Structured Models for Fine-to-Coarse Sentiment Analysis

Ryan McDonald[*] **Kerry Hannan** **Tyler Neylon** **Mike Wells** **Jeff Reynar**
Google, Inc.
76 Ninth Avenue
New York, NY 10011
[*]Contact email: ryanmcd@google.com

Abstract

In this paper we investigate a structured model for jointly classifying the sentiment of text at varying levels of granularity. Inference in the model is based on standard sequence classification techniques using constrained Viterbi to ensure consistent solutions. The primary advantage of such a model is that it allows classification decisions from one level in the text to influence decisions at another. Experiments show that this method can significantly reduce classification error relative to models trained in isolation.

1 Introduction

Extracting sentiment from text is a challenging problem with applications throughout Natural Language Processing and Information Retrieval. Previous work on sentiment analysis has covered a wide range of tasks, including polarity classification (Pang et al., 2002; Turney, 2002), opinion extraction (Pang and Lee, 2004), and opinion source assignment (Choi et al., 2005; Choi et al., 2006). Furthermore, these systems have tackled the problem at different levels of granularity, from the document level (Pang et al., 2002), sentence level (Pang and Lee, 2004; Mao and Lebanon, 2006), phrase level (Turney, 2002; Choi et al., 2005), as well as the speaker level in debates (Thomas et al., 2006). The ability to classify sentiment on multiple levels is important since different applications have different needs. For example, a summarization system for product reviews might require polarity classification at the sentence or phrase level; a question answering system would most likely require the sentiment of paragraphs; and a system that determines which articles from an online news source are editorial in nature would require a document level analysis.

This work focuses on models that jointly classify sentiment on multiple levels of granularity. Consider the following example,

> *This is the first Mp3 player that I have used ... I thought it sounded great ... After only a few weeks, it started having trouble with the earphone connection ... I won't be buying another.*
>
> Mp3 player review from Amazon.com

This excerpt expresses an overall negative opinion of the product being reviewed. However, not all parts of the review are negative. The first sentence merely provides some context on the reviewer's experience with such devices and the second sentence indicates that, at least in one regard, the product performed well. We call the problem of identifying the sentiment of the document and of all its subcomponents, whether at the paragraph, sentence, phrase or word level, *fine-to-coarse sentiment analysis*.

The simplest approach to fine-to-coarse sentiment analysis would be to create a separate system for each level of granularity. There are, however, obvious advantages to building a single model that classifies each level in tandem. Consider the sentence,

> *My 11 year old daughter has also been using it and it is a lot harder than it looks.*

In isolation, this sentence appears to convey negative sentiment. However, it is part of a favorable review

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 432–439,
Prague, Czech Republic, June 2007.

for a piece of fitness equipment, where *hard* essentially means *good workout*. In this domain, *hard*'s sentiment can only be determined in context (i.e., *hard* to assemble versus a *hard* workout). If the classifier knew the overall sentiment of a document, then disambiguating such cases would be easier.

Conversely, document level analysis can benefit from finer level classification by taking advantage of common discourse cues, such as the last sentence being a reliable indicator for overall sentiment in reviews. Furthermore, during training, the model will not need to modify its parameters to explain phenomena like the typically positive word *great* appearing in a negative text (as is the case above). The model can also avoid overfitting to features derived from neutral or objective sentences. In fact, it has already been established that sentence level classification can improve document level analysis (Pang and Lee, 2004). This line of reasoning suggests that a cascaded approach would also be insufficient. Valuable information is passed in both directions, which means any model of fine-to-coarse analysis should account for this.

In Section 2 we describe a simple structured model that jointly learns and infers sentiment on different levels of granularity. In particular, we reduce the problem of joint sentence and document level analysis to a sequential classification problem using constrained Viterbi inference. Extensions to the model that move beyond just two-levels of analysis are also presented. In Section 3 an empirical evaluation of the model is given that shows significant gains in accuracy over both single level classifiers and cascaded systems.

1.1 Related Work

The models in this work fall into the broad class of global structured models, which are typically trained with structured learning algorithms. Hidden Markov models (Rabiner, 1989) are one of the earliest structured learning algorithms, which have recently been followed by discriminative learning approaches such as conditional random fields (CRFs) (Lafferty et al., 2001; Sutton and McCallum, 2006), the structured perceptron (Collins, 2002) and its large-margin variants (Taskar et al., 2003; Tsochantaridis et al., 2004; McDonald et al., 2005; Daumé III et al., 2006). These algorithms are usually applied to sequential labeling or chunking, but have also been applied to parsing (Taskar et al., 2004; McDonald et al., 2005), machine translation (Liang et al., 2006) and summarization (Daumé III et al., 2006).

