable assignments. In both cases, we find the parameters θ that maximizes equations (1) and (2) by stochastic gradient ascent with a fixed step size, η. Note that while the likelihood function maximized in a standard CRF is concave, the introduction of latent variables makes both the soft and hard likelihood functions non-concave. Any gradient-based optimization method is therefore only guaranteed to find some local maxima of equations (1) and (2). Previous work on latent variable models for sentiment

analysis, e.g. [20], has reported on the need for complex initialization of the parameters to overcome the presence of local minima. We did not experience such problems and for all reported experiments we simply initialized θ to the zero vector.

3.4 Inference

We are interested in two kinds of inference during training. The marginal distributions $p_\theta(y^d, y_i^s|\boldsymbol{s})$ and $p_\theta(y^d, y_i^s, y_{i-1}^s|\boldsymbol{s})$ for each document–sentence variables $(y^d, y_i^s)_{i=1}^n$ and document–sentence pair variables $(y^d, y_i^s, y_{i-1}^s)_{i=2}^n$, are needed when computing the gradient of (1), while the most probable joint assignment of all variables (3) is needed when optimizing (2). As with the model described in [18], we use constrained *max-sum* (Viterbi) to solve (3) and constrained *sum-product* (forward-backward) to compute the marginal distributions [2].

When predicting the document and sentence level sentiment, we can either pick the most probable joint variable assignment or individually assign each variable with the label that has the highest marginal probability. It seems intuitively reasonable that the inference used at prediction time should match that used at training time, i.e. to use sum-product in the soft case and max-sum in the hard case. Our experimental results indicates that this is indeed the case, although the differences between the decoding strategies is quite small. Sum-product inference is moreover useful whenever probabilities are needed for individual variable assignments, such as for trading off precision against recall for each label.

In the HCRF model the interpretation of the latent states assigned to the sentence variables, y_i^s, are not tightly constrained by the observations during training as in a standard CRF. We therefore need to find the best mapping from the latent states to the labels that we are interested in. When the number of latent states is small (as is true for our experiments), such a mapping can be easily found by evaluating all possible mappings on a small set of annotated sentences. Alternatively we experimented with seeding the HCRF with values from the DaS baseline, which fixes the assignment of latent variables to labels. This strategy produced nearly identical results.

4 Experiments

We now turn to a set of experiments by which we assessed the viability of the proposed HCRF model compared to the VoteFlip, SaD and DaS baselines described in Section 2. In order to make the underlying statistical models the same across machine learning systems, SaD and DaS were parameterized as log-linear models and optimized for regularized conditional maximum likelihood using stochastic gradient ascent. This makes them identical to the HCRF except that document structure is not modeled as a latent variable. With regards to the HCRF model, we report results for both soft and hard optimization. Except where noted, we report results of max-sum inference for the hard model, and sum-product inference for the soft model as these combinations performed best. We also measured the benefit of observing the document label at test time. This is a common scenario in, e.g., consumer-review summarization and aggregation [13]. Note that for

	Sentence Acc.	POS Sent. F_1	NEG Sent. F_1	NEU Sent. F_1	Document Acc.
VoteFlip	41.5 (-1.8, 1.8)	45.7	48.9	28.0	–
SaD	47.6 (-0.8, 0.9)	52.9	48.4	42.8	–
DaS	47.5 (-0.8, 0.7)	52.1	54.3	36.0	66.6 (-2.4, 2.2)
HCRF (soft)	**53.9 (-2.4, 1.6)**	**57.3**	**58.5**	**47.8**	65.6 (-2.9, 2.6)
HCRF (hard)	**54.4 (-1.0, 1.0)**	**57.8**	**58.8**	**48.5**	64.6 (-2.0, 2.1)
DocOracle	54.8 (-3.0, 3.1)	61.1	58.5	47.0	–
HCRF (soft)	**57.7 (-0.9, 0.8)**	61.5	**62.0**	**51.9**	–
HCRF (hard)	**58.4 (-0.8, 0.7)**	62.0	**62.3**	**53.2**	–

Table 5. Median results and 95% confidence intervals from ten runs over the large data set. Above line: without observed document label. Below line: with observed document label. Boldfaced: significant compared to best comparable baseline, $p < 0.05$.

this data set the baseline of predicting all sentences with the observed document label, denoted DocOracle, is a strong baseline.

