Catching the Drift: Probabilistic Content Models, with Applications to Generation and Summarization

Regina Barzilay
Computer Science and AI Lab
MIT
regina@csail.mit.edu

Lillian Lee
Department of Computer Science
Cornell University
llee@cs.cornell.edu

Abstract

We consider the problem of modeling the *content structure* of texts within a specific domain, in terms of the topics the texts address and the order in which these topics appear. We first present an effective knowledge-lean method for learning content models from un-annotated documents, utilizing a novel adaptation of algorithms for Hidden Markov Models. We then apply our method to two complementary tasks: information ordering and extractive summarization. Our experiments show that incorporating content models in these applications yields substantial improvement over previously-proposed methods.

1 Introduction

The development and application of computational models of text structure is a central concern in natural language processing. Document-level analysis of text structure is an important instance of such work. Previous research has sought to characterize texts in terms of domain-independent rhetorical elements, such as schema items (McKeown, 1985) or rhetorical relations (Mann and Thompson, 1988; Marcu, 1997). The focus of our work, however, is on an equally fundamental but domain-dependent dimension of the structure of text: *content*.

Our use of the term "content" corresponds roughly to the notions of topic and topic change. We desire models that can specify, for example, that articles about earthquakes typically contain information about quake strength, location, and casualties, and that descriptions of casualties usually precede those of rescue efforts. But rather than manually determine the topics for a given domain, we take a *distributional* view, learning them directly from un-annotated texts via analysis of word distribution patterns. This idea dates back at least to Harris (1982), who claimed that "various types of [word] recurrence patterns seem to characterize various types of discourse". Advantages of a distributional perspective include both drastic reduction in human effort and recognition of "topics" that might not occur to a human expert and yet, when explicitly modeled, aid in applications.

Of course, the success of the distributional approach depends on the existence of recurrent patterns. In arbitrary document collections, such patterns might be too variable to be easily detected by statistical means. However, research has shown that texts *from the same domain* tend to exhibit high similarity (Wray, 2002). Cognitive psychologists have long posited that this similarity is not accidental, arguing that formulaic text structure facilitates readers' comprehension and recall (Bartlett, 1932).[1]

In this paper, we investigate the utility of domain-specific *content models* for representing topics and topic shifts. Content models are Hidden Markov Models (HMMs) wherein states correspond to types of information characteristic to the domain of interest (e.g., earthquake magnitude or previous earthquake occurrences), and state transitions capture possible information-presentation orderings within that domain.

We first describe an efficient, knowledge-lean method for learning both a set of topics and the relations between topics directly from un-annotated documents. Our technique incorporates a novel adaptation of the standard HMM induction algorithm that is tailored to the task of modeling content.

Then, we apply techniques based on content models to two complex text-processing tasks. First, we consider *information ordering*, that is, choosing a sequence in which to present a pre-selected set of items; this is an essential step in concept-to-text generation, multi-document summarization, and other text-synthesis problems. In our experiments, content models outperform Lapata's (2003) state-of-the-art ordering method by a wide margin — for one domain and performance metric, the gap was 78 percentage points. Second, we consider *extractive summa-*

[1] But "formulaic" is not necessarily equivalent to "simple", so automated approaches still offer advantages over manual techniques, especially if one needs to model several domains.

rization: the compression of a document by choosing a subsequence of its sentences. For this task, we develop a new content-model-based learning algorithm for sentence selection. The resulting summaries yield 88% match with human-written output, which compares favorably to the 69% achieved by the standard "leading n sentences" baseline.

The success of content models in these two complementary tasks demonstrates their flexibility and effectiveness, and indicates that they are sufficiently expressive to represent important text properties. These observations, taken together with the fact that content models are conceptually intuitive and efficiently learnable from raw document collections, suggest that the formalism can prove useful in an even broader range of applications than we have considered here; exploring the options is an appealing line of future research.

2 Related Work

Knowledge-rich methods Models employing manual crafting of (typically complex) representations of content have generally captured one of three types of knowledge (Rambow, 1990; Kittredge et al., 1991): *domain knowledge* [e.g., that earthquakes have magnitudes], domain-independent *communication knowledge* [e.g., that describing an event usually entails specifying its location]; and *domain communication knowledge* [e.g., that Reuters earthquake reports often conclude by listing previous quakes[2]]. Formalisms exemplifying each of these knowledge types are DeJong's (1982) *scripts*, McKeown's (1985) *schemas*, and Rambow's (1990) *domain-specific schemas*, respectively.