Structured models have previously been used for sentiment analysis. Choi et al. (2005, 2006) use CRFs to learn a global sequence model to classify and assign sources to opinions. Mao and Lebanon (2006) used a sequential CRF regression model to measure polarity on the sentence level in order to determine the *sentiment flow* of authors in reviews. Here we show that fine-to-coarse models of sentiment can often be reduced to the sequential case.

Cascaded models for fine-to-coarse sentiment analysis were studied by Pang and Lee (2004). In that work an initial model classified each sentence as being subjective or objective using a global mincut inference algorithm that considered local labeling consistencies. The top subjective sentences are then input into a standard document level polarity classifier with improved results. The current work differs from that in Pang and Lee through the use of a single joint structured model for both sentence and document level analysis.

Many problems in natural language processing can be improved by learning and/or predicting multiple outputs jointly. This includes parsing and relation extraction (Miller et al., 2000), entity labeling and relation extraction (Roth and Yih, 2004), and part-of-speech tagging and chunking (Sutton et al., 2004). One interesting work on sentiment analysis is that of Popescu and Etzioni (2005) which attempts to classify the sentiment of phrases with respect to possible product features. To do this an iterative algorithm is used that attempts to globally maximize the classification of all phrases while satisfying local consistency constraints.

2 Structured Model

In this section we present a structured model for fine-to-coarse sentiment analysis. We start by examining the simple case with two-levels of granularity – the sentence and document – and show that the problem can be reduced to sequential classification with constrained inference. We then discuss the feature space and give an algorithm for learning the parameters based on large-margin structured learning.

Extensions to the model are also examined.

2.1 A Sentence-Document Model

Let $\mathcal{Y}(d)$ be a discrete set of sentiment labels at the document level and $\mathcal{Y}(s)$ be a discrete set of sentiment labels at the sentence level. As input a system is given a document containing sentences $\boldsymbol{s} = s_1, \ldots, s_n$ and must produce sentiment labels for the document, $y^d \in \mathcal{Y}(d)$, and each individual sentence, $\boldsymbol{y}^s = y^s_1, \ldots, y^s_n$, where $y^s_i \in \mathcal{Y}(s)$ $\forall$ $1 \leq i \leq n$. Define $\boldsymbol{y} = (y^d, \boldsymbol{y}^s) = (y^d, y^s_1, \ldots, y^s_n)$ as the joint labeling of the document and sentences. For instance, in Pang and Lee (2004), y^d would be the polarity of the document and y^s_i would indicate whether sentence s_i is subjective or objective. The models presented here are compatible with arbitrary sets of discrete output labels.

Figure 1 presents a model for jointly classifying the sentiment of both the sentences and the document. In this undirected graphical model, the label of each sentence is dependent on the labels of its neighbouring sentences plus the label of the document. The label of the document is dependent on the label of every sentence. Note that the edges between the input (each sentence) and the output labels are not solid, indicating that they are given as input and are not being modeled. The fact that the sentiment of sentences is dependent not only on the local sentiment of other sentences, but also the global document sentiment – and vice versa – allows the model to directly capture the importance of classification decisions across levels in fine-to-coarse sentiment analysis. The local dependencies between sentiment labels on sentences is similar to the work of Pang and Lee (2004) where soft local consistency constraints were created between every sentence in a document and inference was solved using a min-cut algorithm. However, jointly modeling the document label and allowing for non-binary labels complicates min-cut style solutions as inference becomes intractable.

Learning and inference in undirected graphical models is a well studied problem in machine learning and NLP. For example, CRFs define the probability over the labels conditioned on the input using the property that the joint probability distribution over the labels factors over clique potentials in undirected graphical models (Lafferty et al., 2001).

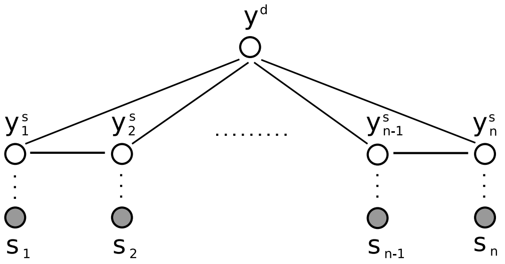

Figure 1: Sentence and document level model.

