Incorporating Content Structure into Text Analysis Applications

Christina Sauper, Aria Haghighi, Regina Barzilay
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
{csauper, aria42, regina}@csail.mit.edu

Abstract

In this paper, we investigate how modeling content structure can benefit text analysis applications such as extractive summarization and sentiment analysis. This follows the linguistic intuition that rich contextual information should be useful in these tasks. We present a framework which combines a supervised text analysis application with the induction of latent content structure. Both of these elements are learned jointly using the EM algorithm. The induced content structure is learned from a large unannotated corpus and biased by the underlying text analysis task. We demonstrate that exploiting content structure yields significant improvements over approaches that rely only on local context.[1]

1 Introduction

In this paper, we demonstrate that leveraging document structure significantly benefits text analysis applications. As a motivating example, consider the excerpt from a DVD review shown in Table 1. This review discusses multiple aspects of a product, such as audio and video properties. While the word "pleased" is a strong indicator of positive sentiment, the sentence in which it appears does not specify the aspect to which it relates. Resolving this ambiguity requires information about global document structure.

A central challenge in utilizing such information lies in finding a relevant representation of content structure for a specific text analysis task. For instance, when performing single-aspect sentiment analysis, the most relevant aspect of content structure is whether a given sentence is objective or subjective (Pang and Lee, 2004). In a multi-aspect setting, however, information about the sentence topic is required to determine the aspect to which a sentiment-bearing word relates (Snyder and Barzilay, 2007). As we can see from even these closely related applications, the content structure representation should be intimately tied to a specific text analysis task.

Audio Audio choices are English, Spanish and French Dolby Digital 5.1 ... Bass is still robust and powerful, giving weight to just about any scene – most notably the film's exciting final fight. Fans should be **pleased** with the presentation.
Extras This single-disc DVD comes packed in a black amaray case with a glossy slipcover. Cover art has clearly been designed to appeal the Twilight crowd ... Finally, we've got a deleted scenes reel. Most of the excised scenes are actually pretty interesting.

Table 1: An excerpt from a DVD review.

In this work, we present an approach in which a content model is learned jointly with a text analysis task. We assume complete annotations for the task itself, but we learn the content model from raw, unannotated text. Our approach is implemented in a discriminative framework using latent variables to represent facets of content structure. In this framework, the original task features (e.g., lexical ones) are conjoined with latent variables to enrich the features with global contextual information. For example, in Table 1, the feature associated with the

[1] Code and processed data presented here are available at http://groups.csail.mit.edu/rbg/code/content_structure.html

Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 377–387,
MIT, Massachusetts, USA, 9-11 October 2010.

word "pleased" should contribute most strongly to the sentiment of the *audio* aspect when it is augmented with a relevant topic indicator.

The coupling of the content model and the task-specific model allows the two components to mutually influence each other during learning. The content model leverages unannotated data to improve the performance of the task-specific model, while the task-specific model provides feedback to improve the relevance of the content model. The combined model can be learned effectively using a novel EM-based method for joint training.

We evaluate our approach on two complementary text analysis tasks. Our first task is a multi-aspect sentiment analysis task, where a system predicts the aspect-specific sentiment ratings (Snyder and Barzilay, 2007). Second, we consider a multi-aspect extractive summarization task in which a system extracts key properties for a pre-specified set of aspects. On both tasks, our method for incorporating content structure consistently outperforms structure-agnostic counterparts. Moreover, jointly learning content and task parameters yields additional gains over independently learned models.

2 Related Work

Prior research has demonstrated the usefulness of content models for discourse-level tasks. Examples of such tasks include sentence ordering (Barzilay and Lee, 2004; Elsner et al., 2007), extraction-based summarization (Haghighi and Vanderwende, 2009) and text segmentation (Chen et al., 2009). Since these tasks are inherently tied to document structure, a content model is essential to performing them successfully. In contrast, the applications considered in this paper are typically developed without any discourse information, focusing on capturing sentence-level relations. Our goal is to augment these models with document-level content information.

Several applications in information extraction and sentiment analysis are close in spirit to our work (Pang and Lee, 2004; Patwardhan and Riloff, 2007; McDonald et al., 2007). These approaches consider global contextual information when determining whether a given sentence is relevant to the underlying analysis task. All assume that relevant sentences have been annotated. For instance, Pang and Lee (2004) refine the accuracy of sentiment analysis by considering only the subjective sentences of a review as determined by an independent classifier. Patwardhan and Riloff (2007) take a similar approach in the context of information extraction. Rather than applying their extractor to all the sentences in a document, they limit it to event-relevant sentences. Since these sentences are more likely to contain information of interest, the extraction performance increases.

