Multi-level Structured Models for Document-level Sentiment Classification

Ainur Yessenalina
Dept. of Computer Science
Cornell University
Ithaca, NY, USA
ainur@cs.cornell.edu

Yisong Yue
Dept. of Computer Science
Cornell University
Ithaca, NY, USA
yyue@cs.cornell.edu

Claire Cardie
Dept. of Computer Science
Cornell University
Ithaca, NY, USA
cardie@cs.cornell.edu

Abstract

In this paper, we investigate structured models for document-level sentiment classification. When predicting the sentiment of a subjective document (e.g., as positive or negative), it is well known that not all sentences are equally discriminative or informative. But identifying the useful sentences automatically is itself a difficult learning problem. This paper proposes a joint two-level approach for document-level sentiment classification that simultaneously extracts useful (i.e., subjective) sentences and predicts document-level sentiment based on the extracted sentences. Unlike previous joint learning methods for the task, our approach (1) does not rely on gold standard sentence-level subjectivity annotations (which may be expensive to obtain), and (2) optimizes directly for document-level performance. Empirical evaluations on movie reviews and U.S. Congressional floor debates show improved performance over previous approaches.

1 Introduction

Sentiment classification is a well-studied and active research area (Pang and Lee, 2008). One of the main challenges for document-level sentiment categorization is that not every part of the document is equally informative for inferring the sentiment of the whole document. Objective statements interleaved with the subjective statements can be confusing for learning methods, and subjective statements with conflicting sentiment further complicate the document categorization task. For example, authors of movie reviews often devote large sections to (largely objective) descriptions of the plot (Pang and Lee, 2004). In addition, an overall positive review might still include some negative opinions about an actor or the plot.

Early research on document-level sentiment classification employed conventional machine learning techniques for text categorization (Pang et al., 2002). These methods, however, assume that documents are represented via a flat feature vector (e.g., a bag-of-words). As a result, their ability to identify and exploit subjectivity (or other useful) information at the sentence-level is limited.

And although researchers subsequently proposed methods for incorporating sentence-level subjectivity information, existing techniques have some undesirable properties. First, they typically require gold standard sentence-level annotations (McDonald et al. (2007), Mao and Lebanon (2006)). But the cost of acquiring such labels can be prohibitive. Second, some solutions for incorporating sentence-level information lack mechanisms for controlling how errors propagate from the subjective sentence identification subtask to the main document classification task (Pang and Lee, 2004). Finally, solutions that attempt to handle the error propagation problem have done so by explicitly optimizing for the best *combination* of document- and sentence-level classification accuracy (McDonald et al., 2007). Optimizing for this compromise, when the real goal is to maximize only the document-level accuracy, can potentially hurt document-level performance.

In this paper, we propose a joint two-level model to address the aforementioned concerns. We formulate our training objective to directly optimize for

document-level accuracy. Further, we do not require gold standard sentence-level labels for training. Instead, our training method treats sentence-level labels as hidden variables and *jointly learns* to predict the document label and those (subjective) sentences that best "explain" it, thus controlling the propagation of incorrect sentence labels. And by directly optimizing for document-level accuracy, our model learns to solve the sentence extraction subtask only to the extent required for accurately classifying document sentiment. A software implementation of our method is also publicly available.[1]

For the rest of the paper, we will discuss related work, motivate and describe our model, present an empirical evaluation on movie reviews and U.S. Congressional floor debates datasets and close with discussion and conclusions.

2 Related Work

Pang and Lee (2004) first showed that sentence-level extraction can improve document-level performance. They used a cascaded approach by first filtering out objective sentences and performing subjectivity extractions using a global min-cut inference. Afterward, the subjective extracts were converted into inputs for the document-level sentiment classifier. One advantage of their approach is that it avoids the need for explicit subjectivity annotations. However, like other cascaded approaches (e.g., Thomas et al. (2006), Mao and Lebanon (2006)), it can be difficult to control how errors propagate from the sentence-level subtask to the main document classification task.

Instead of taking a cascaded approach, one can directly modify the training of flat document classifiers using lower level information. For instance, Zaidan et al. (2007) used human annotators to mark the "annotator rationales", which are text spans that support the document's sentiment label. These annotator rationales are then used to formulate additional constraints during SVM training to ensure that the resulting document classifier is less confident in classifying a document that does not contain the rationale versus the original document. Yessenalina et al. (2010) extended this approach to use automatically generated rationales.

[1] `http://projects.yisongyue.com/svmsle/`

A natural approach to avoid the pitfalls associated with cascaded methods is to use joint two-level models that simultaneously solve the sentence-level and document-level tasks (e.g., McDonald et al. (2007), Zaidan and Eisner (2008)). Since these models are trained jointly, the sentence-level predictions affect the document-level predictions and vice-versa. However, such approaches typically require sentence-level annotations during training, which can be expensive to acquire. Furthermore, the training objectives are usually formulated as a compromise between sentence-level and document-level performance. If the goal is to predict well at the document-level, then these approaches are solving a much harder problem that is not exactly aligned with maximizing document-level accuracy.