In contrast, because our models are based on a distributional view of content, they will freely incorporate information from all three categories as long as such information is manifested as a recurrent pattern. Also, in comparison to the formalisms mentioned above, content models constitute a relatively impoverished representation; but this actually contributes to the ease with which they can be learned, and our empirical results show that they are quite effective despite their simplicity.

In recent work, Duboue and McKeown (2003) propose a method for learning a content planner from a collection of texts together with a domain-specific knowledge base, but our method applies to domains in which no such knowledge base has been supplied.

Knowledge-lean approaches Distributional models of content have appeared with some frequency in research on text segmentation and topic-based language modeling (Hearst, 1994; Beeferman et al., 1997; Chen et al., 1998; Florian and Yarowsky, 1999; Gildea and Hofmann, 1999; Iyer and Ostendorf, 1996; Wu and Khudanpur, 2002). In fact, the methods we employ for learning content models are quite closely related to techniques proposed in that literature (see Section 3 for more details).

However, language-modeling research — whose goal is to predict text probabilities — tends to treat topic as a useful auxiliary variable rather than a central concern; for example, topic-based distributional information is generally interpolated with standard, non-topic-based n-gram models to improve probability estimates. Our work, in contrast, treats content as a primary entity. In particular, our induction algorithms are designed with the explicit goal of modeling document content, which is why they differ from the standard Baum-Welch (or EM) algorithm for learning Hidden Markov Models even though content models are instances of HMMs.

3 Model Construction

We employ an iterative re-estimation procedure that alternates between (1) creating clusters of text spans with similar word distributions to serve as representatives of within-document topics, and (2) computing models of word distributions and topic changes from the clusters so derived.[3]

Formalism preliminaries We treat texts as sequences of pre-defined text spans, each presumed to convey information about a single topic. Specifying text-span length thus defines the granularity of the induced topics. For concreteness, in what follows we will refer to "sentences" rather than "text spans" since that is what we used in our experiments, but paragraphs or clauses could potentially have been employed instead.

Our working assumption is that all texts from a given domain are generated by a single content model. A *content model* is an HMM in which each state s corresponds to a distinct topic and generates sentences relevant to that topic according to a state-specific language model p_s — note that standard n-gram language models can therefore be considered to be degenerate (single-state) content models. State transition probabilities give the probability of changing from a given topic to another, thereby capturing constraints on topic shifts. We can use the *forward algorithm* to efficiently compute the generation probability assigned to a document by a content model and the *Viterbi* algorithm to quickly find the most likely content-model state sequence to have generated a given document; see Rabiner (1989) for details.

In our implementation, we use bigram language models, so that the probability of an n-word sentence $\mathbf{x} = w_1 w_2 \cdots w_n$ being generated by a state s is $p_s(\mathbf{x}) \overset{def}{=}$

[2] This does not qualify as domain knowledge because it is not about earthquakes per se.

[3] For clarity, we omit minor technical details, such as the use of dummy initial and final states. Section 5.2 describes how the free parameters k, T, δ_1, and δ_2 are chosen.

The Athens seismological institute said the temblor's epicenter was located 380 kilometers (238 miles) south of the capital.
Seismologists in Pakistan's Northwest Frontier Province said the temblor's epicenter was about 250 kilometers (155 miles) north of the provincial capital Peshawar.
The temblor was centered 60 kilometers (35 miles) northwest of the provincial capital of Kunming, about 2,200 kilometers (1,300 miles) southwest of Beijing, a bureau seismologist said.

Figure 1: Samples from an earthquake-articles sentence cluster, corresponding to descriptions of location.

$\prod_{i=1}^{n} p_s(w_i|w_{i-1})$. Estimating the state bigram probabilities $p_s(w'|w)$ is described below.