Another approach, taken by Choi and Cardie (2008) and Somasundaran et al. (2009) uses linguistic resources to create a latent model in a task-specific fashion to improve performance, rather than assuming sentence-level task relevancy. Choi and Cardie (2008) address a sentiment analysis task by using a heuristic decision process based on word-level intermediate variables to represent polarity. Somasundaran et al. (2009) similarly uses a bootstrapped local polarity classifier to identify sentence polarity.

McDonald et al. (2007) propose a model which jointly identifies global polarity as well as paragraph- and sentence-level polarity, all of which are observed in training data. While our approach uses a similar hierarchy, McDonald et al. (2007) is concerned with recovering the labels at all levels, whereas in this work we are interested in using latent document content structure as a means to benefit task predictions.

While our method also incorporates contextual information into existing text analysis applications, our approach is markedly different from the above approaches. First, our representation of context encodes more than the relevance-based binary distinction considered in the past work. Our algorithm adjusts the content model dynamically for a given task rather than pre-specifying it. Second, while previous work is fully supervised, in our case relevance annotations are readily available for only a few applications and are prohibitively expensive to obtain for many others. To overcome this drawback, our method induces a content model in an unsupervised fashion and connects it via latent variables to the target model. This design not only eliminates the need for additional annotations, but also allows the algorithm to leverage large quantities of raw data for training the content model. The tight coupling of rel-

evance learning with the target analysis task leads to further performance gains.

Finally, our work relates to supervised topic models in Blei and McAullife (2007). In this work, latent topic variables are used to generate text as well as a supervised sentiment rating for the document. However, this architecture does not permit the usage of standard discriminative models which condition freely on textual features.

3 Model

3.1 Problem Formulation

In this section, we describe a model which incorporates content information into a multi-aspect summarization task.[2] Our approach assumes that at training time we have a collection of labeled documents $\mathcal{D}_L$, each consisting of the document text $\boldsymbol{s}$ and true task-specific labeling $\boldsymbol{y}^*$. For the multi-aspect summarization task, $\boldsymbol{y}^*$ consists of sequence labels (e.g., *value* or *service*) for the tokens of a document. Specifically, the document text $\boldsymbol{s}$ is composed of sentences $s_1, \ldots, s_n$ and the labelings $\boldsymbol{y}^*$ consists of corresponding label sequences $y_1, \ldots, y_n$.[3]

As is common in related work, we model each y_i using a CRF which conditions on the observed document text. In this work, we also assume a content model, which we fix to be the document-level HMM as used in Barzilay and Lee (2004). In this content model, each sentence s_i is associated with a hidden topic variable T_i which generates the words of the sentence. We will use $\boldsymbol{T} = (T_1, \ldots, T_n)$ to refer to the hidden topic sequence for a document. We fix the number of topics to a pre-specified constant K.

3.2 Model Overview

Our model, depicted in Figure 1, proceeds as follows: First the document-level HMM generates a hidden content topic sequence $\boldsymbol{T}$ for the sentences of a document. This content component is parametrized by θ and decomposes in the standard HMM fashion:[4]

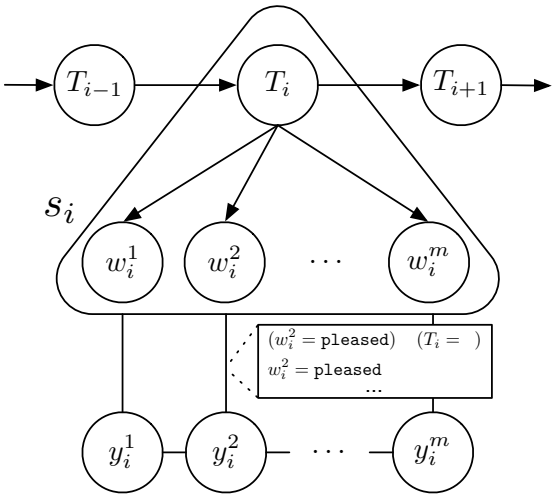


Figure 1: A graphical depiction of our model for sequence labeling tasks. The T_i variable represents the content model topic for the ith sentence s_i. The words of s_i, $(w_i^1, \ldots, w_i^m)$, each have a task label $(y_i^1, \ldots, y_i^m)$. Note that each token label has an undirected edge to a factor containing the words of the current sentence, s_i as well as the topic of the current sentence T_i.

$$P_\theta(\boldsymbol{s}, \boldsymbol{T}) = \prod_{i=1}^{n} P_\theta(T_i|T_{i-1}) \prod_{w \in s_i} P_\theta(w|T_i) \quad (1)$$

Then the label sequences for each sentence in the document are independently modeled as CRFs which condition on both the sentence features and the sentence topic:

$$P_\phi(\boldsymbol{y}|\boldsymbol{s}, \boldsymbol{T}) = \prod_{i=1}^{n} P_\phi(y_i|s_i, T_i) \quad (2)$$

Each sentence CRF is parametrized by ϕ and takes the standard form:

$$P_\phi(y|s,T) \propto \exp\left\{ \sum_j \phi^T \left[f_N(y^j, s, T) + f_E(y^j, y^{j+1}) \right] \right\}$$

[2] In Section 3.6, we discuss how this framework can be used for other text analysis applications.

