Learning Semantic Structures from In-domain Documents

by

Harr Chen

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2011

ARCHIVES

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

MAR 10 2011

LIBRARIES

© Massachusetts Institute of Technology 2011. All rights reserved.

Author . . . . . .
Department of Electrical Engineering and Computer Science
January 28, 2010

Certified by . . . . . .
Regina Barzilay
Associate Professor
Thesis Supervisor

Certified by . . . . . .
David R. Karger
Professor
Thesis Supervisor

Accepted by . . . . . .
Terry P. Orlando
Chairman, Department Committee on Graduate Theses

Learning Semantic Structures from In-domain Documents

by

Harr Chen

Submitted to the Department of Electrical Engineering and Computer Science
on January 28, 2010, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Semantic analysis is a core area of natural language understanding that has typically focused on predicting domain-independent representations. However, such representations are unable to fully realize the rich diversity of technical content prevalent in a variety of specialized domains. Taking the standard supervised approach to domain-specific semantic analysis requires expensive annotation effort for each new domain of interest. In this thesis, we study how multiple granularities of semantic analysis can be learned from *unlabeled* documents within the same domain. By exploiting in-domain *regularities* in the expression of text at various layers of linguistic phenomena, including lexicography, syntax, and discourse, the statistical approaches we propose induce multiple kinds of structure: *relations* at the phrase and sentence level, *content models* at the paragraph and section level, and *semantic properties* at the document level. Each of our models is formulated in a hierarchical Bayesian framework with the target structure captured as latent variables, allowing them to seamlessly incorporate linguistically-motivated prior and posterior constraints, as well as multiple kinds of observations. Our empirical results demonstrate that the proposed approaches can successfully extract hidden semantic structure over a variety of domains, outperforming multiple competitive baselines.

Thesis Supervisor: Regina Barzilay
Title: Associate Professor

Thesis Supervisor: David R. Karger
Title: Professor

Acknowledgments

This thesis is only possible because of the many wonderful people I have had the opportunity to work with over these last few years. Specifically, the work presented in this thesis is the result of collaborations with Edward (Ted) Benson, Branavan, Jacob Eisenstein, Tahira Naseem, and my advisors Regina Barzilay and David R. Karger. I am especially grateful to my advisors: David nurtured my original interest in probabilistic modeling, which Regina then catalyzed into a passion that expanded into natural language processing and beyond. I am also deeply indebted to Branavan, who has put up with my quirks and antics over many late nights in lab and taught me an immeasurable amount in the process.

This thesis, and my graduate school experience, additionally benefited from my committee members Marina Meilă and Michael Collins. In fact, it was through an informal seminar presented by Marina that I first learned of the Mallows model, which forms the basis of a significant portion of this thesis.

I would also like to thank the many other people that I have had the privilege of collaborating with during graduate school, including Kuang Chen, Neil Conway, Joseph Hellerstein, Mark Johnson, Marina Meilă, Tapan Parihk, and Luke Zettlemoyer. My membership in the RBG and Haystack groups also made my time that much more enjoyable.

Finally, this thesis would not have been possible without the support of many caring family members and friends. I will not explicitly list their names here, not because they are any less important than the people already mentioned, but because there are so many that I would certainly omit some by accident. It is these personal connections that I will continue to develop and cherish.

Bibliographic Note

Portions of this thesis are based upon material previously presented in peer-reviewed publications. The content modeling work of Chapter 2 builds from conference [35] and journal [34] publications. A version of the relation discovery work of Chapter 3 is currently under review for publication. Finally, the semantic properties model of Chapter 4 also derives from conference [27] and journal [28] publications.

Contents

1 Introduction **21**

1.1 Tasks 25

1.1.1 Content Modeling 26

1.1.2 Relation Discovery 27

1.1.3 Semantic Property Induction 28

1.2 Contributions 30

1.3 Outline 31

2 Learning Content Structure using Latent Permutations **33**

2.1 Related Work 36

2.1.1 Topic Models 36

2.1.2 Modeling Ordering Constraints in Discourse Analysis 39

2.2 Model 42

2.2.1 Problem Formulation 42

2.2.2 Model Overview 44

2.2.3 The Generalized Mallows Model over Permutations 45

2.2.4 Formal Generative Process 48

2.2.5 Properties of the Model 49

2.3 Inference via Collapsed Gibbs Sampling 51

2.4 Applications 56

2.4.1 Alignment 57

2.4.2 Segmentation 57

2.4.3 Ordering 58

2.5 Experiments 60
2.5.1 General Evaluation Setup 61
2.5.2 Alignment 64
2.5.3 Segmentation 68
2.5.4 Ordering 70
2.5.5 Discussion 75
2.6 Conclusions and Future Work 76

3 Learning Domain Relations using Posterior Regularization 79
3.1 Related Work 82
3.1.1 Extraction with Minimal and Alternative Supervision 82
3.1.2 Open Information Extraction 84
3.1.3 Syntax-driven Extraction 84
3.1.4 Constraint-based Extraction 85
3.2 Model 86
3.2.1 Problem Formulation 87
3.2.2 Model Overview 87
3.2.3 Formal Generative Process 91
3.2.4 Properties of the Model 92
3.3 Inference with Constraints 93
3.3.1 Variational Inference 94
3.3.2 Posterior Regularization 96
3.3.3 Variational Updates for the Model 98
3.4 Declarative Constraints 103
3.4.1 Syntax 104
3.4.2 Prevalence 105
3.4.3 Separation 105
3.5 Experiments 106
3.5.1 Evaluation Setup 106
3.5.2 Comparison against Unsupervised Baselines 110

3.5.3 Constraint Ablation Analysis 114
3.5.4 Comparison against Supervised Approaches 115
3.6 Conclusions and Future Work 119

4 Learning Semantic Properties using Free-text Annotations 123
4.1 Related Work 127
4.1.1 Bayesian Topic Modeling 127
4.1.2 Property Assessment for Review Analysis 128
4.1.3 Multi-document Summarization 131
4.2 Analysis of Free-Text Keyphrase Annotations 132
4.2.1 Incompleteness 133
4.2.2 Inconsistency 134
4.3 Model Description 136
4.3.1 Keyphrase Clustering 138
4.3.2 Document Topic Modeling 140
4.3.3 Generative Process 141
4.4 Inference via Gibbs Sampling 143
4.5 Overview of Experiments 146
4.6 Single-Document Experiments 148
4.6.1 Evaluation Setup 148
4.6.2 Results 152
4.7 Multiple-Document Experiments 161
4.7.1 Data and Setup 161
4.7.2 Aggregation Approaches 161
4.7.3 Results 162
4.8 Conclusions and Future Work 163

5 Conclusions 167
5.1 Future Work 169

A Development and Test Set Statistics for the Semantic Properties Experiments **171**

B Additional Multiple Review Summarization Results for the Semantic Properties Model **173**

List of Figures

1-1 Excerpts from PhoneArena.com cell phone reviews. 22

1-2 Excerpts from newswire articles about earthquakes. 22

2-1 The plate diagram and generative process for our content model, along with a table of notation for reference purposes. Shaded circles in the figure denote observed variables, and squares denote hyperparameters. The dotted arrows indicate that π is constructed deterministically from $\mathbf{v}$ according to algorithm Compute-π, and $\mathbf{z}$ is constructed deterministically from $\mathbf{t}$ and π according to Compute-$\mathbf{z}$. 43

2-2 The collapsed Gibbs sampling inference procedure for estimating our content model's posterior distribution. In each plate diagram, the variable being resampled is shown in a double circle and its Markov blanket is highlighted in black; other variables, which have no impact on the variable being resampled, are grayed out. Variables $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, shown in dotted circles, are never explicitly depended on or re-estimated, because they are marginalized out by the sampler. Each diagram is accompanied by the conditional resampling distribution for its respective variable. 52

3-1 Excerpts from newswire articles about earthquakes. The indicator and argument words for the *damage* relation are highlighted. 80

3-2 As input to the relation discovery model, words w and constituents x of syntactic parses are represented with indicator features ϕ^{i} and argument features ϕ^{a} respectively. A single relation instance is a pair of indicator w and argument x; we filter w to be nouns and verbs and x to be noun phrases. 86

3-3 The generative process for the relation discovery model. In the above, Z indicates a normalization factor that makes the parameters A_w and B_x sum to one. Fixed hyperparameters are subscripted with zero. . . 88

3-4 The plate diagram for the relation discovery model. Shaded circles in the figure denote observed variables, and squares denote hyperparameters. See Figure 3-3 for a full description of the variables. 89

3-5 Unconstrained variational update for $\hat{\zeta}$. 102

3-6 The F-score of the CLUTO clustering baseline as additional noise clusters are added compared to the relation discovery model's performance. 113

3-7 Comparison of the semi-supervised variant of our relation discovery model to two supervised baselines, a CRF sequence model and an SVM trained on sentences, on both datasets (top vs. bottom) and metrics (left vs. right). The x-axis measures the number of labeled documents provided to each system. Token-level performance for the SVM is not reported since the SVM does not predict individual relation tokens. . 117

4-1 Excerpts from online restaurant reviews with pros/cons phrase lists. Both reviews assert that the restaurant serves healthy food, but use different keyphrases. Additionally, the first review discusses the restaurant's good service, but is not annotated as such in its keyphrases. . . 124

4-2 Examples of the many different paraphrases related to the property *good price* that appear in the pros/cons keyphrases of reviews used for our inconsistency analysis. 134

4-3 Cumulative occurrence counts for the top ten keyphrases associated with the *good service* property. The percentages are out of a total of 1,210 separate keyphrase occurrences for this property. 135

4-4 The plate diagram for our semantic properties model. Shaded circles denote observed variables and squares denote hyperparameters. The dotted arrows indicate that η is constructed deterministically from x and h. We use ϵ to refer to a small constant probability mass. 137

4-5 A surface plot of the keyphrase similarity matrix from a set of restaurant reviews, computed according to Table 4.2. Red indicates high similarity, whereas blue indicates low similarity. In this diagram, the keyphrases have been grouped according to an expert-created clustering, so keyphrases of similar meaning are close together. The strong series of similarity "blocks" along the diagonal hint at how this information could induce a reasonable clustering. 139

4-6 Summary of reviews for the movie *Pirates of the Caribbean: At World's End* on PRÉCIS. This summary is based on 27 documents. The list of pros and cons are generated automatically using the system described in this chapter. The generation of numerical ratings is based on the algorithm described by Snyder and Barzilay [122]. 147

List of Tables

2.1 Statistics of the datasets used to evaluate our content model. All values except vocabulary size and document count are per-document averages. 60

2.2 Comparison of the alignments produced by our content model and a series of baselines and model variations, for both 10 and 20 topics, evaluated against clean and noisy sets of section headings. Higher scores are better. Within the same K, the methods which our model significantly outperforms are indicated with $*$ for $p < 0.001$ and $\diamond$ for $p < 0.01$. 66

2.3 Comparison of the segmentations produced by our content model and a series of baselines and model variations, for both 10 and 20 topics, evaluated against clean and noisy sets of section headings. Lower scores are better. BayesSeg and U&I are given the true number of segments, so their segments counts reflect the reference structures' segmentations. In contrast, $\overline{\text{U\&I}}$ automatically predicts the number of segments. . . . 71

2.4 Statistics of the training and test sets used for the content model ordering experiments. All values except vocabulary are the average per document. The training set statistics are reproduced from Table 2.1 for ease of reference. 72

2.5 Comparison of the orderings produced by our content model and a series of baselines and model variations, for both 10 and 20 topics, evaluated on the respective test sets. Higher scores are better. 74

3.1 Summary of constraints we consider for the relation discovery model. Each constraint takes the form $\mathbb{E}_q[f(z,a,i)] \leq b$ or $\mathbb{E}_q[f(z,a,i)] \geq b$ as indicated in the table; D denotes the number of corpus documents, $\forall k$ means one constraint per relation type, and $\forall w$ means one constraint per token in the corpus. 104

3.2 Corpus statistics for the datasets used for the relation model experiments. Sentence and token counts are per-document averages. 106

3.3 The manually annotated relation types identified in the finance and earthquake datasets with example instance arguments. 107

3.4 Comparison of our relation discovery model, with and without domain-specific constraints (DSC), to a series of unsupervised baselines and model variants on both domains. Lines are numbered for ease of reference in the text. For all scores higher is better. 112

4.1 Incompleteness and inconsistency in the restaurant domain for six prevalent semantic properties. The incompleteness figures are the recall, precision, and F-score of the author annotations (manually clustered into properties) against the gold standard property annotations. Inconsistency is measured by the number of different keyphrase realizations with at least five occurrences associated with each property, and the percentage frequency with which the most commonly occurring keyphrases is used to annotate a property. The averages in the bottom row are weighted according to frequency of property occurrence. 132

4.2 The two sources of information used to compute the similarity matrix for our semantic properties model. The final similarity scores are linear combinations of these two values. Note that co-occurrence similarity contains second-order co-occurrence information. 139

4.3 Statistics of the datasets used to evaluate our semantic properties model. 148

4.4 Values of the hyperparameters used for each domain across all experiments for the semantic properties model. 151

4.5 A summary of the baselines and variations against which our semantic properties model is compared. 153

4.6 Comparison of the property predictions made by our semantic properties model and a series of baselines and model variations in the restaurant domain, evaluated against expert semantic annotations. The results are divided according to experiment. The methods against which our model has significantly better results using approximate randomization are indicated with $*$ for $p \leq 0.05$, and $\diamond$ for $p \leq 0.1$. 154

4.7 Comparison of the property predictions made by our semantic properties model and a series of baselines and model variations in three product domains, as evaluated against author free-text annotations. The results are divided according to experiment. The methods against which our model has significantly better results using approximate randomization are indicated with $*$ for $p \leq 0.05$, and $\diamond$ for $p \leq 0.1$. Methods which perform significantly better than our model with $p \leq 0.05$ are indicated with $\dagger$. 155

4.8 Rand Index scores of our semantic properties model's clusters, learned from keyphrases and text jointly, compared against clusters learned only from keyphrase similarity. Evaluation of cluster quality is based on the gold standard clustering. 160

4.9 Comparison of the aggregated property predictions made by our semantic properties model and a series of baselines that use free-text annotations. The methods against which our model has significantly better results using approximate randomization are indicated with $*$ for $p \leq 0.05$. 162

A.1 Breakdown by property for the development and test sets used for the evaluations in section 4.6.2. 172

B.1 Comparison of the aggregated property predictions made by our semantic properties model and a series of baselines that only use free-text annotations. Aggregation requires three of five reviews to predict a property, rather than two as in Section 4.7. The methods against which our model has significantly better results using approximate randomization are indicated with $*$ for $p \leq 0.05$. 173

Chapter 1

Introduction

Building semantic representations of raw text is a core problem of natural language understanding. Such representations can come in a variety of forms — extracting phrase-level relations describing important entities and their interactions, structuring sentences and paragraphs into semantically cohesive topics, or clustering documents into groups implying similar semantic properties. These representations all provide a window into the *meaning* of the text in a structured manner, facilitating database-style access and processing. Semantic analysis also serves as a stepping stone toward other important language processing tasks, such as question answering, information retrieval, machine translation, and summarization.

Traditional approaches to semantic analysis typically target *domain-independent* output representations [4, 55, 123]. Under the usual supervised setup, such systems can take advantage of a number of annotated corpora. For phrase-level analysis, for example, resources such as PropBank [101], VerbNet [75], and FrameNet [48] provide detailed breakdowns of a number of predicate classes into their canonical argument representations. These resources are produced using heterogeneous corpora with annotations describing domain-independent semantic structure, *e.g.*, in FrameNet a *building* frame (predicate) is associated with an *agent* argument who uses *components* to construct a *created_entity*. Note that this frame refers to arbitrary notions of building across different domains, encompassing everything from architectural constructions to abstract assembling of ideas or concepts.

The newly-released Katana LX is the successor to the Katana II and latest addition to Sanyos Katana line ... We really like the minimalist design of the LX and our only complaint is the keypad and cheap materials that Sanyo continues to use ... Sanyos calling card has long been excellent reception ... The battery is rated at a more than respectable 4.8 hours (288 minutes) ...
From the minute you pick the Centro up its obvious that the device is totally different than the Treo ... Overall, we have found the design of the Centro to be excellent. It is as close to perfect as we have encountered for a full featured smartphone ... From our perspective the call quality of the Centro was good ... The battery is rated at 4 hours of talk time ...
The new Touch Cruise, successor to the P3300 Artemis should make you feel like a discoverer wherever you go ... Its overall appearance has changed, making the phone much more beautiful and classy compared to the last model ... A nice surprise was the call quality ... The battery will give you up to 7 hours of talk time ...

Figure 1-1: Excerpts from PhoneArena.com cell phone reviews.

A strong earthquake with a preliminary magnitude of 5.1 rocked part of Sumba island in eastern Indonesia ... There were no immediate reports of damage or casualties ... He located the quake's epicenter in the Sawu sea, between Sumba and Timor island, at a depth of 61 kilometers (38 miles) ...
Dozens of people were feared buried in the rubble of collapsed buildings Sunday after a strong earthquake with a preliminary magnitude of 6.0 rocked western Turkey. At least 14 people were killed and 193 others wounded ... Sunday's quake hit at 5:57 p.m. (1557 GMT) ...
A strong earthquake with numerous aftershocks knocked over buildings and killed at least 23 people in mountainous southwestern Yunnan province Tuesday morning ... The quake with a preliminary magnitude of 6.5 struck at about 6:46 am (2246 GMT) ... Beds shook in the provincial capital of Kunming, about 100 kilometers (60 miles) southeast of the epicenter in Wuding County ...

Figure 1-2: Excerpts from newswire articles about earthquakes.

In-domain Semantic Analysis Frequently, however, the sort of semantic knowledge we wish to glean from text is very specialized and domain-specific. For example, consider the excerpts from a set of cell phone reviews presented in Figure 1-1. The kinds of semantic information that would aptly characterize the information in these reviews is very particular to the cell phone domain — for instance, aspects of the phone's exterior design, battery information, and the audio quality in calls. Now consider the excerpts from newswire articles about earthquake incidents in Figure 1-2. There, target extractions include earthquake magnitude and epicenter, affected regions, information about casualties and damage, and date and time. These do not overlap with the semantic properties germane to the cell phone review domain.

Furthermore, these kinds of domain-specific, technical extractions are not encoded in domain-independent resources, which focus instead on broad classes of entity and clause interactions prevalent across heterogeneous corpora. For example, the nearest analogue in FrameNet for the phone-specific *battery life* is a generic *duration* relation, describing a *period* for any arbitrary *eventuality.* Such a broad relation encompasses semantic knowledge irrelevant to battery life, such as the phone's timeline for availability or its maximum video recording length.

Taking a supervised approach is costly for domain-specific analysis, due to the need to annotate target structures repeatedly for every new domain. Research in the domain-specific task of information extraction, for example, has primarily relied on several key annotated corpora, including terrorism news reports, citation text, seminar announcements, and corporate acquisition articles [61, 81, 92]. However, constructing such training instances requires extensive human labor, typically by annotators with expert domain knowledge following carefully designed guidelines [137]. We cannot expect that such an annotation enterprise can be undertaken for every possible domain of interest. Furthermore, an *exploratory* mechanism for understanding the structure of complex, unfamiliar domains is itself beneficial, particularly when domain experts are unavailable.

Linguistic Intuitions In this thesis, we focus on learning domain-specific and domain-relevant semantic analyses from raw text in a single domain *without* labeled data. We demonstrate that such a setup is feasible for learning a wide range of semantic structures at different levels of granularity. The common insight behind our various approaches is that documents within a single domain exhibit strong patterns of textual *regularity* that can drive the learning process [132]. These regularities occur at multiple layers of linguistic phenomena. For example, consider again the excerpts from cell phone reviews shown in Figure 1-1. At the lexicographic layer, a number of domain-specific terms provide strong cues for identifying different aspects of the review. Words such as "battery" and "hours," for instance, are indicative of battery-related information, whereas "call quality" and "reception" are cues for reception characteristics. We also observe regularities in the way relations are expressed at the syntactic level, consistent with prior studies showing that the bulk of relation instances are verbalized using a small set of syntactic patterns [6, 115]. For example, the verbalization of battery life is contained within the object of a clause whose subject is consistently the word "battery"; in two cases, the verb is also the same word "rated." At the document structure level, these documents are organized in similar ways, progressing through a discussion of exterior design, call quality, and battery life (along with many other topics not shown).

Computational Approach Unsupervised learning approaches have exploited various forms of these regularities in the past to deduce various forms of semantic structure [5, 8, 11, 23, 63, 83]. Clustering-based approaches [8, 83] use manually defined similarity metrics to drive induction. With the breadth of regularities we consider, spanning lexicography, syntax, and discourse, it is difficult to produce an appropriate domain-independent metric that can properly balance different sources of knowledge.

Instead, we take a *generative* Bayesian approach that uses latent variables to stochastically represent semantic structure, akin to various previous unsupervised models [11, 23, 63]. Compared to previous work, the approaches we propose utilize a broader range of knowledge sources to drive higher-accuracy induction, including:

- **Deeper regularities** that capture nuances of semantic structure coherence not exploited by past approaches,
- **Declaratively-specified constraints** that explicitly express linguistically-informed preferences for certain kinds of output structures, and
- **Noisy annotations** in the form of user-generated semantic metadata provided by the documents' authors in conjunction with the text.

To properly incorporate these disparate knowledge sources, our approaches take full advantage of the flexibility of the Bayesian framework by using:

- **Prior distributions** that are selected for their ability to directly encode linguistic intuitions,
- **Posterior constraints** to explicitly eliminate analyses that contradict declarative knowledge about the appropriate output structures, and
- **Model structures** that generate different sets of observed data from shared latent structures, allowing multiple information sources to be easily unified in a principled manner.

Furthermore, our proposed approaches enjoy all the other benefits of a fully Bayesian generative framework, such as a rich variety of principled approximate inference techniques for performing parameter estimation.

1.1 Tasks

Since we are interested in inducing a range of semantic structures using different kinds of constraints to guide unsupervised learning, we tackle three tasks at different levels of textual granularity. The common thread linking each of these models is the incorporation of linguistic knowledge in a principled manner. Here, we describe the different tasks, identify the regularities and constraints we exploit in modeling, and summarize our approaches and empirical findings.

1.1.1 Content Modeling

Task Description The goal of content modeling is to uncover the hidden topic structure of documents at the paragraph and section level. Content models assign each discourse unit of a document (typically a paragraph in our work) to a numeric topic value; discourse units in different documents with the same topic assignment should be semantically related. We note that these topics should reflect domain-specific structure, rather than the broad discourse relations encoded by domain-independent annotated resources such as the Penn Discourse Treebank [107]. For example, for the cell phone reviews possible topics could correspond to *hardware design*, *call quality*, and *battery*. Content modeling is an important step in a variety of linguistic processing pipelines, contributing in particular to summarization [11] and text analysis applications [119].

Regularities The regularities we rely on for content modeling are at the word and document structure levels. First, the same topic should share similar word choices across documents, *e.g.*, a *battery* topic would be associated with words "battery" and "hours." At the document structure level, we observe that in-domain documents exhibit recurring patterns in topic organization. For example, a cell phone review will dwell longer on hardware design and less on infrequently used features such as voice dialing, and will typically present the former before the latter. This *global* regularity in ordering is richer than the Markovian constraints exploited by previous work [11, 63, 108].

Approach and Findings The content model that we propose directly captures regularity in organizational regularity by biasing the model toward content structures coherent in topic organization. In the generative process, the hidden variables encode both the frequency of topics and their order of occurrence within a document. The coherence bias is instituted via careful choices of the hidden variable priors. In particular, topic frequencies are drawn from a common multinomial distribution and topic orderings from a common *Generalized Mallows Model*, a simple and compact

distribution over topic permutations that places highest probability mass on similar orderings. Parameter estimation is accomplished with an efficient collapsed Gibbs sampler that learns the hidden variables given document text. We apply the model to inter-document paragraph alignment, intra-document topic segmentation, and inductive information ordering, and find that the model outperforms previous state-of-the-art baselines on each task. Our experiments also show that the Generalized Mallows Model component is successful at capturing ordering regularity. In particular, replacing it with a more or less tightly constrained distribution over permutations degrades performance.

1.1.2 Relation Discovery

Task Description The task of relation discovery is to find important clusters of phrases across documents that characterize domain-relevant attributes of the text. For example, possible relations for the cell phone reviews include *battery life* and *call quality*, with corresponding values "4.8 hours" and "excellent." Note the difference in granularity between content modeling and relation modeling — content topics tend to characterize general themes of a section of text (*e.g.*, this paragraph discusses the *battery*), while relations identify individual facts and attributes (*e.g.*, this phrase gives the precise value of the *battery life*). Relation analysis provides a database-oriented view of text that supports a variety of applications such as question answering and structured search.

Regularities Regularities in relation expression occur in lexicography, syntax, and document structure. At the lexicographic level, instances of a single relation share word attributes, such as having numerals or proper nouns. Furthermore, these instances are typically situated near words in the same sentence that are likely to be "evocative" of the relation, such as "battery" for *battery life*. We also observe that the relation-evoking word and relation phrase will typically be linked with particular syntactic patterns, *e.g.*, the word "battery" being the grammatical subject of the verb whose object is *battery life*. Finally, as with content modeling, across well-structured

documents the same relation information tends to occur in the same relative locations — for example, call quality information at the beginning of documents, and battery life toward the end. Previous work [5, 104] has not exploited regularities at all these levels of textual phenomena simultaneously.

Approach and Findings Our approach to relation discovery integrates all these sources of evidence in a unified model through a combination of modeling techniques. Relations are encoded as clusters of related *indicator* words and *argument* phrases across documents. Indicators represent relation-evoking words, while arguments comprise the actual phrase values of the relations. We propose a generative model whose structure encourages coherence in lexicographic and syntactic properties of the indicator and argument values, as well as consistency in the relative position of a relation within each document. Using the *posterior regularization* technique, we impose additional domain-independent declarative constraints on relation expression during variational inference, particularly on how indicators and arguments are syntactically linked. Our results show that the model is effective at discovering relation types and their instances, and that further performance improvements can be achieved by introducing simple additional domain-specific declarative knowledge. We also find that the declarative constraints are crucial for high accuracy, as removing any of them hurts performance.

1.1.3 Semantic Property Induction

Task Description Semantic property induction concerns finding clusters of documents that express semantically similar domain-specific characteristics. For example, we may identify that some subset of the cell phone reviews implies the property of *good audio quality*, while some other subset supports *bad price*. Unlike both relation and content modeling, the goal here is to glean semantic properties relevant to a whole document. Identifying semantic properties provides a way of concisely distilling documents to their key points, thus facilitating rapid browsing and multi-document summarization.

In our semantic property setup, we assume access to an additional source of noisy information in the form of *free-text* annotations. For reviews, these come in the form of pros/cons keyphrases written by the document author to summarize their own review; for blog postings, tags are an example of such annotations. We use these free-text annotations as a complementary source of observed information to improve semantic property induction.

Regularities Here, the regularities that drive learning are at both the word and keyphrase levels. Documents expressing similar properties should use similar words, such as "expensive" or "overpriced" for *bad price.* They should also have similar keyphrase annotations, such as "way too expensive" or "not a good value." A key challenge of using free-text annotations is that we do not know *a priori* which different keyphrases express the same semantic property.

Approach and Findings The model we propose addresses uncertainty in keyphrase clusters and document text by modeling both jointly, learning a single set of semantic properties from both kinds of observed data. The key technical challenge in modeling keyphrases is that they are fraught with inconsistent phrase usage and are frequently incomplete representations of the corresponding text. To properly learn from such annotations, we propose a Bayesian model that induces a hidden clustering of the keyphrases linked to the topic word distributions, and a Gibbs sampling algorithm for parameter estimation of the model. By jointly inferring the clustering and the word distributions, our model learns parallel views of each topic as both keyphrases and words. Our results show that this technique is effective for both predicting properties of individual documents, as well as summarizing multiple documents, compared to baselines using standard supervised approaches. Furthermore, we demonstrate that joint learning of keyphrase clusters and word models is superior to learning either in isolation.

1.2 Contributions

Using Broad Linguistic Knowledge for Semantic Analysis The main contribution of this thesis is a demonstration that multiple forms of linguistically-motivated regularities, situated in a unified, principled probabilistic formalism, can drive the induction of a wide range of useful domain-specific semantic structures. While such regularities have been used for unsupervised learning in the past, the models we propose apply a wider range of previously explored constraints and exploit new linguistic insights to drive learning. Our content model looks at *global* regularities in content selection and ordering, above the level of the local transition properties exploited by previous work [11, 63]. Our relation model uses both model structure and posterior constraints to simultaneously learn from regularities at all three levels of lexicography, syntax, and document structure, more than what has been studied by earlier approaches [5, 83, 104]. Our semantic properties model discovers latent topics that have parallel views as both language models and keyphrase annotation distributions, a richer representation than previous word-only topic modeling approaches [23, 125] that nonetheless does not require manual expert annotations [21]. As a consequence of our rich modeling, our models are able to learn more robust and accurate structures that outperform a variety of previous approaches.

Technical Modeling Insights A secondary contribution of this thesis is the technical modeling ideas underlying our approaches:

- Our content model is the first NLP work to apply the Generalized Mallows Model, a popular probability model over permutations previously used for learning rankings, to the problem of modeling linguistic orderings. We expect that the Mallows model can be reused as a component of other ordering-sensitive linguistic tasks, such as semantic role labeling, multilingual part-of-speech tagging, and grounded language acquisition.
- Our relation model demonstrates that domain-independent meta-constraints, applied via posterior regularization, can guide the induction of relation types

in conjunction with a generative process over words and documents. This work suggests that the posterior regularization framework is a promising technique for encoding domain-independent declarative linguistic knowledge for other tasks, such as sentiment analysis and summarization.

- Our results with the semantic properties model show that the signal contained within either long prose or summary phrases can help disambiguate the noise in the other. This finding makes a case for the joint generative modeling of multiple sources of evidence.

1.3 Outline

The remainder of this thesis proceeds as follows.

- **Chapter 2** describes our approach to content modeling, focusing on how the Generalized Mallows Model is a particularly suitable choice of distribution for topic orderings.
- **Chapter 3** presents the details of our relation discovery model, and explains how the posterior regularization technique is applied for enforcing declarative constraints.
- **Chapter 4** explains our model for learning semantic properties, focusing on the noise inherent to free-text annotations and how they can be mitigated with joint modeling.
- **Chapter 5** summarizes the main points of this thesis and presents avenues for future work.

Chapter 2

Learning Content Structure using Latent Permutations

In this chapter, we describe an unsupervised approach to learning document-level content structure, a central problem of discourse analysis. This structure encompasses the topics that are addressed and the order in which these topics appear across documents in a single domain. Modeling content structure is particularly germane for domains that exhibit recurrent patterns in content organization, such as news and encyclopedia articles. Our model aims to induce, for example, that articles about cities typically contain information about History, Economy, and Transportation, and that descriptions of History usually precede those of Transportation.

Previous work [11, 46] has demonstrated that content models can be learned from raw unannotated text, and are useful in a variety of text processing tasks such as summarization and information ordering. However, the expressive power of these approaches is limited: by taking a Markovian view on content structure, they only model *local* regularities in topic organization, such as topic transitions. This shortcoming is substantial since many discourse regularities described in the literature are *global* in nature [58, 120].

Our model of content structure explicitly represents two important global constraints on topic selection.[1] The first constraint posits that each document follows a

[1]We will use "topic" to refer interchangeably to both the discourse unit and language model

progression of coherent, nonrecurring topics [64]. Following the example above, this constraint captures the notion that a single topic, such as History, is expressed in a contiguous block within the document, rather than spread over disconnected sections. The second constraint states that documents from the same domain tend to exhibit regularity in organization, *i.e.*, they present similar topics in similar orders [7, 132]. This constraint guides toward selecting sequences with similar topic *ordering*, such as placing History before Transportation. While these constraints are not universal across all genres of human discourse, they are applicable to many important domains, ranging from newspaper text to product reviews.[2]

We present a latent topic model over in-domain documents that encodes these discourse constraints by positing a single distribution over the *entirety* of a document's content ordering. Specifically, we represent content structure as a *permutation* over topics. This naturally enforces the first constraint since a permutation does not allow topic repetition. To learn the distribution over permutations, we employ the *Generalized Mallows Model* (GMM). This model concentrates probability mass on permutations close to a *centroid* permutation. Permutations drawn from this distribution are likely to be similar, allowing it to capture the regularities expressed by the second constraint. A major benefit of the GMM is its compact parameterization using a set of real-valued *dispersion* values. These dispersion parameters allow the model to learn how strongly to bias each document's topic ordering toward the centroid permutation. Furthermore, the number of parameters grows linearly with the number of topics, thus sidestepping tractability problems typically associated with the large discrete space of permutations.