In this work we will use structured linear classifiers (Collins, 2002). We denote the score of a labeling $\boldsymbol{y}$ for an input $\boldsymbol{s}$ as $\mathit{score}(\boldsymbol{y}, \boldsymbol{s})$ and define this score as the sum of scores over each clique,

$$\begin{aligned} \mathit{score}(\boldsymbol{y}, \boldsymbol{s}) &= \mathit{score}((y^d, \boldsymbol{y}^s), \boldsymbol{s}) \\ &= \mathit{score}((y^d, y^s_1, \ldots, y^s_n), \boldsymbol{s}) \\ &= \sum_{i=2}^{n} \mathit{score}(y^d, y^s_{i-1}, y^s_i, \boldsymbol{s}) \end{aligned}$$

where each clique score is a linear combination of features and their weights,

$$\mathit{score}(y^d, y^s_{i-1}, y^s_i, \boldsymbol{s}) = \mathbf{w} \cdot \mathbf{f}(y^d, y^s_{i-1}, y^s_i, \boldsymbol{s}) \quad (1)$$

and $\mathbf{f}$ is a high dimensional feature representation of the clique and $\mathbf{w}$ a corresponding weight vector. Note that $\boldsymbol{s}$ is included in each score since it is given as input and can always be conditioned on.

In general, inference in undirected graphical models is intractable. However, for the common case of sequences (a.k.a. linear-chain models) the Viterbi algorithm can be used (Rabiner, 1989; Lafferty et al., 2001). Fortunately there is a simple technique that reduces inference in the above model to sequence classification with a constrained version of Viterbi.

2.1.1 Inference as Sequential Labeling

The inference problem is to find the highest scoring labeling $\boldsymbol{y}$ for an input $\boldsymbol{s}$, i.e.,

$$\arg\max_{\boldsymbol{y}} \ \mathit{score}(\boldsymbol{y}, \boldsymbol{s})$$

If the document label y^d is fixed, then inference in the model from Figure 1 reduces to the sequential case. This is because the search space is only over the sentence labels y^s_i, whose graphical structure forms a chain. Thus the problem of finding the

```
Input: $\boldsymbol{s} = s_1, \ldots, s_n$
1.  $\boldsymbol{y}$ = null
2.  for each $y^d \in \mathcal{Y}(d)$
3.      $\boldsymbol{y}^s = \arg\max_{\boldsymbol{y}^s}\ \mathit{score}((y^d, \boldsymbol{y}^s), \boldsymbol{s})$
4.      $\boldsymbol{y}' = (y^d, \boldsymbol{y}^s)$
5.      if $\mathit{score}(\boldsymbol{y}', \boldsymbol{s}) > \mathit{score}(\boldsymbol{y}, \boldsymbol{s})$ or $\boldsymbol{y}$ = null
6.          $\boldsymbol{y} = \boldsymbol{y}'$
7.  return $\boldsymbol{y}$
```

Figure 2: Inference algorithm for model in Figure 1. The argmax in line 3 can be solved using Viterbi's algorithm since y^d is fixed.

highest scoring sentiment labels for all sentences, given a particular document label y^d, can be solved efficiently using Viterbi's algorithm.

The general inference problem can then be solved by iterating over each possible y^d, finding $\boldsymbol{y^s}$ maximizing $\mathit{score}((y^d, \boldsymbol{y^s}), \boldsymbol{s})$ and keeping the single best $\boldsymbol{y} = (y^d, \boldsymbol{y^s})$. This algorithm is outlined in Figure 2 and has a runtime of $O(|\mathcal{Y}(d)||\mathcal{Y}(s)|^2 n)$, due to running Viterbi $|\mathcal{Y}(d)|$ times over a label space of size $|\mathcal{Y}(s)|$. The algorithm can be extended to produce exact k-best lists. This is achieved by using k-best Viterbi techniques to return the k-best global labelings for each document label in line 3. Merging these sets will produce the final k-best list.

It is possible to view the inference algorithm in Figure 2 as a constrained Viterbi search since it is equivalent to flattening the model in Figure 1 to a sequential model with sentence labels from the set $\mathcal{Y}(s) \times \mathcal{Y}(d)$. The resulting Viterbi search would then need to be constrained to ensure consistent solutions, i.e., the label assignments agree on the document label over all sentences. If viewed this way, it is also possible to run a constrained forward-backward algorithm and learn the parameters for CRFs as well.