[3] Note that each y_i is a label sequence across the words in s_i, rather than an individual label.

[4] We also utilize a hierarchical emission model so that each topic distribution interpolates between a topic-specific distribution as well as a shared background model; this is intended to capture domain-specific stop words.

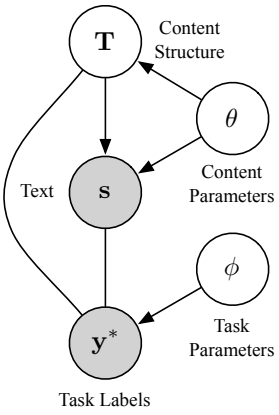


Figure 2: A graphical depiction of the generative process for a labeled document at training time (See Section 3); shaded nodes indicate variables which are observed at training time. First the latent underlying content structure $\boldsymbol{T}$ is drawn. Then, the document text $\boldsymbol{s}$ is drawn conditioned on the content structure utilizing content parameters θ. Finally, the observed task labels for the document are modeled given $\boldsymbol{s}$ and $\boldsymbol{T}$ using the task parameters ϕ. Note that the arrows for the task labels are undirected since they are modeled discriminatively.

where $f_N(\cdot)$ and $f_E(\cdot)$ are feature functions associated with CRF nodes and transitions respectively.

Allowing the CRF to condition on the sentence topic T_i permits predictions to be more sensitive to content. For instance, using the example from Table 1, we could have a feature that indicates the word "pleased" conjoined with the segment topic (see Figure 1). These topic-specific features serve to disambiguate word usage.

This joint process, depicted graphically in Figure 2, is summarized as:

$$P(\boldsymbol{T}, \boldsymbol{s}, \boldsymbol{y}^*) = P_\theta(\boldsymbol{T}, \boldsymbol{s}) P_\phi(\boldsymbol{y}^* | \boldsymbol{s}, \boldsymbol{T}) \qquad (3)$$

Note that this probability decomposes into a document-level HMM term (the content component) as well as a product of CRF terms (the task component).

3.3 Learning

During learning, we would like to find the document-level HMM parameters θ and the summarization task CRF parameters ϕ which maximize the likelihood of the labeled documents. The only observed elements of a labeled document are the document text $\boldsymbol{s}$ and the aspect labels $\boldsymbol{y}^*$. This objective is given by:

$$\begin{aligned}\mathcal{L}_L(\phi, \theta) &= \sum_{(\boldsymbol{s}, \boldsymbol{y}^*) \in \mathcal{D}_L} \log P(\boldsymbol{s}, \boldsymbol{y}^*) \\ &= \sum_{(\boldsymbol{s}, \boldsymbol{y}^*) \in \mathcal{D}_L} \log \sum_{\boldsymbol{T}} P(\boldsymbol{T}, \boldsymbol{s}, \boldsymbol{y}^*)\end{aligned}$$

We use the EM algorithm to optimize this objective.

E-Step The E-Step in EM requires computing the posterior distribution over latent variables. In this model, the only latent variables are the sentence topics $\boldsymbol{T}$. To compute this term, we utilize the decomposition in Equation (3) and rearrange HMM and CRF terms to obtain:

$$\begin{aligned}P(\boldsymbol{T}, \boldsymbol{s}, \boldsymbol{y}^*) &= P_\theta(\boldsymbol{T}, \boldsymbol{s}) P_\phi(\boldsymbol{y}^* | \boldsymbol{T}, \boldsymbol{s}) \\ &= \left(\prod_{i=1}^{n} P_\theta(T_i | T_{i-1}) \prod_{w \in s_i} P_\theta(w | T_i) \right) \cdot \\ &\quad \left(\prod_{i=1}^{n} P_\phi(y_i^* | s_i, T_i) \right) \\ &= \prod_{i=1}^{n} P_\theta(T_i | T_{i-1}) \cdot \\ &\quad \left(\prod_{w \in s_i} P_\theta(w | T_i) P_\phi(y_i^* | s_i, T_i) \right)\end{aligned}$$

We note that this expression takes the same form as the document-level HMM, except that in addition to emitting the words of a sentence, we also have an observation associated with the sentence sequence labeling. We treat each $P_\phi(y_i^* | s_i, T_i)$ as part of the node potential associated with the document-level HMM. We utilize the Forward-Backward algorithm as one would with the document-level HMM in isolation, except that each node potential incorporates this CRF term.