Initial topic induction As in previous work (Florian and Yarowsky, 1999; Iyer and Ostendorf, 1996; Wu and Khudanpur, 2002), we initialize the set of "topics", distributionally construed, by partitioning all of the sentences from the documents in a given domain-specific collection into clusters. First, we create k clusters via complete-link clustering, measuring sentence similarity by the cosine metric using word bigrams as features (Figure 1 shows example output).[4] Then, given our knowledge that documents may sometimes discuss new and/or irrelevant content as well, we create an "etcetera" cluster by merging together all clusters containing fewer than T sentences, on the assumption that such clusters consist of "outlier" sentences. We use m to denote the number of clusters that results.

Determining states, emission probabilities, and transition probabilities Given a set $c_1, c_2, \ldots, c_m$ of m clusters, where c_m is the "etcetera" cluster, we construct a content model with corresponding states $s_1, s_2, \ldots, s_m$; we refer to s_m as the *insertion state*.

For each state s_i, $i < m$, bigram probabilities (which induce the state's sentence-emission probabilities) are estimated using smoothed counts from the corresponding cluster:

$$p_{s_i}(w'|w) \stackrel{def}{=} \frac{f_{c_i}(ww') + \delta_1}{f_{c_i}(w) + \delta_1|V|},$$

where $f_{c_i}(y)$ is the frequency with which word sequence y occurs within the sentences in cluster c_i, and V is the vocabulary. But because we want the insertion state s_m to model digressions or unseen topics, we take the novel step of forcing its language model to be complementary to those of the other states by setting

$$p_{s_m}(w'|w) \stackrel{def}{=} \frac{1 - \max_{i:i<m} p_{s_i}(w'|w)}{\sum_{u \in V}(1 - \max_{i:i<m} p_{s_i}(u|w))}.$$

Note that the contents of the "etcetera" cluster are ignored at this stage.

Our state-transition probability estimates arise from considering how sentences from the same article are distributed across the clusters. More specifically, for two clusters c and c', let $D(c, c')$ be the number of documents in which a sentence from c immediately precedes one from c', and let $D(c)$ be the number of documents containing sentences from c. Then, for any two states s_i and s_j, $i, j \leq m$, we use the following smoothed estimate of the probability of transitioning from s_i to s_j:

$$p(s_j|s_i) = \frac{D(c_i, c_j) + \delta_2}{D(c_i) + \delta_2 m}.$$

Viterbi re-estimation Our initial clustering ignores sentence order; however, contextual clues may indicate that sentences with high lexical similarity are actually on different "topics". For instance, Reuters articles about earthquakes frequently finish by mentioning previous quakes. This means that while the sentence "The temblor injured dozens" at the beginning of a report is probably highly salient and should be included in a summary of it, the same sentence at the end of the piece probably refers to a different event, and so should be omitted.

A natural way to incorporate ordering information is iterative re-estimation of the model parameters, since the content model itself provides such information through its transition structure. We take an EM-like *Viterbi* approach (Iyer and Ostendorf, 1996): we re-cluster the sentences by placing each one in the (new) cluster c_i, $i \leq m$, that corresponds to the state s_i most likely to have generated it according to the Viterbi decoding of the training data. We then use this new clustering as the input to the procedure for estimating HMM parameters described above. The cluster/estimate cycle is repeated until the clusterings stabilize or we reach a predefined number of iterations.

4 Evaluation Tasks

We apply the techniques just described to two tasks that stand to benefit from models of content and changes in topic: information ordering for text generation and information selection for single-document summarization. These are two complementary tasks that rely on disjoint model functionalities: the ability to order a set of pre-selected information-bearing items, and the ability to do the selection itself, extracting from an ordered sequence of information-bearing items a representative subsequence.

4.1 Information Ordering

The information-ordering task is essential to many text-synthesis applications, including concept-to-text genera-

[4] Following Barzilay and Lee (2003), proper names, numbers and dates are (temporarily) replaced with generic tokens to help ensure that clusters contain sentences describing the same event type, rather than same actual event.

tion and multi-document summarization; While accounting for the full range of discourse and stylistic factors that influence the ordering process is infeasible in many domains, probabilistic content models provide a means for handling important aspects of this problem. We demonstrate this point by utilizing content models to select appropriate sentence orderings: we simply use a content model trained on documents from the domain of interest, selecting the ordering among all the presented candidates that the content model assigns the highest probability to.