2.1.2 Feature Space

In this section we define the feature representation for each clique, $\mathbf{f}(y^d, y^s_{i-1}, y^s_i, \boldsymbol{s})$. Assume that each sentence s_i is represented by a set of binary predicates $\mathcal{P}(s_i)$. This set can contain any predicate over the input $\boldsymbol{s}$, but for the present purposes it will include all the unigram, bigram and trigrams in the sentence s_i conjoined with their part-of-speech (obtained from an automatic classifier). Back-offs of each predicate are also included where one or more word is discarded. For instance, if $\mathcal{P}(s_i)$ contains the predicate *a:DT_great:JJ_product:NN*, then it would also have the predicates *a:DT_great:JJ_*:NN*, *a:DT_*:JJ_product:NN*, **:DT_great:JJ_product:NN*, *a:DT_*:JJ_*:NN*, etc. Each predicate, p, is then conjoined with the label information to construct a binary feature. For example, if the sentence label set is $\mathcal{Y}(s) = \{\text{subj}, \text{obj}\}$ and the document set is $\mathcal{Y}(d) = \{\text{pos}, \text{neg}\}$, then the system might contain the following feature,

$$\mathbf{f}_{(j)}(y^d, y^s_{i-1}, y^s_i, \boldsymbol{s}) = \begin{cases} 1 & \text{if } p \in \mathcal{P}(s_i) \\ & \text{and } y^s_{i-1} = \text{obj} \\ & \text{and } y^s_i = \text{subj} \\ & \text{and } y^d = \text{neg} \\ 0 & \text{otherwise} \end{cases}$$

Where $\mathbf{f}_{(j)}$ is the j^{th} dimension of the feature space. For each feature, a set of back-off features are included that only consider the document label y^d, the current sentence label y^s_i, the current sentence and document label y^s_i and y^d, and the current and previous sentence labels y^s_i and y^s_{i-1}. Note that through these back-off features the joint models feature set will subsume the feature set of any individual level model. Only features observed in the training data were considered. Depending on the data set, the dimension of the feature vector $\mathbf{f}$ ranged from 350K to 500K. Though the feature vectors can be sparse, the feature weights will be learned using large-margin techniques that are well known to be robust to large and sparse feature representations.

2.1.3 Training the Model

Let $\mathcal{Y} = \mathcal{Y}(d) \times \mathcal{Y}(s)^n$ be the set of all valid sentence-document labelings for an input $\boldsymbol{s}$. The weights, $\mathbf{w}$, are set using the MIRA learning algorithm, which is an inference based online large-margin learning technique (Crammer and Singer, 2003; McDonald et al., 2005). An advantage of this algorithm is that it relies only on inference to learn the weight vector (see Section 2.1.1). MIRA has been shown to provide state-of-the-art accuracy for many language processing tasks including parsing, chunking and entity extraction (McDonald, 2006).

The basic algorithm is outlined in Figure 3. The algorithm works by considering a single training instance during each iteration. The weight vector $\mathbf{w}$ is updated in line 4 through a quadratic programming problem. This update modifies the weight vector so

```
Training data: $\mathcal{T} = \{(\boldsymbol{y}_t, \boldsymbol{s}_t)\}_{t=1}^{T}$
$\mathbf{w}^{(0)} = 0;\ i = 0$
for $n : 1..N$
    for $t : 1..T$
        $\mathbf{w}^{(i+1)} = \arg\min_{\mathbf{w}*} \left\|\mathbf{w}* - \mathbf{w}^{(i)}\right\|$
        s.t. $\mathit{score}(\boldsymbol{y}_t, \boldsymbol{s}_t) - \mathit{score}(\boldsymbol{y}', \boldsymbol{s}) \geq L(\boldsymbol{y}_t, \boldsymbol{y}')$
            relative to $\mathbf{w}*$
        $\forall \boldsymbol{y}' \in \mathcal{C} \subset \mathcal{Y}$, where $|\mathcal{C}| = k$
        $i = i + 1$
return $\mathbf{w}^{(N \times T)}$
```

Figure 3: MIRA learning algorithm.

that the score of the correct labeling is larger than the score of every labeling in a constraint set $\mathcal{C}$ with a margin proportional to the loss. The constraint set $\mathcal{C}$ can be chosen arbitrarily, but it is usually taken to be the k labelings that have the highest score under the old weight vector $\mathbf{w}^{(i)}$ (McDonald et al., 2005). In this manner, the learning algorithm can update its parameters relative to those labelings closest to the decision boundary. Of all the weight vectors that satisfy these constraints, MIRA chooses the one that is as close as possible to the previous weight vector in order to retain information about previous updates.