M-Step We perform separate M-Steps for content and task parameters. The M-Step for the content parameters is identical to the document-level HMM

content model: topic emission and transition distributions are updated with expected counts derived from E-Step topic posteriors.

The M-Step for the task parameters does not have a closed-form solution. Recall that in the M-Step, we maximize the log probability of all random variables given expectations of latent variables. Using the decomposition in Equation (3), it is clear that the only component of the joint labeled document probability which relies upon the task parameters is $\log P_\phi(\boldsymbol{y}^*|\boldsymbol{s}, \boldsymbol{T})$. Thus for the M-Step, it is sufficient to optimize the following with respect to ϕ:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{T}|\boldsymbol{s},\boldsymbol{y}^*} \log P_\phi(\boldsymbol{y}^*|\boldsymbol{s}, \boldsymbol{T}) \\
&= \sum_{i=1}^{n} \mathbb{E}_{T_i|s_i,y_i^*} \log P_\phi(y_i^*|s_i, T_i) \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K} P(T_i = k|s_i, y_i^*) \log P_\phi(y_i^*|s_i, T_i)
\end{aligned}$$

The first equality follows from the decomposition of the task component into independent CRFs (see Equation (2)). Optimizing this objective is equivalent to a weighted version of the conditional likelihood objective used to train the CRF in isolation. An intuitive explanation of this process is that there are multiple CRF instances, one for each possible hidden topic T. Each utilizes different content features to explain the sentence sequence labeling. These instances are weighted according to the posterior over T obtained during the E-Step. While this objective is non-convex due to the summation over T, we can still optimize it using any gradient-based optimization solver; in our experiments, we used the LBFGS algorithm (Liu et al., 1989).

3.4 Inference

We must predict a label sequence y for each sentence s of the document. We assume a loss function over a sequence labeling y and a proposed labeling $\hat{y}$, which decomposes as:

$$L(y, \hat{y}) = \sum_j L(y^j, \hat{y}^j)$$

where each position loss is sensitive to the kind of error which is made. Failing to extract a token is penalized to a greater extent than extracting it with an incorrect label:

$$L(y^j, \hat{y}^j) = \begin{cases} 0 & \text{if } \hat{y}^j = y^j \\ c & \text{if } y^j \neq \text{NONE and } \hat{y}^j = \text{NONE} \\ 1 & \text{otherwise} \end{cases}$$

In this definition, NONE represents the background label which is reserved for tokens which do not correspond to labels of interest. The constant c represents a user-defined trade-off between precision and recall errors. For our multi-aspect summarization task, we select $c = 4$ for Yelp and $c = 5$ for Amazon to combat the high-precision bias typical of conditional likelihood models.

At inference time, we select the single labeling which minimizes the expected loss with respect to model posterior over label sequences:

$$\begin{aligned}
\hat{y} &= \min_{\hat{y}} \mathbb{E}_{y|\boldsymbol{s}} L(y, \hat{y}) \\
&= \min_{\hat{y}} \sum_{j=1} \mathbb{E}_{y^j|\boldsymbol{s}} L(y^j, \hat{y}^j)
\end{aligned}$$

In our case, we must marginalize out the sentence topic T:

$$\begin{aligned}
P(y^j|s) &= \sum_T P(y^j, T|s) \\
&= \sum_T P_\theta(T|s) P_\phi(y^j|s, T)
\end{aligned}$$

This minimum risk criterion has been widely used in NLP applications such as parsing (Goodman, 1999) and machine translation (DeNero et al., 2009). Note that the above formulation differs from the standard CRF due to the latent topic variables. Otherwise the inference task could be accomplished by directly obtaining posteriors over each y^j state using the Forward-Backwards algorithm on the sentence CRF.

Finding $\hat{y}$ can be done efficiently. First, we obtain marginal token posteriors as above. Then, the expected loss of a token prediction is computed as follows:

$$\sum_{\hat{y}^j} P(y^j|s) L(y^j, \hat{y}^j)$$

Once we obtain expected losses of each token prediction, we compute the minimum risk sequence labeling by running the Viterbi algorithm. The potential for each position and prediction is given by

the negative expected loss. The maximal scoring sequence according to these potentials minimizes the expected risk.