We position the GMM within a larger hierarchical Bayesian model that explains how a set of in-domain documents is generated. For each document, the model posits that a topic ordering is drawn from the GMM, and that a set of topic frequencies is drawn from a multinomial distribution. Together, these draws specify the document's entire topic structure, in the form of topic assignments for each textual unit. As with

views of a topic.

[2]An example of a domain where the first constraint is violated is dialogue. Texts in such domains follow the stack structure that allows topics to recur throughout a conversation [62].

other topic models, such as our semantic properties model (Chapter 4), words are then drawn from language models indexed by topic. This model structure encourages probability mass to be placed on content structures that are highly regular within a domain, both in the relative frequency of topics in a document as well as their ordering. To estimate the model posterior, we perform Gibbs sampling over the topic structures and GMM dispersion parameters while analytically integrating out the remaining hidden variables.

We apply our model to three complex document-level tasks. First, in the *alignment* task, we aim to discover paragraphs across different documents that share the same topic. In our experiments, our permutation-based model outperforms the Hidden Topic Markov Model [63] by a wide margin — the gap averaged 28% percentage points in F-score. Second, we consider the *segmentation* task, where the goal is to partition each document into a sequence of topically coherent segments. The model yields an average P_k measure of 0.231, a 7.9% percentage point improvement over a competitive Bayesian segmentation method that does not take global constraints into account [44]. Third, we apply our model to the *ordering* task, that is, sequencing a held out set of textual units into a coherent document. As with the previous two applications, the difference between our model and a state-of-the-art baseline is substantial: our model achieves an average *Kendall's* τ of 0.602, compared to a value of 0.267 for the HMM-based content model [11].[3]

The success of the permutation-based model in these three complementary tasks demonstrates its flexibility and effectiveness, and attests to the versatility of the general document structure induced by our model. We find that encoding global ordering constraints into topic models makes them more suitable for discourse-level analysis, in contrast to the local decision approaches taken by previous work. Furthermore, in most of our evaluation scenarios, our full model yields significantly better results than its simpler variants that either use a fixed ordering or are order-agnostic.

The remainder of this chapter proceeds as follows. In Section 2.1, we describe how

[3]See Sections 2.5.3 and 2.5.4 for definitions of the segmentation and ordering evaluation metrics, respectively.

our approach relates to previous work in both topic modeling and statistical discourse processing. We provide a problem formulation in Section 2.2.1 followed by an overview of our content model in Section 2.2.2. At the heart of this model is the distribution over topic permutations, for which we provide background in Section 2.2.3, before employing it in a formal description of the model's probabilistic generative story in Section 2.2.4. Section 2.3 discusses the estimation of the model's posterior distribution given example documents using a collapsed Gibbs sampling procedure. Techniques for applying our model to the three tasks of alignment, segmentation, and ordering are explained in Section 2.4. We then evaluate our model's performance on each of these tasks in Section 2.5 before concluding by touching upon directions for future work in Section 2.6.

2.1 Related Work

We describe two bodies of previous work related to our approach. From the algorithmic perspective our work falls into a broad class of topic models. While earlier work on topic modeling took the bag of words view of documents, many recent approaches have expanded topic models to capture some structural constraints. In Section 2.1.1, we describe these extensions and highlight their differences from our model. On the linguistic side, our work relates to research on modeling text structure in statistical discourse processing. We summarize this work in Section 2.1.2, drawing comparisons with the functionality supported by our model.

2.1.1 Topic Models

Probabilistic topic models, originally developed in the context of language modeling, have today become popular for a range of NLP applications, such as text classification and document browsing. Topic models posit that a latent state variable controls the generation of each word. Their parameters are estimated using approximate inference techniques such as Gibbs sampling and variational methods. In traditional topic models such as *Latent Dirichlet Allocation* (LDA) [23, 59], documents are treated

as bags of words, where each word receives a separate topic assignment and words assigned to the same topic are drawn from a shared language model. While the bag of words representation is sufficient for some applications, in many cases this structure-unaware view is too limited. Previous research has considered extensions of LDA models in two orthogonal directions, covering both *intrasentential* and *extrasentential* constraints.

Modeling Intrasentential Constraints

One promising direction for improving topic models is to augment them with constraints on topic assignments of adjoining words within sentences. For example, Griffiths et al. [60] propose a model that jointly incorporates both *syntactic* and *semantic* information in a unified generative framework and constrains the syntactic classes of adjacent words. In their approach, the generation of each word is controlled by two hidden variables, one specifying a semantic topic and the other specifying a syntactic class. The syntactic class hidden variables are chained together as a Markov model, whereas semantic topic assignments are assumed to be independent for every word.

As another example of intrasentential constraints, Wallach [130] proposes a way to incorporate word order information, in the form of bigrams, into an LDA-style model. In this approach, the generation of each word is conditioned on both the previous word and the topic of the current word, while the word topics themselves are generated from per-document topic distributions as in LDA. This formulation models text structure at the level of word transitions, as opposed to the work of Griffiths et al. [60] where structure is modeled at the level of hidden syntactic class transitions.

Our focus is on modeling high-level document structure in terms of its semantic content. As such, our work is complementary to methods that impose structure on intrasentential units; it should be possible to combine our model with constraints on adjoining words.

Modeling Extrasentential Constraints

Given the intuitive connection between the notion of topic in LDA and the notion of topic in discourse analysis, it is natural to assume that LDA-like models can be useful for discourse-level tasks such as segmentation and topic classification. This hypothesis motivated research on models where topic assignment is guided by structural considerations [108, 63, 125], particularly relationships between the topics of adjacent *textual units*. Depending on the application, a textual unit may be a sentence, paragraph, or speaker utterance. A common property of these models is that they bias topic assignments to cohere within local segments of text.

Models in this category vary in terms of the mechanisms used to encourage local topic coherence. For instance, the model of Purver et al. [108] biases the topic distributions of adjacent utterances (textual units) in discourse transcripts to be similar. Their model generates each utterance from a mixture of topic language models. The parameters of this topic mixture distribution is assumed to follow a type of Markovian transition process — specifically, with high probability an utterance u will have the same topic distribution as the previous utterance $u - 1$; otherwise, a new topic distribution is drawn for u. Thus, each textual unit's topic distribution only depends on the previous textual unit, controlled by a parameter indicating whether a new topic distribution is drawn.

In a similar vein, the *Hidden Topic Markov Model* (HTMM) [63] posits a generative process where each sentence (textual unit) is assigned a *single* topic, so that all of the sentence's words are drawn from a single language model. As with the model of Purver et al., topic transitions between adjacent textual units are modeled in a Markovian fashion — specifically, sentence i has the same topic as sentence $i - 1$ with high probability, or receives a new topic assignment drawn from a shared topic multinomial distribution.

In both HTMM and our model, the assumption of a *single* topic per textual unit allows sections of text to be related across documents by topic. In contrast, Purver et al.'s model is tailored for the task of segmentation, so each utterance is drawn

from a mixture of topics. Thus, their model does not capture how utterances are topically *aligned* across in-domain documents. More importantly, both HTMM and the model of Purver et al. are only able to make local decisions regarding topic transitions, and thus have difficulty respecting long-range discourse constraints such as topic contiguity. Our model instead takes a *global* view on topic assignments for all textual units by explicitly generating an entire document's topic ordering from one joint distribution. As we show later in this chapter, this global view yields significant performance gains.

More recently, the *Multi-Grain Latent Dirichlet Allocation* (MGLDA) model [125] has also studied topic assignments at the level of sub-document textual units. In MGLDA, a set of *local* topic distributions is induced for each sentence, dependent on a window of local context around the sentence. Individual words are then drawn either from these local topics or from document-level topics as in standard LDA. MGLDA represents local context using a *sliding window*, where each window frame comprises overlapping short spans of sentences. In this way, local topic distributions are shared between sentences in close proximity.

MGLDA can represent more complex topical dependencies than the models of Purver et al. and Gruber et al., because the window can incorporate a much wider swath of local context than two adjacent textual units. However, MGLDA is unable to encode longer range constraints, such as contiguity and ordering similarity, because sentences not in close proximity are only loosely connected through a series of intervening window frames. In contrast, our work is specifically oriented toward these long-range constraints, necessitating a whole-document notion of topic assignment.

2.1.2 Modeling Ordering Constraints in Discourse Analysis

The global constraints encoded by our model are closely related to research in discourse on information ordering with applications to text summarization and generation [9, 46, 73, 79]. The emphasis of that body of work is on learning ordering constraints from data, with the goal of reordering new text from the same domain. These methods build on the assumption that recurring patterns in topic ordering can

be discovered by analyzing patterns in word distribution. The key distinction between prior methods and our approach is that existing ordering models are largely driven by local constraints with limited ability to capture global structure. Below, we describe two main classes of probabilistic ordering models studied in discourse processing.

Discriminative Models

Discriminative approaches aim directly to predict an ordering for a given set of sentences. Modeling the ordering of all sentences simultaneously leads to a complex structure prediction problem. In practice, however, a more computationally tractable two-step approach is taken: first, probabilistic models are used to estimate pairwise sentence ordering preferences; next, these local decisions are combined to produce a consistent global ordering [3, 79]. Training data for pairwise models is constructed by considering all pairs of sentences in a document, with supervision labels based on how they are actually ordered. Prior work has demonstrated that a wide range of features are useful in these classification decisions [24, 68, 73, 79]. For instance, Lapata [79] demonstrated that lexical features, such as verb pairs from the input sentences, serve as a proxy for plausible sequences of actions, and thus are effective predictors of well-formed orderings. During the second stage, these local decisions are integrated into a global order that maximizes the number of consistent pairwise classifications. Since finding such an ordering is NP-hard [40], various approximations are used in practice [3, 79].

While these two-step discriminative approaches can effectively leverage information about local transitions, they do not provide any means for representing global constraints. In more recent work, Barzilay and Lapata [10] demonstrated that certain global properties can be captured in the discriminative framework using a reranking mechanism. In this set-up, the system learns to identify the best global ordering given a set of n possible candidate orderings. The accuracy of this ranking approach greatly depends on the quality of selected candidates. Identifying such candidates is a challenging task given the large search space of possible alternatives.

The approach presented in this work differs from existing discriminative models

in two ways. First, our model represents a distribution over *all* possible global orderings. Thus, we can use sampling mechanisms that consider this whole space rather than being limited to a subset of candidates as with ranking models. The second difference arises out of the generative nature of our model. Rather than focusing on the ordering task, our order-aware model effectively captures a layer of hidden variables that explain the underlying structure of document content. Thus, it can be effectively applied to a wider variety of applications, including those where sentence ordering is already observed, by appropriately adjusting the observed and hidden components of the model.

Generative Models

Our work is closer in technique to generative models that treat topics as hidden variables. One instance of such work is the Hidden Markov Model (HMM)-based content model [11]. In their model, states correspond to topics and state transitions represent ordering preferences; each hidden state's emission distribution is then a language model over words. Thus, similar to our approach, these models implicitly represent patterns at the level of topical structure. The HMM is then used in the ranking framework to select an ordering with the highest probability.

In more recent work, Elsner et al. [46] developed a search procedure based on simulated annealing that finds a high likelihood ordering. In contrast to ranking-based approaches, their search procedure can cover the entire ordering space. On the other hand, as we show in Section 2.4.3, we can define an ordering objective that can be maximized very efficiently over all possible orderings during prediction once the model parameters have been learned. Specifically, for a bag of p paragraphs, only $O(pK)$ calculations of paragraph probabilities are necessary, where K is the number of topics.

Another distinction between our proposed model and prior work is in the way global ordering constraints are encoded. In a Markovian model, it is possible to induce some global constraints by introducing additional local constraints. For instance, topic contiguity can be enforced by selecting an appropriate model topology (*e.g.*, by

augmenting hidden states to record previously visited states). However, other global constraints, such as similarity in overall ordering across documents, are much more challenging to represent. By explicitly modeling the topic permutation distribution, we can easily capture this kind of global constraint, ultimately resulting in more accurate topic models and orderings. As we show later in this chapter, our model substantially outperforms the approach of Barzilay and Lee [11] on the information ordering task to which they applied the HMM-based content model.

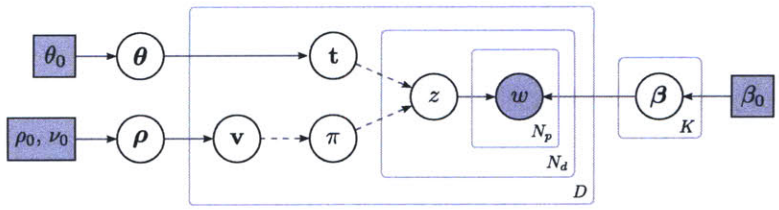
2.2 Model

In this section, we describe our problem formulation and proposed model.

2.2.1 Problem Formulation

Our content modeling problem can be formalized as follows. We take as input a corpus $\{d_1, \ldots d_D\}$ of in-domain documents, and a specification of a number of topics K.[4] Each document d is comprised of an ordered sequence of N_d paragraphs $(p_{d,1}, \ldots, p_{d,N_d})$. As output, we predict a single *topic assignment* $z_{d,p} \in \{1, \ldots, K\}$ for each paragraph p.[5] These $\mathbf{z}$ values should reflect the underlying content organization of each document — related content discussed within each document, and across separate documents, should receive the same z value.

Our formulation shares some similarity with the standard LDA setup in that a common set of topics is assigned across a collection of documents. The difference is that in LDA each word's topic assignment is conditionally independent, following the bag of words view of documents, whereas our constraints on how topics are assigned let us connect word distributional patterns to document-level topic structure.



$\boldsymbol{\theta}$ – parameters of distribution over topic counts

$\boldsymbol{\rho}$ – parameters of distribution over topic orderings

$\mathbf{t}$ – vector of topic counts

$\mathbf{v}$ – vector of inversion counts

π – topic ordering

z – paragraph topic assignment

$\boldsymbol{\beta}$ – language model parameters of each topic

w – document words

K – number of topics

D – number of documents in corpus

N_d – number of paragraphs in document d

N_p – number of words in paragraph p

$\boldsymbol{\theta} \sim \text{Dirichlet}(\theta_0)$

for $j = 1 \ldots K-1$
  $\rho_j \sim \text{GMM}_0(\rho_0, \nu_0)$

for $k = 1 \ldots K$
  $\boldsymbol{\beta}_k \sim \text{Dirichlet}(\beta_0)$

for each document d
  $\mathbf{t}_d \sim \text{Multinomial}(\boldsymbol{\theta})$
  $\mathbf{v}_d \sim \text{GMM}(\boldsymbol{\rho})$
  $\pi_d = \text{Compute-}\pi(\mathbf{v}_d)$
  $\mathbf{z}_d = \text{Compute-}\mathbf{z}(\mathbf{t}_d, \pi_d)$

  for each paragraph p in d
    for each word w in p
      $w \sim \text{Multinomial}(\boldsymbol{\beta}_{z_{d,p}})$

Algorithm: Compute-π
Input: Inversion count vector $\mathbf{v}$
Output: Permutation π

Create an empty list π
$\pi[1] \leftarrow K$
for $j = K-1$ down to 1
  for $i = K-1$ down to $\mathbf{v}[j]$
    $\pi[i+1] \leftarrow \pi[i]$
  $\pi[\mathbf{v}[j]] \leftarrow j$

Algorithm: Compute-$\mathbf{z}$
Input: Topic counts $\mathbf{t}$, permutation π
Output: Paragraph topic vector $\mathbf{z}$

Create an empty list $\mathbf{z}$
$end \leftarrow 1$
for $k = K$ to 1
  for $i = 1$ to $\mathbf{t}[\pi[k]]$
    $\mathbf{z}[end] \leftarrow \pi[k]$
    $end \leftarrow end + 1$

Figure 2-1: The plate diagram and generative process for our content model, along with a table of notation for reference purposes. Shaded circles in the figure denote observed variables, and squares denote hyperparameters. The dotted arrows indicate that π is constructed deterministically from $\mathbf{v}$ according to algorithm Compute-π, and $\mathbf{z}$ is constructed deterministically from $\mathbf{t}$ and π according to Compute-$\mathbf{z}$.

2.2.2 Model Overview

We propose a generative Bayesian model that explains how a corpus of D documents can be produced from a set of hidden variables. At a high level, the model first selects how frequently each topic is expressed in the document, and how the topics are ordered. These topics then determine the selection of words for each paragraph. Notation used in this and subsequent sections is summarized in Figure 2-1.

For each document d with N_d paragraphs, we separately generate a *bag of topics* $\mathbf{t}_d$ and a *topic ordering* π_d. The unordered bag of topics $\mathbf{t}_d$, which contains N_d elements, expresses how many paragraphs of the document are assigned to each of the K topics. Equivalently, $\mathbf{t}_d$ can be viewed as a vector of occurrence counts for each topic, with zero counts for topics that do not appear at all. Variable $\mathbf{t}_d$ is constructed by taking N_d samples from a distribution over topics $\boldsymbol{\theta}$, a multinomial representing the probability of each topic being expressed. Sharing $\boldsymbol{\theta}$ between documents captures the notion that certain topics are more likely across most documents in the corpus.

The topic ordering variable π_d is a permutation over the numbers 1 through K that defines the order in which topics appear in the document. We draw π_d from the *Generalized Mallows Model*, a distribution over permutations that we explain in Section 2.2.3. As we will see, this particular distribution biases the permutation selection to be close to a single centroid, reflecting the discourse constraint of preferring similar topic structures across documents.

Together, a document's bag of topics $\mathbf{t}_d$ and ordering π_d determine the topic assignment $z_{d,p}$ for each of its paragraphs. For example, in a corpus with $K = 4$, a seven-paragraph document d with $\mathbf{t}_d = \{1, 1, 1, 1, 2, 4, 4\}$ and $\pi_d = (2, 4, 3, 1)$ would induce the topic sequence $\mathbf{z}_d = (2, 4, 4, 1, 1, 1, 1)$. The induced topic sequence $\mathbf{z}_d$ can never assign the same topic to two unconnected portions of a document, thus satisfying the constraint of topic contiguity.

We assume that each topic k is associated with a language model $\boldsymbol{\beta}_k$. The words

[4] A nonparametric extension of this model would be to also learn K.

[5] In well structured documents, paragraphs tend to be internally topically consistent [64], so predicting one topic per paragraph is sufficient. However, we note that our approach can be applied with no modifications to other levels of textual granularity such as sentences.

of a paragraph assigned to topic k are then drawn from that topic's language model $\boldsymbol{\beta}_k$. This portion is similar to standard LDA in that each topic relates to its own language model. However, unlike LDA, our model enforces topic coherence for an entire paragraph rather than viewing a paragraph as a mixture of topics.

Before turning to a more formal discussion of the generative process, we first provide background on the permutation model for topic ordering.

2.2.3 The Generalized Mallows Model over Permutations

A central challenge of the approach we have presented is modeling the distribution over possible topic orderings. For this purpose we use the *Generalized Mallows Model* (GMM) [51, 77, 82, 95], which exhibits two appealing properties in the context of this task. First, the model concentrates probability mass on some *centroid ordering* and small perturbations (permutations) of that ordering. This characteristic matches our constraint that documents from the same domain exhibit structural similarity. Second, its parameter set scales linearly with the number of elements being ordered, making it sufficiently constrained and tractable for inference.

We first describe the standard *Mallows Model* over orderings [87]. The Mallows Model takes two parameters, a *centroid ordering* σ and a *dispersion parameter* ρ. It then sets the probability of any other ordering π to be proportional to $e^{-\rho d(\pi,\sigma)}$, where $d(\pi, \sigma)$ represents some *distance metric* between orderings π and σ. Frequently, this metric is the *Kendall τ distance*, the minimum number of swaps of adjacent elements needed to transform ordering π into the centroid ordering σ. Thus, orderings which are close to the centroid ordering will have high probability, while those in which many elements have been moved will have less probability mass.

The Generalized Mallows Model, first introduced by Fligner and Verducci [51], refines the standard Mallows Model by adding an additional set of dispersion parameters. These parameters break apart the distance $d(\pi, \sigma)$ between orderings into a set of independent components. Each component can then separately vary in its sensitivity to perturbation. To tease apart the distance function into components, the GMM distribution considers the *inversions* required to transform the centroid ordering into

an observed ordering. We first discuss how these inversions are parameterized in the GMM, then turn to the distribution's definition and characteristics.

Inversion Representation of Permutations

Typically, permutations are represented directly as an ordered sequence of elements — for example, $(3, 1, 2)$ represents permuting the initial order by placing the third element first, followed by the first element, and then the second. The GMM utilizes an alternative permutation representation defined by a vector $(v_1, \ldots, v_{K-1})$ of *inversion counts* with respect to the identity permutation $(1, \ldots, K)$. Term v_j counts the number of times when a value greater than j appears before j in the permutation. Note that the jth inversion count v_j can only take on integer values from 0 to $K - j$ inclusive. Thus the inversion count vector has only $K - 1$ elements, as v_K is always zero. For instance, given the standard form permutation $(3, 1, 5, 6, 2, 4)$, $v_2 = 3$ because 3, 5, and 6 appear before 2, and $v_3 = 0$ because no numbers appear before it; the entire inversion count vector would be $(1, 3, 0, 2, 0)$. Likewise, our previous example permutation $(2, 4, 3, 1)$ maps to inversion counts $(3, 0, 1)$. The sum of all components of an entire inversion count vector is simply that ordering's Kendall τ distance from the centroid ordering.

A significant appeal of the inversion representation is that every valid, distinct vector of inversion counts corresponds to a distinct permutation and vice versa. To see this, note that for each permutation we can straightforwardly compute its inversion counts. Conversely, given a sequence of inversion counts, we can construct the unique corresponding permutation. We insert items into the permutation, working backwards from item K. Assume that we have already placed items $j + 1$ through K in the proper order. To insert item j, we note that exactly v_j of items $j + 1$ to K must precede it, meaning that it must be inserted after position v_j in the current order (see the Compute-π algorithm in Figure 2-1). Since there is only one place where j can be inserted that fulfills the inversion counts, induction shows that exactly one permutation can be constructed to satisfy the given inversion counts.

In our model, we take the centroid topic ordering to always be the identity ordering

$(1, \ldots, K)$. Because the topic numbers in our task are completely symmetric and not linked to any extrinsic meaning, fixing the global ordering to a specific arbitrary value does not sacrifice any representational power. In the general case of the GMM, the centroid ordering is a parameter of the distribution.

Probability Mass Function

The GMM assigns probability mass to a particular order based on how that order is permuted from the centroid ordering. More precisely, it associates a distance with every permutation, where the centroid ordering has distance zero and permutations with many inversions with respect to this canonical ordering have larger distance. The distance assignment is based on $K-1$ real-valued dispersion parameters $(\rho_1, \ldots, \rho_{K-1})$. The distance of a permutation with inversion counts $\mathbf{v}$ is then defined to be $\sum_j \rho_j v_j$. The GMM's probability mass function is exponential in this distance:

$$\begin{aligned} \mathrm{GMM}(\mathbf{v}; \boldsymbol{\rho}) &= \frac{e^{-\sum_j \rho_j v_j}}{\psi(\boldsymbol{\rho})} \\ &= \prod_{j=1}^{K-1} \frac{e^{-\rho_j v_j}}{\psi_j(\rho_j)}, \end{aligned} \tag{2.1}$$

where $\psi(\boldsymbol{\rho}) = \prod_j \psi_j(\rho_j)$ is a normalization factor with value:

$$\psi_j(\rho_j) = \frac{1 - e^{-(K-j+1)\rho_j}}{1 - e^{-\rho_j}}. \tag{2.2}$$

Setting all ρ_j equal to a single value ρ recovers the standard Mallows Model with a Kendall τ distance function. The factorization of the GMM into independent probabilities per inversion count makes this distribution particularly easy to apply; we will use GMM_j to refer to the jth multiplicand of the probability mass function, which is the marginal distribution over v_j:

$$\mathrm{GMM}_j(v_j; \rho_j) = \frac{e^{-\rho_j v_j}}{\psi_j(\rho_j)}. \tag{2.3}$$

Due to the exponential form of the distribution, requiring that $\rho_j > 0$ constrains the GMM to assign highest probability mass to each v_j being zero, *i.e.*, the distributional mode is the canonical identity permutation. A higher value for ρ_j assigns more probability mass to v_j being close to zero, biasing j to have fewer inversions.

Conjugate Prior

A major benefit of the GMM is its membership in the exponential family of distributions; this means that it is particularly amenable to a Bayesian representation, as it admits a natural independent conjugate prior for each parameter ρ_j [93, 94]:

$$\mathrm{GMM}_0(\rho_j \mid v_{j,0}, \nu_0) \propto e^{(-\rho_j v_{j,0} - \log \psi_j(\rho_j))\nu_0}. \tag{2.4}$$

This prior distribution takes two parameters ν_0 and $v_{j,0}$. Intuitively, the prior states that over ν_0 previous trials, the total number of inversions observed was $\nu_0 v_{j,0}$. This distribution can be easily updated with the observed v_j to derive a posterior distribution.

Because each v_j has a different range, it is inconvenient to set the prior hyperparameters $v_{j,0}$ directly. In our work, we take the novel approach of assigning a common prior value for each parameter ρ_j, which we denote as ρ_0. Then we set each $v_{j,0}$ such that the maximum likelihood estimate of ρ_j is ρ_0. By differentiating the likelihood of the GMM with respect to ρ_j, it is straightforward to verify that this works out to setting:

$$v_{j,0} = \frac{1}{e^{\rho_0} - 1} - \frac{K - j + 1}{e^{(K-j+1)\rho_0} - 1}. \tag{2.5}$$

2.2.4 Formal Generative Process

We now fully specify the details of our content model, whose plate diagram appears in Figure 2-1. We observe a corpus of D documents, where each document d is an ordered sequence of N_d paragraphs and each paragraph is represented as a bag of words. The number of topics K is assumed to be pre-specified. The model induces a set of hidden variables that probabilistically explain how the words of the corpus were

produced. Our final desired output is the posterior distributions over the paragraphs' hidden topic assignment variables. In the following, variables subscripted with 0 are fixed prior hyperparameters.

1. For each topic k, draw a language model $\boldsymbol{\beta}_k \sim \text{Dirichlet}(\beta_0)$. As with LDA, these are topic-specific word distributions.
2. Draw a topic distribution $\boldsymbol{\theta} \sim \text{Dirichlet}(\theta_0)$, which expresses how likely each topic is to appear regardless of position.
3. Draw the topic ordering distribution parameters $\rho_j \sim \text{GMM}_0(\rho_0, \nu_0)$ for $j = 1$ to $K - 1$. These parameters control how rapidly probability mass decays for having more inversions for each topic. A separate ρ_j for every topic allows us to learn that some topics are more likely to be reordered than others.
4. For each document d with N_d paragraphs:
   (a) Draw a bag of topics $\mathbf{t}_d$ by sampling N_d times from Multinomial($\boldsymbol{\theta}$).
   (b) Draw a topic ordering π_d, by sampling a vector of inversion counts $\mathbf{v}_d \sim \text{GMM}(\boldsymbol{\rho})$, and then applying algorithm Compute-π from Figure 2-1 to $\mathbf{v}_d$.
   (c) Compute the vector of topic assignments $\mathbf{z}_d$ for document d's paragraphs by sorting $\mathbf{t}_d$ according to π_d, as in algorithm Compute-$\mathbf{z}$ from Figure 2-1.[6]
   (d) For each paragraph p in document d:
      i. Sample each word w in p according to the language model of p: $w \sim \text{Multinomial}(\boldsymbol{\beta}_{z_{d,p}})$.

2.2.5 Properties of the Model

In this section we describe the rationale behind using the GMM to represent the ordering component of our content model.

[6]Multiple permutations can contribute to the probability of a single document's topic assignments $\mathbf{z}_d$, if there are topics that do not appear in $\mathbf{t}_d$. As a result, our current formulation is biased toward assignments with fewer topics per document. In practice, we do not find this to negatively impact model performance.

- **Representational Power** The GMM concentrates probability mass around one centroid permutation, reflecting our preferred bias toward document structures with similar topic orderings. Furthermore, the parameterization of the GMM using a vector of dispersion parameters $\boldsymbol{\rho}$ allows for flexibility in how strongly the model biases toward a single ordering — at one extreme ($\boldsymbol{\rho} = \infty$) only one ordering has nonzero probability, while at the other ($\boldsymbol{\rho} = 0$) all orderings are equally likely. Because $\boldsymbol{\rho}$ is comprised of independent dispersion parameters $(\rho_1, \ldots, \rho_{K-1})$, the distribution can assign different penalties for displacing different topics. For example, we may learn that middle sections (in the case of Cities, sections such as Economy and Culture) are more likely to vary in position across documents than early sections (such as Introduction and History).

- **Computational Benefits** The parameterization of the GMM using a vector of dispersion parameters $\boldsymbol{\rho}$ is compact and tractable. Since the number of parameters grows linearly with the number of topics, the model can efficiently handle longer documents with greater diversity of content.

  Another computational advantage of this model is its seamless integration into a larger Bayesian model. Due to its membership in the exponential family and the existence of its conjugate prior, inference does not become significantly more complex when the GMM is used in a hierarchical context. In our case, the entire document generative model also accounts for topic frequency and the words within each topic.

  One final beneficial effect of the GMM is that it breaks the symmetry of topic assignments by fixing the distribution centroid. Specifically, topic assignments are not invariant to relabeling, because the probability of the underlying permutation would change. In contrast, many topic models assign the same probability to any relabeling of the topic assignments. Our model thus sidesteps the problem of topic *identifiability*, the issue where a model may have multiple maxima with the same likelihood due to the underlying symmetry of the hidden

variables. Non-identifiable models such as standard LDA may cause sampling procedures to jump between maxima or produce draws that are difficult to aggregate across runs.

Finally, we will show in Section 2.5 that the benefits of the GMM extend from the theoretical to the empirical: representing permutations using the GMM almost always leads to superior performance compared to alternative approaches.

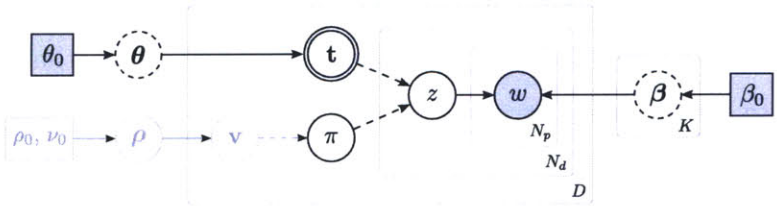
2.3 Inference via Collapsed Gibbs Sampling

The variables that we aim to infer are the paragraph topic assignments $\mathbf{z}$, which are determined by the bag of topics $\mathbf{t}$ and ordering π for each document. Thus, our goal is to estimate the joint marginal distributions of $\mathbf{t}$ and π given the document text while integrating out all remaining hidden parameters:

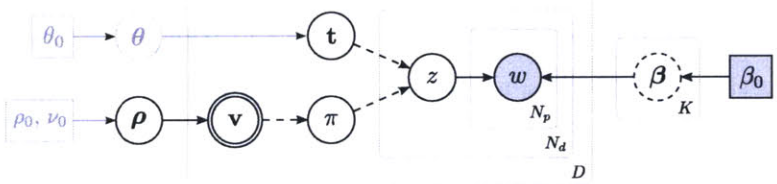
$$p(\mathbf{t}, \pi, \mid \mathbf{w}). \tag{2.6}$$

We accomplish this inference task through Gibbs sampling [19, 54]. A Gibbs sampler builds a Markov chain over the hidden variable state space whose stationary distribution is the actual posterior of the joint distribution. Each new sample is drawn from the distribution of a single variable conditioned on previous samples of the other variables. We can "collapse" the sampler by integrating over some of the hidden variables in the model, in effect reducing the state space of the Markov chain. Collapsed sampling has been previously demonstrated to be effective for LDA and its variants [59, 106, 125]. It is typically preferred over explicit Gibbs sampling of all hidden variables because of the smaller search space and generally shorter mixing time.