Recently, researchers within both Natural Language Processing (e.g., Petrov and Klein (2007), Chang et al. (2010), Clarke et al. (2010)) and other fields (e.g., Felzenszwalb et al. (2008), Yu and Joachims (2009)) have analyzed joint multi-level models (i.e., models that simultaneously solve the main prediction task along with important subtasks) that are trained using limited or no explicit lower level annotations. Similar to our approach, the lower level labels are treated as hidden or latent variables during training. Although the training process is non-trivial (and in particular requires a good initialization of the hidden variables), it avoids the need for human annotations for the lower level subtasks. Some researchers have also recently applied hidden variable models to sentiment analysis, but they were focused on classifying either phrase-level (Choi and Cardie, 2008) or sentence-level polarity (Nakagawa et al., 2010).

3 Extracting Hidden Explanations

In this paper, we take the view that each document has a subset of sentences that best explains its sentiment. Consider the "annotator rationales" generated by human judges for the movie reviews dataset (Zaidan et al., 2007). Each rationale is a text span that was identified to support (or explain) its parent document's sentiment. Thus, these rationales can be interpreted as (something close to) a ground truth labeling of the explanatory segments. Using a dataset where each document contains only its rationales,

Algorithm 1 Inference Algorithm for (2)

Input: x
Output: (y, s)
$s_+ \leftarrow \text{argmax}_{s \in S(x)} \vec{w}^T \Psi(x, +1, s)$
$s_- \leftarrow \text{argmax}_{s \in S(x)} \vec{w}^T \Psi(x, -1, s)$
if $\vec{w}^T \Psi(x, +1, s_+) > \vec{w}^T \Psi(x, -1, s_-)$ **then**
    Return $(+1, s_+)$
else
    Return $(-1, s_-)$
end if

cross validation experiments using an SVM classifier yields 97.44% accuracy – as opposed to 86.33% accuracy when using the full text of the original documents. Clearly, extracting the best supporting segments can offer a tremendous performance boost.

We are interested in settings where human-extracted explanations such as annotator rationales might not be readily available, or are imperfect. As such, we will formulate the set of extracted sentences as latent or hidden variables in our model. Viewing the extracted sentences as latent variables will pose no new challenges during prediction, since the model is expected to predict all labels at test time. We will leverage recent advances in training latent variable SVMs (Yu and Joachims, 2009) to arrive at an effective training procedure.

4 Model

In this section, we present a two-level document classification model. Although our model makes predictions at both the document and sentence levels, it will be trained (and evaluated) only with respect to document-level performance. We begin by presenting the feature structure and inference method. We will then describe a supervised training algorithm based on structural SVMs, and finally discuss some extensions and design decisions.

Let x denote a document, $y = \pm 1$ denote the sentiment (for us, a binary positive or negative polarity) of a document, and s denote a subset of explanatory sentences in x. Let $\Psi(x, y, s)$ denote a joint feature map that outputs features describing the quality of predicting sentiment y using explanation s for document x. We focus on linear models, so given a (learned) weight vector $\vec{w}$, we can write the quality of predicting y (with explanation s) as

$$F(x, y, s; \vec{w}) = \vec{w}^T \Psi(x, y, s), \quad (1)$$

and a document-level sentiment classifier as

$$h(x; \vec{w}) = \underset{y=\pm 1}{\text{argmax}} \max_{s \in S(x)} F(x, y, s; \vec{w}), \quad (2)$$

where $S(x)$ denotes the collection of feasible explanations (e.g., subsets of sentences) for x.

Let x^j denote the j-th sentence of x. We propose the following instantiation of (1),

$$\vec{w}^T \Psi(x, y, s) = \frac{1}{N(x)} \sum_{j \in s} y \cdot \vec{w}_{pol}^T \psi_{pol}(x^j) + \vec{w}_{subj}^T \psi_{subj}(x^j), \quad (3)$$

where the first term in the summation captures the quality of predicting polarity y on sentences in s, the second term captures the quality of predicting s as the subjective sentences, and $N(x)$ is a normalizing factor (which will be discussed in more detail in Section 4.3). We represent the weight vector as

$$\vec{w} = \begin{bmatrix} \vec{w}_{pol} \\ \vec{w}_{subj} \end{bmatrix}, \quad (4)$$

and $\psi_{pol}(x^j)$ and $\psi_{subj}(x^j)$ denote the polarity and subjectivity features of sentence x^j, respectively. Note that ψ_{pol} and ψ_{subj} are disjoint by construction, i.e., $\psi_{pol}^T \psi_{subj} = 0$. We will present extensions in Section 4.5.

For example, suppose ψ_{pol} and ψ_{subj} were both bag-of-words feature vectors. Then we might learn a high weight for the feature corresponding to the word "think" in ψ_{subj} since that word is indicative of the sentence being subjective (but not necessarily indicating positive or negative polarity).

4.1 Making Predictions

Algorithm 1 describes our inference procedure. Recall from (2) that our hypothesis function predicts the sentiment label that maximizes (3). To do this, we compare the best set of sentences that explains a positive polarity prediction with the best set that explains a negative polarity prediction.