3.5 Leveraging unannotated data

Our model allows us to incorporate unlabeled documents, denoted $\mathcal{D}_U$, to improve the learning of the content model. For an unlabeled document we only observe the document text $\boldsymbol{s}$ and assume it is drawn from the same content model as our labeled documents. The objective presented in Section 3.3 assumed that all documents were labeled; here we supplement this objective by capturing the likelihood of unlabeled documents according to the content model:

$$\begin{aligned}\mathcal{L}_U(\theta) &= \sum_{\boldsymbol{s} \in \mathcal{D}_U} \log P_\theta(\boldsymbol{s}) \\ &= \sum_{\boldsymbol{s} \in \mathcal{D}_U} \log \sum_{\boldsymbol{T}} P_\theta(\boldsymbol{s}, \boldsymbol{T})\end{aligned}$$

Our overall objective function is to maximize the likelihood of both our labeled and unlabeled data. This objective corresponds to:

$$\mathcal{L}(\phi, \theta) = \mathcal{L}_U(\theta) + \mathcal{L}_L(\phi, \theta)$$

This objective can also be optimized using the EM algorithm, where the E-Step for labeled and unlabeled documents is outlined above.

3.6 Generalization

The approach outlined can be applied to a wider range of task components. For instance, in Section 4.1 we apply this approach to multi-aspect sentiment analysis. In this task, the target y consists of numeric sentiment ratings $(y_1, \ldots, y_K)$ for each of K aspects. The task component consists of independent linear regression models for each aspect sentiment rating. For the content model, we associate a topic with each paragraph; $\boldsymbol{T}$ consists of assignments of topics to each document paragraph.

The model structure still decomposes as in Figure 2, but the details of learning are slightly different. For instance, because the task label (aspect sentiment ratings) is not localized to any region of the document, all content model variables influence the target response. Conditioned on the target label, all topic variables become correlated. Thus when learning, the E-Step requires computing a posterior over paragraph topic tuples $\boldsymbol{T}$:

$$P(\boldsymbol{T}|\boldsymbol{y}, \boldsymbol{s}) \propto P(\boldsymbol{s}, \boldsymbol{T}) P(\boldsymbol{y}|\boldsymbol{T}, \boldsymbol{s})$$

For the case of our multi-aspect sentiment task, this computation can be done exactly by enumerating $\boldsymbol{T}$ tuples, since the number of sentences and possible topics is relatively small. If summation is intractable, the posterior may be approximated using variational techniques (Bishop, 2006), which is applicable to a broad range of potential applications.

4 Experimental Set-Up

We apply our approach to two text analysis tasks that stand to benefit from modeling content structure: multi-aspect sentiment analysis and multi-aspect review summarization.

4.1 Tasks

In the following section, we define each task in detail, explain the task-specific adaptation of the model and describe the data sets used in the experiments. Table 2 summarizes statistics for all the data sets.

For all tasks, when using a content model with a task model, we utilize a new set of features which include all the original features as well as a copy of each feature conjoined with the content topic assignment (see Figure 1). We also include a feature which indicates whether a given word was most likely emitted from the underlying topic or from a background distribution.

Multi-Aspect Sentiment Ranking The goal of multi-aspect sentiment classification is to predict a set of numeric ranks that reflects the user satisfaction for each aspect (Snyder and Barzilay, 2007). One of the challenges in this task is to attribute sentiment-bearing words to the aspects they describe. Information about document structure has the potential to greatly reduce this ambiguity.

Following standard sentiment ranking approaches (Wilson et al., 2004; Pang and Lee, 2005; Goldberg and Zhu, 2006; Snyder and Barzilay, 2007), we employ ordinary linear regression to independently map bag-of-words representations into predicted aspect ranks. In addition to commonly used lexical features, this set is augmented

Task	Labeled		Unlabeled	Avg. Size	
	Train	Test		Words	Sents
Multi-aspect sentiment	600	65	—	1,027	20.5
Multi-aspect summarization					
Amazon	35	24	12,684	214	11.7
Yelp	48	48	33,015	178	11.2

Table 2: This table summarizes the size of each corpus. In each case, the unlabeled texts of both labeled and unlabeled documents are used for training the content model, while only the labeled training corpus is used to train the task model. Note that the entire data set for the multi-aspect sentiment analysis task is labeled.

with content features as described above. For this application, we fix the number of HMM states to be equal to the predefined number of aspects.

We test our sentiment ranker on a set of DVD reviews from the website IGN.com.[5] Each review is accompanied by 1-10 scale ratings in four categories that assess the quality of a movie's content, video, audio, and DVD extras. In this data set, segments corresponding to each of the aspects are clearly delineated in each document. Therefore, we can compare the performance of the algorithm using automatically induced content models against the gold standard structural information.