The SaD, DaS and HCRF methods all depend on three hyper-parameters during training — the stochastic gradient ascent learning rate, η; the regularization trade-off parameter, σ^2; and the number of epochs to run. We allowed a maximum of 75 epochs and picked values for the hyper-parameters that maximized development set macro-averaged F_1 on the document level for HCRFs and DaS, and on the sentence level with SaD. Since the latter uses document labels as a proxy for sentence labels, no manual sentence-level supervision was used during any point of training; only when evaluating the results, the sentence-level annotations were used to identify the latent states. These three models use identical feature sets when possible (as discussed in Section 3.1). The single exception being that SaD and DaS do not contain structured features (such as adjacent sentence label features) since they are not structured predictors. For all models, we mapped feature template instantiations to feature space elements using a 19-bit hash function. Except for the lexicon-based model, training for all models is stochastic in nature. To account for this, we performed ten runs of each model with different random seeds. In each run a different split of the training data was used for tuning the hyper-parameters. Results were then gathered by applying each model to the test data described in Section 2.1 and bootstrapping median and confidence intervals of the statistic of interest. Since sentence level predictions are not i.i.d, a hierarchical bootstrap was used [9].

4.1 Results and analysis

Table 5 shows the results for each model in terms of sentence and document level accuracy as well as F_1-scores for each sentence sentiment category. From these results it is clear that the HCRF models significantly outperform all the baselines with quite a wide margin. When document labels are provided at test time, results are even better compared to the machine learning baselines, but compared to the DocOracle baseline the error reductions are more modest. These differences are all statistically significant at $p < 0.05$ according to bootstrapped confidence interval tests.

Specifically, the HCRF with hard estimation reduces the error compared to the pure lexicon approach by 22% and by 13% compared to the best machine learning baseline.

	POS docs.	NEG docs.	NEU docs.
VoteFlip	59/19/27	16/61/23	40/51/32
SaD	67/18/45	15/60/36	43/42/45
DaS	67/20/35	14/68/29	45/49/41
HCRF (soft)	69/14/45	07/70/37	33/49/55
HCRF (hard)	69/14/47	06/71/36	34/48/56
DocOracle	69/00/00	00/77/00	00/00/67
HCRF (soft)	70/01/39	02/76/29	20/36/66
HCRF (hard)	72/00/44	00/76/23	03/38/66

Table 6. Sentence results per document category (columns). Each cell contains positive/negative/neutral sentence-level F_1-scores.

	Small	Medium	Large
VoteFlip	41.5 (-1.8, 1.8)	41.5 (-1.8, 1.8)	41.5 (-1.8, 1.8)
SaD	42.4 (-2.0, 1.3)	46.3 (-1.2, 1.0)	47.6 (-0.8, 0.9)
DaS	43.8 (-0.9, 0.8)	46.8 (-0.6, 0.7)	47.5 (-0.8, 0.7)
HCRF (soft)	44.9 (-1.7, 1.5)	**50.0 (-1.2, 1.2)**	**53.9 (-2.4, 1.6)**
HCRF (hard)	43.0 (-1.2, 1.3)	**49.1 (-1.4, 1.5)**	**54.4 (-1.0, 1.0)**
DocOracle	54.8 (-3.0, 3.1)	54.8 (-3.0, 3.1)	54.8 (-3.0, 3.1)
HCRF (soft)	54.5 (-1.0, 0.9)	54.9 (-1.0, 0.8)	**57.7 (-0.9, 0.8)**
HCRF (hard)	48.6 (-1.6, 1.4)	54.3 (-1.9, 1.8)	**58.4 (-0.8, 0.7)**

Table 7. Sentence accuracy for varying training size. Lower and upper offset limits of the 95% confidence interval in parentheses. Bold: significant compared to all comparable baselines, $p < 0.05$.