The loss function $L(\boldsymbol{y}, \boldsymbol{y}')$ is a positive real valued function and is equal to zero when $\boldsymbol{y} = \boldsymbol{y}'$. This function is task specific and is usually the hamming loss for sequence classification problems (Taskar et al., 2003). Experiments with different loss functions for the joint sentence-document model on a development data set indicated that the hamming loss over sentence labels multiplied by the 0-1 loss over document labels worked best.

An important modification that was made to the learning algorithm deals with how the k constraints are chosen for the optimization. Typically these constraints are the k highest scoring labelings under the current weight vector. However, early experiments showed that the model quickly learned to discard any labeling with an incorrect document label for the instances in the training set. As a result, the constraints were dominated by labelings that only differed over sentence labels. This did not allow the algorithm adequate opportunity to set parameters relative to incorrect document labeling decisions. To combat this, k was divided by the number of document labels, to get a new value k'. For each document label, the k' highest scoring labelings were extracted. Each of these sets were then combined to produce the final constraint set. This allowed constraints to be equally distributed amongst different document labels.

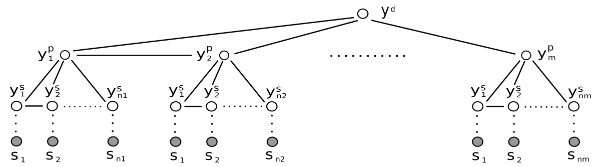

Figure 4: An extension to the model from Figure 1 incorporating paragraph level analysis.

Based on performance on the development data set the number of training iterations was set to $N = 5$ and the number of constraints to $k = 10$. Weight averaging was also employed (Collins, 2002), which helped improve performance.

2.2 Beyond Two-Level Models

To this point, we have focused solely on a model for two-level fine-to-coarse sentiment analysis not only for simplicity, but because the experiments in Section 3 deal exclusively with this scenario. In this section, we briefly discuss possible extensions for more complex situations. For example, longer documents might benefit from an analysis on the paragraph level as well as the sentence and document levels. One possible model for this case is given in Figure 4, which essentially inserts an additional layer between the sentence and document level from the original model. Sentence level analysis is dependent on neighbouring sentences as well as the paragraph level analysis, and the paragraph analysis is dependent on each of the sentences within it, the neighbouring paragraphs, and the document level analysis. This can be extended to an arbitrary level of fine-to-coarse sentiment analysis by simply inserting new layers in this fashion to create more complex hierarchical models.

The advantage of using hierarchical models of this form is that they are nested, which keeps inference tractable. Observe that each pair of adjacent levels in the model is equivalent to the original model from Figure 1. As a result, the scores of the every label at each node in the graph can be calculated with a straight-forward bottom-up dynamic programming algorithm. Details are omitted

	Sentence Stats				Document Stats		
	Pos	Neg	Neu	Tot	Pos	Neg	Tot
Car	472	443	264	1179	98	80	178
Fit	568	635	371	1574	92	97	189
Mp3	485	464	214	1163	98	89	187
Tot	1525	1542	849	3916	288	266	554

Table 1: Data statistics for corpus. Pos = positive polarity, Neg = negative polarity, Neu = no polarity.

for space reasons.

Other models are possible where dependencies occur across non-neighbouring levels, e.g., by inserting edges between the sentence level nodes and the document level node. In the general case, inference is exponential in the size of each clique. Both the models in Figure 1 and Figure 4 have maximum clique sizes of three.

3 Experiments

3.1 Data

To test the model we compiled a corpus of 600 online product reviews from three domains: car seats for children, fitness equipment, and Mp3 players. Of the original 600 reviews that were gathered, we discarded duplicate reviews, reviews with insufficient text, and spam. All reviews were labeled by online customers as having a positive or negative polarity on the document level, i.e., $\mathcal{Y}(d) = \{\text{pos}, \text{neg}\}$. Each review was then split into sentences and every sentence annotated by a single annotator as either being positive, negative or neutral, i.e., $\mathcal{Y}(s) = \{\text{pos}, \text{neg}, \text{neu}\}$. Data statistics for the corpus are given in Table 1.