We now specify the structure of $S(x)$. In this paper, we use a cardinality constraint,

$$S(x) = \{s \subseteq \{1, \ldots, |x|\} \ : \ |s| \leq f(|x|)\}, \quad (5)$$

Algorithm 2 Training Algorithm for OP 1

1: Input: $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ *//training data*
2: Input: C *//regularization parameter*
3: Input: $(s_1, \ldots, s_N)$ *//initial guess*
4: $\vec{w} \leftarrow$ *SSVMSolve*$(C, \{(x_i, y_i, s_i)\}_{i=1}^N)$
5: **while** $\vec{w}$ not converged **do**
6: **for** $i = 1, \ldots, N$ **do**
7: $s_i \leftarrow \text{argmax}_{s \in S(x_i)} \vec{w}^T \Psi(x_i, y_i, s)$
8: **end for**
9: $\vec{w} \leftarrow$ *SSVMSolve*$(C, \{(x_i, y_i, s_i)\}_{i=1}^N)$
10: **end while**
11: Return $\vec{w}$

where $f(|x|)$ is a function that depends only on the number of sentences in x. For example, a simple function is $f(|x|) = |x| \cdot 0.3$, indicating that at most 30% of the sentences in x can be subjective.

Using this definition of $S(x)$, we can then compute the best set of subjective sentences for each possible y by computing the joint subjectivity and polarity score of each sentence x^j in isolation,

$$y \cdot \vec{w}_{pol}^T \psi_{pol}(x^j) + \vec{w}_{subj}^T \psi_{subj}(x^j),$$

and selecting the top $f(|x|)$ as s (or fewer, if there are fewer than $f(|x|)$ that have positive joint score).

4.2 Training

For training, we will use an approach based on latent variable structural SVMs (Yu and Joachims, 2009).

Optimization Problem 1.

$$\min_{\vec{w}, \xi \geq 0} \frac{1}{2}\|\vec{w}\|^2 + \frac{C}{N}\sum_{i=1}^{N} \xi_i \tag{6}$$

s.t. $\forall i$:

$$\max_{s \in S_i} \vec{w}^T \Psi(x_i, y_i, s) \geq \max_{s' \in S(x_i)} \vec{w}^T \Psi(x_i, -y_i, s') + 1 - \xi_i \tag{7}$$

OP 1 optimizes the standard SVM training objective for binary classification. Each training example has a corresponding constraint (7), which is quantified over the best possible explanation of the training polarity label. Note that we never observe the true explanation for the training labels; they are the hidden or latent variables. The hidden variables are also ignored in the objective function.

As a result, one can interpret OP 1 to be directly optimizing a trade-off between model complexity (as measured using the 2-norm) and document-level classification error in the training set. This has two main advantages over related training approaches. First, it solves the multi-level problem jointly as opposed to separately, which avoids introducing difficult to control propagation errors. Second, it does not require solving the sentence-level task perfectly, and also does not require precise sentence-level training labels. In other words, our goal is to learn to identify the informative (subjective) sentences that best explain the training labels to the extent required for good document classification performance.

OP 1 is non-convex because of the constraints (7). To solve OP 1, we use the combination of the CCCP algorithm (Yuille and Rangarajan, 2003) with cutting plane training of structural SVMs (Joachims et al., 2009), as proposed in Yu and Joachims (2009). Suppose each constraint (7) is replaced by

$$\vec{w}^T \Psi(x_i, y_i, s_i) \geq \max_{s' \in S(x_i)} \vec{w}^T \Psi(x_i, -y_i, s') + 1 - \xi_i,$$

where s_i is some fixed explanation (e.g., an initial guess of the best explanation). Then OP 1 reduces to a standard structural SVM, which can be solved efficiently (Joachims et al., 2009). Algorithm 2 describes our training procedure. Starting with an initial guess s_i for each training example, the training procedure alternates between solving an instance of the resulting structural SVM (called *SSVMSolve* in Algorithm 2) using the currently best known explanations s_i (Line 9), and making a new guess of the best explanations (Line 7). Yu and Joachims (2009) showed that this alternating procedure for training latent variable structural SVMs is an instance of the CCCP procedure (Yuille and Rangarajan, 2003), and so is guaranteed to converge to a local optimum.

For our experiments, we do not train until convergence, but instead use performance on a validation set to choose the halting iteration. Since OP 1 is non-convex, a good initialization is necessary. To generate the initial explanations, one can use an off-the-shelf sentiment classifier such as OpinionFinder[2] (Wilson et al., 2005). For some datasets, there exist documents with annotated sentences, which we

[2] `http://www.cs.pitt.edu/mpqa/opinionfinderrelease/`

can treat either as the ground truth or another (very good) initial guess of the explanatory sentences.

4.3 Feature Representation

Like any machine learning approach, we must specify a useful set of features for the ψ vectors described above. We will consider two types of features.

Bag-of-words. Perhaps the simplest approach is to define ψ using a bag-of-words feature representation, with one feature corresponding to each word in the active lexicon of the corpus. Using such a feature representation might allow us to learn which words have high polarity (e.g., "great") and which are indicative of subjective sentences (e.g., "opinion").

Sentence properties. We can incorporate many useful features to describe sentence subjectivity. For example, subjective sentences might densely populate the end of a document, or exhibit spatial coherence (so features describing previous sentences might be useful for classifying the current sentence). Such features cannot be compactly incorporated into flat models that ignore the document structure.