When document labels are provided at test time, the corresponding error reductions are 29% and 21%. In the latter case the reduction compared to the strong DocOracle baseline is only 8%. However, the probabilistic predictions of the HCRF are much more informative than this simple baseline. Hard estimation for the HCRF slightly outperforms soft estimation.

In terms of document accuracy the DaS model seem to slightly outperform the latent variable models. This is contrary to the results reported in [36], in which latent variables on the sentence level was shown to improve document predictions. Note, however, that our model is restricted when it comes to document level classification, due to the lack of connection between the document node and the input nodes in the graphical model. If we let the document sentiment be directly dependent on the input, which corresponds to a probabilistic formulation of the one in [36], we would expect the document accuracy to improve. Still, experiments with such connected HCRF models actually showed a slight decrease in document level accuracy compared to the disconnected models, while sentence level accuracy dropped even below the SaD and DaS models. By initializing the HCRF models with the parameters of the DaS model, results where better, but still not on par with the disconnected models.

Looking in more detail at Table 5, we observe that all models perform best in terms of F_1 on positive and negative sentences, while all models perform much worse on neutral sentences. This is not surprising, as neutral documents are particularly bad proxies for sentence level sentiment, as can be seen from the distributions of sentence-level sentiment per document category in Table 2. The lexicon based approach has difficulties with neutral sentences, since the lexicon contains only positive and negative words and there is no way of determining if a mention of a word in the lexicon should be considered as sentiment bearing in a given context.

A shortcoming of the HCRF model compared to the baselines is illustrated by Table 6: it tends to over-predict positive (negative) sentences in positive (negative) documents and to under-predict positive sentences in neutral documents. In other words, it only predicts well on the two dominant sentence-level categories for each document category. This is a problem shared by the baselines, but it is more prominent in the HCRF model. A plausible explanation comes from the optimization criteria, i.e. document-level likelihood, and the nature of the document-level annotations, since in order to learn whether a review