4.2 Extractive Summarization

Content models can also be used for single-document summarization. Because ordering is not an issue in this application[5], this task tests the ability of content models to adequately represent domain topics independently of whether they do well at ordering these topics.

The usual strategy employed by domain-specific summarizers is for humans to determine *a priori* what types of information from the originating documents should be included (e.g., in stories about earthquakes, the number of victims) (Radev and McKeown, 1998; White et al., 2001). Some systems avoid the need for manual analysis by learning content-selection rules from a collection of articles paired with human-authored summaries, but their learning algorithms typically focus on within-sentence features or very coarse structural features (such as position within a paragraph) (Kupiec et al., 1999). Our content-model-based summarization algorithm combines the advantages of both approaches; on the one hand, it learns all required information from un-annotated document-summary pairs; on the other hand, it operates on a more abstract and global level, making use of the topical structure of the entire document.

Our algorithm is trained as follows. Given a content model acquired from the full articles using the method described in Section 3, we need to learn which topics (represented by the content model's states) should appear in our summaries. Our first step is to employ the Viterbi algorithm to tag all of the summary sentences and all of the sentences from the original articles with a *Viterbi topic label*, or *V-topic* — the name of the state most likely to have generated them. Next, for each state s such that at least three full training-set articles contained V-topic s, we compute the probability that the state generates sentences that should appear in a summary. This probability is estimated by simply (1) counting the number of document-summary pairs in the parallel training data such that both the originating document and the summary contain sentences assigned V-topic s, and then (2) normalizing this count by the number of full articles containing sentences with V-topic s.

Domain	Average Length	Standard Deviation	Vocabulary Size	Token/ type
Earthquakes	10.4	5.2	1182	13.2
Clashes	14.0	2.6	1302	4.5
Drugs	10.3	7.5	1566	4.1
Finance	13.7	1.6	1378	12.8
Accidents	11.5	6.3	2003	5.6

Table 1: Corpus statistics. Length is in sentences. Vocabulary size and type/token ratio are computed after replacement of proper names, numbers and dates.

To produce a length-ℓ summary of a new article, the algorithm first uses the content model and Viterbi decoding to assign each of the article's sentences a V-topic. Next, the algorithm selects those ℓ states, chosen from among those that appear as the V-topic of one of the article's sentences, that have the highest probability of generating a summary sentence, as estimated above. Sentences from the input article corresponding to these states are placed in the output summary.[6]

5 Evaluation Experiments

5.1 Data

For evaluation purposes, we created corpora from five domains: earthquakes, clashes between armies and rebel groups, drug-related criminal offenses, financial reports, and summaries of aviation accidents.[7] Specifically, the first four collections consist of AP articles from the North American News Corpus gathered via a TDT-style document clustering system. The fifth consists of narratives from the National Transportation Safety Board's database previously employed by Jones and Thompson (2003) for event-identification experiments. For each such set, 100 articles were used for training a content model, 100 articles for testing, and 20 for the development set used for parameter tuning. Table 1 presents information about article length (measured in sentences, as determined by the sentence separator of Reynar and Ratnaparkhi (1997)), vocabulary size, and token/type ratio for each domain.

5.2 Parameter Estimation

Our training algorithm has four free parameters: two that indirectly control the number of states in the induced content model, and two parameters for smoothing bigram probabilities. All were tuned separately for each domain on the corresponding held-out development set using Powell's grid search (Press et al., 1997). The parameter values were selected to optimize system performance

[5] Typically, sentences in a single-document summary follow the order of appearance in the original document.

[6] If there are more than ℓ sentences, we prioritize them by the summarization probability of their V-topic's state; we break any further ties by order of appearance in the document.

[7] `http://www.sls.csail.mit.edu/~regina/struct`

on the information-ordering task[8]. We found that across all domains, the optimal models were based on "sharper" language models (e.g., $\delta_1 < 0.0000001$). The optimal number of states ranged from 32 to 95.