Multi-Aspect Review Summarization The goal of this task is to extract informative phrases that identify information relevant to several predefined aspects of interest. In other words, we would like our system to both extract important phrases (e.g., *cheap food*) and label it with one of the given aspects (e.g., *value*). For concrete examples and lists of aspects for each data set, see Figures 3b and 3c. Variants of this task have been considered in review summarization in previous work (Kim and Hovy, 2006; Branavan et al., 2009).

This task has elements of both information extraction and phrase-based summarization — the phrases we wish to extract are broader in scope than in standard template-driven IE, but at the same time, the type of selected information is restricted to the defined aspects, similar to query-based summarization. The difficulty here is that phrase selection is highly context-dependent. For instance, in TV reviews such as in Figure 3b, the highlighted phrase "easy to read" might refer to either the menu or the remote; broader context is required for correct labeling.

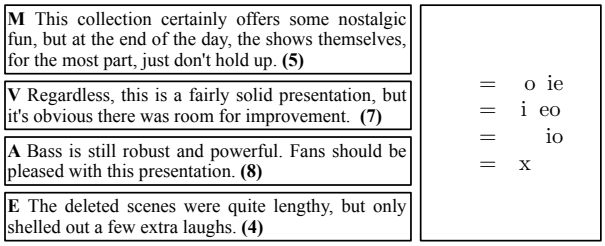
We evaluated our approach for this task on two data sets: Amazon TV reviews (Figure 3b) and Yelp restaurant reviews (Figure 3c). To eliminate noisy reviews, we only retain documents that have been rated "helpful" by the users of the site; we also remove reviews which are abnormally short or long.

Each data set was manually annotated with aspect labels using Mechanical Turk, which has been used in previous work to annotate NLP data (Snow et al., 2008). Since we cannot select high-quality annotators directly, we included a control document which had been previously annotated by a native speaker among the documents assigned to each annotator. The work of any annotator who exhibited low agreement on the control document annotation was excluded from the corpus. To test task annotation agreement, we use Cohen's Kappa (Cohen, 1960). On the Amazon data set, two native speakers annotated a set of four documents. The agreement between the judges was 0.54. On the Yelp data set, we simply computed the agreement between all pairs of reviewers who received the same control documents; the agreement was 0.49.

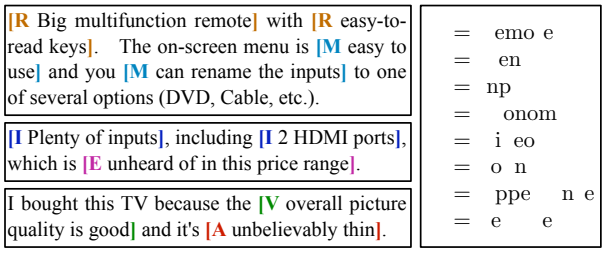
4.2 Baseline Comparison and Evaluation

Baselines For all the models, we obtain a baseline system by eliminating content features and only using a task model with the set of features described above. We also compare against a simplified variant of our method wherein a content model is induced in isolation rather than learned jointly in the context of the underlying task. In our experiments, we refer to the two methods as the No Content Model (NoCM) and Independent Content Model (IndepCM) settings, respectively. The Joint Content

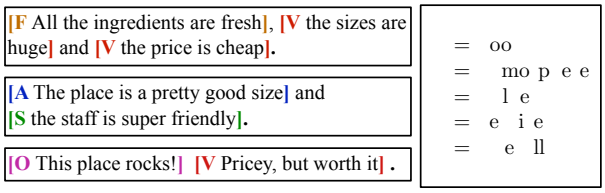
[5] http://dvd.ign.com/index/reviews.html



(a) Sample labeled text from the multi-aspect sentiment corpus



(b) Sample labeled text from the Amazon multi-aspect summarization corpus



(c) Sample labeled text from the Yelp multi-aspect summarization corpus

Figure 3: Excerpts from the three corpora with the corresponding labels. Note that sentences from the multi-aspect summarization corpora generally focus on only one or two aspects. The multi-aspect sentiment corpus has labels per paragraph rather than per sentence.

Model (JointCM) setting refers to our full model described in Section 3, where content and task components are learned jointly.

Evaluation Metrics For multi-aspect sentiment ranking, we report the average L_2 (squared difference) and L_1 (absolute difference) between system prediction and true 1-10 sentiment rating across test documents and aspects.

For the multi-aspect summarization task, we measure average token precision and recall of the label assignments (Multi-label). For the Amazon corpus, we also report a coarser metric which measures extraction precision and recall while ignoring labels (Binary labels) as well as ROUGE (Lin, 2004). To compute ROUGE, we control for length by limiting each system to predict the same number of tokens as the original labeled document.

Our metrics of statistical significance vary by task. For the sentiment task, we use Student's t-test. For the multi-aspect summarization task, we perform chi-square analysis on the ROUGE scores as well as on precision and recall separately, as is commonly done in information extraction (Freitag, 2004; Weeds et al., 2004; Finkel and Manning, 2009).