For our experiments, we normalize each ψ_{subj} and ψ_{pol} to have unit 2-norm.

Joint Feature Normalization. Another design decision is the choice of normalization $N(x)$ in (3). Two straightforward choices are $N(x) = f(|x|)$ and $N(x) = \sqrt{f(|x|)}$, where $f(|x|)$ is the size constraint as described in (5). In our experiments we tried both and found the square root normalization to work better in practice; therefore all the experimental results are reported using $N(x) = \sqrt{f(|x|)}$. The appendix contains an analysis that sheds light on when square root normalization can be useful.

4.4 Incorporating Proximity Information

As mentioned in Section 4.3, it is possible (and likely) for subjective sentences to exhibit spatial coherence (e.g., they might tend to group together). To exploit this structure, we will expand the feature space of ψ_{subj} to include both the words of the current and previous sentence as follows,

$$\psi_{subj}(x, j) = \begin{bmatrix} \psi_{subj}(x^j) \\ \psi_{subj}(x^{j-1}) \end{bmatrix}.$$

The corresponding weight vector can be written as

$$\vec{w}'_{subj} = \begin{bmatrix} \vec{w}_{subj} \\ \vec{w}_{prevSubj} \end{bmatrix}.$$

By adding these features, we are essentially assuming that the words of the previous sentence are predictive of the subjectivity of the current sentence.

Alternative approaches include explicitly accounting for this structure by treating subjective sentence extraction as a sequence-labeling problem, such as in McDonald et al. (2007). Such structure formulations can be naturally encoded in the joint feature map. Note that the inference procedure in Algorthm 1 is still tractable, since it reduces to comparing the best sequence of subjective/objective sentences that explains a positive sentiment versus the best sequence that explains a negative sentiment. For this study, we chose not to examine this more expressive yet more complex structure.

4.5 Extensions

Though our initial model (3) is simple and intuitive, performance can depend heavily on the quality of latent variable initialization and the quality of the feature structure design. Consider the case where the initialization contains only objective sentences that do not convey any sentiment. Then all the features initially available during training are generated from these objective sentences and are thus useless for sentiment classification. In other words, too much useful information has been suppressed for the model to make effective decisions. To hedge against learning poor models due to using a poor initialization and/or a suboptimal feature structure, we now propose extensions that incorporate information from the entire document.

We identify the following desirable properties that any such extended model should satisfy:

(A) The model should be linear.

(B) The model should be trained jointly.

(C) The component that models the entire document should influence which sentences are extracted.

The first property stems from the fact that our approach relies on linear models. The second property is desirable since joint training avoids error propagation that can be difficult to control. The third property deals with the information suppression issue.

4.5.1 Regularizing Relative to a Prior

We first consider a model that satisfies properties (A) and (C). Using the representation in (4), we propose a training procedure that regularize $\vec{w}_{pol}$ relative to a prior model. Suppose we have a weight vector $\vec{w}_0$ which indicated the a priori guess of the contribution of each corresponding feature, then we can train our model using OP 2,

Optimization Problem 2.

$$\min_{\vec{w},\xi\geq 0} \frac{1}{2}\|\vec{w}-\vec{w}_0\|^2 + \frac{C}{N}\sum_{i=1}^{N}\xi_i$$

s.t. $\forall i$:

$$\max_{s\in S_i} \vec{w}^T\Psi(x_i,y_i,s) \geq \max_{s'\in S(x_i)} \vec{w}^T\Psi(x_i,-y_i,s') + 1 - \xi_i$$

For our experiments, we use

$$\vec{w}_0 = \begin{bmatrix} \vec{w}_{doc} \\ 0 \end{bmatrix},$$

where $\vec{w}_{doc}$ denotes a weight vector trained to classify the polarity of entire documents. Then one can interpret OP 2 as enforcing that the polarity weights $\vec{w}_{pol}$ not be too far from $\vec{w}_{doc}$. Note that $\vec{w}_0$ must be available before training. Therefore this approach does not satisfy property (B).

4.5.2 Extended Feature Space

One simple way to satisfy all three aforementioned properties is to jointly model not only polarity and subjectivity of the extracted sentences, but also polarity of the entire document. Let $\vec{w}_{doc}$ denote the weight vector used to model the polarity of entire document x (so the document polarity score is then $\vec{w}_{doc}^T\psi_{pol}(x)$). We can also incorporate this weight vector into our structured model to compute a smoothed polarity score of each sentence via $\vec{w}_{doc}^T\psi_{pol}(x^j)$. Following this intuition, we propose the following structured model,

$$\vec{w}^T\Psi(x,y,s) = \frac{y}{N(x)}\left(\sum_{j\in s}\left(\vec{w}_{pol}^T\psi_{pol}(x^j) + \vec{w}_{doc}^T\psi_{pol}(x^j)\right)\right) + \frac{1}{N(x)}\left(\sum_{j\in s}\vec{w}_{subj}^T\psi_{subj}(x^j)\right) + y\cdot\vec{w}_{doc}^T\psi_{pol}(x)$$

where the weight vector is now

$$\vec{w} = \begin{bmatrix} \vec{w}_{pol} \\ \vec{w}_{subj} \\ \vec{w}_{doc} \end{bmatrix}.$$

Training this model via OP 1 achieves that $\vec{w}_{doc}$ is (1) used to model the polarity of the entire document, and (2) used to compute a smoothed estimate of the polarity of the extracted sentences. This satisfies all three properties (A), (B), and (C), although other approaches are also possible.