5.3 Ordering Experiments

5.3.1 Metrics

The intent behind our *ordering* experiments is to test whether content models assign high probability to acceptable sentence arrangements. However, one stumbling block to performing this kind of evaluation is that we do not have data on ordering quality: the set of sentences from an N-sentence document can be sequenced in $N!$ different ways, which even for a single text of moderate length is too many to ask humans to evaluate. Fortunately, we do know that at least the original sentence order (*OSO*) in the source document must be acceptable, and so we should prefer algorithms that assign it high probability relative to the bulk of all the other possible permutations. This observation motivates our first evaluation metric: the *rank* received by the OSO when all permutations of a given document's sentences are sorted by the probabilities that the model under consideration assigns to them. The best possible rank is 0, and the worst is $N! - 1$.

An additional difficulty we encountered in setting up our evaluation is that while we wanted to compare our algorithms against Lapata's (2003) state-of-the-art system, her method doesn't consider all permutations (see below), and so the rank metric cannot be computed for it. To compensate, we report the *OSO prediction rate*, which measures the percentage of test cases in which the model under consideration gives highest probability to the OSO from among all possible permutations; we expect that a good model should predict the OSO a fair fraction of the time. Furthermore, to provide some assessment of the quality of the predicted orderings themselves, we follow Lapata (2003) in employing *Kendall's* τ, which is a measure of how much an ordering differs from the OSO—the underlying assumption is that most reasonable sentence orderings should be fairly similar to it. Specifically, for a permutation σ of the sentences in an N-sentence document, $\tau(\sigma)$ is computed as

$$\tau(\sigma) = 1 - 2\frac{S(\sigma)}{\binom{N}{2}},$$

where $S(\sigma)$ is the number of swaps of adjacent sentences necessary to re-arrange σ into the OSO. The metric ranges from -1 (inverse orders) to 1 (identical orders).

5.3.2 Results

For each of the 500 unseen test texts, we exhaustively enumerated all sentence permutations and ranked them using a content model from the corresponding domain. We compared our results against those of a bigram language model (the baseline) and an improved version of the state-of-the-art probabilistic ordering method of Lapata (2003), both trained on the same data we used. Lapata's method first learns a set of pairwise sentence-ordering preferences based on features such as noun-verb dependencies. Given a new set of sentences, the latest version of her method applies a Viterbi-style approximation algorithm to choose a permutation satisfying many preferences (Lapata, personal communication).[9]

Table 2 gives the results of our ordering-test comparison experiments. Content models outperform the alternatives almost universally, and often by a very wide margin. We conjecture that this difference in performance stems from the ability of content models to capture *global* document structure. In contrast, the other two algorithms are local, taking into account only the relationships between adjacent word pairs and adjacent sentence pairs, respectively. It is interesting to observe that our method achieves better results despite not having access to the linguistic information incorporated by Lapata's method. To be fair, though, her techniques were designed for a larger corpus than ours, which may aggravate data sparseness problems for such a feature-rich method.

Table 3 gives further details on the rank results for our content models, showing how the rank scores were distributed; for instance, we see that on the Earthquakes domain, the OSO was one of the top five permutations in 95% of the test documents. Even in Drugs and Accidents — the domains that proved relatively challenging to our method — in more than 55% of the cases the OSO's rank did not exceed ten. Given that the maximal possible rank in these domains exceeds three million, we believe that our model has done a good job in the ordering task.

We also computed learning curves for the different domains; these are shown in Figure 2. Not surprisingly, performance improves with the size of the training set for all domains. The figure also shows that the relative difficulty (from the content-model point of view) of the different domains remains mostly constant across varying training-set sizes. Interestingly, the two easiest domains, Finance and Earthquakes, can be thought of as being more formulaic or at least more redundant, in that they have the highest token/type ratios (see Table 1) — that is, in these domains, words are repeated much more frequently on average.

[8] See Section 5.5 for discussion of the relation between the ordering and the summarization task.

[9] Finding the optimal such permutation is NP-complete.

Domain	System	Rank	OSO pred.	τ
Earthquakes	Content	**2.67**	**72%**	**0.81**
	Lapata	(N/A)	24%	0.48
	Bigram	485.16	4%	0.27
Clashes	Content	**3.05**	**48%**	**0.64**
	Lapata	(N/A)	27%	0.41
	Bigram	635.15	12%	0.25
Drugs	Content	**15.38**	**38%**	0.45
	Lapata	(N/A)	27%	**0.49**
	Bigram	712.03	11%	0.24
Finance	Content	**0.05**	**96%**	**0.98**
	Lapata	(N/A)	18%	0.75
	Bigram	7.44	66%	0.74
Accidents	Content	**10.96**	**41%**	**0.44**
	Lapata	(N/A)	10%	0.07
	Bigram	973.75	2%	0.19

Table 2: Ordering results (averages over the test cases).