	L_1	L_2
NoCM	1.37	3.15
IndepCM	1.28†*	2.80†*
JointCM	**1.25**†	**2.65**†*
Gold	1.18†*	2.48†*

Table 3: The error rate on the multi-aspect sentiment ranking. We report mean L_1 and L_2 between system prediction and true values over all aspects. Marked results are statistically significant with $p < 0.05$: * over the previous model and † over NoCM.

	F_1	F_2	Prec.	Recall
NoCM	28.8%	34.8%	22.4%	40.3%
IndepCM	37.9%	43.7%	31.1%†*	**48.6%**†*
JointCM	**39.2%**	**44.4%**	**32.9%**†*	**48.6%**†

Table 4: Results for multi-aspect summarization on the Yelp corpus. Marked precision and recall are statistically significant with $p < 0.05$: * over the previous model and † over NoCM.

5 Results

In this section, we present the results of the methods on the tasks described above (see Tables 3, 4, and 5).

Baseline Comparisons Adding a content model significantly outperforms the NoCM baseline on both tasks. The highest F_1 error reduction – 14.7% – is achieved on multi-aspect summarization on the Yelp corpus, followed by the reduction of 11.5% and 8.75%, on multi-aspect summarization on the Amazon corpus and multi-aspect sentiment ranking, respectively.

We also observe a consistent performance boost when comparing against the IndepCM baseline. This result confirms our hypothesis about the ad-

	Multi-label				Binary labels				
	F_1	F_2	Prec.	Recall	F_1	F_2	Prec.	Recall	ROUGE
NoCM	18.9%	18.0%	20.4%	17.5%	35.1%	33.6%	38.1%	32.6%	43.8%
IndepCM	24.5%	23.8%	**25.8%**†*	23.3%†*	43.0%	41.8%	**45.3%**†*	40.9%†*	47.4%†*
JointCM	**28.2%**	**31.3%**	24.3%†	**33.7%**†*	**47.8%**	**53.0%**	41.2%†	**57.1%**†*	**47.6%**†*

Table 5: Results for multi-aspect summarization on the Amazon corpus. Marked ROUGE, precision, and recall are statistically significant with $p < 0.05$: * over the previous model and † over NoCM.

vantages of jointly learning the content model in the context of the underlying task.

Comparison with additional context features One alternative to an explicit content model is to simply incorporate additional features into NoCM as a proxy for contextual information. In the multi-aspect summarization case, this can be accomplished by adding unigram features from the sentences before and after the current one.[6]

When testing this approach, however, the performance of NoCM actually decreases on both Amazon (to 15.0% F_1) and Yelp (to 24.5% F_1) corpora. This result is not surprising for this particular task – by adding these features, we substantially increase the feature space without increasing the amount of training data. An advantage of our approach is that our learned representation of context is coarse, and we can leverage large quantities of unannotated training data.

Impact of content model quality on task performance In the multi-aspect sentiment ranking task, we have access to gold standard document-level content structure annotation. This affords us the ability to compare the ideal content structure, provided by the document authors, with one that is learned automatically. As Table 3 shows, the manually created document structure segmentation yields the best results. However, the performance of our JointCM model is not far behind the gold standard content structure.

The quality of the induced content model is determined by the amount of training data. As Figure 4 shows, the multi-aspect summarizer improves with the increase in the size of raw data available for learning content model.

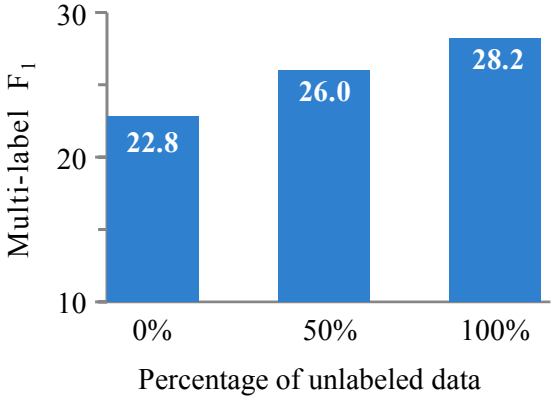


Figure 4: Results on the Amazon corpus using the complete annotated set with varying amounts of additional unlabeled data.[7]

Compensating for annotation sparsity We hypothesize that by incorporating rich contextual information, we can reduce the need for manual task annotation. We test this by reducing the amount of annotated data available to the model and measuring performance at several quantities of unannotated data. As Figure 5 shows, the performance increase achieved by doubling the amount of annotated data can also be achieved by adding only 12.5% of the unlabeled data.