5 Experiments

5.1 Experimental Setup

We evaluate our methods using the Movie Reviews and U.S. Congressional Floor Debates datasets, following the setup used in previous work for comparison purposes.[3]

Movie Reviews. We use the movie reviews dataset from Zaidan et al. (2007) that was originally released by Pang and Lee (2004). This version contains annotated rationales for each review, which we use to generate an additional initialization during training (described below). We follow exactly the experimental setup used in Zaidan et al. (2007).[4]

U.S. Congressional Floor Debates. We also use the U.S. Congressional floor debates transcripts from Thomas et al. (2006). The data was extracted from GovTrack (http://govtrack.us), which has all available transcripts of U.S. floor debates in the House of Representatives in 2005. As in previous work, only debates with discussions of "controversial" bills were considered (where the losing side had at least 20% of the speeches). The goal is to predict the vote ("yea" or "nay") for the speaker of each speech segment. For our experiments, we evaluate our methods using the speaker-based speech-segment classification setting as described in Thomas et al. (2006).[5]

[3] Datasets in the required format for *SVM*^{sle} are available at `http://www.cs.cornell.edu/~ainur/data.html`

[4] Since the rationale annotations are available for nine out of 10 folds, we used the 10-th fold as the blind test set. We trained nine different models on subsets of size eight, used the remaining fold as the validation set, and then measured the average performance on the final test set.

[5] In the other setting described in Thomas et al. (2006) (segment-based speech-segment classification), around 39% of

Table 1: Summary of the experimental results for the Movie Reviews (top) and U.S. Congressional Floor Debates (bottom) datasets using *SVM*^{sle}, *SVM*^{sle} w/ Prior and *SVM*$^{sle}_{fs}$ with and without proximity features.

INITIALIZATION	*SVM*^{sle}	+ Prox.Feat.	*SVM*^{sle} w/ Prior	+ Prox.Feat.	*SVM*$^{sle}_{fs}$	+ Prox.Feat.
Random 30%	87.22	85.44	87.61	87.56	89.50	88.22
Last 30%	89.72 $*$	88.83	90.50 $*$	90.00 $*$	91.06 $\star$	91.22 $\star$
OpinionFinder	91.28 $\star$	90.89 $\star$	91.72 $\star$	93.22 $\star$	92.50 $\star$	92.39 $\star$
Annot.Rationales	91.61 $\star$	92.00 $\star$	92.67 $\star$	92.00 $\star$	92.28 $\star$	93.22 $\star$

INITIALIZATION	*SVM*^{sle}	+ Prox.Feat.	*SVM*^{sle} w/ Prior	+ Prox.Feat.	*SVM*$^{sle}_{fs}$	+ Prox.Feat.
Random 30%	78.84	73.14	78.49	76.40	77.33	73.84
Last 30%	73.26	73.95	71.51	73.60	67.79	73.37
OpinionFinder	77.33	79.53	77.09	78.60	77.67	77.09

– For Movie Reviews, the SVM baseline accuracy is 88.56%. A $\star$ (or $*$) indicates statically significantly better performance than baseline according to the paired t-test with $p < 0.001$ (or $p < 0.05$).
– For U.S. Congressional Floor Debates, the SVM baseline accuracy is 70.00%. Statistical significance cannot be calculated because the data comes in a single split.

Since our training procedure solves a non-convex optimization problem, it requires an initial guess of the explanatory sentences. We use an explanatory set size (5) of 30% of the number of sentences in each document, $L = \lceil 0.3 \cdot |x| \rceil$, with a lower cap of 1. We generate initializations using OpinionFinder (Wilson et al., 2005), which were shown to be a reasonable substitute for human annotations in the Movie Reviews dataset (Yessenalina et al., 2010).[6]

We consider two additional (baseline) methods for initialization: using a random set of sentences, and using the last 30% of sentence in the document. In the Movie Reviews dataset, we also use sentences containing human-annotator rationales as a final initialization option. No such manual annotations are available for the Congressional Debates.

5.2 Experimental Results

We evaluate three versions of our model: the initial model (3) which we call *SVM*^{sle} (**SVM**s for **S**entiment classification with **L**atent **E**xplanations), *SVM*^{sle} regularized relative to a prior as described in Section 4.5.1 which we refer to as *SVM*^{sle} w/ Prior,[7] and the feature smoothing model described in Section 4.5.2 which we call *SVM*$^{sle}_{fs}$. Due to the difficulty of selecting a good prior, we expect *SVM*$^{sle}_{fs}$ to exhibit the most robust performance.