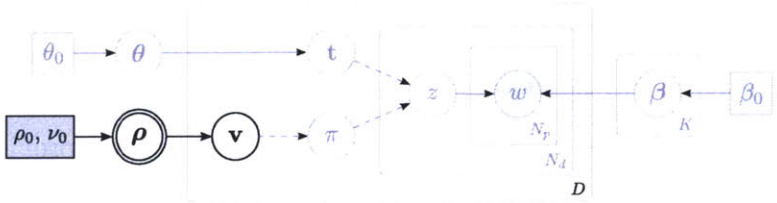
Our sampler analytically integrates out all but three sets of hidden variables: bags of topics $\mathbf{t}$, orderings π, and permutation inversion parameters $\boldsymbol{\rho}$. After a burn-in period, we treat the last samples of $\mathbf{t}$ and π as a draw from the posterior. When samples of the marginalized variables $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ are necessary, they can be estimated



$$
\begin{aligned}
p(t_{d,i} = t \mid \ldots) &\propto p(t_{d,i} = t \mid \mathbf{t}_{-(d,i)}, \theta_0)\; p(\mathbf{w}_d \mid \mathbf{t}_d, \pi_d, \mathbf{w}_{-d}, \mathbf{z}_{-d}, \beta_0) \\
&\propto \left[\frac{N(\mathbf{t}_{-(d,i)}, t) + \theta_0}{|\mathbf{t}_{-(d,i)}| + K\theta_0} \right] p(\mathbf{w}_d \mid \mathbf{z}, \mathbf{w}_{-d}, \beta_0),
\end{aligned}
$$



$$
\begin{aligned}
p(v_{d,j} = v \mid \ldots) &\propto p(v_{d,j} = v \mid \rho_j)\; p(\mathbf{w}_d \mid \mathbf{t}_d, \pi_d, \mathbf{w}_{-d}, \mathbf{z}_{-d}, \beta_0) \\
&= \mathrm{GMM}_j(v; \rho_j)\; p(\mathbf{w}_d \mid \mathbf{z}, \mathbf{w}_{-d}, \beta_0),
\end{aligned}
$$



$$
p(\rho_j \mid \ldots) = \mathrm{GMM}_0\left(\rho_j;\ \frac{\sum_d v_{d,j} + v_{j,0}\nu_0}{N + \nu_0},\ N + \nu_0\right),
$$

Figure 2-2: The collapsed Gibbs sampling inference procedure for estimating our content model's posterior distribution. In each plate diagram, the variable being resampled is shown in a double circle and its Markov blanket is highlighted in black; other variables, which have no impact on the variable being resampled, are grayed out. Variables $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, shown in dotted circles, are never explicitly depended on or re-estimated, because they are marginalized out by the sampler. Each diagram is accompanied by the conditional resampling distribution for its respective variable.

based on the topic assignments as we show in Section 2.4.3. Figure 2-2 summarizes the Gibbs sampling steps of our inference procedure.

Document Probability

As a preliminary step, consider how to calculate the probability of a single document's words $\mathbf{w}_d$ given the document's paragraph topic assignments $\mathbf{z}_d$ and the remaining documents and their topic assignments. Note that this probability is decomposable into a product of probabilities over individual paragraphs where paragraphs with different topics have conditionally independent word probabilities. Let $\mathbf{w}_{-d}$ and $\mathbf{z}_{-d}$ indicate the words and topic assignments to documents other than d, and W be the vocabulary size. The probability of the words in d is then:

$$\begin{aligned} p(\mathbf{w}_d \mid \mathbf{z}, \mathbf{w}_{-d}, \beta_0) &= \prod_{k=1}^{K} \int_{\boldsymbol{\beta}_k} p(\mathbf{w}_d \mid \mathbf{z}_d, \boldsymbol{\beta}_k)\; p(\boldsymbol{\beta}_k \mid \mathbf{z}, \mathbf{w}_{-d}, \beta_0)\; \mathrm{d}\boldsymbol{\beta}_k \\ &= \prod_{k=1}^{K} \mathrm{DCM}(\{\mathbf{w}_{d,i} : z_{d,i} = k\} \mid \{\mathbf{w}_{-d,i} : z_{-d,i} = k\}, \beta_0), \end{aligned} \tag{2.7}$$

where DCM($\cdot$) refers to the *Dirichlet compound multinomial* distribution, the result of integrating over multinomial parameters with a Dirichlet prior [17]. For a Dirichlet prior with parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_W)$, the DCM assigns the following probability to a series of observations $\mathbf{x} = \{x_1, \ldots, x_n\}$:

$$\mathrm{DCM}(\mathbf{x}; \boldsymbol{\alpha}) = \frac{\Gamma(\sum_j \alpha_j)}{\prod_j \Gamma(\alpha_j)} \prod_{i=1}^{W} \frac{\Gamma(N(\mathbf{x}, i) + \alpha_i)}{\Gamma(|\mathbf{x}| + \sum_j \alpha_j)}, \tag{2.8}$$

where $N(\mathbf{x}, i)$ refers to the number of times word i appears in $\mathbf{x}$. Here, $\Gamma(\cdot)$ is the Gamma function, a generalization of the factorial for real numbers. Some algebra shows that the DCM's posterior probability density function conditioned on a series of observations $\mathbf{y} = \{y_1, \ldots, y_n\}$ can be computed by updating each α_i with counts of how often word i appears in $\mathbf{y}$:

$$\mathrm{DCM}(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\alpha}) = \mathrm{DCM}(\mathbf{x}; \alpha_1 + N(\mathbf{y}, 1), \ldots, \alpha_W + N(\mathbf{y}, W)). \tag{2.9}$$

Equations 2.7 and 2.9 will be used to compute the conditional distributions of the hidden variables. We now turn to how each individual random variable is resampled.

Bag of Topics

First we consider how to resample $t_{d,i}$, the ith topic draw for document d conditioned on all other parameters being fixed (note this is *not* the topic of the ith paragraph, as we reorder topics using π_d, which is generated separately):

$$\begin{aligned} p(t_{d,i} = t \mid \ldots) &\propto p(t_{d,i} = t \mid \mathbf{t}_{-(d,i)}, \theta_0)\; p(\mathbf{w}_d \mid \mathbf{t}_d, \pi_d, \mathbf{w}_{-d}, \mathbf{z}_{-d}, \beta_0) \\ &\propto \left[\frac{N(\mathbf{t}_{-(d,i)}, t) + \theta_0}{|\mathbf{t}_{-(d,i)}| + K\theta_0} \right] p(\mathbf{w}_d \mid \mathbf{z}, \mathbf{w}_{-d}, \beta_0), \end{aligned} \tag{2.10}$$

where $\mathbf{t}_d$ is updated to reflect $t_{d,i} = t$, and $\mathbf{z}_d$ is deterministically computed in the last step using Compute-$\mathbf{z}$ from Figure 2-1 with inputs $\mathbf{t}_d$ and π_d. The first step reflects an application of Bayes rule to factor out the term for $\mathbf{w}_d$; we then drop superfluous terms from the conditioning. In the second step, the former term arises out of the DCM, by updating the parameters θ_0 with observations $\mathbf{t}_{-(d,i)}$ as in Equation 2.9 and dropping constants. The latter document probability term is computed using Equation 2.7. The new $t_{d,i}$ is selected by sampling from this probability computed over all possible topic assignments.

Ordering

The parameterization of a permutation π_d as a series of inversion values $v_{d,j}$ reveals a natural way to decompose the search space for Gibbs sampling. For each document d, we resample $v_{d,j}$ for $j = 1$ to $K - 1$ independently and successively according to its conditional distribution:

$$\begin{aligned} p(v_{d,j} = v \mid \ldots) &\propto p(v_{d,j} = v \mid \rho_j)\; p(\mathbf{w}_d \mid \mathbf{t}_d, \pi_d, \mathbf{w}_{-d}, \mathbf{z}_{-d}, \beta_0) \\ &= \mathrm{GMM}_j(v; \rho_j)\; p(\mathbf{w}_d \mid \mathbf{z}, \mathbf{w}_{-d}, \beta_0), \end{aligned} \tag{2.11}$$

where π_d is updated to reflect $v_{d,j} = v$, and $\mathbf{z}_d$ is computed deterministically according to $\mathbf{t}_d$ and π_d. The first term refers to Equation 2.3; the second is computed using Equation 2.7. This probability is computed for every possible value of v, which ranges from 0 to $K - j$, and term $v_{d,j}$ is sampled according to the resulting probabilities.

GMM Parameters

For each $j = 1$ to $K - 1$, we resample ρ_j from its posterior distribution:

$$p(\rho_j \mid \ldots) = \mathrm{GMM}_0 \left(\rho_j;\ \frac{\sum_d v_{d,j} + v_{j,0}\nu_0}{N + \nu_0},\ N + \nu_0 \right), \tag{2.12}$$

where GMM_0 is evaluated according to Equation 2.4. The normalization constant of this distribution is unknown, meaning that we cannot directly compute and invert the cumulative distribution function to sample from this distribution. However, the distribution itself is univariate and unimodal, so we can expect that an MCMC technique such as *slice sampling* [97] should perform well. In practice, MATLAB's built-in slice sampler provides a robust draw from this distribution.[7]

Computational Issues

During inference, directly computing document probabilities on the basis of Equation 2.7 results in many redundant calculations that slow the runtime of each iteration considerably. To improve the computational performance of our proposed inference procedure, we apply some memoization techniques during sampling. Within a single iteration, for each document, the Gibbs sampler requires computing the document's probability given its topic assignments (Equation 2.7) many times, but each computation frequently conditions on only slight variations of those topic assignments. A naïve approach would compute a probability for every paragraph each time a document probability is desired, performing redundant calculations when topic assignment sequences with shared subsequences are repeatedly considered.

Instead, we use lazy evaluation to build a three-dimensional cache, indexed by

[7]In particular, we use the `slicesample` function from the MATLAB Statistics Toolbox.

tuple (i, j, k), as follows. Each time a document probability is requested, it is broken into *independent* subspans of paragraphs, where each subspan takes on one contiguous topic assignment. This is possible due to the way Equation 2.7 factorizes into independent per-topic multiplicands. For a subspan starting at paragraph i, ending at paragraph j, and assigned topic k, the cache is consulted using key (i, j, k). For example, topic assignments $\mathbf{z}_d = (2, 4, 4, 1, 1, 1, 1)$ would result in cache lookups at $(1, 1, 2)$, $(2, 3, 4)$, and $(4, 7, 1)$. If a cached value is unavailable, the correct probability is computed using Equation 2.7 and the result is stored in the cache at location (i, j, k). Moreover, we also record values at every intermediate cache location (i, l, k) for $l = i$ to $j - 1$, because these values are computed as subproblems while evaluating Equation 2.7 for (i, j, k). The cache is reset before proceeding to the next document since the conditioning changes between documents. For each document, this caching guarantees that there are at most $O(N_d^2 K)$ paragraph probability calculations. In practice, because most individual Gibbs steps are small, this bound is very loose and the caching mechanism reduces computation time by several orders of magnitude.

We also maintain caches of word-topic and paragraph-topic assignment frequencies, allowing us to rapidly compute the counts used in equations 2.7 and 2.10. This form of caching is also used by Griffiths and Steyvers [59].

2.4 Applications

In this section, we describe how our model can be applied to three challenging discourse-level tasks: *aligning* paragraphs of similar topical content between documents, *segmenting* each document into topically cohesive sections, and *ordering* new unseen paragraphs into a coherent document. In particular, we show that the posterior samples produced by our inference procedure from Section 2.3 can be used to derive a solution for each of these tasks.

2.4.1 Alignment

For the alignment task we wish to find how the paragraphs of each document topically relate to paragraphs of other documents. Essentially, this is a cross-document clustering task – an alignment assigns each paragraph of a document into one of K topically related groupings. For instance, given a set of cell phone reviews, one group may represent text fragments that discuss Price, while another group consists of fragments about Reception.

Our model can be readily employed for this task: we can view the topic assignment for each paragraph z as a cluster label. For example, for two documents d_1 and d_2 with topic assignments $\mathbf{z}_{d_1} = (2, 4, 4, 1, 1, 1, 1)$ and $\mathbf{z}_{d_2} = (4, 4, 3, 3, 2, 2, 2)$, paragraph 1 of d_1 is grouped together with paragraphs 5 through 7 of d_2, and paragraphs 2 and 3 of d_1 with 1 and 2 of d_2. The remaining paragraphs assigned to topics 1 and 3 form their own separate per-document clusters.

Previously developed methods for cross-document alignment have been primarily driven by similarity functions that quantify lexical overlap between textual units [8, 98]. These methods do not explicitly model document structure, but they specify some global constraints that guide the search for an optimal alignment. Pairs of textual units are considered in isolation for making alignment decisions. In contrast, our approach allows us to take advantage of global structure and shared language models across all related textual units without requiring manual specification of matching constraints.

2.4.2 Segmentation

Segmentation is a well-studied discourse task where the goal is to divide a document into topically cohesive contiguous sections. Previous approaches have typically relied on *lexical cohesion* — that is, similarity in word choices within a document subspan — to guide the choice of segmentation boundaries [22, 44, 52, 66, 86, 129, 108, 128]. Our model relies on this same notion in determining the language models of topics, but connecting topics across documents and constraining how those topics appear

allow it to better learn the words that are most indicative of topic cohesion.

The output samples from our model's inference procedure map straightforwardly to segmentations — contiguous spans of paragraphs that are assigned the same topic number are taken to be one segment. For example, a seven-paragraph document d with topic assignments $\mathbf{z}_d = (2, 4, 4, 1, 1, 1, 1)$ would be segmented into three sections, comprised of paragraph 1, paragraphs 2 and 3, and paragraphs 4 through 7. Note that the segmentation ignores the specific values used for topic assignments, and only heeds the paragraph boundaries at which topic assignments change.

2.4.3 Ordering

A third application of our model is to the problem of creating structured documents from collections of unordered text segments. This text ordering task is an important step in broader NLP tasks such as text summarization and generation. For this task, we assume we are provided with well structured documents from a single domain as training examples; once trained, the model is used to induce an ordering of previously unseen collections of paragraphs from the same domain.

During training, our model learns a canonical ordering of topics for documents within the collection, via the language models associated with each topic. Because the GMM concentrates probability mass around the canonical $(1, \ldots, K)$ topic ordering, we expect that highly probable words in the language models of *lower*-numbered topics tend to appear *early* in a document, whereas highly probable words in the language models of *higher*-numbered topics tend to appear *late* in a document. Thus, we structure new documents according to this intuition — paragraphs with words tied to low topic numbers should be placed earlier than paragraphs with words relating to high topic numbers.

Formally, given an unseen document d comprised of an unordered set of paragraphs $\{p_1, \ldots, p_n\}$, we order paragraphs according to the following procedure. First, we find the most probable topic assignment $\hat{z}_i$ *independently* for each paragraph p_i, according

to parameters $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ learned during the training phase:

$$\hat{z}_i = \arg\max_k p(z_i = k \mid p_i, \boldsymbol{\beta}, \boldsymbol{\theta}) = \arg\max_k p(p_i \mid z_i = k, \boldsymbol{\beta}_k) p(z_i = k \mid \boldsymbol{\theta}). \tag{2.13}$$

Second, we sort the paragraphs by topic assignment $\hat{z}_i$ in ascending order — since $(1 \ldots K)$ is the GMM's canonical ordering, this yields the most likely ordering conditioned on a single estimated topic assignment for each paragraph. Due to possible ties in topic assignments, the resulting document may be a partial ordering; if a full ordering is required, ties are broken arbitrarily.

A key advantage of this proposed approach is that it is closed-form and computationally efficient. Though the training phase requires running the inference procedure of Section 2.3, once the model parameters are learned, predicting an ordering for a new set of p paragraphs requires computing only pK probability scores. In contrast, previous approaches have only been able to *rank* a small subset of all possible document reorderings [10], or performed a search procedure through the space of orderings to find an optimum [46].[8]

The objective function of Equation 2.13 depends on posterior estimates of $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ given the training documents. Since our collapsed Gibbs sampler integrates out these two hidden variables, we need to back out the values of $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ from the known posterior samples of $\mathbf{z}$. This can easily be done by computing the posterior expectation of each distribution based on the word-topic and topic-document assignment frequencies, respectively, as is done by Griffiths and Steyvers [59]. The probability mass $\hat{\beta}_k^w$ of word w in the language model of topic k is given by:

$$\hat{\beta}_k^w = \frac{N_\beta(k, w) + \beta_0}{N_\beta(k) + W\beta_0}, \tag{2.14}$$

where $N_\beta(k, w)$ the total number of times word w was assigned to topic k, and $N_\beta(k)$

[8]The approach we describe is not the same as finding the most probable paragraph ordering according to data likelihood, which is how the optimal ordering is derived for the HMM-based content model. Our proposed ordering technique essentially approximates that objective by using a *per-paragraph* maximum a posteriori estimate of the topic assignments rather than the full posterior topic assignment distribution. This approximation makes for a much faster prediction algorithm that performs well empirically.

Articles about large cities from Wikipedia

Corpus	Language	Documents	Sections	Paragraphs	Vocabulary	Tokens
CitiesEn	English	100	13.2	66.7	42,000	4,920
CitiesEn500	English	500	10.5	45.9	95,400	3,150
CitiesFr	French	100	10.4	40.7	31,000	2,630

Articles about chemical elements from Wikipedia

Corpus	Language	Documents	Sections	Paragraphs	Vocabulary	Tokens
Elements	English	118	7.7	28.1	18,000	1,920

Cell phone reviews from PhoneArena.com

Corpus	Language	Documents	Sections	Paragraphs	Vocabulary	Tokens
Phones	English	100	6.6	24.0	13,500	2,750

Table 2.1: Statistics of the datasets used to evaluate our content model. All values except vocabulary size and document count are per-document averages.

is the total number of words assigned to topic k, according to the posterior sample of $\mathbf{z}$. We can derive a similar estimate for $\hat{\theta}_k$, the prior likelihood of topic k:

$$\hat{\theta}_k = \frac{N_\theta(k) + \theta_0}{N_\theta + K\theta_0}, \tag{2.15}$$

where $N_\theta(k)$ is the total number of paragraphs assigned to topic k according to the sample of $\mathbf{z}$, and N_θ is the total number of paragraphs in the entire corpus.

2.5 Experiments

In this section, we evaluate the performance of our model on the three tasks presented in Section 2.4: cross-document alignment, document segmentation, and information ordering. We first describe some preliminaries common to all three tasks, covering the datasets, reference comparison structures, model variants, and inference algorithm settings shared by each evaluation. We then provide a detailed examination of how our model performs on each individual task.

2.5.1 General Evaluation Setup

Datasets

In our experiments we use five datasets, briefly described below (for additional statistics, see Table 2.1):

- *CitiesEn*: Articles from the English Wikipedia about the world's 100 largest cities by population. Common topics include History, Culture, and Demographics. These articles are typically of substantial size and share similar content organization patterns.

- *CitiesEn500*: Articles from the English Wikipedia about the world's 500 largest cities by population. This collection is a superset of *CitiesEn*. Many of the lower-ranked cities are not well known to English Wikipedia editors — thus, compared to *CitiesEn* these articles are shorter on average and exhibit greater variability in content selection and ordering.

- *CitiesFr*: Articles from the French Wikipedia about the same 100 cities as in *CitiesEn*.

- *Elements*: Articles from the English Wikipedia about chemical elements in the periodic table,[9] including topics such as Biological Role, Occurrence, and Isotopes.

- *Phones*: Reviews extracted from PhoneArena.com, a popular cell phone review website. Topics in this corpus include Design, Camera, and Interface. These reviews are written by expert reviewers employed by the site, as opposed to lay users.[10]

This heterogeneous collection of datasets allows us to examine the behavior of the model under diverse test conditions. These sets vary in how the articles were gener-

[9] All 118 elements at http://en.wikipedia.org/wiki/Periodic_table, including undiscovered element 117.

[10] In the *Phones* set, 35 documents are very short "express" reviews without section headings; we include them in the input to the model, but did not evaluate on them.

ated, the language in which the articles were written, and the subjects they discuss. As a result, patterns in topic organization vary greatly across domains. For instance, within the *Phones* corpus, the articles are very formulaic, due to the centralized editorial control of the website, which establishes consistent standards followed by the expert reviewers. On the other hand, Wikipedia articles exhibit broader structural variability due to the collaborative nature of Wikipedia editing, which allows articles to evolve independently. While Wikipedia articles within the same category often exhibit similar section orderings, many have idiosyncratic inversions. For instance, in the *CitiesEn* corpus, both the Geography and History sections typically occur toward the beginning of a document, but History can appear either before or after Geography across different documents.

Each corpus we consider has been manually divided into sections by their authors, including a short textual *heading* for each section. In Sections 2.5.2 and 2.5.3, we discuss how these author-created sections with headings are used to generate reference annotations for the alignment and segmentation tasks. Note that we only use the headings for evaluation; none of the heading information is provided to any of the methods under consideration. For the tasks of alignment and segmentation, evaluation is performed on the datasets presented in Table 2.1. For the ordering task, however, this data is used for training, and evaluation is performed using a separate held-out set of documents. The details of this held-out dataset are given in Section 2.5.4.

Model Variants

For each evaluation, besides comparing to baselines from the literature, we also consider two variants of our proposed model. In particular, we investigate the impact of the Mallows component of the model by alternately relaxing and tightening the way it constrains topic orderings:

- *Constrained*: In this variant, we require all documents to follow the exact same canonical ordering of topics. That is, no topic permutation inversions are allowed, though documents may skip topics as before. This case can be viewed

as a special case of the general model, where the Mallows inversion prior ρ_0 approaches infinity. From an implementation standpoint, we simply fix all inversion counts $\mathbf{v}$ to zero during inference.[11]

- *Uniform*: This variant assumes a uniform distribution over all topic permutations, instead of biasing toward a small related set. Again, this is a special case of the full model, with inversion prior ρ_0 set to zero, and the strength of that prior ν_0 approaching infinity, thus forcing each item of ρ to always be zero.

Note that both of these variants still enforce the long-range constraint of topic contiguity, and vary from the full model only in how they capture topic ordering similarity.

Evaluation Procedure and Parameter Settings

For each evaluation of our model and its variants, we run the collapsed Gibbs sampler from five random seed states, and take the 10,000th iteration of each chain as a sample. Results presented are the average over these five samples.

Dirichlet prior hyperparameters for the bag of topics θ_0 and language models β_0 are set to 0.1. For the GMM, we set the prior dispersion hyperparameter ρ_0 to 1, and the effective sample size prior ν_0 to be 0.1 times the number of documents. These values are minimally tuned, and similar results are achieved for alternative settings of θ_0 and β_0. Parameters ρ_0 and ν_0 control the strength of the bias toward structural regularity, trading off between the *Constrained* and *Uniform* model variants. The values we have chosen are a middle ground between those two extremes.

Our model also takes a parameter K that controls the upper bound on the number of latent topics. Note that our algorithm can select fewer than K topics for each document, so K does not determine the number of segments in each document. In

[11] At first glance, the *Constrained* model variant appears to be equivalent to an HMM where each state i can transition to either i or $i+1$. However, this is not the case — some topics may appear zero times in a document, resulting in multiple possible transitions from each state. Furthermore, the transition probabilities would be dependent on position within the document — for example, at earlier absolute positions within a document, transitions to high-index topics are unlikely, because that would require all subsequent paragraphs to have a high-index topic.

general, a higher K results in a finer-grained division of each document into different topics, which may result in more precise topics, but may also split topics that should be together. We report results in each evaluation using both $K = 10$ and 20.

2.5.2 Alignment

We first evaluate the model on the task of cross-document alignment, where the goal is to group textual units from different documents into topically cohesive clusters. For instance, in the Cities-related domains, one such cluster may include Transportation-related paragraphs. Before turning to the results we first present details of the specific evaluation setup targeted to this task.

Alignment Evaluation Setup

Reference Annotations To generate a sufficient amount of reference data for evaluating alignments we use section headings provided by the authors. We assume that two paragraphs are aligned if and only if their section headings are identical. These headings constitute noisy annotations in the Wikipedia datasets: the same topical content may be labeled with different section headings in different articles (*e.g.*, for *CitiesEn*, "Places of interest" in one article and "Landmarks" in another), so we call this reference structure the *noisy headings* set.

It is not clear *a priori* what effect this noise in the section headings may have on evaluation accuracy. To empirically estimate this effect, we also use some manually annotated alignments in our experiments. Specifically, for the *CitiesEn* corpus, we manually annotated each article's paragraphs with a consistent set of section headings, providing us an additional reference structure to evaluate against. In this *clean headings* set, we found approximately 18 topics that were expressed in more than one document.

Metrics To quantify our alignment output we compute a *recall* and *precision* score of a candidate alignment against a reference alignment. Recall measures, for each unique section heading in the reference, the maximum number of paragraphs with

that heading that are assigned to one *particular* topic.[12] The final score is computed by summing over each section heading and dividing by the total number of paragraphs. High recall indicates that paragraphs of the same section headings are generally being assigned to the same topic.

Conversely, precision measures, for each topic number, the maximum number of paragraphs with that topic assignment that share the *same* section heading. Precision is summed over each topic and normalized by the total number of paragraphs. High precision means that paragraphs assigned to a single topic usually correspond to the same section heading.

Recall and precision trade off against each other — more finely grained topics will tend to improve precision at the cost of recall. At the extremes, perfect recall occurs when every paragraph is assigned the same topic, and perfect precision when each paragraph is its own topic.

We also present one summary *F-score* in our results, which is the harmonic mean of recall and precision.

Statistical significance in this setup is measured with *approximate randomization* [100], a nonparametric test that can be directly applied to nonlinearly computed metrics such as F-score. This test has been used in prior evaluations for information extraction and machine translation [37, 114].

Baselines For this task, we compare against two baselines:

- *Hidden Topic Markov Model* (HTMM) [63]: As explained in Section 2.1, this model represents topic change between adjacent textual units in a Markovian fashion. HTMM can only capture local constraints, so it would allow topics to recur non-contiguously throughout a document. We use the publicly available implementation,[13] with priors set according to the recommendations made in the original work.

[12]This greedy mapping procedure is akin to the many-to-one evaluation used for unsupervised part-of-speech induction [71].

[13]http://code.google.com/p/openhtmm/

		CitiesEn Clean headings			*CitiesEn* Noisy headings			*CitiesEn500* Noisy headings		
		Recall	Prec	F-score	Recall	Prec	F-score	Recall	Prec	F-score
$K = 10$	Clustering	0.578	0.439	$*$ 0.499	0.611	0.331	$*$ 0.429	0.609	0.329	$*$ 0.427
	HTMM	0.446	0.232	$*$ 0.305	0.480	0.183	$*$ 0.265	0.461	0.269	$*$ 0.340
	Constrained	0.579	0.471	$*$ 0.520	0.667	0.382	$*$ 0.485	0.643	0.385	$*$ 0.481
	Uniform	0.520	0.440	$*$ 0.477	0.599	0.343	$*$ 0.436	0.582	0.344	$*$ 0.432
	Our model	**0.639**	**0.509**	**0.566**	**0.705**	**0.399**	**0.510**	**0.722**	**0.426**	**0.536**
$K = 20$	Clustering	0.486	0.541	$*$ 0.512	0.527	0.414	$*$ 0.464	0.489	0.391	$*$ 0.435
	HTMM	0.260	0.217	$*$ 0.237	0.304	0.187	$*$ 0.232	0.351	0.234	$*$ 0.280
	Constrained	0.458	0.519	$*$ 0.486	0.553	0.415	$*$ 0.474	0.515	0.394	$*$ 0.446
	Uniform	0.499	0.551	$*$ 0.524	0.571	0.423	$*$ 0.486	0.557	0.422	$*$ 0.480
	Our model	**0.578**	**0.636**	**0.606**	**0.648**	**0.489**	**0.557**	**0.620**	**0.473**	**0.537**

		CitiesFr Noisy headings			*Elements* Noisy headings			*Phones* Noisy headings		
		Recall	Prec	F-score	Recall	Prec	F-score	Recall	Prec	F-score
$K = 10$	Clustering	0.588	0.283	$*$ 0.382	0.524	0.361	$*$ 0.428	0.599	0.456	$*$ 0.518
	HTMM	0.338	0.190	$*$ 0.244	0.430	0.190	$*$ 0.264	0.379	0.240	$*$ 0.294
	Constrained	0.652	0.356	0.460	0.603	0.408	$*$ 0.487	**0.745**	**0.506**	**0.602**
	Uniform	0.587	0.310	$*$ 0.406	0.591	0.403	$*$ 0.479	0.656	0.422	$*$ 0.513
	Our model	**0.657**	**0.360**	**0.464**	**0.685**	**0.460**	**0.551**	0.738	0.493	0.591
$K = 20$	Clustering	0.453	0.317	$*$ 0.373	0.477	0.402	$*$ 0.436	0.486	0.507	$*$ 0.496
	HTMM	0.253	0.195	$*$ 0.221	0.248	0.243	$*$ 0.246	0.274	0.229	$*$ 0.249
	Constrained	0.584	0.379	$*$ 0.459	0.510	0.421	$*$ 0.461	0.652	**0.576**	**0.611**
	Uniform	0.571	0.373	$*$ 0.451	0.550	0.479	$\diamond$ 0.512	0.608	0.471	$*$ 0.538
	Our model	**0.633**	**0.431**	**0.513**	**0.569**	**0.498**	**0.531**	**0.683**	0.546	0.607

Table 2.2: Comparison of the alignments produced by our content model and a series of baselines and model variations, for both 10 and 20 topics, evaluated against clean and noisy sets of section headings. Higher scores are better. Within the same K, the methods which our model significantly outperforms are indicated with $*$ for $p < 0.001$ and $\diamond$ for $p < 0.01$.

- *Clustering*: We use a repeated bisection algorithm to find a clustering of the paragraphs that maximizes the sum of the pairwise cosine similarities of the items in each cluster.[14] This clustering was implemented using the CLUTO toolkit.[15] Note that this approach is completely structure-agnostic, treating documents as bags of paragraphs rather than sequences of paragraphs. These types of clustering techniques have been shown to deliver competitive performance for cross-document alignment tasks [8].

Alignment Results

Table 2.2 presents the results of the alignment evaluation. On all of the datasets, the best performance is achieved by our model or its variants, by a statistically significant and usually substantial margin.

The comparative performance of the baseline methods is consistent across domains – surprisingly, clustering performs better than the more complex HTMM model. This observation is consistent with previous work on cross-document alignment and multi-document summarization, which use clustering as their main component [12, 109]. Despite the fact that HTMM captures some dependencies between adjacent paragraphs, it is not sufficiently constrained. Manual examination of the actual topic assignments reveals that HTMM often assigns the same topic for disconnected paragraphs within a document, violating the topic contiguity constraint.

In all but one domain the full GMM-based approach yields the best performance compared to its variants. The one exception is in the *Phone* domain. There the *Constrained* baseline achieves the best result for both K by a small margin. These results are to be expected, given the fact that this domain exhibits a highly rigid topic structure across all documents. A model that permits permutations of topic ordering, such as the GMM, is too flexible for such highly formulaic domains.

We also qualitatively examine the posterior estimates of the ρ distribution for the *CitiesEn* and *Elements* domains. We generally find across evaluations that ρ

[14]This particular clustering technique substantially outperforms the agglomerative and graph partitioning-based clustering approaches for our task.

[15]http://glaros.dtc.umn.edu/gkhome/views/cluto/

took on the highest values toward the first and last topics — for example, ρ_1 and $\rho_1 9$ (for $K = 20$) would typically exceed 1, whereas $\rho_1 0$ would be much smaller, *e.g.*, around 0.2. This implies that the model is learning lower variability toward the beginning and end of documents, which reflects reality: a document is likely to exhibit higher variability in its middle sections (for example, the Demographics, Economy, and Transportation sections of *Cities*) than the extremes (the Introduction and Sister Cities).

Finally, we observe that the evaluations based on manual and noisy annotations exhibit an almost entirely consistent ranking of the methods under consideration (see the clean and noisy headings results for *CitiesEn* in Table 2.2). This consistency indicates that the noisy headings are sufficient for gaining insight into the comparative performance of the different approaches.

2.5.3 Segmentation

Next we consider the task of text segmentation. We test whether the model is able to identify the boundaries of topically coherent text segments.

Segmentation Evaluation Setup

Reference Segmentations As described in Section 2.5.1, all of the datasets used in this evaluation have been manually divided into sections by their authors. These annotations are used to create reference segmentations for evaluating our model's output. Recall from Section 2.5.2 that we also built a clean reference structure for the *CitiesEn* set. That structure encodes a "clean" segmentation of each document because it adjusts the granularity of section headings to be consistent across documents. Thus, we also compare against the segmentation specified by the *CitiesEn* clean section headings.

Metrics Segmentation quality is evaluated using the standard penalty metrics P_k and WindowDiff [14, 103]. Both pass a sliding window over the documents and compute the probability of the words at the end of the windows being improperly

segmented with respect to each other. As with previous work, window size is set to half the average length of a segment in the reference segmentation. WindowDiff is stricter, and requires that the number of segmentation boundaries between the endpoints of the window be correct as well.[16]

Baselines We first compare to BayesSeg [44],[17] a Bayesian segmentation approach that is the current state-of-the-art for this task. Interestingly, our model reduces to their approach when every document is considered completely in isolation, with no topic sharing between documents. Connecting topics across documents makes for a much more difficult inference problem than the one tackled by Eisenstein and Barzilay. At the same time, their algorithm cannot capture structural relatedness across documents.

Since BayesSeg is designed to be operated with a specification of a number of segments, we provide this baseline with the benefit of knowing the correct number of segments for each document, which is not provided to our system. We run this baseline using the authors' publicly available implementation;[18] its priors are set using a built-in mechanism that automatically re-estimates hyperparameters.

We also compare our method with the algorithm of [128], which is commonly used as a point of reference in the evaluation of segmentation algorithms. This algorithm computes the optimal segmentation by estimating changes in the predicted language models of segments under different partitions. We used the publicly available implementation of the system,[19] which does not require parameter tuning on a held-out development set. In contrast to BayesSeg, this algorithm has a mechanism for predicting the number of segments, but can also take a pre-specified number of segments. In our comparison, we consider both versions of the algorithm – U&I denotes the case when the correct number of segments is provided to the model and

[16]Statistical significance testing is not standardized and usually not reported for the segmentation task, so we omit these tests in our results.