Domain	Rank range		
	[0-4]	[5-10]	> 10
Earthquakes	95%	1%	4%
Clashes	75%	18%	7%
Drugs	47%	8%	45%
Finance	100%	0%	0%
Accidents	52%	7%	41%

Table 3: Percentage of cases for which the content model assigned to the OSO a rank within a given range.

5.4 Summarization Experiments

The evaluation of our summarization algorithm was driven by two questions: (1) Are the summaries produced of acceptable quality, in terms of selected content? and (2) Does the content-model representation provide additional advantages over more locally-focused methods?

To address the first question, we compare summaries created by our system against the "lead" baseline, which extracts the first ℓ sentences of the original text — despite its simplicity, the results from the annual Document Understanding Conference (DUC) evaluation suggest that most single-document summarization systems cannot beat this baseline. To address question (2), we consider a summarization system that learns extraction rules directly from a parallel corpus of full texts and their summaries (Kupiec et al., 1999). In this system, summarization is framed as a sentence-level binary classification problem: each sentence is labeled by the publicly-available BoosTexter system (Schapire and Singer, 2000) as being either "in" or "out" of the summary. The features considered for each sentence are its unigrams and its location within the text, namely *beginning third*, *middle third* and *end third*.[10] Hence, relationships between sentences are not explicitly modeled, making this system a good basis for comparison.

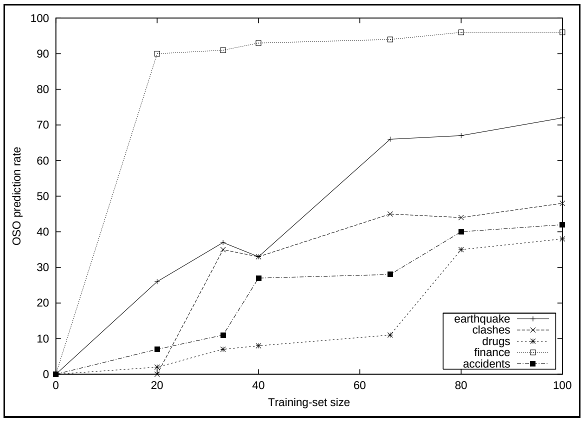

Figure 2: Ordering-task performance, in terms of OSO prediction rate, as a function of the number of documents in the training set.

We evaluated our summarization system on the Earthquakes domain, since for some of the texts in this domain there is a condensed version written by AP journalists. These summaries are mostly extractive[11]; consequently, they can be easily aligned with sentences in the original articles. From sixty document-summary pairs, half were randomly selected to be used for training and the other half for testing. (While thirty documents may not seem like a large number, it is comparable to the size of the training corpora used in the competitive summarization-system evaluations mentioned above.) The average number of sentences in the full texts and summaries was 15 and 6, respectively, for a total of 450 sentences in each of the test and (full documents of the) training sets.

At runtime, we provided the systems with a full document and the desired output length, namely, the length in sentences of the corresponding shortened version. The resulting summaries were judged as a whole by the fraction of their component sentences that appeared in the human-written summary of the input text.

The results in Table 4 confirm our hypothesis about the benefits of content models for text summarization — our model outperforms both the sentence-level, locally-focused classifier and the "lead" baseline. Furthermore, as the learning curves shown in Figure 3 indicate, our method achieves good performance on a small subset of parallel training data: in fact, the accuracy of our method on one third of the training data is higher than that of the

[10] This feature set yielded the best results among the several possibilities we tried.

[11] Occasionally, one or two phrases or, more rarely, a clause were dropped.

System	Extraction accuracy
Content-based	**88%**
Sentence classifier (words + location)	76%
Leading n sentences	69%

Table 4: Summarization-task results.