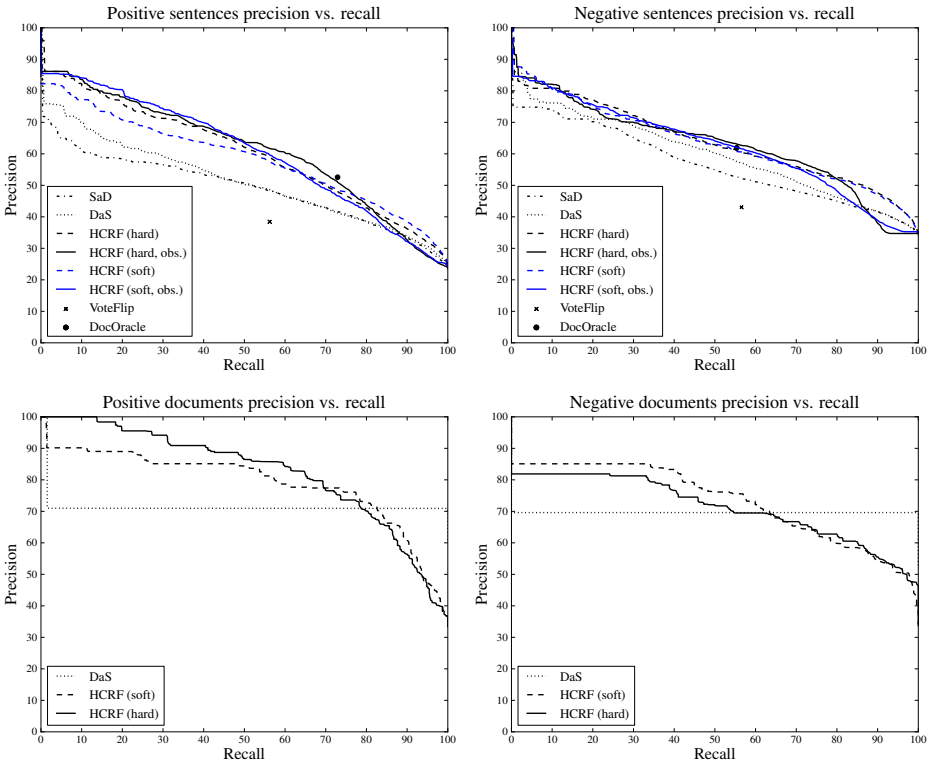

Fig. 3. Interpolated precision-recall curves with respect to positive and negative sentence level sentiment (top) and document level sentiment (bottom). Curves shown correspond to bootstrapped median of average precision over ten runs.

is positive, negative or neutral, it will often suffice to find the dominant sentence-level sentiment and to identify the non-relevant sentences of the review. Therefore, the model might need more constraints in order to learn to predict the minority sentence-level sentiment categories. More refined document labels and/or additional constraints during optimization might be avenues for future research with regard to these issues. Increasing the amount of training data is another potential route to reducing this problem.

4.2 The impact of more data

In order to study the impact when varying the size of the training data, we created additional training sets, denoted *Small* and *Medium*, by sampling 1,500 and 15,000 documents, respectively, from the full training set, denoted *Large*. We then performed the same experiment and evaluation as with the full training set with these smaller sets. Different training set samples was used for each run of the experiment. From Table 7, we observe that adding more training data improves all models. For the small data set there is no significant difference between the learning based models, but starting with the medium data set, the HCRF models outperform the baselines. Furthermore, while

	Sentence Acc.	POS Sent. F_1	NEG Sent. F_1	NEU Sent. F_1	Document Acc.
VoteFlip	41.5 (-1.9, 2.0)	48.2	47.7	25.0	–
SaD	49.0 (-1.2, 1.2)	57.7	59.7	11.1	–
DaS	48.3 (-0.9, 0.9)	57.3	60.7	–	87.5 (-1.5, 1.6)
HCRF (soft)	**57.6 (-1.3, 1.2)**	**63.6**	**66.9**	**39.4**	88.4 (-1.9, 1.6)
HCRF (hard)	**53.7 (-1.5, 1.7)**	**62.8**	**68.8**	–	87.8 (-1.5, 1.5)
DocOracle	57.3 (-4.0, 3.6)	67.1	72.5	–	–
HCRF (soft)	**60.6 (-1.0, 1.0)**	68.2	71.5	**38.2**	–
HCRF (hard)	57.6 (-1.4, 1.6)	66.2	71.7	16.0	–

Table 8. Median results and 95% confidence intervals from ten runs over the large data set with excluded neutral documents. Above line: without observed document label. Below line: with observed document label. Boldfaced: significant compared to best comparable baseline, $p < 0.05$.

the improvements are relatively small for the baselines, the improvement is substantial for the HCRF models. Thus, we expect that the gap between the latent variable models and the baselines will continue to increase with increasing training set size.

4.3 Trading off precision against recall

Though max-sum inference slightly outperforms sum-product inference for the hard HCRF in terms of classification performance, using sum-product inference for prediction has the advantage that we can tune per-label precision–recall based on the sentence-level marginal distributions. Such flexibility is another reason for preferring statistical approaches to rule-based approaches such as VoteFlip and the DocOracle baseline. Figure 3 contains sentence-level precision–recall curves for HCRF (hard), with and without observed document label, SaD and DaS, together with the fixed points of VoteFlip and DocOracle. Curves are also shown for positive and negative document-level precision–recall. Each curve correspond to the bootstrapped median of average-precision over ten runs.