[17]We do not evaluate on the corpora used in their work, since our model relies on content similarity across documents in the corpus.

[18]http://groups.csail.mit.edu/rbg/code/bayesseg/

[19]http://www2.nict.go.jp/x/x161/members/mutiyama/software.html#textseg

$\overline{\text{U\&I}}$ denotes when the model estimates the optimal number of segments.

Segmentation Results

Table 2.3 presents the segmentation experiment results. On every dataset our model outperforms the BayesSeg and U&I baselines by a substantial margin regardless of K. This result provides strong evidence that learning connected topic models over in-domain documents leads to improved segmentation performance.

The best performance is generally obtained by the full version of our model, with three exceptions. In two cases (*CitiesEn* with $K = 10$ using clean headings on the WindowDiff metric, and *CitiesFr* with $K = 10$ on the P_k metric), the variant that performs better than the full model only does so by a minute margin. Furthermore, in both of those instances, the corresponding evaluation with $K = 20$ using the full model leads to the best overall results for the respective domains.

The only case when a variant outperforms our full model by a notable margin is the *Phones* dataset. This result is not unexpected given the formulaic nature of this dataset as discussed earlier.

As expected, using more topics leads to more segments. Between our model variants, we observe that the number of segments found by the *Constrained* approach typically finds the least number of segments. This result is intuitive, since the constrained model requires one common ordering and thus is more likely to assign the same topic to two adjacent sections that are swapped with respect to other documents. The uniform model, in contrast, can still learn different orderings, and thus exhibits similar segment count across experiments. Interestingly, in the one domain where the orderings are in fact always the same (*Phones*), the *Constrained* model variant finds about the same number of segments as the full model.

2.5.4 Ordering

The final task on which we evaluate our model is that of finding a coherent ordering of a set of textual units. Unlike the previous tasks, where prediction is based on

		CitiesEn Clean headings			*CitiesEn* Noisy headings			*CitiesEn500* Noisy headings		
		P_k	WD	# Segs	P_k	WD	# Segs	P_k	WD	# Segs
BayesSeg		0.321	0.376	12.3	0.317	0.376	13.2	0.282	0.335	10.5
U&I		0.337	0.404	12.3	0.337	0.405	13.2	0.292	0.350	10.5
$\overline{\text{U\&I}}$		0.353	0.375	5.8	0.357	0.378	5.8	0.321	0.346	5.4
$K = 10$	Constrained	0.260	**0.281**	7.7	0.267	0.288	7.7	0.221	0.244	6.8
	Uniform	0.268	0.300	8.8	0.273	0.304	8.8	0.227	0.257	7.8
	Our model	**0.253**	0.283	9.0	**0.257**	**0.286**	9.0	**0.196**	**0.225**	8.1
$K = 20$	Constrained	0.274	0.314	10.9	0.274	0.313	10.9	0.226	0.261	9.1
	Uniform	0.234	0.294	14.0	0.234	0.290	14.0	0.203	0.256	12.3
	Our model	**0.221**	**0.278**	14.2	**0.222**	**0.278**	14.2	**0.196**	**0.247**	12.1

		CitiesFr Noisy headings			*Elements* Noisy headings			*Phones* Noisy headings		
		P_k	WD	# Segs	P_k	WD	# Segs	P_k	WD	# Segs
BayesSeg		0.274	0.332	10.4	0.279	0.316	7.7	0.392	0.457	9.6
U&I		0.282	0.336	10.4	0.248	0.286	7.7	0.412	0.463	9.6
$\overline{\text{U\&I}}$		0.321	0.342	4.4	0.294	0.312	4.8	0.423	0.435	4.7
$K = 10$	Constrained	0.230	0.244	6.4	0.227	0.244	5.4	0.312	**0.347**	8.0
	Uniform	**0.214**	**0.233**	7.3	0.226	0.250	6.6	0.332	0.367	7.5
	Our model	0.216	**0.233**	7.4	**0.201**	**0.226**	6.7	**0.309**	0.349	8.0
$K = 20$	Constrained	0.230	0.250	7.9	0.231	0.257	6.6	**0.295**	**0.348**	10.8
	Uniform	0.203	0.234	10.4	0.209	0.248	8.7	0.327	0.381	9.4
	Our model	**0.201**	**0.230**	10.8	**0.203**	**0.243**	8.6	0.302	0.357	10.4

Table 2.3: Comparison of the segmentations produced by our content model and a series of baselines and model variations, for both 10 and 20 topics, evaluated against clean and noisy sets of section headings. Lower scores are better. BayesSeg and U&I are given the true number of segments, so their segments counts reflect the reference structures' segmentations. In contrast, $\overline{\text{U\&I}}$ automatically predicts the number of segments.

Corpus	Set	Documents	Sections	Paragraphs	Vocabulary	Tokens
CitiesEn	Training	100	13.2	66.7	42,000	4,920
	Testing	65	11.2	50.3	42,000	3,460
CitiesFr	Training	100	10.4	40.7	31,000	2,630
	Testing	68	7.7	28.2	31,000	1,580
Phones	Training	100	6.6	24.0	13,500	2,750
	Testing	64	9.6	39.3	13,500	4,540

Table 2.4: Statistics of the training and test sets used for the content model ordering experiments. All values except vocabulary are the average per document. The training set statistics are reproduced from Table 2.1 for ease of reference.

hidden variable distributions, ordering is observed in a document. Moreover, the GMM model uses this information during the inference process. Therefore, we need to divide our datasets into training and test portions.

In the past, ordering algorithms have been applied to textual units of various granularities, most commonly sentences and paragraphs. Our ordering experiments operate at the level of a relatively larger unit — sections. We believe that this granularity is suitable to the nature of our model, because it captures patterns at the level of topic distributions rather than local discourse constraints. The ordering of sentences and paragraphs has been studied in the past [10, 73] and these two types of models can be effectively combined to induce a full ordering [46].

Ordering Evaluation Setup

Training and Test Datasets We use the *CitiesEn*, *CitiesFr* and *Phones* datasets as training documents for parameter estimation as described in Section 2.4. We introduce additional sets of documents from the same domains as test sets. Table 2.4 provides statistics on the training and test set splits (note that out-of-vocabulary terms in the test sets are discarded).[20]

Even though we perform ordering at the section level, these collections still pose a challenging ordering task: for example, the average number of sections in a *CitiesEn*

[20]The *Elements* dataset is limited to 118 articles, preventing us from splitting it into reasonably sized training and test sets. Therefore we do not consider it for our ordering experiments. For the Cities-related sets, the test documents are shorter because they were about cities of lesser population. On the other hand, for *Phones* the test set does not include short "express" reviews and thus exhibits higher average document length.

test document is 11.2, comparable to the 11.5 sentences (the unit of reordering) per document of the National Transportation Safety Board corpus used in previous work [11, 46].

Metrics We report the *Kendall's* τ rank correlation coefficient for our ordering experiments.[21] This metric measures how much an ordering differs from the reference order — the underlying assumption is that most reasonable sentence orderings should be fairly similar to it. Specifically, for a permutation π of the sections in an N-section document, $\tau(\pi)$ is computed as

$$\tau(\pi) = 1 - 2\frac{d(\pi, \sigma)}{\binom{N}{2}}, \tag{2.16}$$

where $d(\pi, \sigma)$ is, as before, the Kendall τ distance, the number of swaps of adjacent textual units necessary to rearrange π into the reference order. The metric ranges from -1 (inverse orders) to 1 (identical orders). Note that a *random* ordering will yield a zero score in expectation. This measure has been widely used for evaluating information ordering [11, 46, 79] and has been shown to correlate with human assessments of text quality [80].

Baselines and Model Variants Our ordering method is compared against the original HMM-based content modeling approach of Barzilay and Lee [11]. This baseline delivers state-of-the art performance in a number of datasets and is similar in spirit to our model — it also aims to capture patterns at the level of topic distribution (see Section 2.1). Again, we use the publicly available implementation[22] with parameters adjusted according to the values used in their previous work. This content modeling implementation provides an A* search procedure that we use to find the optimal permutation.

We do not include in our comparison local coherence models [10, 46]. These models are designed for sentence-level analysis — in particular, they use syntactic

[21] Observe that this is a renormalized version of the Kendall's τ distance mentioned earlier in this chapter.

[22] http://people.csail.mit.edu/regina/code.html

		CitiesEn	*CitiesFr*	*Phones*
HMM-based Content Model		0.245	0.305	0.256
$K = 10$	Constrained	**0.587**	**0.596**	0.676
	Our model	0.571	0.541	**0.678**
$K = 20$	Constrained	**0.583**	**0.575**	**0.711**
	Our model	0.575	0.571	0.678

Table 2.5: Comparison of the orderings produced by our content model and a series of baselines and model variations, for both 10 and 20 topics, evaluated on the respective test sets. Higher scores are better.

information and thus cannot be directly applied for section-level ordering. As we state above, these models are orthogonal to topic-based analysis; combining the two approaches is a promising direction for future work.

Note that the *Uniform* model variant is not applicable to this task, since it does not make any claims to a preferred underlying topic ordering. In fact, from a document likelihood perspective, for any proposed paragraph order the reverse order would have the same probability under the *Uniform* model. Thus, the only model variant we consider here is *Constrained*.

Ordering Results

Table 2.5 summarizes ordering results for the GMM- and HMM-based content models. Across all datasets, our model outperforms content modeling by a very large margin. For instance, on the *CitiesEn* dataset, the gap between the two models reaches 35%. This difference is expected. In previous work, content models were applied to short formulaic texts. In contrast, documents in our collection exhibit higher variability than the original collections. The HMM does not provide explicit constraints on generated global orderings. This may prevent it from effectively learning non-local patterns in topic organization.

We also observe that the *Constrained* variant outperforms our full model. While the difference between the two is small, it is fairly consistent across domains. Since it is not possible to predict idiosyncratic variations in the test documents' topic orderings, a more constrained model can better capture the prevalent ordering patterns that are consistent across the domain.

2.5.5 Discussion

Our experiments with the three separate tasks reveal some common trends in the results. First, we observe that our single unified model of document structure can be readily and successfully applied to multiple discourse-level tasks, whereas previous work has proposed separate approaches for each task. This versatility speaks to the power of our topic-driven representation of document structure. Second, within each task our model outperforms state-of-the-art baselines by substantial margins across a wide variety of evaluation scenarios. These results strongly support our hypothesis that augmenting topic models with discourse-level constraints broadens their applicability to discourse-level analysis tasks.

Looking at the performance of our model across different tasks, we make a few notes about the importance of the individual topic constraints. Topic contiguity is a consistently important constraint, allowing both of our model variants to outperform alternative baseline approaches. In most cases, introducing a bias toward similar topic ordering, without requiring identical orderings, provides further benefits when encoded in the model. Our more flexible models achieve superior performance in the segmentation and alignment tasks. In the case of ordering, however, this extra flexibility does not pay off, as the model distributes its probability mass away from strong ordering patterns likely to occur in unseen data.

We can also identify the properties of a dataset that most strongly affect the performance of our model. The *Constrained* model variant performs slightly better than our full model on rigidly formulaic domains, achieving highest performance on the *Phones* dataset. When we know *a priori* that a domain is formulaic in structure, it is worthwhile to choose the model variant that suitably enforces formulaic topic orderings. Fortunately, this adaptation can be achieved in the proposed framework using the prior of the Generalized Mallows Model — recall that the *Constrained* variant is a special case of the full model.

However, the performance of our model is invariant with respect to other dataset characteristics. Across the two languages we considered, the model and baselines

exhibit the same comparative performance for each task. Moreover, this consistency also holds between the general-interest cities articles and the highly technical chemical elements articles. Finally, between the smaller *CitiesEn* and larger *CitiesEn500* datasets, we observe that our results are consistent.

2.6 Conclusions and Future Work

In this chapter, we have shown how an unsupervised topic-based approach can capture content structure. Our resulting content model constrains topic assignments in a way that requires global modeling of entire topic sequences. We showed that the Generalized Mallows Model is a theoretically and empirically appealing way of capturing the ordering component of this topic sequence. Our results demonstrate the importance of augmenting statistical models of text analysis with structural constraints motivated by discourse theory. Furthermore, our success with the GMM suggests that it could potentially be applied to the modeling of ordering constraints in other NLP applications.

There are multiple avenues of future extensions to this work. First, our empirical results demonstrated that for certain domains providing too much flexibility in the model may in fact be detrimental to predictive accuracy. In those cases, a more tightly constrained variant of our model yields superior performance. An interesting extension of our current model would be to allow additional flexibility in the prior of the GMM by drawing it from another level of hyperpriors. From a technical perspective, this form of hyperparameter re-estimation would involve defining an appropriate hyperprior for the Generalized Mallows Model and adapting its estimation into our present inference procedure.

Additionally, there may be cases when the assumption of *one* canonical topic ordering for an entire corpus is too limiting, *e.g.,* if a dataset consists of topically related articles from multiple sources, each with its own editorial standards. Our model can be extended to allow for *multiple* canonical orderings by positing an additional level of hierarchy in the probabilistic model, *i.e.*, document structures can be generated from

a mixture of several Generalized Mallows Models, each with its own distributional mode. In this case, the model would take on the additional burden of learning how topics are permuted between these multiple canonical orderings. Such a change to the model would greatly complicate inference as re-estimating a Generalized Mallows Model canonical ordering is in general NP-hard. However, recent advances in statistics have produced efficient approximate algorithms with theoretically guaranteed correctness bounds [2] and exact methods that are tractable for typical cases [95].

More generally, the model presented in this chapter assumes two specific global constraints on content structure. While domains that satisfy these constraints are plentiful, there are domains where our modeling assumptions do not hold. For example, in dialogue it is well known that topics recur throughout a conversation [62], thereby violating our first constraint. Nevertheless, texts in such domains still follow certain organizational conventions, *e.g.* the stack structure for dialogue. Our results suggest that explicitly incorporating domain-specific global structural constraints into a content model would likely improve the accuracy of structure induction.

Another direction of future work is to combine the *global* topic structure of our model with *local* coherence constraints. As previously noted, our model is agnostic toward the relationships between sentences within a single topic. In contrast, models of local coherence take advantage of a wealth of additional knowledge, such as syntax, to make decisions about information flow across adjoining sentences. Such a linguistically rich model would provide a powerful representation of all levels of textual structure, and could be used for an even greater variety of applications than we have considered here.

Chapter 3

Learning Domain Relations using Posterior Regularization

The previous chapter described an unsupervised method for learning high-level topic structure in a set of in-domain documents. In this chapter, we present a novel approach for learning finer-grained *relation* structure. A relation is a specific type of information relevant to and mentioned across documents in the same domain. For example, given a collection of news articles about earthquakes, our method discovers relations such as the earthquake's location and resulting damage. In contrast to content modeling, relation instances are represented as extracted short phrases rather than entire sections of documents. Automatic relation discovery allows database representations to be rapidly constructed for new domains with little or no domain-specific expertise. This capability becomes increasingly important as clusters of similar in-domain documents describing latent structured information become more abundant, in forms such as Wikipedia article categories, financial reports, and biographies.

In contrast to previous work in relation extraction, our approach learns from *domain-independent meta-constraints* on relation expression, rather than supervision specific to particular relations and their instances. Specifically, we leverage the linguistic intuition that documents in a single domain exhibit regularities in how they express their relations. These regularities occur both in the relations' lexical and syntactic realizations as well as at the level of document structure. For instance, consider

A strong earthquake rocked the Philippine island of Mindoro early Tuesday, [destroying]$_{ind}$ [some homes]$_{arg}$ ...
A strong earthquake hit the China-Burma border early Wednesday ... The official Xinhua News Agency said [some houses]$_{arg}$ were [damaged]$_{ind}$ ...
A strong earthquake with a preliminary magnitude of 6.6 shook northwestern Greece on Saturday, ... [destroying]$_{ind}$ [hundreds of old houses]$_{arg}$ ...

Figure 3-1: Excerpts from newswire articles about earthquakes. The indicator and argument words for the *damage* relation are highlighted.

the *damage* relation excerpted from earthquake articles in Figure 3-1. Lexically, we observe similar words in the instances and their contexts, such as "destroying" and "houses." Syntactically, in two instances the relation instance is the dependency child of the word "destroying." On the discourse level, these instances appear toward the beginning of their respective documents. In general, valid relations in many domains are characterized by these coherence properties.

We capture these regularities using a Bayesian model where the underlying relations are represented as latent variables. The model takes as input a constituent-parsed corpus and generates them from the hidden relation structure variables. These variables encode each relation instance as a relation-evoking *indicator* word (*e.g.*, "destroying") and corresponding *argument* constituent (*e.g.*, "some homes").[1] Our model generates each indicator and argument instance of a single relation type from shared relation-specific distributions, allowing the model's generative process to encourage coherence in the local features and placement of relation instances.

By itself, the generative process is unable to capture certain intuitive constraints on relation expression. For example, a single pair of indicator word and argument constituent should typically exhibit particular syntactic characteristics, such as the argument being a dependency child of the indicator. These kinds of *declarative constraints* are captured by applying *posterior regularization* [57] during inference. This technique provides a principled and efficient way of enforcing soft declarative constraints, encoded as inequalities on arbitrary functions of the model's posterior distribution, without complicating the model structure itself. We use posterior regulariza-

[1]We do not use the word "argument" in the syntactic sense—a relation's argument may or may not be the syntactic dependency argument of its indicator.

tion to enforce three classes of declarative constraints: 1) indicators and arguments should usually be connected in the sentence's syntax tree via specific patterns, 2) a single relation should be prevalent across an entire corpus, and 3) different relations' instances should rarely overlap in the text. We also explore how constraints tailored to individual domains can further bias learning toward relevant relations.

We evaluate our approach on two domains previously studied for high-level document structure analysis, news articles about earthquakes and financial markets. Our results demonstrate that we can successfully identify domain-relevant relations, outperforming a previous state-of-the-art unsupervised semantic parser in both sentence- and token-level accuracy by substantial margins. Compared to supervised approaches, we find that a substantial number of annotated training examples is frequently required for comparable performance with our unsupervised model on sentence-level accuracy.

We also study the importance and effectiveness of the declaratively-specified constraints. In particular, we find that 1) a small set of soft declarative constraints is effective across domains, 2) removing any of the constraints degrades model performance, and 3) additional domain-specific constraints can yield further benefits.

This chapter is organized as follows. Section 3.1 contrasts our approach to previous information extraction and relation discovery setups, particularly to those that also utilize declarative constraints. We define our problem formulation in Section 3.2.1 and present the generative model in the remainder of Section 3.2. A key technical feature of our approach is the use of posterior regularization during variational inference, which we describe in detail in Sections 3.3.1 and 3.3.2. We then derive the variational updates specific to our model in Section 3.3.3. Section 3.4 examines the specific declarative constraints that we apply during inference. Finally, we present experiments and results in Section 3.5 before concluding in Section 3.6.

3.1 Related Work

Our approach relates to a growing body of work in information extraction with reduced supervision. Some of this work, described in Section 3.1.1, assumes a setup where the relation types are known, but supervision is either minimal or comes in a form other than fully annotated relation instances. In contrast, open information extraction setups, described in Section 3.1.2, do not assume a known set of relation types. They instead aim to extract *all* possible relations in a heterogeneous corpus by leveraging redundancies in relation expression. We also examine previous work that learns extractions and paraphrases based primarily on syntactic context in Section 3.1.3. Finally, in Section 3.1.4 we draw comparisons to previous work that has explicitly utilized constraints during inference to drive extraction.

3.1.1 Extraction with Minimal and Alternative Supervision

Recent research in information extraction has taken large steps toward reducing the need for labeled data. Many of these approaches use *bootstrapping*, where extractions are built up iteratively from a small seed set of annotated example outputs [1, 29, 30, 47, 113, 116, 117, 133]. These approaches take a similar high-level iterative strategy; we give specifics on several such methods here. Among the earliest such work was the DIPRE system for extracting relations from the web [29]. DIPRE takes as input a small seed set of target relation extractions, such as a list of (author, book title) pairs. It searches the given web corpus for these particular relation instances and builds a set of *patterns* that characterize the relation. Each pattern specifies the ordering of the relation (*e.g.,* whether author or title comes first), the URL prefix of the page containing the relation, and the text immediately surrounding and within the relation. These patterns are used to match other plausible book-title instances in the corpus, which are added to the seed set. The process repeats until a full set of extractions is obtained. Snowball [1] and StatSnowball [140] extend DIPRE by utilizing richer linguistic context in defining patterns, such as named entity tags, and by using confidence scores and probabilities to filter out unlikely candidate patterns.

ExDisco [133] uses a similar overall strategy, but starts from an initial set of seed patterns rather than example extractions. For example, for corporate management changes a likely seed pattern would be "*company verb person*," where *company* and *person* correspond to named entity classes and *verb* is a specified list of verbs such as *appoint*, *promote*, and *nominate*. ExDisco then learns new patterns by repeatedly performing two steps: 1) dividing the corpus into relevant and irrelevant documents based on the patterns, 2) adding the most prevalent patterns in the relevant documents to the pattern set. A human then reviews the final patterns and groups them into relation types.

Another line of work has reduced the need for supervision in extraction by utilizing various forms of database matching instead of labeled data. For example, Mintz et al. [96] and Yao et al. [134] train extraction models using the existing Freebase knowledge base derived from Wikipedia. In this setup, the knowledge base provides a large set of candidate relation types that may or may not be mentioned in the textual corpus. These systems use various alignment techniques to relate the knowledge base and the corpus, which then allows candidate relation instances to be extracted from the text.

Our approach is distinct from the bootstrapping and database matching approaches in both the form of supervision and the target output. First, rather than assuming access to minimal examples or a pre-existing collection of known relation types, we learn from *meta-qualities*, such as low variability in syntactic patterns, that characterize a good relation. We hypothesize that these properties hold across relations in different domains, thus reducing the need for supervision specific to relation types. Second, our method's goal is to discover the underlying types that are important to a domain. Both the bootstrapping- and database-based approaches rely on having some knowledge of target relation types, which may be costly for a human practitioner to collect for new complex domains.

3.1.2 Open Information Extraction

Another class of information extraction approaches strives to learn general domain-independent knowledge bases by exploiting redundancies in large web and news corpora [5, 6, 65, 121, 135]. These approaches do not assume knowledge of target relation types; instead, they aim to find the most prevalent relations across the corpus. For example, Shinyama and Sekine [121] learn relation types by hierarchically clustering a news corpus into documents about similar topics, then within each topic cluster into documents about the same event. This clustering uses features based on the documents' prevailing syntactic patterns. The output relations are based on the most prevalent patterns within each cluster. Banko et al. [5] extracts relations from the web by training an extraction classifier based on a small subset of the corpus, using heuristics rather than annotated training data to identify positive examples of relations. This classifier is then applied to the full web corpus to derive a large set of all possible relations; these relations are filtered to those that occur in the most sentences.

In contrast to these heterogeneous-corpus approaches, our focus is on learning the relations salient in a single domain. In our setup, each document expresses relations that are relevant for that document's subject; for example, a daily financial news report describes market activity specific to that day's trading session. Thus, we cannot rely on cross-document redundancy to identify salient relation instances. Our setup is more germane to specialized domains expressing information not broadly available on the web, such as personal medical records or internal corporate documents.

3.1.3 Syntax-driven Extraction

Earlier work in unsupervised information extraction has also leveraged syntactic meta-knowledge independent of specific relation types. Riloff [115] uses a set of declaratively-specified syntactic templates to produce a set of candidate extractions that a human then filters. Several approaches [36, 83, 117, 124, 139] perform automatic clusterings of syntactic tree fragments and contexts to cluster entity pairs into

relations. Our approach incorporates a broader range of constraints and balances soft syntactic constraints with other forms of linguistic regularity learned from the data. In particular, we incorporate arbitrary local features of relation entities and document-level information in addition to syntax, thereby requiring more sophisticated machinery for modeling and inference.

3.1.4 Constraint-based Extraction

A recent line of work has recognized the appeal of applying declarative constraints to extraction. In a supervised setting, Roth and Yih [118] induce relations by using linear programming to impose global constraints on the output of a set of classifiers trained on local features. Specifically, these classifiers are independently trained for identifying candidate relations and their entities; a linear program then induces the final relation structure while imposing constraints, such as the *work_for* relation needing to take person and organization arguments. Chang et al. [33] takes a semi-supervised approach that combines labeled and unlabeled data. They propose an objective function of model parameters that balances likelihood on labeled instances with constraint violation on unlabeled instances. These constraints state, for example, that a single relation type can only occur at most once in a document.

Recent work has also explored how certain kinds of supervision can be formulated as constraints on model posteriors. Such constraints are not declarative in the same way as our constraints, but instead based on annotations of words' majority relation labels [89] and pre-existing databases with the desired output schema [15]. In these approaches, a framework for applying constraints is used to bias the predictions to cohere with the provided forms of supervision.

In contrast to previous work, our approach explores a different class of constraints that does not rely on information specific to particular relation types and their instances. Moreover, our model is fully unsupervised and does not require any annotated examples for training. The constraints we apply are designed with generalizability across domains in mind. In our results, we also show that combining domain-specific constraints akin to those used by previous work with our domain-independent

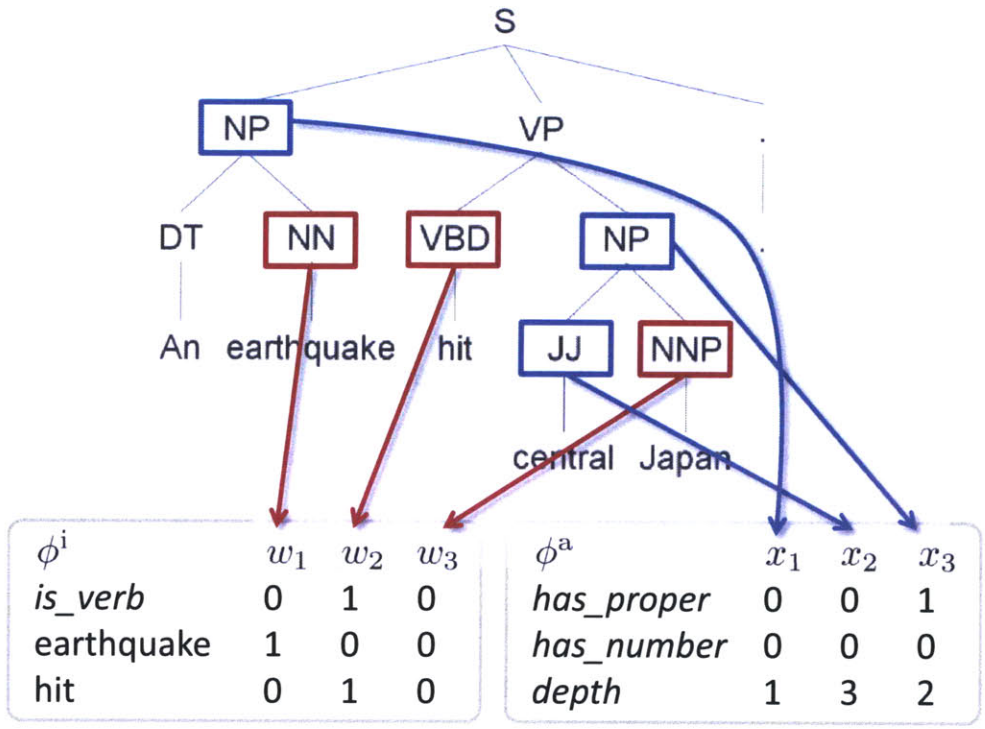


Figure 3-2: As input to the relation discovery model, words w and constituents x of syntactic parses are represented with indicator features ϕ^{i} and argument features ϕ^{a} respectively. A single relation instance is a pair of indicator w and argument x; we filter w to be nouns and verbs and x to be noun phrases.

constraints is beneficial for performance.

3.2 Model

Our work performs in-domain relation discovery by leveraging regularities in relation expression at the lexical, syntactic, and discourse levels. These regularities are captured via two components: a probabilistic model that explains how documents are generated from latent relation variables and a technique for biasing inference to adhere to declaratively-specified constraints on relation expression. This section describes the generative process, while Sections 3.3 and 3.4 discuss declarative constraints.

3.2.1 Problem Formulation

Our input is a corpus of constituent-parsed documents and a number K of relation types. In particular, each document d is comprised of an ordered sequence of sentences, and each sentence is a bag of words $\mathbf{w}$ and constituent phrases $\mathbf{x}$ delineated by the syntax tree. Note that words and constituents form two overlapping, complementary views of the corpus.

The output is K clusters of semantically related relation instances. We represent each instance as a pair of *indicator* word w and *argument* sequence x from the same sentence.[2] The indicator's role is to anchor a relation and identify its type. We only allow content-bearing nouns or verbs to be indicators. For instance, in the earthquake domain a likely indicator for *damage* would be "destroyed." The argument is the actual relation value, *e.g.*, "some homes," and corresponds to a noun phrase.

Along with the document parse trees, we also take as input a set of features $\phi^{\mathrm{i}}(w)$ and $\phi^{\mathrm{a}}(x)$ describing each potential indicator word w and argument constituent x, respectively. An example feature representation is shown in Figure 3-2. These features can encode words, part-of-speech tags, context, and so on. Indicator and argument feature definitions need not be the same (*e.g.*, *has_number* is important for arguments but irrelevant for indicators).[3]

3.2.2 Model Overview

Our model associates each hidden relation type k with a set of *feature distributions* θ_k and a *location distribution* λ_k. Each relation instance's indicator and argument, and its position within a document, are drawn from these distributions. By sharing distributions within each relation, the model places high probability mass on clusters of instances that are coherent in features and position. Furthermore, we allow at most one instance per document and relation, so as to target relations that are relevant to the entire document.

[2]One limitation of this formulation is that the indicator word must be explicit.

[3]We consider only categorical features here, though the extension to continuous or ordinal features is straightforward.

θ^{i}	–	parameters of feature distrs over indicator words
θ^{bi}	–	parameters of feature distrs over non-indicator words
θ^{a}	–	parameters of feature distrs over argument constituents
θ^{ba}	–	parameters of feature distrs over non-argument constituents
λ	–	parameters of distr over locations within document
s, z	–	segment/sentence containing relation instance
i	–	relation indicator word
a	–	relation argument constituent
$\phi^{\mathrm{i}}(w)$	–	features of potential indicators
$\phi^{\mathrm{a}}(x)$	–	features of potential arguments
K	–	number of relations
D	–	number of documents in corpus
$\lvert\Phi^{\mathrm{i}}\rvert$	–	number of indicator features
$\lvert\Phi^{\mathrm{a}}\rvert$	–	number of argument features
W_d	–	number of words in d
$W_{d,z}$	–	number of words in sentence z of d
C_d	–	number of constituents in d
$C_{d,z}$	–	number of constituents in sentence z of d
$S_{d,s}$	–	number of sentences in segment s of d

for $k = 1 \ldots K$
  $\lambda_k \sim \mathrm{Dirichlet}(\lambda_0)$

  for each indicator feature ϕ^{i}
    $\theta^{\mathrm{i}}_{k,\phi^{\mathrm{i}}} \sim \mathrm{Dirichlet}(\theta_0)$
    $\theta^{\mathrm{bi}}_{k,\phi^{\mathrm{i}}} \sim \mathrm{Dirichlet}(\theta_0)$

  for each argument feature ϕ^{a}
    $\theta^{\mathrm{a}}_{k,\phi^{\mathrm{a}}} \sim \mathrm{Dirichlet}(\theta_0)$
    $\theta^{\mathrm{ba}}_{k,\phi^{\mathrm{a}}} \sim \mathrm{Dirichlet}(\theta_0)$

for each document d
  for $k = 1 \ldots K$
    $s_{d,k} \sim \mathrm{Multinomial}(\lambda_k)$
    $z_{d,k} \sim \mathrm{Uniform}(S_{d,s_{d,k}})$
    $i_{d,k} \sim \mathrm{Uniform}(W_{d,z_{d,k}})$
    $a_{d,k} \sim \mathrm{Uniform}(C_{d,z_{d,k}})$

  for each word w in d
    for each indicator feature ϕ^{i}
      $A_w = \frac{1}{Z}\prod_{k=1}^{K}\theta_{k,\phi^{\mathrm{i}}}$,
        $\theta_{k,\phi^{\mathrm{i}}} = \theta^{\mathrm{i}}_{k,\phi^{\mathrm{i}}}$ if $i_{d,k} = w$,
        $\theta_{k,\phi^{\mathrm{i}}} = \theta^{\mathrm{bi}}_{k,\phi^{\mathrm{i}}}$ otherwise

      $\phi^{\mathrm{i}}(w) \sim \mathrm{Multinomial}(A_w)$

  for each constituent x in d
    for each argument feature ϕ^{a}
      $B_x = \frac{1}{Z}\prod_{k=1}^{K}\theta_{k,\phi^{\mathrm{a}}}$,
        $\theta_{k,\phi^{\mathrm{a}}} = \theta^{\mathrm{a}}_{k,\phi^{\mathrm{a}}}$ if $a_{d,k} = x$,
        $\theta_{k,\phi^{\mathrm{a}}} = \theta^{\mathrm{ba}}_{k,\phi^{\mathrm{a}}}$ otherwise

      $\phi^{\mathrm{a}}(x) \sim \mathrm{Multinomial}(B_x)$

Figure 3-3: The generative process for the relation discovery model. In the above, Z indicates a normalization factor that makes the parameters A_w and B_x sum to one. Fixed hyperparameters are subscripted with zero.