6 Conclusion

In this paper, we demonstrate the benefits of incorporating content models in text analysis tasks. We also introduce a framework to allow the joint learning of an unsupervised latent content model with a supervised task-specific model. On multiple tasks and datasets, our results empirically connect model quality and task performance, suggesting that fur-

[6]This type of feature is not applicable to our multi-aspect sentiment ranking task, as we already use unigram features from the entire document.

[7]Because we append the unlabeled versions of the labeled data to the unlabeled set, even with 0% additional unlabeled data, there is a small data set to train the content model.

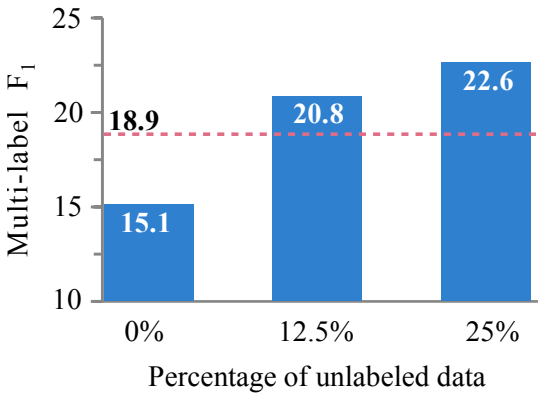


Figure 5: Results on the Amazon corpus using half of the annotated training documents. The content model is trained with 0%, 12.5%, and 25% of additional unlabeled data.[7] The dashed horizontal line represents NoCM with the complete annotated set.

ther improvements in content modeling may yield even further gains.

Acknowledgments

The authors acknowledge the support of the NSF (CAREER grant IIS-0448168) and NIH (grant 5-R01-LM009723-02). Thanks to Peter Szolovits and the MIT NLP group for their helpful comments. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors, and do not necessarily reflect the views of the funding organizations.

References

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of the NAACL/HLT*, pages 113–120.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc.

David M. Blei and Jon D. McAullife. 2007. Supervised Topic Models. In *NIPS*.

S. R. K. Branavan, Harr Chen, Jacob Eisenstein, and Regina Barzilay. 2009. Learning document-level semantic properties from free-text annotations. *JAIR*, 34:569–603.

Harr Chen, S. R. K. Branavan, Regina Barzilay, and David R. Karger. 2009. Content modeling using latent permutations. *JAIR*, 36:129–163.

Yejin Choi and Claire Cardie. 2008. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the EMNLP*, pages 793–801.

J. Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37.

John DeNero, David Chiang, and Kevin Knight. 2009. Fast consensus decoding over translation forests. In *Proceedings of the ACL/IJCNLP*, pages 567–575.

Micha Elsner, Joseph Austerweil, and Eugene Charniak. 2007. A unified local and global model for discourse coherence. In *Proceedings of the NAACL/HLT*, pages 436–443.

Jenny Rose Finkel and Christopher D. Manning. 2009. Joint parsing and named entity recognition. In *Proceedings of the NAACL*.

Dayne Freitag. 2004. Trained named entity recognition using distributional clusters. In *Proceedings of the EMNLP*, pages 262–269.

Andrew B. Goldberg and Xiaojin Zhu. 2006. Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In *Proceedings of the NAACL/HLT Workshop on TextGraphs*, pages 45–52.

Joshua Goodman. 1999. Semiring parsing. *Computational Linguistics*, 25(4):573–605.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of the NAACL/HLT*, pages 362–370.

Soo-Min Kim and Eduard Hovy. 2006. Automatic identification of pro and con reasons in online reviews. In *Proceedings of the COLING/ACL*, pages 483–490.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the ACL*, pages 74–81.

Dong C. Liu, Jorge Nocedal, Dong C. Liu, and Jorge Nocedal. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528.

Ryan McDonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. 2007. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the ACL*, pages 432–439.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, pages 271–278.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*, pages 115–124.

Siddharth Patwardhan and Ellen Riloff. 2007. Effective information extraction with semantic affinity patterns and relevant regions. In *Proceedings of the EMNLP/CoNLL*, pages 717–727.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the EMNLP*.

Benjamin Snyder and Regina Barzilay. 2007. Multiple aspect ranking using the good grief algorithm. In *Proceedings of the NAACL/HLT*, pages 300–307.

Swapna Somasundaran, Galileo Namata, Janyce Wiebe, and Lise Getoor. 2009. Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In *Proceedings of the EMNLP*, pages 170–179.

Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of the COLING*, page 1015.

Theresa Wilson, Janyce Wiebe, and Rebecca Hwa. 2004. Just how mad are you? finding strong and weak opinion clauses. In *Proceedings of the AAAI*, pages 761–769.