All sentences were annotated based on their context within the document. Sentences were annotated as neutral if they conveyed no sentiment or had indeterminate sentiment from their context. Many neutral sentences pertain to the circumstances under which the product was purchased. A common class of sentences were those containing product features. These sentences were annotated as having positive or negative polarity if the context supported it. This could include punctuation such as exclamation points, smiley/frowny faces, question marks, etc. The supporting evidence could also come from another sentence, e.g., *"I love it. It has 64Mb of memory and comes with a set of earphones"*.

3.2 Results

Three baseline systems were created,

- **Document-Classifier** is a classifier that learns to predict the document label only.

- **Sentence-Classifier** is a classifier that learns to predict sentence labels in isolation of one another, i.e., without consideration for either the document or neighbouring sentences sentiment.

- **Sentence-Structured** is another sentence classifier, but this classifier uses a sequential chain model to learn and classify sentences. The third baseline is essentially the model from Figure 1 without the top level document node. This baseline will help to gage the empirical gains of the different components of the joint structured model on sentence level classification.

The model described in Section 2 will be called **Joint-Structured**. All models use the same basic predicate space: unigram, bigram, trigram conjoined with part-of-speech, plus back-offs of these (see Section 2.1.2 for more). However, due to the structure of the model and its label space, the feature space of each might be different, e.g., the document classifier will only conjoin predicates with the document label to create the feature set. All models are trained using the MIRA learning algorithm.

Results for each model are given in the first four rows of Table 2. These results were gathered using 10-fold cross validation with one fold for development and the other nine folds for evaluation. This table shows that classifying sentences in isolation from one another is inferior to accounting for a more global context. A significant increase in performance can be obtained when labeling decisions between sentences are modeled (Sentence-Structured). More interestingly, even further gains can be had when document level decisions are modeled (Joint-Structured). In many cases, these improvements are highly statistically significant.

On the document level, performance can also be improved by incorporating sentence level decisions – though these improvements are not consistent. This inconsistency may be a result of the model overfitting on the small set of training data. We suspect this because the document level error rate on the Mp3 training set converges to zero much more rapidly for the Joint-Structured model than the Document-Classifier. This suggests that the Joint-Structured model might be relying too much on the sentence level sentiment features – in order to minimize its error rate – instead of distributing the weights across all features more evenly.

One interesting application of sentence level sentiment analysis is summarizing product reviews on retail websites like Amazon.com or review aggregators like Yelp.com. In this setting the correct polarity of a document is often known, but we wish to label sentiment on the sentence or phrase level to aid in generating a cohesive and informative summary. The joint model can be used to classify sentences in this setting by constraining inference to the known fixed document label for a review. If this is done, then sentiment accuracy on the sentence level increases substantially from 62.6% to 70.3%.

Finally we should note that experiments using CRFs to train the structured models and logistic regression to train the local models yielded similar results to those in Table 2.

3.2.1 Cascaded Models

Another approach to fine-to-coarse sentiment analysis is to use a cascaded system. In such a system, a sentence level classifier might first be run on the data, and then the results input into a document level classifier – or vice-versa.[1] Two cascaded systems were built. The first uses the Sentence-Structured classifier to classify all the sentences from a review, then passes this information to the document classifier as input. In particular, for every predicate in the original document classifier, an additional predicate that specifies the polarity of the sentence in which this predicate occurred was created. The second cascaded system uses the document classifier to determine the global polarity, then passes this information as input into the Sentence-Structured model, constructing predicates in a similar manner.

The results for these two systems can be seen in the last two rows of Table 2. In both cases there is a slight improvement in performance suggesting that an iterative approach might be beneficial. That is, a system could start by classifying documents, use the document information to classify sentences, use the sentence information to classify documents, and repeat until convergence. However, experiments showed that this did not improve accuracy over a single iteration and often hurt performance.

Improvements from the cascaded models are far less consistent than those given from the joint structure model. This is because decisions in the cascaded system are passed to the next layer as the “gold” standard at test time, which results in errors from the first classifier propagating to errors in the second. This could be improved by passing a lattice of possibilities from the first classifier to the second with corresponding confidences. However, solutions such as these are really just approximations of the joint structured model that was presented here.

4 Future Work

One important extension to this work is to augment the models for partially labeled data. It is realistic to imagine a training set where many examples do not have every level of sentiment annotated. For example, there are thousands of online product reviews with labeled document sentiment, but a much smaller amount where sentences are also labeled. Work on learning with hidden variables can be used for both CRFs (Quattoni et al., 2004) and for inference based learning algorithms like those used in this work (Liang et al., 2006).