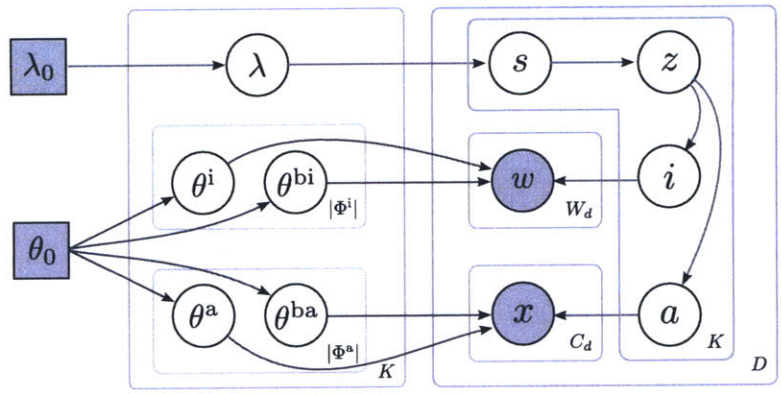


Figure 3-4: The plate diagram for the relation discovery model. Shaded circles in the figure denote observed variables, and squares denote hyperparameters. See Figure 3-3 for a full description of the variables.

There are three steps to the generative process. First, we draw feature and location distributions for each relation. Second, an instance is selected for every pair of document d and relation k. Third, the indicator features of each word and argument features of each constituent are generated based on the relation parameters and instances. Figure 3-3 presents a reference for the generative process, whose plate diagram is depicted in Figure 3-4.

Generating Relation Parameters

Each relation k is associated with four feature distribution parameter vectors: θ_k^{i} for indicator words, θ_k^{bi} for non-indicator words, θ_k^{a} for argument constituents, and θ_k^{ba} for non-argument constituents. Each of these is a set of multinomial parameters per feature drawn from a symmetric Dirichlet prior. A likely indicator word should have features that are highly probable according to θ_k^{i}, and likewise for arguments and θ_k^{a}. Parameters θ_k^{bi} and θ_k^{ba} represent *background* distributions for non-relation words and constituents.[4] By drawing each instance of a single relation from these distributions, we encourage the relation to be coherent in local lexical and syntactic properties.

Each relation type k is also associated with a parameter vector λ_k over document

[4]We use separate background distributions for each relation to make inference more tractable. Because the background distributions collect so many words and constituents, they will tend to be similar despite not being shared across relations.

segments. Documents are divided into L equal-length segments; λ_k states how likely relation k is for each segment, with one null outcome for the case when the relation does not occur in the document. Because λ_k is shared within a relation, its instances will tend to occur in the same relative positions across documents. The model can learn, for example, that a particular relation typically occurs in the first quarter of a document (if $L = 4$). Parameters λ_k are generated by a symmetric Dirichlet prior.

Generating Relation Instantiations

For every relation type k and document d, we first choose which portion of the document (if any) contains the instance by drawing a document segment $s_{d,k}$ from λ_k. Our model only draws one instance per pair of k and d, so each generated instance within a document is a separate relation. We then choose the specific sentence $z_{d,k}$ uniformly from within the segment, and the indicator word position $i_{d,k}$ and argument constituent position $a_{d,k}$ uniformly from within that sentence.

Generating Text

Finally, we draw the feature values. We make a Naïve Bayes assumption between features, drawing each independently conditioned on relation structure. For a word w, we want all relations to be able to influence its generation. Toward this end, we compute the element-wise product of feature parameters across relations $k = 1, \ldots, K$, using indicator parameters θ_k^{i} if relation k selected w as an indicator word (if $i_{d,k} = w$) and background parameters θ_k^{bi} otherwise. The result is then normalized to form a valid multinomial that produces word w's features.[5] Constituents are drawn similarly from every relations' argument distributions. By generating features using this element-wise product, we allow a single word to play a role in multiple relations.

[5]Only the word's features are generated; the word identity itself is not explicitly generated, but will typically be one of the features.

3.2.3 Formal Generative Process

We now formally specify the full generative process. We take as observed a corpus of D documents where each document d is comprised of words $\mathbf{w}$, filtered to content-bearing nouns and verbs. The corpus is also constituent-parsed into a set of candidate constituents $\mathbf{x}$, filtered to noun phrases. Each word w and constituent x is associated with an observed set of indicator features $\phi^{\mathrm{i}}(w)$ and argument features $\phi^{\mathrm{a}}(x)$, respectively. The number of hidden relations K is assumed to be pre-specified. Additionally, the model is provided with a fixed hyperparameter L specifying the number of segments each document is divided into. The model's independence assumptions are represented by this factorization into conditional probability distributions:

$$\prod_{k=1}^{K} p(\lambda_k)p(\theta_k^{\mathrm{i}})p(\theta_k^{\mathrm{bi}})p(\theta_k^{\mathrm{a}})p(\theta_k^{\mathrm{ba}}) \prod_{d} p(s_{d,k} \mid \lambda_k)p(z_{d,k} \mid s_{d,k})p(a_{d,k} \mid z_{d,k})p(i_{d,k} \mid z_{d,k})$$
$$\times \prod_{w \in d} \prod_{\phi^{\mathrm{i}}} p(\phi^{\mathrm{i}}(w) \mid \theta^{\mathrm{i}}, \theta^{\mathrm{bi}}, \mathbf{i}_d) \prod_{x \in d} \prod_{\phi^{\mathrm{a}}} p(\phi^{\mathrm{a}}(x) \mid \theta^{\mathrm{a}}, \theta^{\mathrm{ba}}, \mathbf{a}_d) \tag{3.1}$$

Our final desired output is the posterior distributions over the relation structure $p(\mathbf{s}, \mathbf{z}, \mathbf{i}, \mathbf{a} \mid \mathbf{w}, \mathbf{x})$. In the following description, variables subscripted with 0 are fixed prior hyperparameters.

1. For each relation type k:

   (a) For each indicator feature ϕ^{i} draw feature distribution parameters $\theta^{\mathrm{i}}_{k,\phi^{\mathrm{i}}}, \theta^{\mathrm{bi}}_{k,\phi^{\mathrm{i}}} \sim \mathrm{Dir}(\theta_0)$. These two parameter sets represent parameters for indicator and non-indicator words of relation k, respectively.

   (b) For each argument feature ϕ^{a} draw feature distributions $\theta^{\mathrm{a}}_{k,\phi^{\mathrm{a}}}, \theta^{\mathrm{ba}}_{k,\phi^{\mathrm{a}}} \sim \mathrm{Dir}(\theta_0)$. As with indicators, these two parameter sets represent feature distributions for argument and non-argument constituents of relation k.

   (c) Draw location distribution $\lambda_k \sim \mathrm{Dir}(\lambda_0)$. This distribution over $L+1$ values controls where in a document relation k should occur; a peaky λ distribution encourages similar placement (*e.g.,* toward the beginning) of a single relation's instances across documents. The first L outcomes

correspond to L equally-sized segments of each document; the last outcome indicates that the relation does not occur in the document.

2. For each document d:

   (a) For each relation type k:

      i. Select a document segment $s_{d,k} \sim \text{Mult}(\lambda_k)$.

      ii. If $s_{d,k}$ is the null case (*i.e.,* relation k does not occur in document d), set $z_{d,k}$, $a_{d,k}$, and $i_{d,k}$ all to null.

      iii. Otherwise, select a sentence index $z_{d,k}$ uniformly from the sentences contained in segment $s_{d,k}$. Then draw indicator position $i_{d,k}$ and argument position $a_{d,k}$ uniformly from the words and constituents, respectively, of sentence $z_{d,k}$.

   (b) For each potential indicator word w in document d:

      i. Draw each indicator feature $\phi^{\text{i}}(w) \sim \text{Mult}\left(\frac{1}{Z}\prod_{k=1}^{K}\theta_{k,\phi^{\text{i}}}\right)$, where $\theta_{k,\phi^{\text{i}}}$ is $\theta^{\text{i}}_{k,\phi^{\text{i}}}$ if $i_{d,k} = w$ and $\theta^{\text{bi}}_{k,\phi^{\text{i}}}$ otherwise. The element-wise product between different relations' distributions allows every relation to influence the generation of each word. Here, Z is a normalization factor that makes the multiplied parameters sum to one, which is required for them to form valid multinomial parameters.

   (c) For each potential argument constituent x in document d:

      i. Draw each argument feature $\phi^{\text{a}}(x) \sim \text{Mult}\left(\frac{1}{Z}\prod_{k=1}^{K}\theta_{k,\phi^{\text{a}}}\right)$, where $\theta_{k,\phi^{\text{a}}}$ is $\theta^{\text{a}}_{k,\phi^{\text{a}}}$ if $a_{d,k} = x$ and $\theta^{\text{ba}}_{k,\phi^{\text{a}}}$ otherwise.

3.2.4 Properties of the Model

The generative process presented above leverages relation regularities in local features and document placement. By utilizing lexical and syntactic features in ϕ^{i} and ϕ^{a}, the model will tend to identify relation instances that cohere lexically and syntactically (*e.g.*, a small set of words as the indicator). Furthermore, using a shared relation placement distribution λ encourages relation instances to occur in similar locations

at the document structure level.

However, the generative process is unable to specify syntactic preferences about how indicators and arguments are related within a sentence, since they are generated separately. For example, intuitively we expect that indicators and arguments should usually not cross clause boundaries, a constraint that the generative process is unable to institute. Furthermore, this generative process allows different relations to overlap in their indicators and arguments, which would often be undesirable.[6] Ideally we would like to impose a constraint that biases against repeated overlap.

Incorporating these constraints directly in the model structure would complicate the model structure and introduce new parameters, thus making inference less tractable. Instead, we impose constraints during the posterior inference process to bias parameter learning, as explained in the next section.

3.3 Inference with Constraints

Given our generative model and a set of documents, the goal of inference is to derive a posterior distribution over the hidden parameters describing the relation structure:

$$p(\mathbf{s}, \mathbf{z}, \mathbf{a}, \mathbf{i} \mid \mathbf{w}, \mathbf{x}). \tag{3.2}$$

We are interested primarily in the hidden relation structure as opposed to the relation parameters, so our posterior objective integrates out θ and λ. For most nontrivial models, including ours, it is not possible to find the posterior analytically due to the complex dependencies in the model. Instead we appeal to an approximate inference technique, specifically, variational inference [19, 72].[7] In variational inference, we find an approximate posterior distribution $q(\mathbf{s}, \mathbf{z}, \mathbf{a}, \mathbf{i})$ that is close in KL-divergence to the true posterior. This is made tractable by restricting q to come from a restricted set

[6] In fact, a true *maximum a posteriori* estimate of the model parameters would find the same most salient relation over and over again for every k, rather than finding K different relations.

[7] Gibbs sampling, which we use for the content structure and semantic property models presented in the other chapters, does not admit the declarative constraint machinery that we require for this work.

of distributions, as explained in Section 3.3.1.

During inference, we apply various declarative constraints by imposing inequalities on expectations of the posterior using the *posterior regularization* technique [57]. In Section 3.3.2 we present the technical details of this approach for arbitrary constraints; Section 3.4 explains the specific linguistically-motivated constraints we consider.

3.3.1 Variational Inference

We first review the general variational inference setup for graphical models, then describe how posterior regularization is incorporated. Let θ and $\mathbf{x}$ denote the parameters and observed data, respectively, of an arbitrary model. We are interested in estimating the posterior distribution $p(\theta \mid \mathbf{x})$. This is not directly tractable, so instead we try to find a distribution $q(\theta) \in \mathcal{Q}$ that is "close" to the true posterior distribution. In this context, closeness is defined by the *KL-divergence* between q and the true posterior:

$$\arg\min_q \mathrm{KL}(q(\theta) \parallel p(\theta \mid \mathbf{x})) = \arg\min_q \left(\int q(\theta) \log \frac{q(\theta)}{p(\theta, \mathbf{x})} d\theta + \log p(\mathbf{x}) \right). \tag{3.3}$$

Because the term $\log p(\mathbf{x})$ is fixed with respect to q, we can drop it from the minimization.

If we do not restrict the set of valid distributions $q \in \mathcal{Q}$, this formulation does not simplify our problem: the q distribution that minimizes the KL divergence is simply the true unknown posterior $p(\theta \mid x)$. Instead, we make a *mean-field* independence assumption stating that some partition of the model parameters θ is independent. Specifically, let $\theta_1, \ldots, \theta_n$ be a partition of the variables of θ. Then the assumption states that $\mathcal{Q}$ is restricted to distributions q that factorize as follows:

$$q(\theta) = \prod_{i=1}^{n} q(\theta_i). \tag{3.4}$$

For notational convenience, we will abbreviate $q(\theta_i)$ simply as q_i.

Thanks to this mean-field assumption, the optimization problem of equation 3.3

can be tractably tackled by coordinate descent: we minimize the objective with respect to each factor of the partition, *i.e.*, a single q_i, at a time while holding every other factor fixed. For factor q_i, the objective of equation 3.3 yields the following expression after rearranging terms and dropping constants [19]:

$$\arg\max_{q_i} \mathbb{E}_{q_i}[\log \tilde{p}(\theta, \mathbf{x})] - \mathbb{E}_{q_i}[\log q_i], \tag{3.5}$$

where $\tilde{p}(\theta, \mathbf{x})$ denotes:

$$\tilde{p}(\theta, \mathbf{x}) \propto \exp \mathbb{E}_{q_j \neq q_i}[\log p(\theta, \mathbf{x})]. \tag{3.6}$$

Here, $\mathbb{E}_{q_j \neq q_i}[\cdot]$ denotes an expectation with respect to every factor of q except q_i. We recognize that equation 3.5 is simply the negative of the KL-divergence between q_i and $\tilde{p}(\theta, \mathbf{x})$. Thus, equation 3.5 is maximized when $q_i = \tilde{p}(\theta, \mathbf{x})$, where the KL-divergence reaches its minimum of zero. Consequently the update for q_i is:

$$q_i \propto \exp \mathbb{E}_{q_j \neq q_i}[\log p(\theta, \mathbf{x})]. \tag{3.7}$$

For many common distributions, including all conjugate prior/likelihood distribution pairs in the exponential family, equation 3.7 yields a closed form exact update for q_i. As we will see, however, this closed form update cannot be used for some updates when posterior constraints are in effect, and when the expectation of equation 3.7 or its normalization factor is not analytically tractable. In those cases, we will use numerical optimization techniques directly on the objective function of equation 3.5 with respect to the parameters of q_i.

Each update of a factor of q decreases or leaves unchanged the value of the KL-divergence in equation 3.3, so iteratively applying these updates is guaranteed to converge toward a local minimum. The parameter space will typically be non-convex for nontrivial problems, so random restarts can be used to find a better global solution.

3.3.2 Posterior Regularization

Declarative constraints can be imposed during inference through *posterior regularization* [57], which we explain here. In this section, assume an arbitrary graphical model with observed data $\mathbf{x}$ and parameters divided into two groups, θ and $\mathbf{z}$, the latter of which will be constrained. Recall that variational inference makes a mean-field assumption that restricts the set of approximating distributions $\mathcal{Q}$ to those satisfying a factorization of the form in equation 3.4. Posterior regularization further restricts $\mathcal{Q}$; it requires that members of $\mathcal{Q}$ must satisfy declarative constraints formulated as inequalities on expectation functions of the posterior distribution:

$$\mathbb{E}_q[f(\mathbf{z})] \leq b. \tag{3.8}$$

Here, $f(\mathbf{z})$ is a deterministic function of $\mathbf{z}$ and b is a user-specified threshold. Note that inequalities in the opposite direction can be applied by negating $f(\mathbf{z})$ and b, and equality constraints can be effected by using inequality constraints in both directions. For tractability reasons explained later in this section, we will require that function $f(\mathbf{z})$ is always linear over the variables of $\mathbf{z}$.

Constraints of the form in equation 3.8 provide a flexible mechanism for specifying a wide range of declarative knowledge. For example, one of the constraints we will apply is of the form $\mathbb{E}_q[f(z)] \geq b$ where $f(z)$ counts the number of indicator/argument pairs that are syntactically connected in a pre-specified manner (*e.g.*, the indicator and argument modify the same verb) and b is a fixed threshold. Such a constraint would bias learning to prefer relation structures that are syntactically plausible according to our linguistic knowledge, while still finding relation instances with other kinds of syntactic patterns given strong enough support by the data. In general, this inequality formulation can capture nearly any constraint expressible as a linear function of the hidden structure that must meet some threshold.

We now derive the new form for variational updates in the presence of constraints. Let $\mathcal{C}$ be the set of inequality constraints, with functions $f_c(\mathbf{z})$ and thresholds b_c. For notational simplicity, assume that the only mean-field factorization assumption is

between θ and $\mathbf{z}$, *i.e.*, $q(\theta, \mathbf{z}) = q(\theta)q(\mathbf{z})$; the extension to finer-grained factorizations is straightforward. First, the update for $q(\theta)$ is unchanged from equation 3.5 since its value does not affect whether the constraints, defined only on $q(\mathbf{z})$, are satisfied. For $\mathbf{z}$, we perform the optimization of equation 3.5 in the presence of $\mathcal{C}$:

$$\arg\max_{q(\mathbf{z})} \mathbb{E}_{q(\mathbf{z})}[\log \tilde{p}(\theta, \mathbf{z}, \mathbf{x})] - \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})] \quad s.t. \quad \mathbb{E}_{q(z)}[f_c(z)] \leq b_c, \ \forall c \in \mathcal{C}, \tag{3.9}$$

where $\tilde{p}(\theta, \mathbf{z}, \mathbf{x}) \propto \exp \mathbb{E}_{q(\theta)}[\log p(\theta, \mathbf{z}, \mathbf{x})]$.

Directly optimizing equation 3.9 is difficult due to the complexity of the constraints, but Graça et al. [57] show that the expression's *dual* formulation is typically tractable:

$$\arg\min_{\kappa} \sum_{c \in \mathcal{C}} \kappa_c b_c + \log \sum_{\mathbf{z}} \tilde{p}(\theta, \mathbf{z}, \mathbf{x}) \exp\left(-\sum_{c \in \mathcal{C}} \kappa_c f_c(\mathbf{z})\right) \quad s.t. \quad \kappa_c \geq 0, \ \forall c \in \mathcal{C}. \tag{3.10}$$

Here, κ is a newly introduced $|\mathcal{C}|$-dimensional vector of dual variables, one for each of the original constraints. In equation 3.10 the sum over all configurations of the hidden parameters $\mathbf{z}$ appears intractable at first, since most models have exponentially many values of $\mathbf{z}$. However, note that $\tilde{p}(\theta, \mathbf{z}, \mathbf{x})$ decomposes into a product over cliques of random variables according to the original factorization of the probabilistic model. Furthermore, since we assume that every $f_c(\mathbf{z})$ is linear in the variables of $\mathbf{z}$, the term $\exp\left(-\sum_{c \in \mathcal{C}} \kappa_c f_c(\mathbf{z})\right)$ also factorizes in the same way. Therefore the sum over $\mathbf{z}$ decomposes into tractable sums over independent cliques of $\mathbf{z}$.

With the box constraints of equation 3.10, a numerical optimization procedure such as L-BFGS-B [31] can be used to find optimal dual parameters κ^*. The corresponding primal update then takes the following form [57]:

$$q(\mathbf{z}) \propto \tilde{p}(\theta, \mathbf{z}, x) \exp\left(-\sum_{c \in \mathcal{C}} \kappa_c^* f_c(z)\right). \tag{3.11}$$

3.3.3 Variational Updates for the Model

Now that we have developed the general variational inference with posterior regularization framework, we turn to the individual updates for our model. Our full model posterior is defined over the relation feature parameters θ, document location parameters λ, and relation structure variables $\mathbf{s}, \mathbf{z}, \mathbf{a}, \mathbf{i}$. Thus, our variational distribution is of the form $q(\theta, \lambda, \mathbf{s}, \mathbf{z}, \mathbf{a}, \mathbf{i})$; we use the following mean-field factorization:

$$q(\theta, \lambda, \mathbf{s}, \mathbf{z}, \mathbf{a}, \mathbf{i}) = \prod_{k=1}^{K} q(\lambda_k; \hat{\lambda}_k) q(\theta_k^{\mathrm{i}}; \hat{\theta}_k^{\mathrm{i}}) q(\theta_k^{\mathrm{bi}}; \hat{\theta}_k^{\mathrm{bi}}) q(\theta_k^{\mathrm{a}}; \hat{\theta}_k^{\mathrm{a}}) q(\theta_k^{\mathrm{ba}}; \hat{\theta}_k^{\mathrm{ba}}) \times \prod_{d} q(s_{d,k}, z_{d,k}, a_{d,k}, i_{d,k}; \hat{\zeta}_{d,k}) \tag{3.12}$$

In the above, we make each $q(\theta)$ and $q(\lambda)$ a Dirichlet with corresponding *variational parameters* $\hat{\theta}$ and $\hat{\lambda}$, and and each $q(s, z, a, i; \hat{\zeta})$ a multinomial with variational parameters $\hat{\zeta}$.[8] Updating each factor is equivalent to updating the corresponding variational parameters. Note that we do not factorize the distribution of s, z, i, and a for a single document and relation, instead representing their joint distribution with a single set of variational parameters $\hat{\zeta}$. This is tractable because a single relation occurs only once per document, reducing the joint search space of these variables.

All of the declarative constraints we impose on the posterior restrict properties of the relation structure itself, not the relation parameters (see Section 3.4 for details). In other words, their constraint inequalities are defined on s, z, i, and a. Hence the updates for $\hat{\lambda}$ and $\hat{\theta}$ follow from equation 3.5 directly. For $\hat{\zeta}$, we solve the optimization problem in equation 3.10.

[8]Distributions $q(\lambda)$ and $q(s, z, a, i)$ naturally form Dirichlet and multinomial distributions respectively when optimized according to equation 3.5 due to the use of conjugate priors for those variables. However, θ generates feature values through a non-conjugate pointwise product, so its posterior $q(\theta)$ does not necessarily form a Dirichlet naturally. As with previous generative models using product distributions [26], we restrict $q(\theta)$ to Dirichlet for tractability.

Updating $\hat{\lambda}$

Since λ is drawn from its conjugate prior, the update for $\hat{\lambda}$ follows directly from the closed form solution of equation 3.7. After dropping constants, we arrive at:

$$q(\lambda_k) \propto \exp\left(\log p(\lambda_k) + \sum_d \mathbb{E}_{q(s_{d,k}, z_{d,k}, i_{d,k}, a_{d,k})}[\log p(s_{d,k} \mid \lambda_k)]\right) \tag{3.13}$$

By exponentiating both sides it is straightforward to verify that the right hand side becomes an unnormalized Dirichlet density function. The final update takes on the following form:

$$\hat{\lambda}_{k,\ell} = \lambda_0 + \sum_d \sum_{\zeta_{d,k}} \hat{\zeta}_{d,k} \mathcal{I}(\zeta_{d,k}, \ell), \tag{3.14}$$

where $\mathcal{I}(\zeta_{d,k}, \ell)$ is a binary function returning 1 if the segment s implied by $\zeta_{d,k}$ is equal to ℓ. The update intuitively states that the parameters are set to the prior pseudo-counts plus the expected count of each segment ℓ based on the current estimate of the other parameters. This pseudo-count formulation is common to multinomial/Dirichlet combinations in the variational inference setting.

Updating $\hat{\theta}$

Due to the Naïve Bayes assumption between features, each feature's $q(\theta)$ distributions can be updated separately. However, the product between feature parameters of different relations introduces a non-conjugacy in the model, precluding a closed form update as in equation 3.7. Instead we numerically optimize equation 3.5 for each $\hat{\theta}$, as in previous work [26].

We will derive the update for a single $\hat{\theta}$, specifically the indicator feature distribution parameters $\hat{\theta}^{\mathrm{i}}_{k,\phi^{\mathrm{i}}}$; the derivations for other $\hat{\theta}$'s is analogous. For notational clarity we drop the subscript of ϕ^{i} on θ. Assume feature ϕ^{i} can take on values from 1 to N, so $\hat{\theta}^{\mathrm{i}}_k = (\hat{\theta}^{\mathrm{i}}_{k,1}, \ldots, \hat{\theta}^{\mathrm{i}}_{k,N})$ is an N-dimensional parameter vector. The objective from equation 3.5 becomes:

$$\arg\max_{\hat{\theta}^{\mathrm{i}}} \mathbb{E}_q[\log p(\theta, \lambda, \mathbf{s}, \mathbf{z}, \mathbf{i}, \mathbf{a})] - \mathbb{E}_{q(\theta)}[\log q(\theta)]. \tag{3.15}$$

In the following, let $B(\cdot)$ be the Beta function, $\sum_{\zeta_{d,k}}$ be a sum over every possible valid combination of s, z, i, and a for document d and relation k, and $\mathcal{I}(\zeta, w)$ be a binary function that returns 1 if the indicator location i corresponding to the given combination ζ is word w. Plugging in the decomposition of p from equation 3.1 into equation 3.15 and dropping constant terms yields:

$$
\begin{aligned}
&\mathbb{E}_q[\log p(\theta_k^{\mathrm{i}})] + \sum_{d,w} \mathbb{E}_q[\log p(\phi^{\mathrm{i}}(w) \mid \theta_k^{\mathrm{i}}, \theta_k^{\mathrm{bi}}, s_{d,k}, z_{d,k}, i_{d,k}, a_{d,k})] - \mathbb{E}_{q(\theta_k^{\mathrm{i}})}[\log q(\theta_k^{\mathrm{i}})] \\
&= -\log B(\theta_0) + \sum_{n=1}^{N} (\theta_0 - 1)\mathbb{E}_q[\log \theta_{k,n}^{\mathrm{i}}] \\
&\quad + \sum_{d,w} \sum_{\zeta_{d,1}} \cdots \sum_{\zeta_{d,K}} \prod_{k'=1}^{K} \hat{\zeta}_{d,k'} \mathbb{E}_q \left[\log \prod_{k'=1}^{K} \left(\mathcal{I}(\zeta_{d,k'}, w)\theta_{k',\phi^{\mathrm{i}}(w)}^{\mathrm{i}} + (1 - \mathcal{I}(\zeta_{d,k'}, w))\theta_{k',\phi^{\mathrm{i}}(w)}^{\mathrm{bi}} \right) \right] \\
&\quad - \sum_{d,w} \sum_{\zeta_{d,1}} \cdots \sum_{\zeta_{d,K}} \prod_{k'=1}^{K} \hat{\zeta}_{d,k'} \mathbb{E}_q \left[\log \sum_{n=1}^{N} \prod_{k'=1}^{K} \left(\mathcal{I}(\zeta_{d,k'}, w)\theta_{k',n}^{\mathrm{i}} + (1 - \mathcal{I}(\zeta_{d,k'}, w))\theta_{k',n}^{\mathrm{bi}} \right) \right] \\
&\quad + \log B(\hat{\theta}_k^{\mathrm{i}}) - \sum_{n=1}^{N} (\hat{\theta}_{k,n}^{\mathrm{i}} - 1)\mathbb{E}_q[\log \theta_{k,n}^{\mathrm{i}}].
\end{aligned}
$$

To proceed with the derivation, we make a first-order Taylor series approximation of the form $\log x \leq x - 1$ to the third term of the above expression. Because this term is negative, approximating $\log x$ with $x - 1$ effectively yields a lower bound on the true $\hat{\theta}_k^{\mathrm{i}}$ update objective function, a desirable feature since we are maximizing this objective. Additionally, note that for an N-dimensional Dirichlet distribution $q(\theta; \hat{\theta})$ the expectation $\mathbb{E}_q[\log \theta_i]$ evaluates to $\psi(\hat{\theta}_i) - \psi\left(\sum_j \hat{\theta}_j\right)$, where $\psi(\cdot)$ is the digamma function, and the expectation $\mathbb{E}_q[\theta_i]$ evaluates to $\hat{\theta}_i / \sum_j \hat{\theta}_j$. In the following, let $\hat{\theta}_{k,*} = \sum_{n=1}^{N} \hat{\theta}_{k,n}$. Further simplifications and dropping constants yield the final

objective:

$$
\begin{aligned}
&- \log B(\theta_0) + \sum_{n=1}^{N} (\theta_0 - 1) \left(\psi(\hat{\theta}^{\mathrm{i}}_{k,n}) - \psi\left(\hat{\theta}^{\mathrm{i}}_{k,*}\right) \right) \\
&\quad + \sum_{d,w} \sum_{\zeta_{d,k}} \hat{\zeta}_{d,k} \mathcal{I}(\zeta_{d,k}, w) \left(\psi(\hat{\theta}^{\mathrm{i}}_{k,\phi^{\mathrm{i}}(w)}) - \psi(\hat{\theta}^{\mathrm{i}}_{k,*}) \right) \\
&\quad - \sum_{d,w} \sum_{n=1}^{N} \prod_{k'=1}^{K} \sum_{\zeta_{d,k'}} \hat{\zeta}_{d,k'} \left(\mathcal{I}(\zeta_{d,k'}, w) \frac{\hat{\theta}^{\mathrm{i}}_{k',n}}{\hat{\theta}^{\mathrm{i}}_{k',*}} + (1 - \mathcal{I}(\zeta_{d,k'}, w)) \frac{\hat{\theta}^{\mathrm{bi}}_{k',n}}{\hat{\theta}^{\mathrm{bi}}_{k',*}} \right) \\
&\quad + \log B(\hat{\theta}^{\mathrm{i}}_{k}) - \sum_{n=1}^{N} (\hat{\theta}^{\mathrm{i}}_{k,n} - 1) \left(\psi(\hat{\theta}^{\mathrm{i}}_{k,n}) - \psi(\hat{\theta}^{\mathrm{i}}_{k,*}) \right). \qquad (3.16)
\end{aligned}
$$

The gradient of this expression with respect to $\hat{\theta}^{\mathrm{i}}_{k}$ is straightforward to analytically derive, so we perform this optimization by applying the quasi-Newton L-BFGS numerical optimization procedure. This yields the update to $\hat{\theta}^{\mathrm{i}}_{k}$.

Simplifying Approximation The update for $\hat{\theta}$ requires numerical optimization due to the nonconjugacy introduced by the point-wise product in feature generation. If instead we have every relation type separately generate a copy of the corpus, the $\hat{\theta}$ updates become vastly simpler closed-form expressions in the form of equation 3.7, similar to the $\hat{\lambda}$ update in equation 3.14. For example, $\hat{\theta}_{k,\phi^{\mathrm{i}}}$'s update becomes:

$$
\hat{\theta}_{k,\phi^{\mathrm{i}},n} = \theta_0 + \sum_{d,w} \sum_{\zeta_{d,k}} \hat{\zeta}_{d,k} \mathcal{I}(\zeta_{d,k}, w) \phi^{\mathrm{i}}(w). \qquad (3.17)
$$

Note that this approximation effectively drops the third term from the $\hat{\theta}$ objective in equation 3.16. Empirically, we find that this approximation yields very close parameter estimates to the true updates while vastly improving speed. For this reason our experimental results are reported using this approximation.