Table 1 shows a comparison of our proposed methods on the two datasets. We observe that *SVM*$^{sle}_{fs}$ provides both strong and robust performance. The performance of *SVM*^{sle} is generally better when trained using a prior than not in the Movie Reviews dataset. Both extensions appear to hurt performance in the U.S. Congressional Floor Debates dataset. Using OpinionFinder to initialize our training procedure offers good performance across both datasets, whereas the baseline initializations exhibit more erratic performance behavior.[8] Unsurprisingly, initializing using human annotations (in the Movie Reviews dataset) can offer further improvement. Adding proximity features (as described in Section 4.4) in general seems to improve performance when using a good initialization, and hurts performance otherwise.

the documents in the whole dataset contain only 1-3 sentences, making it an uninteresting setting to analyze with our model.

[6]We select all sentences whose majority vote of OpinionFinder word-level polarities matches the document's sentiment. If there are fewer than L sentences, we add sentences starting from the end of the document. If there are more, we remove sentences starting from the beginning of the document.

[7]We either used the same value of C to train both standard SVM model and *SVM*^{sle} w/ Prior or used the best standard SVM model on the validation set to train *SVM*^{sle} w/ Prior. We chose the combination that works the best on the validation set.

[8]Using the random initialization on the U.S. Congressional Floor Debates dataset offers surprisingly good performance.

Table 2: Comparison of SVM_{fs}^{sle} with previous work on the Movie Reviews dataset. We considered two settings: when human annotations are available (Annot. Labels), and when they are unavailable (No Annot. Labels).

	METHOD	ACC
Baseline	SVM	88.56
Annot. Labels	Zaidan et al. (2007)	92.20
	SVM_{fs}^{sle}	92.28
	SVM_{fs}^{sle}+ Prox.Feat.	93.22
No Annot. Labels	Yessenalina et al. (2010)	91.78
	SVM_{fs}^{sle}	92.50
	SVM_{fs}^{sle}+Prox.Feat.	92.39

Table 3: Comparison of SVM_{fs}^{sle} with previous work on the U.S. Congressional Floor Debates dataset for the speaker-based segment classification task.

	METHOD	ACC
Baseline	SVM	70.00
Prior work	Thomas et al. (2006)	71.28
	Bansal et al. (2008)	75.00
Our work	SVM_{fs}^{sle}	77.67
	SVM_{fs}^{sle}+ Prox.Feat.	77.09

Tables 2 and 3 show a comparison of SVM_{fs}^{sle} with previous work on the Movie Reviews and U.S. Congressional Floor Debates datasets, respectively. For the Movie Reviews dataset, we considered two settings: when human annotations are available, and when they are not (in which case we initialized using OpinionFinder). For the U.S. Congressional Floor Debates dataset we used only the latter setting, since there are no annotations available for this dataset. In all cases we observe SVM_{fs}^{sle} showing improved performance compared to previous results.

Training details. We tried around 10 different values for C parameter, and selected the final model based on the validation set. The training procedure alternates between training a standard structural SVM model and using the subsequent model to re-label the latent variables. We selected the halting iteration of the training procedure using the validation set. When initializing using human annotations for the Movie Reviews dataset, the halting iteration is typically the first iteration, whereas the halting iteration is typically chosen from a later iteration when initializing using OpinionFinder.

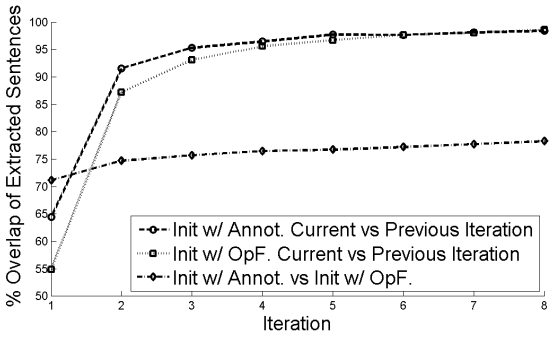

Figure 1: Overlap of extracted sentences from different SVM_{fs}^{sle} models on the Movie Reviews training set.

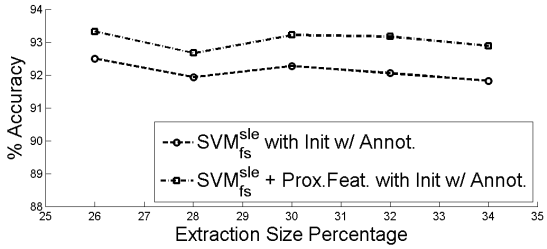

Figure 2: Test accuracy on the Movie Reviews dataset for SVM_{fs}^{sle} while varying extraction size.

Figure 1 shows the per-iteration overlap of extracted sentences from SVM_{fs}^{sle} models initialized using OpinionFinder and human annotations on the Movie Reviews training set. We can see that training has approximately converged after about 10 iterations.[9] We can also see that both models iteratively learn to extract sentences that are more similar to each other than their respective initializations (the overlap between the two initializations is 57%). This is an indicator that our learning problem, despite being non-convex and having multiple local optima, has a reasonably large "good" region that can be approached using different initialization methods.

Varying the extraction size. Figure 2 shows how accuracy on the test set of SVM_{fs}^{sle} changes on the Movie Reviews dataset as a function of varying the extraction size $f(|x|)$ from (5). We can see that performance changes smoothly[10] (and so is robust), and that one might see further improvement from more

[9] The number of iterations required to converge is an upper bound on the number of iterations from which to choose the halting iteration (based on a validation set).

[10] The smoothness will depend on the initialization.