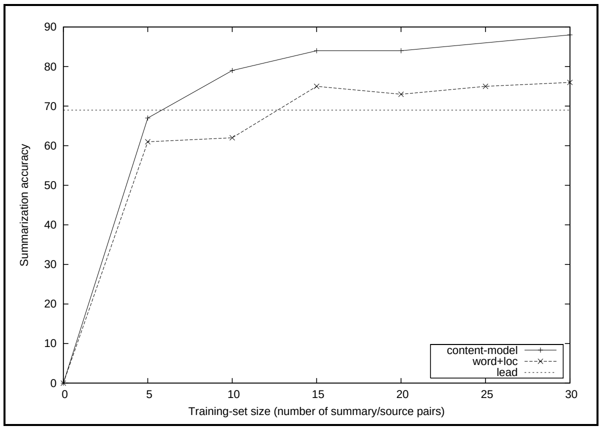

Figure 3: Summarization performance (extraction accuracy) on Earthquakes as a function of training-set size.

sentence-level classifier on the full training set. Clearly, this performance gain demonstrates the effectiveness of content models for the summarization task.

5.5 Relation Between Ordering and Summarization Methods

Since we used two somewhat orthogonal tasks, ordering and summarization, to evaluate the quality of the content-model paradigm, it is interesting to ask whether the same parameterization of the model does well in both cases. Specifically, we looked at the results for different model topologies, induced by varying the number of content-model states. For these tests, we experimented with the Earthquakes data (the only domain for which we could evaluate summarization performance), and exerted direct control over the number of states, rather than utilizing the cluster-size threshold; that is, in order to create exactly m states for a specific value of m, we merged the smallest clusters until m clusters remained.

Table 5 shows the performance of the different-sized content models with respect to the summarization task and the ordering task (using OSO prediction rate). While the ordering results seem to be more sensitive to the number of states, both metrics induce similar ranking on the models. In fact, the same-size model yields top performance on both tasks. While our experiments are limited to only one domain, the correlation in results is encouraging: optimizing parameters on one task promises to yield good performance on the other. These findings provide support for the hypothesis that content models are not only helpful for specific tasks, but can serve as effective representations of text structure in general.

Model size	10	20	40	60	**64**	80
Ordering	11%	28%	52%	50%	**72%**	57%
Summarization	54%	70%	79%	79%	**88%**	83%

Table 5: Content-model performance on Earthquakes as a function of model size. Ordering: OSO prediction rate; Summarization: extraction accuracy.

6 Conclusions

In this paper, we present an unsupervised method for the induction of content models, which capture constraints on topic selection and organization for texts in a particular domain. Incorporation of these models in ordering and summarization applications yields substantial improvement over previously-proposed methods. These results indicate that distributional approaches widely used to model various inter-sentential phenomena can be successfully applied to capture text-level relations, empirically validating the long-standing hypothesis that word distribution patterns strongly correlate with discourse patterns within a text, at least within specific domains.

An important future direction lies in studying the correspondence between our domain-specific model and domain-independent formalisms, such as RST. By automatically annotating a large corpus of texts with discourse relations via a rhetorical parser (Marcu, 1997; Soricut and Marcu, 2003), we may be able to incorporate domain-independent relationships into the transition structure of our content models. This study could uncover interesting connections between domain-specific stylistic constraints and generic principles of text organization.

In the literature, discourse is frequently modeled using a hierarchical structure, which suggests that probabilistic context-free grammars or hierarchical Hidden Markov Models (Fine et al., 1998) may also be applied for modeling content structure. In the future, we plan to investigate how to bootstrap the induction of hierarchical models using labeled data derived from our content models. We would also like to explore how domain-independent discourse constraints can be used to guide the construction of the hierarchical models.

Acknowledgments We are grateful to Mirella Lapata for providing us the results of her system on our data, and to Dominic Jones and Cindi Thompson for supplying us with their document collection. We also thank Eli Barzilay, Sasha Blair-Goldensohn, Eric Breck, Claire Cardie, Yejin Choi, Marcia Davidson, Pablo Duboue, Noémie

Elhadad, Luis Gravano, Julia Hirschberg, Sanjeev Khudanpur, Jon Kleinberg, Oren Kurland, Kathy McKeown, Daniel Marcu, Art Munson, Smaranda Muresan, Vincent Ng, Bo Pang, Becky Passoneau, Owen Rambow, Ves Stoyanov, Chao Wang and the anonymous reviewers for helpful comments and conversations. Portions of this work were done while the first author was a postdoctoral fellow at Cornell University. This paper is based upon work supported in part by the National Science Foundation under grants ITR/IM IIS-0081334 and IIS-0329064 and by an Alfred P. Sloan Research Fellowship. Any opinions, findings, and conclusions or recommendations expressed above are those of the authors and do not necessarily reflect the views of the National Science Foundation or Sloan Foundation.