Updating $\hat{\zeta}$

Because parameters $\hat{\zeta}$ are impacted by constraints, their update is a two step procedure. First, we compute $\tilde{p}(\theta, \lambda, \mathbf{s}, \mathbf{z}, \mathbf{i}, \mathbf{a}, \mathbf{w}, \mathbf{x})$ using equation 3.6, which is equivalent

$$
\begin{aligned}
\hat{\zeta}'_{k,m} &\propto \frac{\hat{\lambda}_s}{S_s W_z C_z} \\
&\times \exp\Bigg\{ \sum_{\phi^{\mathrm{i}}} \Bigg[\psi\left(\hat{\theta}^{\mathrm{i}}_{k,\phi^{\mathrm{i}},\phi^{\mathrm{i}}(w_i)}\right) - \psi\left(\hat{\theta}^{\mathrm{i}}_{k,\phi^{\mathrm{i}},*}\right) + \sum_{w \neq w_i} \left(\psi\left(\hat{\theta}^{\mathrm{bi}}_{k,\phi^{\mathrm{i}},\phi^{\mathrm{i}}(w)}\right) - \psi\left(\hat{\theta}^{\mathrm{bi}}_{k,\phi^{\mathrm{i}},*}\right) \right) \\
&\qquad - \sum_{w} \sum_{n=1}^{N_{\phi^{\mathrm{i}}}} \left(\left(\mathcal{I}(\zeta_{k,m}, w) \frac{\hat{\theta}^{\mathrm{i}}_{k,\phi^{\mathrm{i}},n}}{\hat{\theta}^{\mathrm{i}}_{k,\phi^{\mathrm{i}},*}} + (1 - \mathcal{I}(\zeta_{k,m}, w)) \frac{\hat{\theta}^{\mathrm{bi}}_{k,\phi^{\mathrm{i}},n}}{\hat{\theta}^{\mathrm{bi}}_{k,\phi^{\mathrm{i}},*}} \right) \right. \\
&\qquad \left. \times \prod_{k' \neq k} \sum_{\zeta_{k',m'}} \hat{\zeta}_{k',m'} \left(\mathcal{I}(\zeta_{k',m'}, w) \frac{\hat{\theta}^{\mathrm{i}}_{k',\phi^{\mathrm{i}},n}}{\hat{\theta}^{\mathrm{i}}_{k',\phi^{\mathrm{i}},*}} + (1 - \mathcal{I}(\zeta_{k',m'}, w)) \frac{\hat{\theta}^{\mathrm{bi}}_{k',\phi^{\mathrm{i}},n}}{\hat{\theta}^{\mathrm{bi}}_{k',\phi^{\mathrm{i}},*}} \right) \right) \Bigg] \\
&\quad + \sum_{\phi^{\mathrm{a}}} \Bigg[\psi\left(\hat{\theta}^{\mathrm{a}}_{k,\phi^{\mathrm{a}},\phi^{\mathrm{a}}(x_a)}\right) - \psi\left(\hat{\theta}^{\mathrm{a}}_{k,\phi^{\mathrm{a}},*}\right) + \sum_{x \neq x_a} \left(\psi\left(\hat{\theta}^{\mathrm{ba}}_{k,\phi^{\mathrm{a}},\phi^{\mathrm{a}}(x)}\right) - \psi\left(\hat{\theta}^{\mathrm{ba}}_{k,\phi^{\mathrm{a}},*}\right) \right) \\
&\qquad - \sum_{x} \sum_{n=1}^{N_{\phi^{\mathrm{a}}}} \left(\left(\mathcal{I}(\zeta_{k,m}, x) \frac{\hat{\theta}^{\mathrm{a}}_{k,\phi^{\mathrm{a}},n}}{\hat{\theta}^{\mathrm{a}}_{k,\phi^{\mathrm{a}},*}} + (1 - \mathcal{I}(\zeta_{k,m}, x)) \frac{\hat{\theta}^{\mathrm{ba}}_{k,\phi^{\mathrm{a}},n}}{\hat{\theta}^{\mathrm{ba}}_{k,\phi^{\mathrm{a}},*}} \right) \right. \\
&\qquad \left. \times \prod_{k' \neq k} \sum_{\zeta_{k',m'}} \hat{\zeta}_{k',m'} \left(\mathcal{I}(\zeta_{k',m'}, x) \frac{\hat{\theta}^{\mathrm{a}}_{k',\phi^{\mathrm{a}},n}}{\hat{\theta}^{\mathrm{a}}_{k',\phi^{\mathrm{a}},*}} + (1 - \mathcal{I}(\zeta_{k',m'}, x)) \frac{\hat{\theta}^{\mathrm{ba}}_{k',\phi^{\mathrm{a}},n}}{\hat{\theta}^{\mathrm{ba}}_{k',\phi^{\mathrm{a}},*}} \right) \right) \Bigg] \Bigg\}
\end{aligned}
$$

Figure 3-5: Unconstrained variational update for $\hat{\zeta}$.

to the update we would have made had there been no constraints. We notate the parameters of this distribution as $\hat{\zeta}'_{d,k} = (\hat{\zeta}'_{k,1}, \ldots, \hat{\zeta}'_{k,M})$, with one parameter for every valid combination of s, z, i, and a; we will drop the subscript d below for notational clarity. We find $\hat{\zeta}'$ using the update in equation 3.7 and the model factorization in equation 3.1, yielding:

$$\begin{aligned}\hat{\zeta}'_{k,m} \propto \exp\Bigg(&\mathbb{E}_{q(\lambda)}[\log p(s_k \mid \lambda_k)] + \log p(z_k \mid s_k) + \log p(i_k \mid z_k) + \log p(a_k \mid z_k) \\ &+ \mathbb{E}_{q(\theta^{\mathrm{i}})}[\log p(w_i \mid \theta^{\mathrm{i}})] + \sum_{w \neq w_i} \mathbb{E}_{q(\theta^{\mathrm{bi}})}[\log p(w \mid \theta^{\mathrm{bi}})] \\ &+ \mathbb{E}_{q(\theta^{\mathrm{a}})}[\log p(x_a \mid \theta^{\mathrm{a}})] + \sum_{x \neq x_a} \mathbb{E}_{q(\theta^{\mathrm{ba}})}[\log p(x \mid \theta^{\mathrm{ba}})]\Bigg), \end{aligned} \tag{3.18}$$

where s, z, i, and a are the segment, sentence, indicator, and argument selections corresponding to case m, w_i is the word selected as the indicator by i, and similarly for x_a. Let N_ϕ be the number of possible values for feature ϕ and $\hat{\theta}_{k,\phi,*} = \sum_{n=1}^{N_\phi} \hat{\theta}_{k,\phi,n}$. Plugging in the individual probability distributions and evaluating the expectations yields the closed-form unconstrained update shown in Figure 3-5. In that expression, S_s, W_z, and C_z are respectively the number of sentences in segment s and the number of words and constituents in sentence z, $\mathcal{I}(\zeta, w)$ is a binary function returning 1 if ζ selects word w as the indicator, and $\mathcal{I}(\zeta, x)$ returns 1 if ζ selects x as the argument. Note that we apply the same Taylor series approximation introduced by the $\hat{\theta}$ update.

We then apply L-BFGS-B to the optimization program of equation 3.10 to find the optimal dual parameters κ^*. These values are fed into equation 3.11 to produce the final constrained update.

3.4 Declarative Constraints

The previous section provides us the machinery to incorporate a variety of declarative constraints during inference. The classes of domain-independent constraints we study are summarized in Table 3.1. We will later also introduce domain-specific constraints

	Quantity	$f(\mathbf{s}, \mathbf{z}, \mathbf{i}, \mathbf{a})$	$\leq$ or $\geq$	b
Syntax	$\forall k$	Counts i, a of relation k that match a pattern	$\geq$	$0.8D$
Prevalence	$\forall k$	Counts instances of relation k	$\geq$	$0.8D$
Separation (ind)	$\forall w$	Counts times token w selected as i	$\leq$	2
Separation (arg)	$\forall w$	Counts times token w selected as part of a	$\leq$	1

Table 3.1: Summary of constraints we consider for the relation discovery model. Each constraint takes the form $\mathbb{E}_q[f(z, a, i)] \leq b$ or $\mathbb{E}_q[f(z, a, i)] \geq b$ as indicated in the table; D denotes the number of corpus documents, $\forall k$ means one constraint per relation type, and $\forall w$ means one constraint per token in the corpus.

relevant to the individual experiment datasets. For the syntactic and prevalence constraints, both of which require a proportion threshold, we arbitrarily select 80%, foregoing specific tuning in the spirit of building a domain-independent approach.

3.4.1 Syntax

As previous work has observed, most relations are expressed using a limited number of common syntactic patterns [6, 115]. Our syntactic constraint captures this insight by requiring that a certain proportion of the induced instances for each relation match one of these syntactic patterns:

- The indicator is a verb and the argument's headword is either the child or grandchild of the indicator word in the dependency tree.
- The indicator is a noun and the argument is a modifier or complement.
- The indicator is a noun in a verb's subject and the argument is contained within the corresponding object.

Note that these patterns are very generic, since they should be generally applicable across different domains. In domains where specialist knowledge is available, they can be refined to capture more appropriate patterns.

Encoding the syntactic bias as a threshold-based soft constraint is preferable to using a hard constraint for two reasons. First, our patterns do not necessarily cover all potential relation syntactic patterns, such as coordinations ("*indicator* and *argument*"). We could include those additional rarer patterns in the pattern set, but doing

so would also likely introduce a disproportionate number of false positive instances. Second, most domains do not come with clean human-derived parses, so we will typically need to rely on automatic parsing to produce the input to our model. Using a soft constraint allows our model to be robust to errors induced by the automatic parser.

The syntactic constraint is a prime example of how posterior regularization enables the easy injection of linguistically-motivated declarative knowledge into the learning process. This constraint allows us to directly enforce intuitive domain-independent linguistic knowledge while still permitting predictions that do not follow the patterns when the data evidence is strong enough.

3.4.2 Prevalence

For a relation to be domain-relevant, it should occur in numerous documents across the corpus, so we institute a constraint on the minimum number of times a relation is instantiated. Note that the effect of this constraint could also be achieved by tuning the prior probability of a relation not occurring in a document. However, this prior would need to be adjusted every time the number of documents or feature selection changes; using a constraint is an appealing alternative that both allows for direct control over the space of predictions and is portable across domains.

3.4.3 Separation

The separation constraint encourages diversity in the discovered relation types by restricting the number of times a single word can serve as either an indicator or part of the argument of a relation instance. Specifically, we require that every token of the corpus occurs at most once as a word in a relation's argument in expectation. This constraint serves to discourage identifying a phrase such as "the island of Mindoro" as three separate relation arguments "the island," "Mindoro," and "the island of Mindoro." On the other hand, a single word can sometimes be evocative of multiple relations (*e.g.*, "occurred" signals both *date* and *time* in "occurred on Friday at

Corpus	Documents	Sentences	Tokens	Vocabulary	Token/type ratio
Finance	100	12.1	262.9	2918	9.0
Earthquake	200	9.3	210.3	3155	13.3

Table 3.2: Corpus statistics for the datasets used for the relation model experiments. Sentence and token counts are per-document averages.

3pm"). Thus, we allow each word to serve as an indicator more than once, arbitrarily fixing the limit at two.

3.5 Experiments

We now present experimental results of our model. We compare against unsupervised baselines in Section 3.5.2 and examine the importance of applying posterior constraints in Section 3.5.3. We also study how labeled instances impact performance, comparing against supervised baselines in Section 3.5.4.

3.5.1 Evaluation Setup

Datasets

We evaluate on two datasets, financial market reports and newswire articles about earthquakes, previously used in work on high-level content analysis [11, 80]. The finance articles chronicle daily market movements of currencies and stock indexes, and the earthquake articles document specific earthquakes. Constituent parses are obtained automatically using the Stanford parser [76]. Since some of our constraints are defined in terms of dependency relationships, we also apply the PennConvertor tool [70] to transform the constituent parses into dependency parses. Corpus statistics are summarized in Table 3.2.

We manually annotate relations for both corpora, selecting relation types that are relevant and prevalent in each domain. This yields 15 types for finance and nine for earthquakes. Table 3.3 describes each relation and provides example values from the corpora. Note that the true relations arguments do not necessarily fall on constituent boundaries, though even in those cases there is usually high overlap with

Finance

Bond	104.58 yen, 98.37 yen
Bond Change	down 0.06 yen, unchanged
Dollar	108.42 yen, 119.76 yen
Dollar Change	up 0.52 yen, down 0.01 yen
Dollar Previous	119.47 yen, 97.68 yen
Nikkei	17367.54, 19115.82
Nikkei Change	rose 32.64 points or 0.19 percent, fell 19.51 points or 0.10 percent
Nikkei Previous	19284.36, 16868.36
Nikkei Previous Change	up 19.51 points or 0.10 percent, gained 2.31 points or 0.013 percent
Tokyo Index	1321.22, 1508.10
Tokyo Index Change	down 5.38 points or 0.41 percent, up 0.16 points, insignificant in percentage terms
TOPIX	1321.22, 1517.67
TOPIX Change	fell 26.88 points or 1.99 percent, down 2.36 points or 0.16 percent
Yield	2.360 percent, 3.225 percent
Yield Change	slipped 26.88 points or 1.99 percent, unchanged

Earthquake

Casualties	eight people, at least 70 people, no reports
Damage	about 10000 homes, some buildings, no information
Date	Tuesday, Wednesday, Saturday
Duration	about one minute, about 90 seconds, a minute and a half
Epicenter	the Tumkin Valley in Buryatia, Patuca about 185 miles (300 kilometers) south of Quito, 110 kilometers (65 miles) from shore under the surface of the Flores sea in the Indonesian archipelago
Injuries	more than 500, about 1700, no immediate reports
Location	a remote area in the Altai mountains in northwestern Xinjiang, off Russia's eastern coast, the district of Nabire 3130 kilometers (1950 miles) northeast of Jakarta
Magnitude	5.7, 6, magnitude-4
Time	4:29 p.m., 12:44 a.m., about 6:46 am

Table 3.3: The manually annotated relation types identified in the finance and earthquake datasets with example instance arguments.

a noun phrase. Restricting our model to only constituents as arguments implies that it cannot achieve perfect extraction accuracy. However, this restriction reduces the search space of candidate relations to phrases that are likely to closely match true relation tokens, and hence is a beneficial tradeoff to make.

Domain-specific Constraints

On top of the cross-domain constraints from Section 3.4, we study whether imposing basic domain-specific constraints can be beneficial for performance. The finance dataset is heavily quantitative, so we consider applying a single domain-specific constraint stating that most relation arguments should include a number. Likewise, earthquake articles are typically written with a majority of the relevant information toward the beginning of the document, so its domain-specific constraint is that most relations should occur in the first two sentences of a document. Note that these domain-specific constraints are not specific to individual relations or instances, but rather encode a preference across all relation types. In both cases, we again use an 80% threshold without tuning.

Given a fixed amount of resources, injecting domain-specific knowledge via constraints can be much more cost-effective than producing supervised examples. The constraints we propose here are general tendencies that could be easily intuited directly or identified based on a cursory examination of the data. In contrast, using any sort of supervised method would require at least one annotation per target relation type. Providing such annotations necessitates a much more detailed understanding of the important relation types in a domain, and full analysis of at least one entire document.

Metrics

We measure the quality of the induced relations by comparing them to the manually annotated relation sets. The obvious way to evaluate is by measuring token-level accuracy. In our task, however, annotation conventions for desired output relations can greatly impact token-level performance, and the model cannot learn to fit a

particular convention by looking at example data. For example, earthquakes times are frequently reported in both local and GMT, and depending on annotation convention either or both may be arbitrarily chosen as correct. Moreover, several of our baselines operate at the sentence rather than word level, so a direct token-level comparison would be infeasible.

For these reasons, we evaluate on both the *sentence level* and *token level*. Sentence-level evaluations are less prone to idiosyncrasies in annotation style, and are easily comparable across disparate techniques.

The specific scores we compute are sentence- and word-level precision, recall, and F-score. Precision is measured by mapping every induced relation cluster to the gold relation with the highest overlap, then computing the proportion of predicted sentences or words that are correct. Conversely, for recall we map every gold relation to its highest-overlap predicted relation and find the proportion of gold sentences or words that are predicted. High precision implies that our discovered relation types are internally highly coherent, while high recall means that the relations we predict provide broad coverage of the true relations. This mapping technique is similar at a high level to the one we use to evaluate our content model from Chapter 2, and is again based on the many-to-one scheme used for evaluating unsupervised part-of-speech induction [71].

Features

For indicators, we use the word, part of speech, and word stem as features. For arguments, we use the word, syntactic constituent label, the head word of the parent constituent, and the dependency label of the argument to its parent. Numeric words are mapped onto the same feature value, as are different casings of a single word type. In general, we expect that richer domain-specific features could also improve relation extraction performance.

Training Regimes and Hyperparameters

For each run of our model we perform three random restarts to convergence and select the posterior with lowest final KL-divergence objective value (equation 3.3). We fix K to the true number of annotated relation types for each run of our model and L (the number of document segments) to five. Dirichlet hyperparameters are set to 0.1. These values were not tuned since we do not assume access to a separate development set.

3.5.2 Comparison against Unsupervised Baselines

Our first set of results compare against a series of alternative unsupervised approaches:

- *Clustering* (CLUTO): A straightforward way of identifying candidate groups of sentences bearing the same relation, though not exact relation text spans, is to simply cluster the sentences. We implement a clustering baseline using the CLUTO toolkit with word and part-of-speech features, and with configuration settings identical to the experimental setup for the content model of Chapter 2. As we show in those experiments, clustering delivers competitive performance for finding topically-grouped discourse units; hence, they appear to be a promising approach for finding relation-bearing sentence clusters. As with our model, we set the number of clusters to the true number of relation types.
- *Mallows Content Model* (Mallows): Another baseline is to use the sentence clustering output by the Mallows-based content model of Chapter 2. The datasets we consider here exhibit high-level regularities in content organization, so we may expect that the globally-informed content model could identify plausible clusters of sentence-bearing relations. We set the number of topics K to the true number of relation types.
- *Unsupervised Semantic Parsing* (USP): Our final unsupervised comparison is to USP, an unsupervised deep semantic parser proposed by Poon and Domingos [104]. USP induces a hierarchical lambda calculus representation of an entire corpus. It was shown to outperform previous open information extraction

and unsupervised paraphrase discovery approaches [5, 83]. We use the publicly available implementation of USP,[9] and provide it the required Stanford dependency format as input [42]. Note that this dependency format is richer than the syntactic and dependency parses we use for our model, potentially giving USP an advantage.

USP's output is represented as lambda calculus formulas; to map these to relation spans, we first group lambda forms by a combination of core form, argument form, and the parent's core form.[10] We then filter to the K relations that appear in the most documents, where K is the true number of relation types. For token-level evaluations, we take the dependency tree fragment corresponding to the argument form. For example, in the sentence "a strong earthquake rocked the Philippines island of Mindoro early Tuesday," USP learns that the word "Tuesday" has a core form corresponding to the cluster of words {*Tuesday*, *Wednesday*, *Saturday*}, a parent form corresponding to the cluster of words {*shook*, *rock*, *hit*, *jolt*}, and an argument form (dependency edge label) of TMOD; all phrases with this same combination are grouped as a relation.

Note the first two baselines above only predict the sentences of relation types, whereas the last also predicts word spans within the sentences.

Results

Lines 1 through 5 of Table 3.4 present the results of our main evaluation. Using only domain-independent constraints, our model substantially outperforms all baselines on the earthquake dataset. For the market dataset, our base model is comparable in sentence-level F-score to the CLUTO baseline and outperforms all other baselines. Note however that this CLUTO baseline delivers the *worst* performance of all approaches on the earthquake dataset, and moreover does not provide any mechanism for identifying the appropriate tokens comprising the relation instance within the sentence. Thus, these results support our hypothesis that our relation discovery model

[9] http://alchemy.cs.washington.edu/papers/poon09/

[10] This grouping mechanism yields better average results than only grouping by core form.

		Finance					
		Sentence-level			Token-level		
		Prec	Recall	F-score	Prec	Recall	F-score
1	Model	82.1	59.7	69.2	42.2	23.9	30.5
2	Model+DSC	87.3	81.6	**84.4**	51.8	30.0	**38.0**
3	CLUTO	56.3	92.7	70.0	—	—	—
4	Mallows	40.4	99.3	57.5	—	—	—
5	USP	81.1	51.7	63.1	15.1	26.7	19.3
6	No-sep	97.8	35.4	52.0	86.1	8.7	15.9
7	No-syn	83.3	46.1	59.3	20.8	9.9	13.4
8	Hard-syn	47.7	39.0	42.9	11.6	7.0	8.7

		Earthquake					
		Sentence-level			Token-level		
		Prec	Recall	F-score	Prec	Recall	F-score
1	Model	54.2	68.1	60.4	20.2	16.8	18.3
2	Model+DSC	66.4	65.6	**66.0**	22.6	23.1	**22.8**
3	CLUTO	19.8	58.0	29.5	—	—	—
4	Mallows	18.6	74.6	29.7	—	—	—
5	USP	42.5	34.8	38.2	8.3	25.6	12.6
6	No-sep	42.2	21.9	28.8	16.1	4.6	7.1
7	No-syn	53.8	60.9	57.1	14.0	13.8	13.9
8	Hard-syn	55.0	66.2	60.1	20.1	17.3	18.6

Table 3.4: Comparison of our relation discovery model, with and without domain-specific constraints (DSC), to a series of unsupervised baselines and model variants on both domains. Lines are numbered for ease of reference in the text. For all scores higher is better.

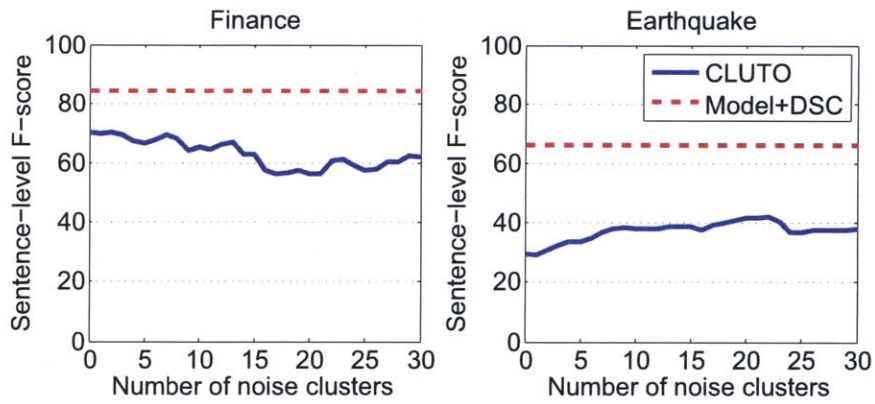


Figure 3-6: The F-score of the CLUTO clustering baseline as additional noise clusters are added compared to the relation discovery model's performance.

using a single set of constraints can be applicable across domains.

We also find that introducing simple domain-specific constraints yields a strong additional performance benefit for both datasets and metrics. In particular, our model with such constraints substantially outperforms the CLUTO baseline on the finance dataset. This result supports our claim that the posterior regularization approach provides a simple and effective framework for injecting low-cost domain knowledge into relation discovery. For the baselines there is no simple way of leveraging these kinds of additional constraints.

The USP baseline delivers reasonable sentence-level performance on both datasets, but gives poor token-level performance. This result speaks to the inherent difficulty in identifying precise token-level boundaries for relation instances, considering that USP is provided with richer Stanford-format dependency parses as input.

The CLUTO and Mallows baselines yield very skewed precision/recall tradeoffs, tending to favor recall greatly over precision. This is primarily because clustering requires that every sentence, including those without relations, is assigned to a relation cluster, thus lowering precision. In an effort to ameliorate this over-prediction effect, we experiment with a variant of the CLUTO baseline where we first cluster sentences into $K + N$ groups, then predict the largest K of those clusters as the relation sentences. These additional N "noise" clusters should absorb the relation-less sentences at the cost of recall. Figure 3-6 presents F-score as a function of the number

of additional noise clusters for both domains. We find that earthquakes benefits from noise clusters up to a certain point, though even with an optimal 22 noise clusters its 42.2 F-score still greatly lags behind our model's 66.0 F-score. For finance, noise clusters are not useful, leaving the CLUTO results unchanged or making them worse.

There is a large gap in F-score between the sentence- and token-level evaluations for our model. A qualitative error analysis of the predictions indicates that our model often picks up on regularities that are difficult to distinguish without relation-specific supervision. For instance, in the earthquake domain a *location* may be annotated as "the Philippine island of Mindoro" while we predict just the word "Mindoro." Additionally, indicators and arguments are sometimes switched—in the frequently-occurring phrase "no reports of damage," "reports" is often chosen as the argument and "damage" as the indicator, a distinction that is difficult to make from unlabeled data. For finance, a *Nikkei change* can be annotated as "rose 32.64 points or 0.19 percent," while our model identifies "32.64 points" and "0.9 percent" as separate relations. In practice, these outputs are all plausible discoveries, and a practitioner desiring specific outputs could impose additional domain-specific constraints to guide relation discovery toward them.

Additionally, we note the difference in results between the two domains. This gap is due to the much more formulaic nature of the finance domain. As Table 3.3 demonstrates, the relations in that domain tend to exhibit much less variability in verbalization than earthquakes; additionally, they tend to occur in more similar contexts. This greater regularity for finance also benefits the unsupervised baselines, allowing them to deliver stronger relative performance to our model than for earthquakes. In the earthquakes case, learning from multiple layers of regularity simultaneously, coupled with the declarative constraints, is crucial to achieving reasonable results.

3.5.3 Constraint Ablation Analysis

To understand the impact of the declarative constraints, we perform an ablation analysis on the constraint sets. We experiment with the following model variants:

- *No-sep*: Removes the separation constraints, which biases against overlapping relations, from the constraint set.
- *No-syn*: Removes the syntactic constraints, which biases toward plausible syntactic patterns, from the constraint set.
- *Hard-syn*: Makes the syntactic constraint hard, *i.e.*, requires that *every* extraction match one of the syntactic patterns specified by the syntactic constraint.

We perform this ablation analysis starting from the domain-independent constraint set. Prevalence constraints are always enforced, as otherwise the prior on not instantiating a relation would need to be tuned.

Results

Lines 6 through 8 of Table 3.4 present the results of this ablation evaluation. The model's performance degrades when either of the two constraint sets are removed, demonstrating that the constraints are in fact beneficial for relation discovery. In particular, note the *no-sep* case for the finance domain—the near-perfect precision but dramatically reduced recall indicates that this variant is discovering one single correct relation over and over again for each hidden relation.

In the *hard-syn* case, performance drops dramatically for the finance dataset while remaining almost unchanged for earthquakes. This suggests that formulating constraints as soft inequalities on posterior expectations gives our model the flexibility to accommodate both the underlying signal in the data and the declarative constraints.

3.5.4 Comparison against Supervised Approaches

Our final set of experiments address the scenario when a small amount of training data is available. For these evaluations we study a *semi-supervised* version of our model. Because our model is formalized in a generative manner, incorporating training examples is straightforward. The provided annotated relation instances are simply encoded as observed variables, and predictions are made transductively on the remaining unlabeled documents. In cases when the training instance's argument does

not fall on constituent boundaries, we adjust the training instance to the closest valid constituent. We modify the syntactic and prevalence constraint thresholds to 80% of the predictions on *unlabeled* documents. In case the training examples violate one of the separation (non-overlap) constraints at a given word, we increase the constraint threshold for that single word to remove the violation. We run our model both with and without domain-specific constraints (DSC).

We compare to two supervised baselines:

- *Conditional Random Field* (CRF): We train a conditional random field model [78] using the same features as our model's argument features.[11] CRFs are a very well-established discriminative sequence model for supervised information extraction. The CRF is trained with a corpus tagged with relation instances at the token level. During test, the CRF predicts a labeling of relation instances over the tokens of the test document. Unlike our model, the CRF ignores syntax except through the feature representation. Consequently, it is not limited to constituent boundaries for predicting relations, but also does not benefit from syntactic constraints or document-level regularity in relation expression.
- *Sentence-level Support Vector Machine* (SVM): As an analogue to the unsupervised CLUTO and Mallows sentence clustering models, we explore how well an SVM can predict which sentences contain a given relation. We train a separate binary classifier using SVM^{light} [69] for each relation type using word and part-of-speech tag features. Positive examples are the sentences in the training data with an instance of the relation, negative examples are all other sentences. This baseline does not predict the exact words of the relation within the sentence.

For both the baselines and our model, we experiment with using up to 10 annotated documents in both domains. At each of those levels of supervision, we average results over 10 randomly drawn training sets.

[11] Our model uses features defined on constituents, so the CRF uses equivalent features defined at the token level. We also tried using additional features based on local context but found that they do not appreciably affect performance for these datasets under the given training regimes.

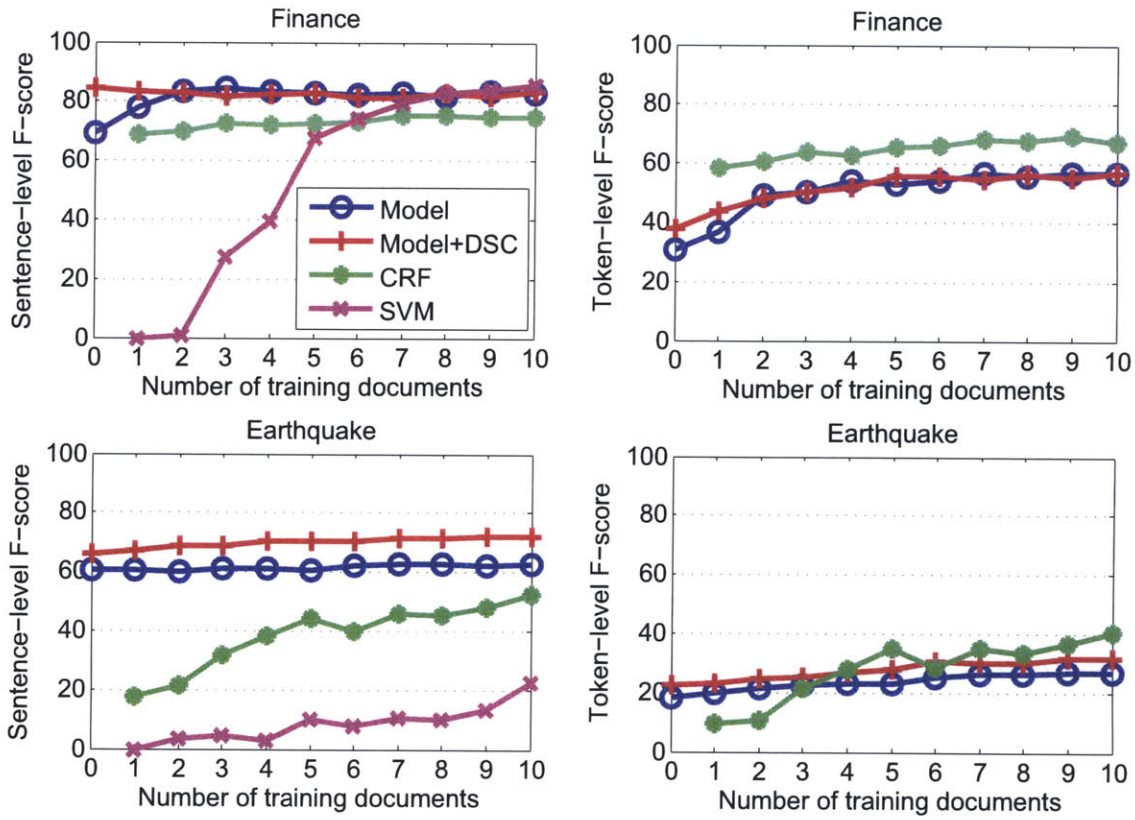


Figure 3-7: Comparison of the semi-supervised variant of our relation discovery model to two supervised baselines, a CRF sequence model and an SVM trained on sentences, on both datasets (top vs. bottom) and metrics (left vs. right). The x-axis measures the number of labeled documents provided to each system. Token-level performance for the SVM is not reported since the SVM does not predict individual relation tokens.

Results

Figure 3-7 depicts F-score of our model and the baselines as a function of the number of annotated documents for both domains and metrics.

At the sentence level, we find that our model, with and without domain-specific constraints, compares very favorably to the supervised baselines. For finance, it takes eight annotated documents (corresponding to roughly 100 individually annotated relation instances) for the SVM to match the semi-supervised model's performance, and at least ten documents for the CRF. For earthquake, using even ten annotated documents (corresponding to about 71 relation instances) is not sufficient for either the CRF or SVM to match our model's performance.

At the token level, the supervised CRF baseline is far more competitive. Using a single labeled document (13 relation instances) yields superior performance to either of our model variants for finance, while four labeled documents (29 relation instances) does the same for earthquakes. This result is not surprising—our model is designed first and foremost as an unsupervised model, and as such makes strong domain-independent assumptions about how underlying patterns of regularities in the text connect to relation expression. Without domain-specific supervision such assumptions are necessary, but they can prevent the model from fully utilizing available labeled instances.

For all of these evaluations we find that labeled data does not yield as great an improvement in our model as compared to the supervised baselines. This is due to multiple reasons. First, our model must build upon already strong unsupervised results, particularly at the sentence level, and simply has less room to improve. Second, at the token level we make a strong assumption that relation instances align precisely to phrase constituents; while this assumption benefits the unsupervised setting by reducing the search space, it effectively imposes an upper bound on performance in the presence of supervised data breaking this assumption. Third, we find that for finance, instance supervision benefits the model with domain-specific constraints more than the model without—in fact, just two labeled documents closes the gap

in performance between the two variants. It appears that the domain-specific constraint already captures much of the information that could be gleaned from labeled instances, thus obviating their usefulness. Finally, our model is framed generatively with a strong Naïve Bayes independence assumption over features. While such an assumption allows for tractable learning in the unsupervised setting, when training data is available a discriminative model can benefit more deeply from the feature values.