Another area of future work is to empirically investigate the use of these models on longer documents that require more levels of sentiment analysis than product reviews. In particular, the relative position of a phrase to a contrastive discourse connective or a cue phrase like “in conclusion” or “to summarize” may lead to improved performance since higher level classifications can learn to weigh information passed from these lower level components more heavily.

5 Discussion

In this paper we have investigated the use of a global structured model that learns to predict sentiment on different levels of granularity for a text. We de-

[1] Alternatively, decisions from the sentence classifier can guide which input is seen by the document level classifier (Pang and Lee, 2004).

	Sentence Accuracy				Document Accuracy			
	Car	Fit	Mp3	Total	Car	Fit	Mp3	Total
Document-Classifier	-	-	-	-	72.8	80.1	**87.2**	80.3
Sentence-Classifier	54.8	56.8	49.4	53.1	-	-	-	-
Sentence-Structured	60.5	61.4	55.7	58.8	-	-	-	-
Joint-Structured	**63.5***	**65.2****	**60.1****	**62.6****	**81.5***	**81.9**	85.0	**82.8**
Cascaded Sentence → Document	60.5	61.4	55.7	58.8	75.9	80.7	86.1	81.1
Cascaded Document → Sentence	59.7	61.0	58.3	59.5	72.8	80.1	87.2	80.3

Table 2: Fine-to-coarse sentiment accuracy. Significance calculated using McNemar's test between top two performing systems. *Statistically significant $p < 0.05$. **Statistically significant $p < 0.005$.

scribed a simple model for sentence-document analysis and showed that inference in it is tractable. Experiments show that this model obtains higher accuracy than classifiers trained in isolation as well as cascaded systems that pass information from one level to another at test time. Furthermore, extensions to the sentence-document model were discussed and it was argued that a nested hierarchical structure would be beneficial since it would allow for efficient inference algorithms.

References

Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan. 2005. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proc. HLT/EMNLP*.

Y. Choi, E. Breck, and C. Cardie. 2006. Joint extraction of entities and relations for opinion recognition. In *Proc. EMNLP*.

M. Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proc. EMNLP*.

K. Crammer and Y. Singer. 2003. Ultraconservative online algorithms for multiclass problems. *JMLR*.

Hal Daumé III, John Langford, and Daniel Marcu. 2006. Search-based structured prediction. In Submission.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*.

P. Liang, A. Bouchard-Cote, D. Klein, and B. Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proc. ACL*.

Y. Mao and G. Lebanon. 2006. Isotonic conditional random fields and local sentiment flow. In *Proc. NIPS*.

R. McDonald, K. Crammer, and F. Pereira. 2005. Online large-margin training of dependency parsers. In *Proc. ACL*.

R. McDonald. 2006. *Discriminative Training and Spanning Tree Algorithms for Dependency Parsing*. Ph.D. thesis, University of Pennsylvania.

S. Miller, H. Fox, L.A. Ramshaw, and R.M. Weischedel. 2000. A novel use of statistical parsing to extract information from text. In *Proc NAACL*, pages 226–233.

B. Pang and L. Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proc. ACL*.

B. Pang, L. Lee, and S. Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *EMNLP*.

A. Popescu and O. Etzioni. 2005. Extracting product features and opinions from reviews. In *Proc. HLT/EMNLP*.

A. Quattoni, M. Collins, and T. Darrell. 2004. Conditional random fields for object recognition. In *Proc. NIPS*.

L. R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, February.

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

C. Sutton and A. McCallum. 2006. An introduction to conditional random fields for relational learning. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press.

C. Sutton, K. Rohanimanesh, and A. McCallum. 2004. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In *Proc. ICML*.

B. Taskar, C. Guestrin, and D. Koller. 2003. Max-margin Markov networks. In *Proc. NIPS*.

B. Taskar, D. Klein, M. Collins, D. Koller, and C. Manning. 2004. Max-margin parsing. In *Proc. EMNLP*.

M. Thomas, B. Pang, and L. Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proc. EMNLP*.

I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. 2004. Support vector learning for interdependent and structured output spaces. In *Proc. ICML*.

P. Turney. 2002. Thumbs up or thumbs down? Sentiment orientation applied to unsupervised classification of reviews. In *EMNLP*.