From these plots, it is evident that the HCRF dominates sentence-level predictions at nearly all levels of precision/recall, especially so for positive sentences. In terms of document-level precision/recall, the HCRF models have substantially higher precision for lower levels of recall, again especially for positive documents, while DaS maintains precision better at higher recall levels. Note how the document level probabilities learned for the DaS model are note very informative for trading off precision against recall.

4.4 Ignoring neutral documents

It is worth mentioning that although the results for all systems seem low ($<$60% sentence level accuracy and $<$70% document accuracy), they are comparable with those in [18] (62.6% sentence level accuracy), which was trained with both document and sentence level supervision and evaluated on a data set that did not contain neutral documents. In fact, the primary reason for the low scores presented in this work is the inclusion of neutral documents and sentences in our data. This makes the task much more difficult than 2-class positive-negative polarity classification, but also more representative of

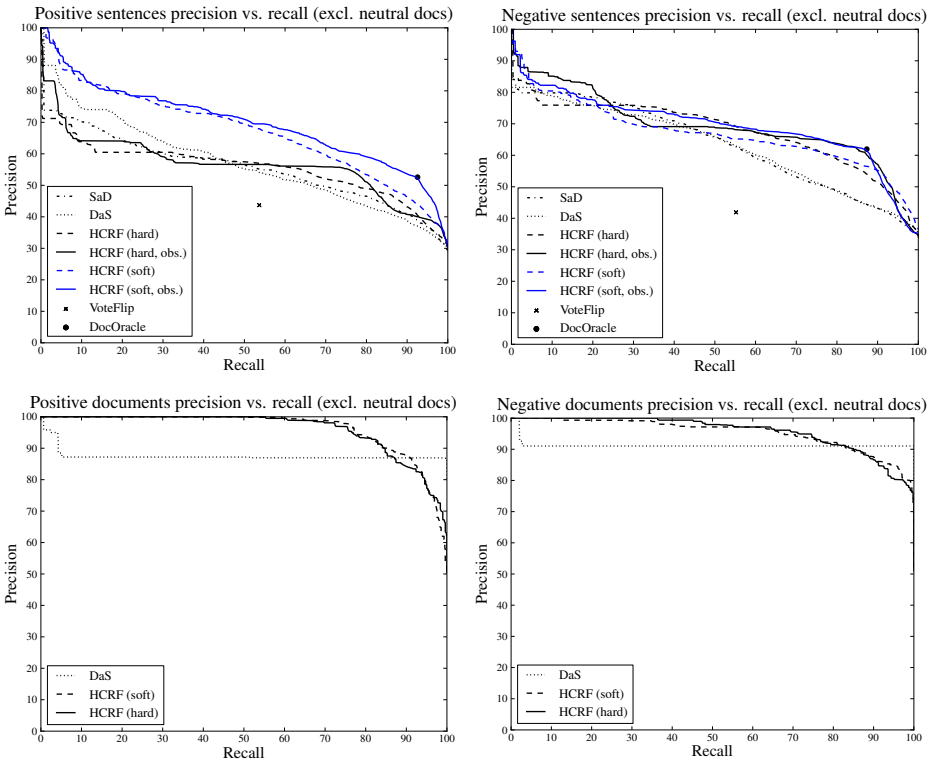

Fig. 4. Interpolated precision-recall curves with respect to positive and negative sentence level sentiment (top) and document level sentiment (bottom) with neutral documents excluded. Curves shown correspond to bootstrapped median of average precision over ten runs.

real-world use-cases. To support this claim, we ran the same experiments as above while excluding neutral documents from the training and test data. Table 8 contains detailed results for the two-class experiments, while Figure 4 shows the corresponding precision–recall curves. In this scenario the best HCRF model achieves a document accuracy of 88.4%, which is roughly on par with reported document accuracies for the two-class task in state-of-the-art systems [4, 20, 36]. Furthermore, as mentioned in Section 2.1, inter-annotator agreement was only 86% for the three-class problem, which can be viewed as an upper bound on sentence-level accuracy. Interestingly, while excluding neutral documents improve accuracies and F_1-scores of positive and negative sentences, which is not unexpected since the task is made simpler, F_1-scores for neutral sentences are much lower. In the DaS and hard HCRF cases, the models completely fail to predict any neutral sentence-level sentiment.