Overall, these results indicate that relatively few labeled instances are sometimes sufficient for a supervised method to outperform its unsupervised counterpart at the token level. As previous work has recognized for dependency parsing, however, the bulk of the labor in constructing annotated instances is in the initial phases of defining appropriate labeling guidelines and training annotators [43]. A similar phenomenon holds for extraction; being able to annotate even one document requires a broad understanding of every relation type germane to the domain, which can be infeasible when there are many unfamiliar, complex domains to process. In light of the supervised results, and our much stronger relative performance at the sentence-level than the token-level, we suggest a human-assisted relation discovery application: use our model to identify promising relation-bearing sentences in a new domain, then have a human annotate those sentences for use by a supervised approach to achieve optimal token-level extraction.

3.6 Conclusions and Future Work

This chapter presented a constraint-based approach to in-domain relation discovery. We have shown that a generative model augmented with declarative constraints on the model posterior can successfully identify domain-relevant relations and their instances. Furthermore, we found that a single set of constraints can be used across divergent domains, and that tailoring constraints specific to a domain can yield further performance benefits.

We see multiple avenues of future work. From the modeling perspective, our gen-

erative process could be enriched in several ways. First, it makes a strong Naïve Bayes assumption between features. Recent work has demonstrated that many unsupervised NLP models can instead benefit from a locally normalized log-linear feature generation process [16]. Incorporating such a representation in our model could potentially allow for richer interdependent features to be used. Second, our document-level location component is rather simple, and could potentially be enriched with an HMM or Mallows content model. Previous work has shown that rich content models are indeed beneficial for text analysis tasks such as extraction, at least in the supervised scenario [119]. Third, indicators and arguments are treated almost symmetrically by the generative process, distinguished only by feature selection and limitations in kinds of word spans. This symmetry leads to occasional situations where indicators and arguments are identified in reverse. In fact, the data typically exhibits much stronger regularity (*e.g.*, more limited word choice) in indicators than in arguments, and a more intelligent model could use this to help differentiate the relation roles more clearly.

From an application perspective, we have focused on datasets that exhibit rich syntactic structure in the form of complete prose. A large body of previous information extraction work focuses instead on semi-structured text, where the text already resembles a database record, such as paper citations, classified ads listings, seminar announcements, and contact information [15, 49, 78, 102]. The kind of syntactic constraints we used here would be inappropriate for such texts, but we believe a similar constraint-based strategy could also be successful. This would require developing new constraints targeting domain-independent properties of semi-structured text.

Finally, the semi-supervised variant of our model could likely be refined. We found that the impact of adding more labeled documents yielded relatively minor performance improvements. However, it is likely that such training examples could be incorporated in a more intelligent way. For example, a few labeled instances of a single relation are likely to cover a wide gamut of its possible indicator words. In the spirit of the labeled features supervision setup [89], we could encode this knowledge as a constraint on the relation's indicator words rather than simply as

observed variables in the model. Developing a domain-independent mechanism for incorporating supervision using constraints is a promising area of future work.

Chapter 4

Learning Semantic Properties using Free-text Annotations

In this chapter, we propose a technique for inducing document-level semantic properties implied by a text. For example, given the text of a restaurant review, we desire to extract a semantic-level characterization of the author's reaction to specific aspects of the restaurant, such as food and service quality (see Figure 4-1). The work in this chapter identifies clusters of *entire documents* that share semantic characteristics, and thus induce structure at a higher level than the phrases of the relation model and the paragraphs of the content model from previous chapters. Learning-based approaches have dramatically increased the scope and robustness of this kind of high-level semantic processing, but they are typically dependent on large expert-annotated datasets, which are costly to produce [137].

We propose to use an alternative source of annotations for learning: free-text keyphrases produced by novice users. As an example, consider the lists of pros and cons that often accompany reviews of products and services. Such end-user annotations are increasingly prevalent online, and they naturally grow to keep pace with subjects of interest and socio-cultural trends. Beyond such pragmatic considerations, free-text annotations are appealing from a linguistic standpoint because they capture the intuitive semantic judgments of non-specialist language users. In many real-world datasets, these annotations are created by the document's original author, providing

pros/cons: *great nutritional value* ... combines it all: an amazing product, quick and friendly service, cleanliness, great nutrition ...
pros/cons: *a bit pricey, healthy* ... is an awesome place to go if you are health conscious. They have some really great low calorie dishes and they publish the calories and fat grams per serving.

Figure 4-1: Excerpts from online restaurant reviews with pros/cons phrase lists. Both reviews assert that the restaurant serves healthy food, but use different keyphrases. Additionally, the first review discusses the restaurant's good service, but is not annotated as such in its keyphrases.

a direct window into the semantic judgments that motivated the document text.

The major obstacle to the computational use of such free-text annotations is that they are inherently noisy — there is no fixed vocabulary, no explicit relationship between annotation keyphrases, and no guarantee that all relevant semantic properties of a document will be annotated. For example, in the pros/cons annotations accompanying the restaurant reviews in Figure 4-1, the same underlying semantic idea is expressed in different ways through the keyphrases "great nutritional value" and "healthy." Additionally, the first review discusses quality of service, but is not annotated as such. In contrast, expert annotations would replace synonymous keyphrases with a single canonical label, and would fully label all semantic properties described in the text. Such expert annotations are typically used in supervised learning methods. As we will demonstrate with our results, traditional supervised approaches perform poorly when free-text annotations are used instead of clean, expert annotations.

This chapter demonstrates a new approach for handling free-text annotation in the context of a hidden-topic analysis of the document text. We show that regularities in document word choice can clarify noise in the annotations, and vice versa. For example, although "great nutritional value" and "healthy" have different surface forms, the text in documents that are annotated by these two keyphrases will likely be similar. Conversely, the same phrase "delicious food" annotated on two different documents allows us to predict the same property for each, even if the document texts are very divergent. By modeling the relationship between document text and

annotations over a large dataset, it is possible to induce a clustering over the annotation keyphrases that can help to overcome the problem of inconsistency. Our model also addresses the problem of incompleteness — when novice annotators fail to label relevant semantic topics — by estimating which topics are predicted by the document text alone.

Central to this approach is the idea that both document text and the associated annotations reflect a single underlying set of semantic properties. In the text, the semantic properties correspond to the induced hidden topics — this is similar to the growing body of work on latent topic models, such as latent Dirichlet allocation (LDA) [23]. However, unlike existing work on topic modeling, we tie hidden topics in the text with clusters of observed keyphrases. By modeling these phenomena jointly, we ensure that the inferred hidden topics are semantically meaningful, and that the clustering over free-text annotations is robust to noise.

Our approach takes the form of a hierarchical Bayesian framework, and includes an LDA-style component in which each word in the text is generated from a mixture of multinomials. In addition, we also incorporate a similarity matrix across the universe of annotation keyphrases, which is constructed based on the orthographic and distributional features of the keyphrases. We model this matrix as being generated from an underlying clustering over the keyphrases, such that keyphrases that are clustered together are likely to produce high similarity scores. To generate the words in each document, we model two distributions over semantic properties — one governed by the annotation keyphrases and their clusters, and a background distribution to cover properties not mentioned in the annotations. The latent topic for each word is drawn from a mixture of these two distributions. After learning model parameters from a noisily-labeled training set, we can apply the model to unlabeled data.

We build a system that extracts semantic properties from reviews of products and services. This system uses as training corpus that includes user-created free-text annotations of the pros and cons in each review. Training yields two outputs: a clustering of keyphrases into semantic properties, and a topic model that is capable of inducing the semantic properties of unlabeled text. The clustering of annotation

keyphrases is relevant for applications such as content-based information retrieval, allowing users to retrieve documents with semantically relevant annotations even if their surface forms differ from the query term. The topic model can be used to infer the semantic properties of unlabeled text.

The topic model can also be used to perform multi-document summarization, capturing the key semantic properties of multiple reviews. Unlike traditional extraction-based approaches to multi-document summarization, our induced topic model abstracts the text of each review into a representation capturing the relevant semantic properties. This enables comparison between reviews even when they use superficially different terminology to describe the same set of semantic properties. This idea is implemented in a review aggregation system that extracts the majority sentiment of multiple reviewers for each product or service. An example of the output produced by this system is shown in Figure 4-6. This system is applied to reviews in 480 product categories, allowing users to navigate the semantic properties of 49,490 products based on a total of 522,879 reviews. The effectiveness of our approach is confirmed by several evaluations.

For the summarization of both single and multiple documents, we compare the properties inferred by our model with expert annotations. Our approach yields substantially better results than alternatives from the research literature; in particular, we find that learning a clustering of free-text annotation keyphrases is essential to extracting meaningful semantic properties from our dataset. In addition, we compare the induced clustering with a gold standard clustering produced by expert annotators. The comparison shows that tying the clustering to the hidden topic model substantially improves its quality, and that the clustering induced by our system coheres well with the clustering produced by expert annotators.

The remainder of the chapter is structured as follows. Section 4.1 compares our approach with previous work on topic modeling, semantic property extraction, and multi-document summarization. Section 4.2 describes the properties of free-text annotations that motivate our approach. The model itself is described in Section 4.3, and a method for parameter estimation is presented in Section 4.4. Section 4.5 de-

scribes the implementation and evaluation of single-document and multi-document summarization systems using these techniques. We summarize our contributions and consider directions for future work in Section 4.8.

4.1 Related Work

The material presented in this section covers three lines of related work. First, we discuss work on Bayesian topic modeling that is related to our technique for learning from free-text annotations. Next, we discuss state-of-the-art methods for identifying and analyzing product properties from the review text. Finally, we situate our summarization work in the landscape of prior research on multi-document summarization.

4.1.1 Bayesian Topic Modeling

Recent work in the topic modeling literature has demonstrated that semantically salient topics can be inferred in an unsupervised fashion by constructing a generative Bayesian model of the document text. One notable example of this line of research is Latent Dirichlet Allocation (LDA) [23]. In the LDA framework, semantic topics are equated to latent distributions of words in a text; thus, each document is modeled as a mixture of topics. This class of models has been used for a variety of language processing tasks including topic segmentation [108], named-entity resolution [18], sentiment ranking [126], and word sense disambiguation [25].

Our method is similar to LDA in that it assigns latent topic indicators to each word in the dataset, and models documents as mixtures of topics. However, the LDA model does not provide a method for linking the latent topics to external observed representations of the properties of interest. In contrast, our model exploits the free-text annotations in our dataset to ensure that the induced topics correspond to semantically meaningful properties.

Combining topics induced by LDA with external supervision was first considered by Blei and McAuliffe [21] in their supervised Latent Dirichlet Allocation (sLDA) model. The induction of the hidden topics is driven by annotated examples provided

during the training stage. From the perspective of supervised learning, this approach succeeds because the hidden topics mediate between document annotations and lexical features. Blei and McAuliffe describe a variational expectation-maximization procedure for approximate maximum-likelihood estimation of the model's parameters. When tested on two polarity assessment tasks, sLDA shows improvement over a model in which topics where induced by an unsupervised model and then added as features to a supervised model.

The key difference between our model and sLDA is that we do not assume access to clean supervision data during training. Since the annotations provided to our algorithm are free-text in nature, they are incomplete and fraught with inconsistency. This substantial difference in input structure motivates the need for a model that simultaneously induces the hidden structure in free-text annotations and learns to predict properties from text.

4.1.2 Property Assessment for Review Analysis

Our model is applied to the task of review analysis. Traditionally, the task of identifying the properties of a product from review texts has been cast as an extraction problem [67, 84, 105]. For example, Hu and Liu [67] employ association mining to identify noun phrases that express key portions of product reviews. The polarity of the extracted phrases is determined using a seed set of adjectives expanded via WordNet relations. A summary of a review is produced by extracting all property phrases present verbatim in the document.

Property extraction was further refined in OPINE [105], another system for review analysis. OPINE employs a novel information extraction method to identify noun phrases that could potentially express the salient properties of reviewed products; these candidates are then pruned using WordNet and morphological cues. Opinion phrases are identified using a set of hand-crafted rules applied to syntactic dependencies extracted from the input document. The semantic orientation of properties is computed using a relaxation labeling method that finds the optimal assignment of polarity labels given a set of local constraints. Empirical results demonstrate that

OPINE outperforms Hu and Liu's system in both opinion extraction and in identifying the polarity of opinion words.

These two feature extraction methods are informed by human knowledge about the way opinions are typically expressed in reviews: for Hu and Liu [67], human knowledge is encoded using WordNet and the seed adjectives; for Popescu et al. [105], opinion phrases are extracted via hand-crafted rules. An alternative approach is to learn the rules for feature extraction from annotated data. To this end, property identification can be modeled in a classification framework [74]. A classifier is trained using a corpus in which free-text pro and con keyphrases are specified by the review authors. These keyphrases are compared against sentences in the review text; sentences that exhibit high word overlap with previously identified phrases are marked as pros or cons according to the phrase polarity. The rest of the sentences are marked as negative examples.

Clearly, the accuracy of the resulting classifier depends on the quality of the automatically induced annotations. Our analysis of free-text annotations in several domains shows that automatically mapping from even manually-extracted annotation keyphrases to a document text is a difficult task, due to variability in keyphrase surface realizations (see Section 4.2). As we argue in the rest of this chapter, it is beneficial to explicitly address the difficulties inherent in free-text annotations. To this end, our work is distinguished in two significant ways from the property extraction methods described above. First, we are able to predict properties beyond those that appear verbatim in the text. Second, our approach also learns the semantic relationships between different keyphrases, allowing us to draw direct comparisons between reviews even when the semantic ideas are expressed using different surface forms.

Working in the related domain of web opinion mining, Lu and Zhai [85] describe a system that generates *integrated opinion summaries*, which incorporate expert-written articles (*e.g.,* a review from an online magazine) and user-generated "ordinary" opinion snippets (*e.g.,* mentions in blogs). Specifically, the expert article is assumed to be structured into segments, and a collection of representative ordinary opinions is aligned to each segment. Probabilistic Latent Semantic Analysis (PLSA)

is used to induce a clustering of opinion snippets, where each cluster is attached to one of the expert article segments. Some clusters may also be unaligned to any segment, indicating opinions that are entirely unexpressed in the expert article. Ultimately, the integrated opinion summary is this combination of a single expert article with multiple user-generated opinion snippets that confirm or supplement specific segments of the review.

Our work's final goal is different — we aim to provide a highly compact summary of a multitude of user opinions by identifying the underlying semantic properties, rather than supplementing a single expert article with user opinions. We specifically leverage annotations that users already provide in their reviews, thus obviating the need for an expert article as a template for opinion integration. Consequently, our approach is more suitable for the goal of producing concise keyphrase summarizations of user reviews, particularly when no review can be taken as authoritative.

The work closest in methodology to our approach is a review summarizer developed by Titov and McDonald [125]. Their method summarizes a review by selecting a list of phrases that express writers' opinions in a set of predefined properties (*e.g.*,, *food* and *ambiance* for restaurant reviews). The system has access to numerical ratings in the same set of properties, but there is no training set providing examples of appropriate keyphrases to extract. Similar to sLDA, their method uses the numerical ratings to bias the hidden topics towards the desired semantic properties. Phrases that are strongly associated with properties via hidden topics are extracted as part of a summary.

There are several important differences between our work and the summarization method of Titov and McDonald. Their method assumes a predefined set of properties and thus cannot capture properties outside of that set. Moreover, consistent numerical annotations are required for training, while our method emphasizes the use of free-text annotations. Finally, since Titov and McDonald's algorithm is extractive, it does not facilitate property comparison across multiple reviews.

4.1.3 Multi-document Summarization

Our approach also relates to a large body of work in multi-document summarization. Researchers have long noted that a central challenge of multi-document summarization is identifying redundant information over input documents [12, 32, 88, 110]. This task is of crucial significance because multi-document summarizers operate over related documents that describe the same facts multiple times. In fact, it is common to assume that repetition of information among related sources is an indicator of its importance [12, 99, 109]. Many of these algorithms first cluster sentences together, and then extract or generate sentence representatives for the clusters.

Identification of repeated information is equally central in our approach — our multi-document summarization method only selects properties that are stated by a plurality of users, thereby eliminating rare and/or erroneous opinions. The key difference between our algorithm and existing summarization systems is the method for identifying repeated expressions of a single semantic property. Since most of the existing work on multi-document summarization focuses on topic-independent newspaper articles, redundancy is identified via sentence comparison. For instance, Radev et al. [109] compare sentences using cosine similarity between corresponding word vectors. Alternatively, some methods compare sentences via alignment of their syntactic trees [12, 90]. Both string- and tree-based comparison algorithms are augmented with lexico-semantic knowledge using resources such as WordNet.

The approach described in this chapter does not perform comparisons at the sentence level. Instead, we first abstract reviews into a set of properties and then compare property overlap across different documents. This approach relates to domain-dependent approaches for text summarization [45, 110, 131]. These methods identify the relations between documents by comparing their abstract representations. In these cases, the abstract representation is constructed using off-the-shelf information extraction tools. A template specifying what types of information to select is crafted manually for a domain of interest. Moreover, the training of information extraction systems requires a corpus manually annotated with the relations of interest.

Property	Incompleteness			Inconsistency	
	Recall	Precision	F-score	Keyphrase Count	Top Keyphrase Coverage %
Good food	0.736	0.968	0.836	23	38.3
Good service	0.329	0.821	0.469	27	28.9
Good price	0.500	0.707	0.586	20	41.8
Bad food	0.516	0.762	0.615	16	23.7
Bad service	0.475	0.633	0.543	20	22.0
Bad price	0.690	0.645	0.667	15	30.6
Average	0.578	0.849	0.688	22.6	33.6

Table 4.1: Incompleteness and inconsistency in the restaurant domain for six prevalent semantic properties. The incompleteness figures are the recall, precision, and F-score of the author annotations (manually clustered into properties) against the gold standard property annotations. Inconsistency is measured by the number of different keyphrase realizations with at least five occurrences associated with each property, and the percentage frequency with which the most commonly occurring keyphrases is used to annotate a property. The averages in the bottom row are weighted according to frequency of property occurrence.

In contrast, our method does not require manual template specification or corpora annotated by experts. While the abstract representations that we induce are not as linguistically rich as extraction templates, they nevertheless enable us to perform in-depth comparisons across different reviews.

4.2 Analysis of Free-Text Keyphrase Annotations

In this section, we explore the characteristics of free-text annotations, aiming to quantify the degree of noise observed in this data. The results of this analysis motivate the development of the learning algorithm described in Section 4.3.

We perform this investigation in the domain of online restaurant reviews using documents downloaded from the popular Epinions[1] website. Users of this website evaluate products by providing both a textual description of their opinion, as well as concise lists of keyphrases (pros and cons) summarizing the review. Pros/cons keyphrases are an appealing source of annotations for online review texts. However, they are contributed independently by multiple users and are thus unlikely to be

[1] `http://www.epinions.com/`

as clean as expert annotations. In our analysis, we focus on two features of free-text annotations: *incompleteness* and *inconsistency*. The measure of incompleteness quantifies the degree of label omission in free-text annotations, while inconsistency reflects the variance of the keyphrase vocabulary used by various annotators.

To test the quality of these user-generated annotations, we compare them against "expert" annotations produced in a more systematic fashion. This annotation effort focused on six properties that were commonly mentioned by the review authors, specifically those shown in Table 4.1. Given a review and a property, the task is to assess whether the review's text supports the property. These annotations were produced by two judges guided by a standardized set of instructions. In contrast to author annotations from the website, the judges conferred during a training session to ensure consistency and completeness. The two judges collectively annotated 170 reviews, with 30 annotated by both. Cohen's Kappa, a measure of inter-annotator agreement that ranges from zero to one, is 0.78 on this joint set, indicating high agreement [39]. On average, each review text was annotated with 2.56 properties.

Separately, one of the judges also standardized the free-text pros/cons annotations for the same 170 reviews. Each review's keyphrases were matched to the same six properties. This standardization allows for direct comparison between the properties judged to be supported by a review's text and the properties described in the same review's free-text annotations. We find that many semantic properties that were judged to be present in the text were not user annotated — on average, the keyphrases expressed 1.66 relevant semantic properties per document, while the text expressed 2.56 properties. This gap demonstrates the frequency with which authors omitted relevant semantic properties from their review annotations.

4.2.1 Incompleteness

To measure incompleteness, we compare the properties stated by review authors in the form of pros and cons against those stated only in the review text, as judged by expert annotators. This comparison is performed using precision, recall and F-score. In this setting, recall is the proportion of semantic properties in the text for which

Property: *good price*
relatively inexpensive, dirt cheap, relatively cheap, great price, fairly priced, well priced, very reasonable prices, cheap prices, affordable prices, reasonable cost

Figure 4-2: Examples of the many different paraphrases related to the property *good price* that appear in the pros/cons keyphrases of reviews used for our inconsistency analysis.

the review author also provided at least one annotation keyphrase; precision is the proportion of keyphrases that conveyed properties judged to be supported by the text; and F-score is their harmonic mean. The results of the comparison are summarized in the left half of Table 4.1.

These incompleteness results demonstrate the significant discrepancy between user and expert annotations. As expected, recall is quite low; more than 40% of property occurrences are stated in the review text without being explicitly mentioned in the annotations. The precision scores indicate that the converse is also true, though to a lesser extent — some keyphrases will express properties not mentioned in text.

Interestingly, precision and recall vary greatly depending on the specific property. They are highest for *good food*, matching the intuitive notion that high food quality would be a key salient property of a restaurant, and thus more likely to be mentioned in both text and annotations. Conversely, the recall for *good service* is lower — for most users, high quality of service is apparently not a key point when summarizing a review with keyphrases.

4.2.2 Inconsistency

The lack of a unified annotation scheme in the restaurant review dataset is apparent — across all reviewers, the annotations feature 26,801 unique keyphrase surface forms over a set of 49,310 total keyphrase occurrences. Clearly, many unique keyphrases express the same semantic property — in Figure 4-2, *good price* is expressed in ten different ways. To quantify this phenomenon, the judges manually clustered a subset of the keyphrases associated with the six previously mentioned properties. Specifically, 121 keyphrases associated with the six major properties were chosen, accounting for

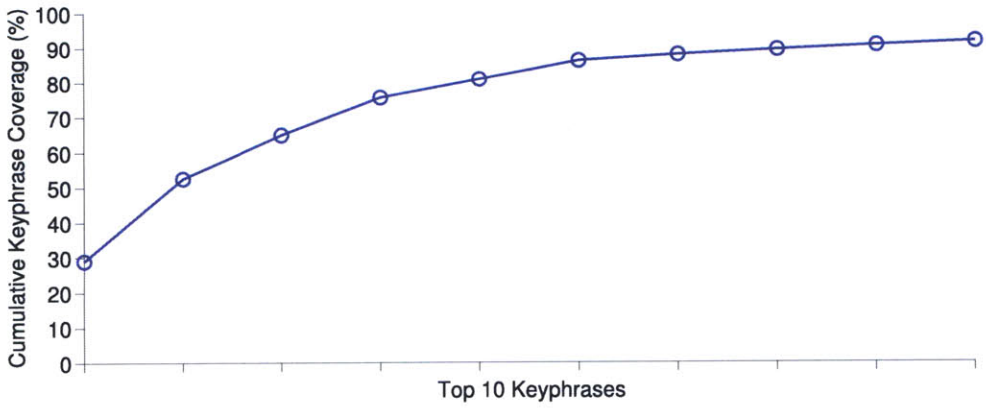


Figure 4-3: Cumulative occurrence counts for the top ten keyphrases associated with the *good service* property. The percentages are out of a total of 1,210 separate keyphrase occurrences for this property.

10.8% of all keyphrase occurrences.

We use these manually clustered annotations to examine the distributional pattern of keyphrases that describe the same underlying property, using two different statistics. First, the number of different keyphrases for each property gives a lower bound on the number of possible paraphrases. Second, we measure how often the most common keyphrase is used to annotate each property, *i.e.*, the *coverage* of that keyphrase. This metric gives a sense of how diffuse the keyphrases within a property are, and specifically whether one single keyphrase dominates occurrences of the property. Note that this value is an overestimate of the true coverage, since we are only considering a tenth of all keyphrase occurrences.

The right half of Table 4.1 summarizes the variability of property paraphrases. Observe that each property is associated with numerous paraphrases, all of which were found multiple times in the actual keyphrase set. Most importantly, the most frequent keyphrase accounted for only about a third of all property occurrences, strongly suggesting that targeting only these labels for learning is a very limited approach. To further illustrate this last point, consider the property of *good service*, whose keyphrase realizations' distributional histogram appears in Figure 4-3. The cumulative percentage frequencies of the most frequent keyphrases associated with

this property are plotted. The top four keyphrases here account for only three quarters of all property occurrences, even within the limited set of keyphrases we consider in this analysis, motivating the need for aggregate consideration of keyphrases.

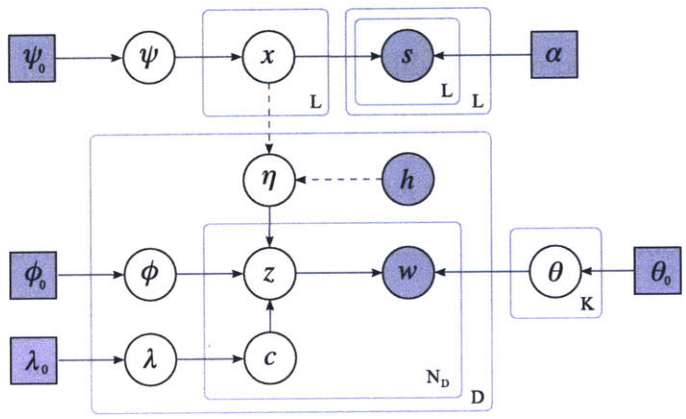
In the next section, we introduce a model that induces a clustering among keyphrases while relating keyphrase clusters to the text, directly addressing these characteristics of the data.

4.3 Model Description

We present a generative Bayesian model for documents annotated with free-text keyphrases. Our model assumes that each annotated document is generated from a set of underlying semantic *topics*. Semantic topics generate the document text by indexing a language model; in our approach, they are also associated with clusters of keyphrases. In this way, the model can be viewed as an extension of Latent Dirichlet Allocation [23], where the latent topics are additionally biased toward the keyphrases that appear in the training data. However, this coupling is flexible, as some words are permitted to be drawn from topics that are not represented by the keyphrase annotations. This permits the model to learn effectively in the presence of incomplete annotations, while still encouraging the keyphrase clustering to cohere with the topics supported by the document text.

Another critical aspect of our model is that we desire the ability to use arbitrary comparisons between keyphrases, in addition to information about their surface forms. To accommodate this goal, we do not treat the keyphrase surface forms as generated from the model. Rather, we acquire a real-valued similarity matrix across the universe of possible keyphrases, and treat this matrix as generated from the keyphrase clustering. This representation permits the use of surface and distributional features for keyphrase similarity, as described in Section 4.3.1.

An advantage of hierarchical Bayesian models is that it is easy to change which parts of the model are observed and which parts are hidden. During training, the keyphrase annotations are observed, so that the hidden semantic topics are coupled



ψ – keyphrase cluster model
x – keyphrase cluster assignment
s – keyphrase similarity values
h – document keyphrases
η – document keyphrase topics
λ – probability of selecting η instead of ϕ
c – selects between η and ϕ for word topics
ϕ – background word topic model
z – word topic assignment
θ – language models of each topic
w – document words

$$\psi \sim \text{Dirichlet}(\psi_0)$$
$$x_\ell \sim \text{Multinomial}(\psi)$$
$$s_{\ell,\ell'} \sim \begin{cases} \text{Beta}(\alpha_=) & \text{if } x_\ell = x_{\ell'} \\ \text{Beta}(\alpha_\neq) & \text{otherwise} \end{cases}$$
$$\eta_d = [\eta_{d,1} \ldots \eta_{d,K}]^{\text{T}}$$
$$\text{where } \eta_{d,k} \propto \begin{cases} 1 & \text{if } x_\ell = k \text{ for any } l \in h \\ \epsilon & \text{otherwise} \end{cases}$$
$$\lambda_d \sim \text{Beta}(\lambda_0)$$
$$c_{d,n} \sim \text{Bernoulli}(\lambda_d)$$
$$\phi_d \sim \text{Dirichlet}(\phi_0)$$
$$z_{d,n} \sim \begin{cases} \text{Multinomial}(\eta_d) & \text{if } c_{d,n} = 1 \\ \text{Multinomial}(\phi_d) & \text{otherwise} \end{cases}$$
$$\theta_k \sim \text{Dirichlet}(\theta_0)$$
$$w_{d,n} \sim \text{Multinomial}(\theta_{z_{d,n}})$$

Figure 4-4: The plate diagram for our semantic properties model. Shaded circles denote observed variables and squares denote hyperparameters. The dotted arrows indicate that η is constructed deterministically from x and h. We use ϵ to refer to a small constant probability mass.

with clusters of keyphrases. To account for words not related to semantic topics, some topics may not have any associated keyphrases. At test time, the model is presented with documents for which the keyphrase annotations are hidden. The model is evaluated on its ability to determine which keyphrases are applicable, based on the hidden topics present in the document text.

The judgment of whether a topic applies to a given unannotated document is based on the probability mass assigned to that topic in the document's background topic distribution. Because there are no annotations, the background topic distribution should capture the entirety of the document's topics. For the task involving reviews of products and services, multiple topics may accompany each document. In this case, each topic whose probability is above a threshold (tuned on the development set) is predicted as being supported.

4.3.1 Keyphrase Clustering

To handle the hidden paraphrase structure of the keyphrases, one component of the model estimates a clustering over keyphrases. The goal is to obtain clusters where each cluster correspond to a well-defined semantic topic — *e.g.*, both "healthy" and "good nutrition" should be grouped into a single cluster. Because our overall joint model is generative, a generative model for clustering could easily be integrated into the larger framework. Such an approach would treat all of the keyphrases in each cluster as being generated from a parametric distribution. However, this representation would not permit many powerful features for assessing the similarity of pairs of keyphrases, such as string overlap or keyphrase co-occurrence in a corpus [91].

For this reason, we represent each keyphrase as a real-valued vector rather than as its surface form. The vector for a given keyphrase includes the similarity scores with respect to every other observed keyphrase (the similarity scores are represented by s in Figure 4-4). We model these similarity scores as generated by the cluster memberships (represented by x in Figure 4-4). If two keyphrases are clustered together, their similarity score is generated from a distribution encouraging high similarity;

Lexical	The cosine similarity between the surface forms of two keyphrases, represented as word frequency vectors.
Co-occurrence	Each keyphrase is represented as a vector of co-occurrence values. This vector counts how many times other keyphrases appear in documents annotated with this keyphrase. For example, the similarity vector for "good food" may include an entry for "very tasty food," the value of which would be the number of documents annotated with "good food" that contain "very tasty food" in their text. The similarity between two keyphrases is then the cosine similarity of their co-occurrence vectors.

Table 4.2: The two sources of information used to compute the similarity matrix for our semantic properties model. The final similarity scores are linear combinations of these two values. Note that co-occurrence similarity contains second-order co-occurrence information.

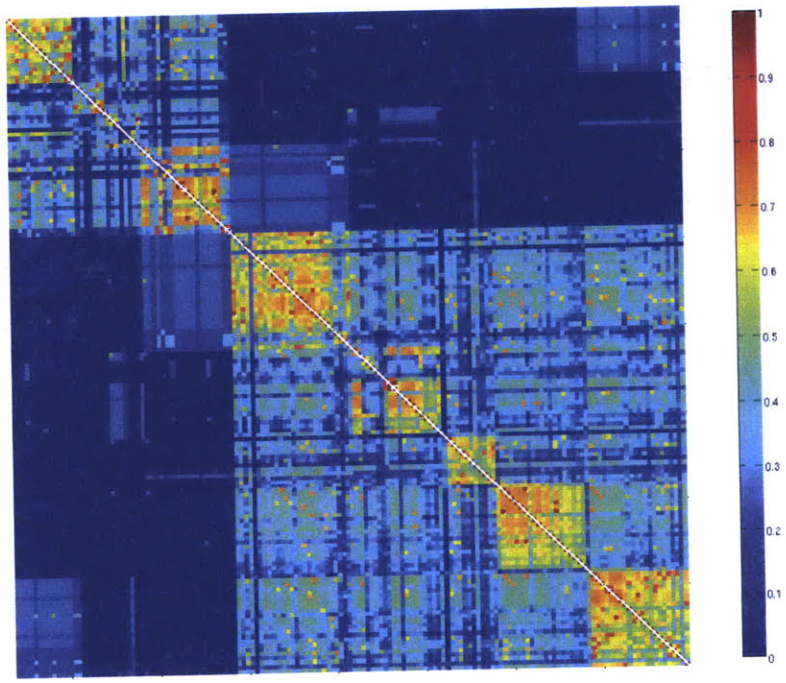


Figure 4-5: A surface plot of the keyphrase similarity matrix from a set of restaurant reviews, computed according to Table 4.2. Red indicates high similarity, whereas blue indicates low similarity. In this diagram, the keyphrases have been grouped according to an expert-created clustering, so keyphrases of similar meaning are close together. The strong series of similarity "blocks" along the diagonal hint at how this information could induce a reasonable clustering.

otherwise, a distribution encouraging low similarity is used.[2]

The features used for producing the similarity matrix are given in Table 4.2, encompassing lexical and distributional similarity measures. Our implemented system takes a linear combination of these two data sources, weighting both sources equally. The resulting similarity matrix for keyphrases from the restaurant domain is shown in Figure 4-5.