Table 4: Example "yea" speech with ***Latent Explanations*** from the U.S. Congressional Floor Debates dataset predicted by SVM^{sle}_{fs} with OpinionFinder initialization. Latent Explanations are preceded by solid circles with numbers denoting their preference order (1 being most preferred by SVM^{sle}_{fs}). The five least subjective sentences are preceded by circles with numbers denoting the subjectivity order (1 being least subjective according to SVM^{sle}_{fs}).

❷ ***Mr. Speaker, I am proud to stand on the house floor today to speak in favor of the Stem Cell Research Enhancement Act, legislation which will bring hope to millions of people suffering from disease in this nation.*** ❸ ***I want to thank Congresswoman Degette and Congressman Castle for their tireless work in bringing this bill to the house floor for a vote.***

① The discovery of embryonic stem cells is a major scientific breakthrough. ⑤ Embryonic stem cells have the potential to form any cell type in the human body. This could have profound implications for diseases such as Alzheimer's, Parkinson's, various forms of brain and spinal cord disorders, diabetes, and many types of cancer. ② According to the Coalition for the Advancement of Medical Research, there are at least 58 diseases which could potentially be cured through stem cell research.

That is why more than 200 major patient groups, scientists, and medical research groups and 80 Nobel Laureates support the Stem Cell Research Enhancement Act. ③ They know that this legislation will give us a chance to find cures to diseases affecting 100 million Americans.

I want to make clear that I oppose reproductive cloning, as we all do. I have voted against it in the past. ❹ ***However, that is vastly different from stem cell research and as an ovarian cancer survivor, I am not going to stand in the way of science.***

Permitting peer-reviewed Federal funds to be used for this research, combined with public oversight of these activities, is our best assurance that research will be of the highest quality and performed with the greatest dignity and moral responsibility. The policy President Bush announced in August 2001 has limited access to stem cell lines and has stalled scientific progress.

As a cancer survivor, I know the desperation these families feel as they wait for a cure. ④ This congress must not stand in the way of that progress. ❺ ***We have an opportunity to change the lives of millions, and I hope we take it.*** ❶ ***I urge my colleagues to support this legislation.***

careful tuning of the size constraint.

Examining an example prediction. Our proposed methods are not designed to extract interpretable explanations, but examining the extracted explanations might still yield meaningful information. Table 4 contains an example speech from the U.S. Congressional Floor Debates test set, with Latent Explanations found by SVM^{sle}_{fs} highlighted in boldface. This speech was made in support of the Stem Cell Research Enhancement Act. For comparison, Table 4 also shows the five least subjective sentences according to SVM^{sle}_{fs}. Notice that most of these "objective" sentences can plausibly belong to speeches made in opposition to bills that limit stem cell research funding. That is, they do not clearly indicate the speaker's stance towards the specific bill in question. We can thus see that our approach can indeed learn to infer sentences that are essential to understanding the document-level sentiment.

6 Discussion

Making good structural assumptions simplifies the development process. Compared to methods that modify the training of flat document classifiers (e.g., Zaidan et al. (2007)), our approach uses fewer parameters, leading to a more compact and faster training stage. Compared to methods that use a cascaded approach (e.g., Pang and Lee (2004)), our approach is more robust to errors in the lower-level subtask due to being a joint model.

Introducing latent variables makes the training procedure more flexible by not requiring lower-level labels, but does require a good initialization (i.e., a reasonable substitute for the lower-level labels). We believe that the widespread availability of off-the-shelf sentiment lexicons and software, despite being developed for a different domain, makes this issue less of a concern, and in fact creates an opportunity for approaches like ours to have real impact.

One can incorporate many types of sentence-level information that cannot be directly incorporated into a flat model. Examples include scores from another sentence-level classifier (e.g., from Nakagawa et. al (2010)) or combining phrase-level polarity scores (e.g., from Choi and Cardie (2008)) for each sentence, or features that describe the position of the sentence in the document.

Most prior work on the U.S. Congressional Floor Debates dataset focused on using relationships between speakers such as agreement (Thomas et al., 2006; Bansal et al., 2008), and used a global min-cut inference procedure. However, they require all

test instances to be known in advance (i.e., their formulations are transductive). Our method is not limited to the transductive setting, and instead exploits a different and complementary structure: the latent explanation (i.e., only some sentences in the speech are indicative of the speaker's vote).

In a sense, the joint feature structure used in our model is the simplest that could be used. Our model makes no explicit structural dependencies between sentences, so the choice of whether to extract each sentence is essentially made independently of other sentences in the document. More sophisticated structures can be used if appropriate. For instance, one can formulate the sentence extraction task as a sequence labeling problem similar to (McDonald et al., 2007), or use a more expressive graphical model such as in (Pang and Lee, 2004; Thomas et al., 2006). So long as the global inference procedure is tractable or has a good approximation algorithm, then the training procedure is guaranteed to converge with rigorous generalization guarantees (Finley and Joachims, 2008). Since any formulation of the extraction subtask will suppress information for the main document-level task, one must take care to properly incorporate smoothing if necessary.