References

F. C. Bartlett. 1932. *Remembering: a study in experimental and social psychology*. Cambridge University Press.

R. Barzilay, L. Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *HLT-NAACL 2003: Main Proceedings*, 16–23.

D. Beeferman, A. Berger, J. Lafferty. 1997. Text segmentation using exponential models. In *Proceedings of EMNLP*, 35–46.

S. F. Chen, K. Seymore, R. Rosenfeld. 1998. Topic adaptation for language modeling using unnormalized exponential models. In *Proceedings of ICASSP*, volume 2, 681–684.

G. DeJong. 1982. An overview of the FRUMP system. In W. G. Lehnert, M. H. Ringle, eds., *Strategies for Natural Language Processing*, 149–176. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

P. A. Duboue, K. R. McKeown. 2003. Statistical acquisition of content selection rules for natural language generation. In *Proceedings of EMNLP*, 121–128.

S. Fine, Y. Singer, N. Tishby. 1998. The hierarchical hidden Markov model: Analysis and applications. *Machine Learning*, 32(1):41–62.

R. Florian, D. Yarowsky. 1999. Dynamic non-local language modeling via hierarchical topic-based adaptation. In *Proceedings of the ACL*, 167–174.

D. Gildea, T. Hofmann. 1999. Topic-based language models using EM. In *Proceedings of EUROSPEECH*, 2167–2170.

Z. Harris. 1982. Discourse and sublanguage. In R. Kittredge, J. Lehrberger, eds., *Sublanguage: Studies of Language in Restricted Semantic Domains*, 231–236. Walter de Gruyter, Berlin; New York.

M. Hearst. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of the ACL*, 9–16.

R. Iyer, M. Ostendorf. 1996. Modeling long distance dependence in language: Topic mixtures vs. dynamic cache models. In *Proceedings of ICSLP*, 236–239.

D. R. Jones, C. A. Thompson. 2003. Identifying events using similarity and context. In *Proceedings of CoNLL*, 135–141.

R. Kittredge, T. Korelsky, O. Rambow. 1991. On the need for domain communication language. *Computational Intelligence*, 7(4):305–314.

J. Kupiec, J. Pedersen, F. Chen. 1999. A trainable document summarizer. In I. Mani, M. T. Maybury, eds., *Advances in Automatic Summarization*, 55–60. MIT Press, Cambridge, MA.

M. Lapata. 2003. Probabilistic text structuring: Experiments with sentence ordering. In *Proceeding of the ACL*, 545–552.

W. C. Mann, S. A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *TEXT*, 8(3):243–281.

D. Marcu. 1997. The rhetorical parsing of natural language texts. In *Proceedings of the ACL/EACL*, 96–103.

K. R. McKeown. 1985. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press, Cambridge, UK.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery. 1997. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, second edition.

L. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

D. R. Radev, K. R. McKeown. 1998. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500.

O. Rambow. 1990. Domain communication knowledge. In *Fifth International Workshop on Natural Language Generation*, 87–94.

J. Reynar, A. Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, 16–19.

R. E. Schapire, Y. Singer. 2000. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 2/3:135–168.

R. Soricut, D. Marcu. 2003. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the HLT/NAACL*, 228–235.

M. White, T. Korelsky, C. Cardie, V. Ng, D. Pierce, K. Wagstaff. 2001. Multi-document summarization via information extraction. In *Proceedings of the HLT Conference*, 263–269.

A. Wray. 2002. *Formulaic Language and the Lexicon*. Cambridge University Press, Cambridge.

J. Wu, S. Khudanpur. 2002. Building a topic-dependent maximum entropy language model for very large corpora. In *Proceedings of ICASSP*, volume 1, 777–780.