5 Related work

Latent-variable structured learning models have been investigated recently in the context of sentiment analysis. Nakagawa et al. [20] presented a sentence level model where the

observed information was the polarity of a sentence and the latent variables the nodes from the syntactic dependency tree of the sentence. They showed that such a model can improve sentence level polarity classification. Yessenalina et al. [36] presented a document level model where the latent variables were binary predictions over sentences indicating whether they would be used to classify the document or disregarded. In both these models, the primary goal was to improve the performance of the model on the supervised annotated signal, i.e., sentence level polarity in the former and document level polarity in the latter. The accuracy of the latent variables was never assessed empirically, even though it was argued that they should equate with the sub-sentence or sub-document sentiment of the text under consideration.

This study inverts the evaluation and attempts to assess the accuracy of the latent structure induced from the observed coarse supervision. In fact, one could argue that learning fine-grained sentiment from document level labels is the more relevant question for multiple reasons: 1) document level annotations are the most common naturally observed sentiment signal, e.g., star-rated consumer reviews, 2) fine-grained annotations often require large annotation efforts [34], which have to be undertaken on a domain-by-domain basis, and 3) document level sentiment analysis is too coarse for most sentiment applications, especially those that rely on aggregation across fine-grained topics [13].

Recent work by Chang et al. [6] had the similar goal of learning and evaluating latent structure from high level (or indirect) supervision, though they did not specifically investigate sentiment analysis. In that work supervision came in the form of coarse binary labels, indicating whether an example was valid or invalid. A typical example would be the task of learning the syntactic structure of a sentence, where the only observed information is a binary variable indicating whether the sentence is grammatical. The primary modeling assumption is that all latent structures for invalid instances were themselves invalid. This allowed for an optimization formulation where invalid structures were constrained to have lower scores than the best latent structure for valid instances. Our task differs in that there is no natural notion of invalid instances – all documents have valid fine-grained sentiment structure. As we have shown, this set-up lends itself more towards latent variable models such as HCRFs or structural SVMs with latent variables [37].

6 Conclusions

In this paper we showed that latent variable structured prediction models can effectively learn fine-grained sentiment from coarse-grained supervision. Empirically, reductions in error of up to 20% were observed relative to both lexicon-based and machine-learning baselines. In the common case when document labels are available at test time as well, we observed error reductions close to 30% and over 20%, respectively, relative to the same baselines. In the latter case, our model reduces error with about 8% relative to the strongest baseline. The model we employed, a hidden conditional random field, leaves open a number of further avenues for investigating weak prior knowledge in fine-grained sentiment analysis, most notably semi-supervised learning when small samples of data annotated with fine-grained information are available.

Acknowledgements

The authors would like to thank the anonymous reviewers of the 33rd European Conference on Information Retrieval (ECIR 2011) for their helpful comments on an earlier version of this paper. We are also grateful to Alexandre Passos, who provided advice regarding the use of the bootstrap procedure, to members of the Natural Language Processing group at Google, who provided insightful comments at an early stage of this research, as well as to Jussi Karlgren for his feedback on a draft version of this paper.

The contribution of the first author was in part funded by the Swedish National Graduate School of Language Technology (GSLT).