Another interesting direction is training models to predict not only sentiment polarity, but also whether a document is objective. For example, one can pose a three class problem ("positive", "negative", "objective"), where objective documents might not necessarily have a good set of (subjective) explanatory sentences, similar to (Chang et al., 2010).

7 Conclusion

We have presented latent variable structured models for the document sentiment classification task. These models do not rely on sentence-level annotations, and are trained jointly (over both the document and sentence levels) to directly optimize document-level accuracy. Experiments on two standard sentiment analysis datasets showed improved performance over previous results.

Our approach can, in principle, be applied to any classification task that is well modeled by jointly solving an extraction subtask. However, as evidenced by our experiments, proper training does require a reasonable initial guess of the extracted explanations, as well as ways to mitigate the risk of the extraction subtask suppressing too much information (such as via feature smoothing).

Acknowledgments

This work was supported in part by National Science Foundation Grants BCS-0904822, BCS-0624277, IIS-0535099; by a gift from Google; and by the Department of Homeland Security under ONR Grant N0014-07-1-0152. The second author was also supported in part by a Microsoft Research Graduate Fellowship. The authors thank Yejin Choi, Thorsten Joachims, Nikos Karampatziakis, Lillian Lee, Chun-Nam Yu, and the anonymous reviewers for their helpful comments.

Appendix

Recall that all the ψ_{subj} and ψ_{pol} vectors have unit 2-norm, which is assumed here to be desirable. We now show that using $N(x) = \sqrt{f(|x|)}$ achieves a similar property for $\Psi(x, y, s)$. We can write the squared 2-norm of $\Psi(x, y, s)$ as

$$|\Psi(x,y,s)|^2 = \frac{1}{N(x)^2}\left[\sum_{j\in s} y\cdot\psi_{pol}(x^j) + \psi_{subj}(x^j)\right]^2$$
$$= \frac{1}{f(|x|)}\left[\left(\sum_{j\in s}\psi_{pol}(x^j)\right)^2 + \left(\sum_{j\in s}\psi_{subj}(x^j)\right)^2\right],$$

where the last equality follows from the fact that

$$\psi_{pol}(x^j)^T\psi_{subj}(x^j) = 0,$$

due to the two vectors using disjoint feature spaces by construction. The summation of the $\psi_{pol}(x^j)$ terms is written as

$$\left(\sum_{j\in s}\psi_{pol}(x^j)\right)^2 = \sum_{j\in s}\sum_{i\in s}\psi_{pol}(x^j)^T\psi_{pol}(x^i)$$
$$\approx \sum_{j\in s}\psi_{pol}(x^j)^T\psi_{pol}(x^j) \qquad (8)$$
$$= \sum_{j\in s} 1 \le f(|x|),$$

where (8) follows from the sparsity assumption that

$$\forall i \neq j:\ \psi_{pol}(x^j)^T\psi_{pol}(x^i) \approx 0.$$

A similar argument applies for the $\psi_{subj}(x^j)$ terms. Thus, by choosing $N(x) = \sqrt{f(|x|)}$ the joint feature vectors $\Psi(x, y, s)$ will have approximately equal magnitude as measured using the 2-norm.

References

Mohit Bansal, Claire Cardie, and Lillian Lee. 2008. The power of negative thinking: Exploiting label disagreement in the min-cut classification framework. In *International Conference on Computational Linguistics (COLING)*.

Ming-Wei Chang, Dan Goldwasser, Dan Roth, and Vivek Srikumar. 2010. Discriminative learning over constrained latent representations. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Yejin Choi and Claire Cardie. 2008. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Empirical Methods in Natural Language Processing (EMNLP)*.

James Clarke, Dan Goldwasser, Ming-Wei Chang, and Dan Roth. 2010. Driving semantic parsing from the world's response. In *ACL Conference on Natural Language Learning (CoNLL)*, July.

Pedro Felzenszwalb, David McAllester, and Deva Ramanan. 2008. A discriminatively trained, multiscale, deformable part model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Thomas Finley and Thorsten Joachims. 2008. Training structural svms when exact inference is intractable. In *International Conference on Machine Learning (ICML)*.

Thorsten Joachims, Thomas Finley, and Chun-Nam Yu. 2009. Cutting plane training of structural svms. *Machine Learning*, 77(1):27–59.

Yi Mao and Guy Lebanon. 2006. Isotonic conditional random fields and local sentiment flow. In *Neural Information Processing Systems (NIPS)*.

Ryan McDonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. 2007. Structured models for fine-to-coarse sentiment analysis. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. 2010. Dependency tree-based sentiment classification using crfs with hidden variables. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Slav Petrov and Dan Klein. 2007. Discriminative log-linear grammars with latent variables. In *Neural Information Processing Systems (NIPS)*.

Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Ainur Yessenalina, Yejin Choi, and Claire Cardie. 2010. Automatically generating annotator rationales to improve sentiment classification. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Chun-Nam Yu and Thorsten Joachims. 2009. Learning structural svms with latent variables. In *International Conference on Machine Learning (ICML)*.

Alan L. Yuille and Anand Rangarajan. 2003. The concave-convex procedure. *Neural Computation*, 15(4):915–936, April.

Omar F. Zaidan and Jason Eisner. 2008. Modeling annotators: a generative approach to learning from annotator rationales. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Omar F. Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.