References

1. Philip Beineke, Trevor Hastie, Christopher Manning, and Shivakumar Vaithyanathan. An exploration of sentiment summarization. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*, 2003.
2. Christopher M. Bishop. *Pattern recognition and machine learning*. Springer New York, 2006.
3. Sasha Blair-Goldensohn, Kerry Hannan, Ryan McDonald, Tyler Neylon, George A. Reis, and Jeff Reynar. Building a sentiment summarizer for local service reviews. In *NLP in the Information Explosion Era*, 2008.
4. John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes, and blenders: Domain adaptation for sentiment classification. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*, 2007.
5. Giuseppe Carenini, Raymond Ng, and Adam Pauls. Multi-document summarization of evaluative text. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*, 2006.
6. Ming-Wei Chang, Vivek Srikumar, Dan Goldwasser, and Dan Roth. Structured output learning with indirect supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.
7. Yejin Choi and Claire Cardie. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2009.
8. Isaac Councill, Ryan McDonald, and Leonid Velikovich. What's great and what's not: Learning to classify the scope of negation for improved sentiment analysis. In *Negation and Speculation in Natural Language Processing*, 2010.
9. Anthony C. Davison and David V. Hinkley. *Bootstrap Methods and Their Applications*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, UK, 1997.
10. Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric Ringger. Pulse: Mining customer opinions from free text. In *Proceedings of the 6th International Symposium on Intelligent Data Analysis (IDA)*, 2005.
11. Andrew B. Goldberg and Xiaojin Zhu. Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In *Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing on the First Workshop on Graph Based Methods for Natural Language Processing*, 2006.
12. Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*, 1997.

13. Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, 2004.
14. Nobuhiro Kaji and Masaru Kitsuregawa. Building lexicon for sentiment analysis from massive collection of HTML documents. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.
15. Soo-Min Kim and Eduard Hovy. Determining the sentiment of opinions. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2004.
16. John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann Publishers Inc., 2001.
17. Yi Mao and Guy Lebanon. Isotonic conditional random fields and local sentiment flow. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.
18. Ryan McDonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*, 2007.
19. Saif Mohammad, Cody Dunne, and Bonnie Dorr. Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2009.
20. Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. Dependency Tree-based Sentiment Classification using CRFs with Hidden Variables. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2010.
21. Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2004.
22. Bo Pang and Lillian Lee. *Opinion mining and sentiment analysis*. Now Publishers, 2008.
23. Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002.
24. Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2005.
25. Ariadna Quattoni, Sybor Wang, Louis-Philippe Morency, Michael Collins, and Trevor Darrell. Hidden conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.
26. Delip Rao and Deepak Ravichandran. Semi-supervised polarity lexicon induction. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*, 2009.
27. Ellen Riloff and Janyce Wiebe. Learning extraction patterns for subjective expressions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2003.
28. Benjamin Snyder and Regina Barzilay. Multiple aspect ranking using the Good Grief algorithm. In *Proceedings of the Joint Conference of the North American Chapter of the Association for Computational Linguistics and Human Language Technologies (NAACL-HLT)*, 2007.
29. Oscar Täckström and Ryan McDonald. Discovering fine-grained sentiment with latent variable structured prediction models. In *Proceedings of the 33rd European Conference on Information Retrieval (ECIR 2011)*, Dublin, Ireland, 2011.
30. Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of the Annual World Wide Web Conference (WWW)*, 2008.

31. Peter Turney. Thumbs up or thumbs down? Sentiment orientation applied to unsupervised classification of reviews. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*, 2002.
32. Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. The viability of web-derived polarity lexicons. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2010.
33. Janyce Wiebe. Learning subjective adjectives from corpora. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2000.
34. Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. In *Language Resources and Evaluation*, volume 39, pages 165–210, 2005.
35. Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2005.
36. Ainur Yessenalina, Yisong Yue, and Claire Cardie. Multi-level structured models for document-level sentiment classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2010.
37. Chun-Nam Yu and Thorsten Joachims. Learning structural svms with latent variables. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2009.
38. Li Zhuang, Feng Jing, and Xiao-Yan Zhu. Movie review mining and summarization. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, 2006.