As described in the next section, when clustering keyphrases, our model takes advantage of the topic structure of documents annotated with those keyphrases, in addition to information about the individual keyphrases themselves. In this sense, it differs from traditional approaches for paraphrase identification [13, 83].

4.3.2 Document Topic Modeling

Our analysis of the document text is based on probabilistic topic models such as LDA [23]. In the LDA framework, each word is generated from a language model that is indexed by the word's topic assignment. Thus, rather than identifying a single topic for a document, LDA identifies a distribution over topics. High probability topic assignments will identify compact, low-entropy language models, so that the probability mass of the language model for each topic is divided among a relatively small vocabulary.

Our model operates in a similar manner, identifying a topic for each word, denoted by z in Figure 4-4. However, where LDA learns a distribution over topics for each document, we deterministically construct a document-specific topic distribution from the clusters represented by the document's keyphrases — this is η in the figure. η assigns equal probability to all topics that are represented in the keyphrase annotations, and very small probability to other topics. Generating the word topics in this way ties together the clustering and language models.

As noted above, sometimes the keyphrase annotation does not represent all of the

[2]Note that while we model each similarity score as an independent draw; clearly this assumption is too strong, due to symmetry and transitivity. Models making similar assumptions about the independence of related hidden variables have previously been shown to be successful (for example, the semi-supervised part-of-speech tagging work of Toutanova and Johnson [127]).

semantic topics that are expressed in the text. For this reason, we also construct another "background" distribution ϕ over topics. The auxiliary variable c indicates whether a given word's topic is drawn from the distribution derived from annotations, or from the background model. Representing c as a hidden variable allows us to stochastically interpolate between the two language models ϕ and η. In addition, any given document will most likely also discuss topics that are not covered by any keyphrase. To account for this, the model is allowed to leave some of the clusters empty, thus leaving some of the topics to be independent of all the keyphrases.

4.3.3 Generative Process

Our model assumes that all observed data is generated through a stochastic process involving hidden parameters. In this section, we formally specify this generative process. This specification guides inference of the hidden parameters based on observed data, which are the following:

- For each of the L keyphrases, a vector $\mathbf{s}_\ell$ of length L denoting a pairwise similarity score in the interval $[0, 1]$ to every other keyphrase.
- For each document d, its bag of words $\mathbf{w}_d$ of length N_d. The nth word of d is $w_{d,n}$.
- For each document d, a set of keyphrase annotations h_d, which includes index ℓ if the document was annotated with keyphrase ℓ.
- The number of clusters K, which should be large enough to encompass topics with actual clusters of keyphrases, as well as word-only topics.

These observed variables are generated according to the following process:

1. Draw a multinomial distribution ψ over the K keyphrase clusters from a symmetric Dirichlet prior with parameter ψ_0.[3]
2. For $\ell = 1 \ldots L$:

[3]Variables subscripted with zero are fixed hyperparameters.

(a) Draw the ℓth keyphrase's cluster assignment x_ℓ from Multinomial(ψ).

3. For $(\ell, \ell') = (1 \ldots L, 1 \ldots L)$:

   (a) If $x_\ell = x_{\ell'}$, draw $s_{\ell,\ell'}$ from Beta($\alpha_=$) $\equiv$ Beta(2, 1), encouraging scores to be biased toward values close to one.

   (b) If $x_\ell \neq x_{\ell'}$, draw $s_{\ell,\ell'}$ from Beta($\alpha_\neq$) $\equiv$ Beta(1, 2), encouraging scores to be biased toward values close to zero.

4. For $k = 1 \ldots K$:

   (a) Draw language model θ_k from a symmetric Dirichlet prior with parameter θ_0.

5. For $d = 1 \ldots D$:

   (a) Draw a background topic model ϕ_d from a symmetric Dirichlet prior with parameter ϕ_0.

   (b) Deterministically construct an annotation topic model η_d, based on keyphrase cluster assignments $\mathbf{x}$ and observed document annotations h_d. Specifically, let $\mathbf{H}$ be the set of topics represented by phrases in h_d. Distribution η_d assigns equal probability to each element of $\mathbf{H}$, and a very small probability mass to other topics.[4]

   (c) Draw a weighted coin λ_d from Beta(λ_0), which will determine the balance between annotation η_d and background topic models ϕ_d.

   (d) For $n = 1 \ldots N_d$:

      i. Draw a binary auxiliary variable $c_{d,n}$ from Bernoulli(λ_d), which determines whether the topic of the word $w_{d,n}$ is drawn from the annotation topic model η_d or the background model ϕ_d.

      ii. Draw a topic assignment $z_{d,n}$ from the appropriate multinomial as indicated by $c_{d,n}$.

[4]Making a hard assignment of zero probability to the other topics creates problems for parameter estimation. A probability of 10^{-4} was assigned to all topics not represented by the keyphrase cluster memberships.

iii. Draw word $w_{d,n}$ from Multinomial($\theta_{z_{d,n}}$), that is, the language model indexed by the word's topic.

4.4 Inference via Gibbs Sampling

To make predictions on unseen data, we need to estimate the parameters of the model. In Bayesian inference, we estimate the distribution for each parameter, conditioned on the observed data and hyperparameters. Such inference is intractable in the general case, but sampling approaches allow us to approximately construct distributions for each parameter of interest.

Gibbs sampling is perhaps the most generic and straightforward sampling technique. Conditional distributions are computed for each hidden variable, given all the other variables in the model. By repeatedly sampling from these distributions in turn, it is possible to construct a Markov chain whose stationary distribution is the posterior of the model parameters [53]. The use of sampling techniques in natural language processing has been previously investigated by many researchers, including Finkel et al. [49] and Goldwater et al. [56].

We now present sampling equations for each of the hidden variables in Figure 4-4. The prior over keyphrase clusters ψ is sampled based on the hyperprior ψ_0 and the keyphrase cluster assignments $\mathbf{x}$. We write $p(\psi \mid \ldots)$ to mean the probability conditioned on all the other variables.

$$\begin{aligned}
p(\psi \mid \ldots) &\propto p(\psi \mid \psi_0) p(\mathbf{x} \mid \psi), \\
&= p(\psi \mid \psi_0) \prod_{\ell} p(x_\ell \mid \psi) \\
&= \text{Dirichlet}(\psi; \psi_0) \prod_{\ell} \text{Multinomial}(x_\ell; \psi) \\
&= \text{Dirichlet}(\psi; \psi'),
\end{aligned}$$

where ψ'_i is $\psi_0 + \text{count}(x_\ell = i)$. This conditional distribution is derived based on the conjugacy of the multinomial to the Dirichlet distribution. The first line follows

from Bayes' rule, and the second line from the conditional independence of cluster assignments $\mathbf{x}$ given keyphrase distribution ψ.

Resampling equations for ϕ_d and θ_k can be derived in a similar manner:

$$p(\phi_d \mid \ldots) \propto \text{Dirichlet}(\phi_d; \phi'_d),$$
$$p(\theta_k \mid \ldots) \propto \text{Dirichlet}(\theta_k; \theta'_k),$$

where $\phi'_{d,i} = \phi_0 + \text{count}(z_{n,d} = i \wedge c_{n,d} = 0)$ and $\theta'_{k,i} = \theta_0 + \sum_d \text{count}(w_{n,d} = i \wedge z_{n,d} = k)$. In building the counts for ϕ'_i, we consider only cases in which $c_{n,d} = 0$, indicating that the topic $z_{n,d}$ is indeed drawn from the background topic model ϕ_d. Similarly, when building the counts for θ'_k, we consider only cases in which the word $w_{d,n}$ is drawn from topic k.

To resample λ, we employ the conjugacy of the Beta prior to the Bernoulli observation likelihoods, adding counts of $\mathbf{c}$ to the prior λ_0.

$$p(\lambda_d \mid \ldots) \propto \text{Beta}(\lambda_d; \lambda'_d),$$

where $\lambda'_d = \lambda_0 + \begin{bmatrix} \sum_n \text{count}(c_{d,n} = 1) \\ \sum_n \text{count}(c_{d,n} = 0) \end{bmatrix}$.

The keyphrase cluster assignments are represented by $\mathbf{x}$, whose sampling distribution depends on ψ, $\mathbf{s}$, and $\mathbf{z}$, via η:

$$\begin{aligned} p(x_\ell \mid \ldots) &\propto p(x_\ell \mid \psi) p(\mathbf{s} \mid x_\ell, \mathbf{x}_{-\ell}, \alpha) p(\mathbf{z} \mid \eta, \psi, \mathbf{c}) \\ &\propto p(x_\ell \mid \psi) \left[\prod_{\ell' \neq \ell} p(s_{\ell,\ell'} \mid x_\ell, x_{\ell'}, \alpha) \right] \left[\prod_d^D \prod_{c_{d,n}=1} p(z_{d,n} \mid \eta_d) \right] \\ &= \text{Multinomial}(x_\ell; \psi) \left[\prod_{\ell' \neq \ell} \text{Beta}(s_{\ell,\ell'}; \alpha_{x_\ell, x_{\ell'}}) \right] \left[\prod_d^D \prod_{c_{d,n}=1} \text{Multinomial}(z_{d,n}; \eta_d) \right]. \end{aligned}$$

The leftmost term of the above equation is the prior on x_ℓ. The next term encodes the dependence of the similarity matrix $\mathbf{s}$ on the cluster assignments; with slight abuse of notation, we write $\alpha_{x_\ell, x_{\ell'}}$ to denote $\alpha_=$ if $x_\ell = x_{\ell'}$, and $\alpha_\neq$ otherwise. The third term

is the dependence of the word topics $z_{d,n}$ on the topic distribution η_d. We compute the final result of this probability expression for each possible setting of x_ℓ, and then sample from the normalized multinomial.

The word topics $\mathbf{z}$ are sampled according to the topic distribution η_d, the background distribution ϕ_d, the observed words $\mathbf{w}$, and the auxiliary variable $\mathbf{c}$:

$$
\begin{aligned}
p(z_{d,n} \mid \ldots) &\propto p(z_{d,n} \mid \phi, \eta_d, c_{d,n}) p(w_{d,n} \mid z_{d,n}, \theta) \\
&= \begin{cases} \text{Multinomial}(z_{d,n}; \eta_d)\text{Multinomial}(w_{d,n}; \theta_{z_{d,n}}) & \text{if } c_{d,n} = 1, \\ \text{Multinomial}(z_{d,n}; \phi_d)\text{Multinomial}(w_{d,n}; \theta_{z_{d,n}}) & \text{otherwise.} \end{cases}
\end{aligned}
$$

As with x, each $z_{d,n}$ is sampled by computing the conditional likelihood of each possible setting within a constant of proportionality, and then sampling from the normalized multinomial.

Finally, we sample the auxiliary variable $c_{d,n}$, which indicates whether the hidden topic $z_{d,n}$ is drawn from η_d or ϕ_d. $\mathbf{c}$ depends on its prior λ and the hidden topic assignments $\mathbf{z}$:

$$
\begin{aligned}
p(c_{d,n} \mid \ldots) &\propto p(c_{d,n} \mid \lambda_d) p(z_{d,n} \mid \eta_d, \phi_d, c_{d,n}) \\
&= \begin{cases} \text{Bernoulli}(c_{d,n}; \lambda_d)\text{Multinomial}(z_{d,n}; \eta_d) & \text{if } c_{d,n} = 1, \\ \text{Bernoulli}(c_{d,n}; \lambda_d)\text{Multinomial}(z_{d,n}; \phi_d) & \text{otherwise.} \end{cases}
\end{aligned}
$$

Again, we compute the likelihood of $c_{d,n} = 0$ and $c_{d,n} = 1$ within a constant of proportionality, and then sample from the normalized Bernoulli distribution.

Finally, our model requires values for fixed hyperparameters θ_0, λ_0, ψ_0, and ϕ_0, which are tuned in the standard way based on development set performance.

One of the main applications of our model is to predict the properties supported by documents that are not annotated with keyphrases. At test time, we would like to compute a posterior estimate of ϕ_d for an unannotated test document d. Since annotations are not present, property prediction is based only on the text component of the model. For this estimate, we use the same Gibbs sampling procedure, restricted

to $z_{d,n}$ and ϕ_d, with the stipulation that $c_{d,n}$ is fixed at zero so that $z_{d,n}$ is always drawn from ϕ_d. In particular, we treat the language models as known; to more accurately integrate over all possible language models, we use the final 1000 samples of the language models from training as opposed to using a point estimate. For each topic, if its probability in ϕ_d exceeds a certain threshold, that topic is predicted. This threshold is tuned independently for each topic on a development set. The empirical results we present in Section 4.5 are obtained in this manner.

4.5 Overview of Experiments

Our model for document analysis is implemented in PRÉCIS,[5] a system that performs single- and multi-document review summarization. The goal of PRÉCIS is to provide users with effective access to review data via mobile devices. PRÉCIS contains information about 49,490 products and services ranging from childcare products to restaurants and movies. For each of these products, the system contains a collection of reviews downloaded from consumer websites such as Epinions, CNET, and Amazon. PRÉCIS compresses data for each product into a short list of pros and cons that are supported by the majority of reviews. An example of a summary of 27 reviews for the movie *Pirates of the Caribbean: At World's End* is shown in Figure 4-6. In contrast to traditional multi-document summarizers, the output of the system is not a sequence of sentences, but rather a list of phrases indicative of product properties. This summarization format follows the format of pros/cons summaries that individual reviewers provide on multiple consumer websites. Moreover, the brevity of the summary is particularly suitable for presenting on small screens such as those of mobile devices.

To automatically generate the combined pros/cons list for a product or service, we first apply our model to each review. The model is trained independently for each product domain (*e.g.,* movies) using a corresponding subset of reviews with free-text annotations. These annotations also provide a set of keyphrases that contribute to

[5]PRÉCIS is accessible at `http://groups.csail.mit.edu/rbg/projects/precis/`.

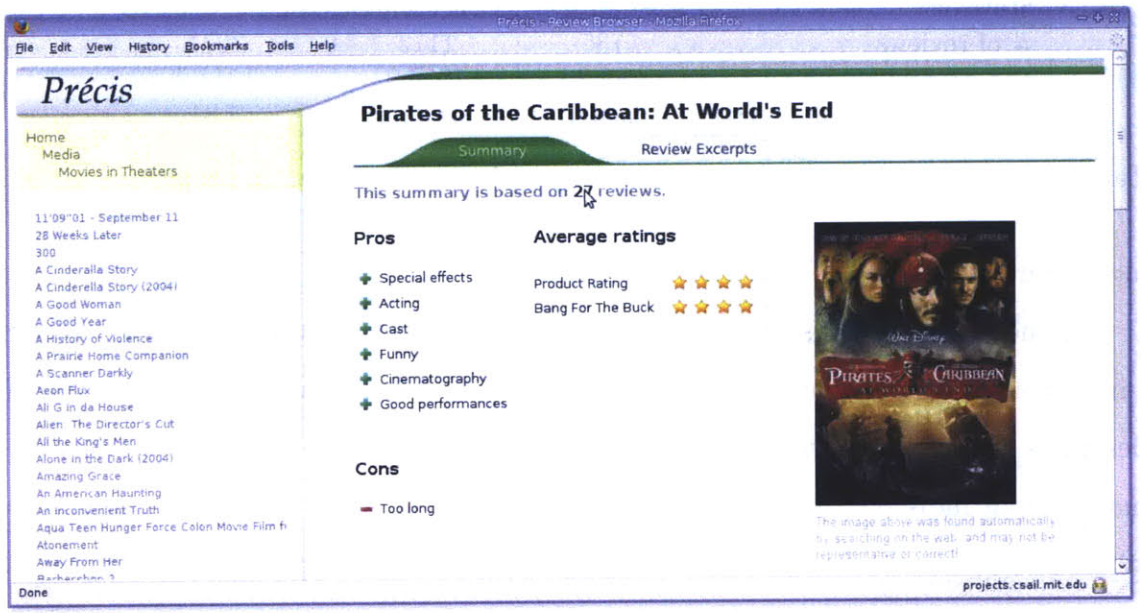


Figure 4-6: Summary of reviews for the movie *Pirates of the Caribbean: At World's End* on PRÉCIS. This summary is based on 27 documents. The list of pros and cons are generated automatically using the system described in this chapter. The generation of numerical ratings is based on the algorithm described by Snyder and Barzilay [122].

the clusters associated with product properties. Once the model is trained, it labels each review with a set of properties. Since the set of possible properties is the same for all reviews of a product, the comparison among reviews is straightforward — for each property, we count the number of reviews that support it, and select the property as part of a summary if it is supported by the majority of the reviews. The set of semantic properties is converted into a pros/cons list by presenting the most common keyphrase for each property.

This aggregation technology is applicable in two scenarios. The system can be applied to unannotated reviews, inducing semantic properties from the document text; this conforms to the traditional way in which learning-based systems are applied to unlabeled data. However, our model is valuable even when individual reviews do include pros/cons keyphrase annotations. Due to the high degree of paraphrasing, direct comparison of keyphrases is challenging (see Section 4.2). By inferring a clustering over keyphrases, our model permits comparison of keyphrase annotations on a

Statistic	Restaurants	Cell Phones	Digital Cameras
# of reviews	5735	1112	3971
avg. review length	786.3	1056.9	1014.2
avg. keyphrases / review	3.42	4.91	4.84

Table 4.3: Statistics of the datasets used to evaluate our semantic properties model.

more semantic level.

The next two sections provide a set of evaluations of our model's ability to capture the semantic content of document text and keyphrase annotations. Section 4.6 describes an evaluation of our system's ability to extract meaningful semantic summaries from individual documents, and also assesses the quality of the paraphrase structure induced by our model. Section 4.7 extends this evaluation to our system's ability to summarize multiple review documents.

4.6 Single-Document Experiments

First, we evaluate our model with respect to its ability to reproduce the annotations present in individual documents, based on the document text. We compare against a wide variety of baselines and variations of our model, demonstrating the appropriateness of our approach to this task. In addition, we explicitly evaluate the quality of the paraphrase structure induced by our model by comparing against a gold standard clustering of keyphrases provided by expert annotators.

4.6.1 Evaluation Setup

In this section, we describe the datasets and evaluation techniques used for experiments with our system and other automatic methods. We also comment on how hyperparameters are tuned for our model, and how sampling is initialized.

Datasets

We evaluate our system on reviews from three domains: restaurants, cell phones, and digital cameras. These reviews were downloaded from the Epinions website; we used

user-authored pros and cons associated with reviews as keyphrases (see Section 4.2). Statistics for the datasets are provided in Table 4.3. For each of the domains, we selected 50% of the documents for training.

We consider two strategies for constructing test data. First, we consider evaluating the semantic properties inferred by our system against expert annotations of the semantic properties present in each document. To this end, we use the expert annotations originally described in Section 4.2 as a test set; to reiterate, these were annotations of 170 reviews in the restaurant domain, of which we now hold out 50 as a development set. The review texts were annotated with six properties according to standardized guidelines. This strategy enforces consistency and completeness in the ground truth annotations, differentiating them from free-text annotations.

Unfortunately, our ability to evaluate against expert annotations is limited by the cost of producing such annotations. To expand evaluation to other domains, we use the author-written keyphrase annotations that are present in the original reviews. Such annotations are noisy — while the presence of a property annotation on a document is strong evidence that the document supports the property, the inverse is not necessarily true. That is, the *lack* of an annotation does not necessarily imply that its respective property does not hold — *e.g.,* a review with no *good service*-related keyphrase may still praise the service in the body of the document.

For experiments using free-text annotations, we overcome this pitfall by restricting the evaluation of predictions of individual properties to only those documents that are annotated with that property or its antonym. For instance, when evaluating the prediction of the *good service* property, we will only select documents which are either annotated with *good service* or *bad service*-related keyphrases.[6] For this reason, each semantic property is evaluated against a unique subset of documents. The details of these development and test sets are presented in Appendix A.

To ensure that free-text annotations can be reliably used for evaluation, we compare with the results produced on expert annotations whenever possible. As shown in

[6]This determination is made by mapping author keyphrases to properties using an expert-generated gold standard clustering of keyphrases. It is much cheaper to produce an expert clustering of keyphrases than to obtain expert annotations of the semantic properties in every document.

Section 4.6.2, the free-text evaluations produce results that cohere well with those obtained on expert annotations, suggesting that such labels can be used as a reasonable proxy for expert annotation evaluations.

Evaluation Methods

Our first evaluation leverages the expert annotations described in Section 4.2. One complication is that expert annotations are marked on the level of semantic properties, while the model makes predictions about the appropriateness of individual keyphrases. We address this by representing each expert annotation with the most commonly-observed keyphrase from the manually-annotated cluster of keyphrases associated with the semantic property. For example, an annotation of the semantic property *good food* is represented with its most common keyphrase realization, "great food." Our evaluation then checks whether this keyphrase is within any of the clusters of keyphrases predicted by the model.

The evaluation against author free-text annotations is similar to the evaluation against expert annotations. In this case, the annotation takes the form of individual keyphrases rather than semantic properties. As noted, author-generated keyphrases suffer from inconsistency. We obtain a consistent evaluation by mapping the author-generated keyphrase to a cluster of keyphrases as a determined by the expert annotator, and then again selecting the most common keyphrase realization of the cluster. For example, the author may use the keyphrase "tasty," which maps to the semantic cluster *good food*; we then select the most common keyphrase realization, "great food." As in the expert evaluation, we check whether this keyphrase is within any of the clusters predicted by the model.

Model performance is quantified using recall, precision, and F-score. These are computed in the standard manner, based on the model's representative keyphrase predictions compared against the corresponding references. As with our document structure work, approximate randomization [100, 136] is used for statistical significance testing. To reiterate, this test repeatedly performs random swaps of individual results from each candidate system, and checks whether the resulting performance

Hyperparameters	Restaurants	Cell Phones	Cameras
θ_0	0.0001	0.0001	0.0001
ψ_0	0.001	0.0001	0.1
ϕ_0	0.001	0.0001	0.001

Table 4.4: Values of the hyperparameters used for each domain across all experiments for the semantic properties model.

gap remains at least as large. We use this test because it is valid for comparing non-linear functions of random variables, such as F-scores, unlike other common methods such as the sign test. Previous work that used this test include evaluations at the Message Understanding Conference [37, 38]; more recently, Riezler and Maxwell [114] advocated for its use in evaluating machine translation systems.

Parameter Tuning and Initialization

To improve the model's convergence rate, we perform two initialization steps for the Gibbs sampler. First, sampling is done only on the keyphrase clustering component of the model, ignoring document text. Second, we fix this clustering and sample the remaining model parameters. These two steps are run for 5,000 iterations each. The full joint model is then sampled for 100,000 iterations. Inspection of the parameter estimates confirms model convergence. On a 2GHz dual-core desktop machine, a multithreaded C++ implementation of model training takes about two hours for each dataset.

Our model needs to be provided with the number of clusters K.[7] We set K large enough for the model to learn effectively on the development set. For the restaurant data we set K to 20. For cell phones and digital cameras, K was set to 30 and 40, respectively. These values were tuned using the development set. However, we found that as long as K was large enough to accommodate a significant number of keyphrase clusters, and a few additional to account for topics with no keyphrases, the specific value of K does not affect the model's performance.

All other hyperparameters were adjusted based on development set performance,

[7]This requirement could conceivably be removed by modeling the cluster indices as being drawn from a Dirichlet process prior.

shown in Table 4.4. In all cases, λ_0 was set to $(1, 1)$, making Beta(λ_0) the uniform distribution. The optimal hyperparameter values tend to be very low, indicating that the model performs best when very peaked parameter estimates are preferred. This also encourages empty clusters to be formed when K is set to a higher than necessary value.

As previously mentioned, we obtain document properties by examining the probability mass of the topic distribution assigned to each property. A probability threshold is set for each property via the development set, optimizing for maximum F-score.

4.6.2 Results

In this section, we report the performance of our model, comparing it with an array of increasingly sophisticated baselines and model variations. We first demonstrate that learning a clustering of annotation keyphrases is crucial for accurate semantic prediction. Next, we investigate the impact of paraphrasing quality on model accuracy by considering the expert-generated gold standard clustering of keyphrases as another comparison point; we also consider alternative automatically computed sources of paraphrase information.

For ease of comparison, the results of all the experiments are shown in Table 4.6 and Table 4.7, with a summary of the baselines and model variations in Table 4.5.

Comparison against Simple Baselines

Our first evaluation compares our model to four naïve baselines. All four treat keyphrases as independent, ignoring their latent paraphrase structure.

- *Random:* Each keyphrase is supported by a document with probability of one half. The results of this baseline are computed in expectation, rather than actually run. This baseline is expected to have a recall of 0.5, because in expectation it will select half of the correct keyphrases. Its precision is the average proportion of annotations in the test set against the number of possible annotations. That is, in a test set of size n with m properties, if property i

Random	Each keyphrase is supported by a document with probability of one half.
Keyphrase in text	A keyphrase is supported by a document if it appears verbatim in the text.
Keyphrase classifier	A separate support vector machine classifier is trained for each keyphrase. Positive examples are documents that are labeled by the author with the keyphrase; all other documents are considered to be negative examples. A keyphrase is supported by a document if that keyphrase's classifier returns a positive prediction.
Heuristic keyphrase classifier	Similar to *keyphrase classifier*, except heuristic methods are used in an attempt to reduce noise from the training documents. Specifically we wish to remove sentences that discuss other keyphrases from the positive examples. The heuristic removes from the positive examples all sentences that have no word overlap with the given keyphrase.
Model cluster in text	A keyphrase is supported by a document if it or any of its paraphrases appear in the text. Paraphrasing is based on our model's keyphrase clusters.
Model cluster classifier	A separate classifier is trained for each cluster of keyphrases. Positive examples are documents that are labeled by the author with any keyphrase from the cluster; all other documents are negative examples. All keyphrases of a cluster are supported by a document if that cluster's classifier returns a positive prediction. Keyphrase clustering is based on our model.
Heuristic model cluster classifier	Similar to *model cluster classifier*, except heuristic methods are used to reduce noise from the training documents. Specifically we wish to remove from the positive examples sentences that discuss keyphrases from other clusters. The heuristic removes from the positive examples all sentences that have no word overlap with any of the keyphrases from the given cluster. Keyphrase clustering is based on our model.
Gold cluster model	A variation of our model where the clustering of keyphrases is fixed to an expert-created gold standard. Only the text modeling parameters are learned.
Gold cluster in text	Similar to *model cluster in text*, except the clustering of keyphrases is according to the expert-produced gold standard.
Gold cluster classifier	Similar to *model cluster classifier*, except the clustering of keyphrases is according to the expert-produced gold standard.
Heuristic gold cluster classifier	Similar to *heuristic model cluster classifier*, except the clustering of keyphrases is according to the expert-produced gold standard.
Independent cluster model	A variation of our model where the clustering of keyphrases is first learned from keyphrase similarity information only, separately from the text. The resulting *independent* clustering is then fixed while the text modeling parameters are learned. This variation's key distinction from our full model is the lack of joint learning of keyphrase clustering and text topics.
Independent cluster in text	Similar to *model cluster in text*, except that the clustering of keyphrases is according to the independent clustering.
Independent cluster classifier	Similar to *model cluster classifier*, except that the clustering of keyphrases is according to the independent clustering.
Heuristic independent cluster classifier	Similar to *heuristic model cluster classifier*, except the clustering of keyphrases is according to the independent clustering.

Table 4.5: A summary of the baselines and variations against which our semantic properties model is compared.

	Method	Restaurants		
		Recall	Prec.	F-score
1	Our model	0.920	0.353	**0.510**
2	Random	0.500	0.346	0.409 *
3	Keyphrase in text	0.048	0.500	0.087 *
4	Keyphrase classifier	0.769	0.353	**0.484** *
5	Heuristic keyphrase classifier	0.839	0.340	**0.484** *
6	Model cluster in text	0.227	0.385	0.286 *
7	Model cluster classifier	0.721	0.402	**0.516**
8	Heuristic model cluster classifier	0.731	0.366	0.488 *
9	Gold cluster model	0.936	0.344	**0.502**
10	Gold cluster in text	0.339	0.360	0.349 *
11	Gold cluster classifier	0.693	0.366	0.479 *
12	Heuristic gold cluster classifier	1.000	0.326	0.492 ◇
13	Independent cluster model	0.745	0.363	**0.488** ◇
14	Independent cluster in text	0.220	0.340	0.266 *
15	Independent cluster classifier	0.586	0.384	0.464 *
16	Heuristic independent cluster classifier	0.592	0.386	0.468 *

Table 4.6: Comparison of the property predictions made by our semantic properties model and a series of baselines and model variations in the restaurant domain, evaluated against expert semantic annotations. The results are divided according to experiment. The methods against which our model has significantly better results using approximate randomization are indicated with $*$ for $p \leq 0.05$, and $\diamond$ for $p \leq 0.1$.

appears n_i times, then expected precision is $\sum_{i=1}^{m} \frac{n_i}{mn}$. For instance, for the restaurants gold standard evaluation, the six tested properties appeared a total of 249 times over 120 documents, yielding an expected precision of 0.346.

- *Keyphrase in text:* A keyphrase is supported by a document if it appears verbatim in the text. Precision should be high while recall will be low, because the model is unable to detect paraphrases of the keyphrase in the text. For instance, for the first review from Figure 4-1, "cleanliness" would be supported because it appears in the text; however, "healthy" would not be supported, even though the synonymous "great nutrition" does appear.

- *Keyphrase classifier:* A separate discriminative classifier is trained for each keyphrase. Positive examples are documents that are labeled by the author with the keyphrase; all other documents are considered to be negative examples. Consequently, for any particular keyphrase, documents labeled with syn-

Method	Restaurants			Cell Phones			Digital Cameras		
	Recall	Prec.	F-score	Recall	Prec.	F-score	Recall	Prec.	F-score
1 Our model	0.923	0.623	**0.744**	0.971	0.537	**0.692**	0.905	0.586	**0.711**
2 Random	0.500	0.500	0.500 *	0.500	0.489	0.494 *	0.500	0.501	0.500 *
3 Keyphrase in text	0.077	0.906	0.142 *	0.171	0.529	0.259 *	0.715	0.642	0.676 *
4 Keyphrase classif.	0.905	0.527	**0.666** *	1.000	0.500	**0.667**	0.942	0.540	**0.687** ◇
5 Heur. keyphr. classif.	0.997	0.497	0.664 *	0.845	0.474	0.607 *	0.845	0.531	0.652 *
6 Model cluster in text	0.416	0.613	0.496 *	0.829	0.547	0.659 ◇	0.812	0.596	0.687 *
7 Model cluster classif.	0.859	0.711	**0.778** †	0.876	0.561	**0.684**	0.927	0.568	0.704
8 Heur. model classif.	0.910	0.567	0.698 *	1.000	0.464	0.634 ◇	0.942	0.568	**0.709**
9 Gold cluster model	0.992	0.500	0.665 *	0.924	0.561	**0.698**	0.962	0.510	0.667 *
10 Gold cluster in text	0.541	0.604	0.571 *	0.914	0.497	0.644 *	0.903	0.522	0.661 *
11 Gold cluster classif.	0.865	0.720	**0.786** †	0.810	0.559	0.661	0.874	0.674	**0.761**
12 Heur. gold classif.	0.997	0.499	0.665 *	0.969	0.468	0.631 ◇	0.971	0.508	0.667 *
13 Indep. cluster model	0.984	0.528	0.687 *	0.838	0.564	**0.674**	0.945	0.519	**0.670** *
14 Indep. cluster in text	0.382	0.569	0.457 *	0.724	0.481	0.578 *	0.469	0.476	0.473 *
15 Indep. cluster classif.	0.753	0.696	**0.724**	0.638	0.472	0.543 *	0.496	0.588	0.538 *
16 Heur. indep. classif.	0.881	0.478	0.619 *	1.000	0.464	0.634 ◇	0.969	0.501	0.660 *

Table 4.7: Comparison of the property predictions made by our semantic properties model and a series of baselines and model variations in three product domains, as evaluated against author free-text annotations. The results are divided according to experiment. The methods against which our model has significantly better results using approximate randomization are indicated with $*$ for $p \leq 0.05$, and $\diamond$ for $p \leq 0.1$. Methods which perform significantly better than our model with $p \leq 0.05$ are indicated with $\dagger$.

onymous keyphrases would be among the negative examples. A keyphrase is supported by a document if that keyphrase's classifier returns a positive prediction.

We use support vector machines, built using SVM^{light} [69] with the same features as our model, *i.e.,*word counts.[8] To partially circumvent the imbalanced positive/negative data problem, we tuned prediction thresholds on a development set to maximize F-score, in the same manner that we tuned thresholds for our model.

- *Heuristic keyphrase classifier:* This baseline is similar to *keyphrase classifier* above, but attempts to mitigate some of the noise inherent in the training data. Specifically, any given positive example document may contain text unrelated to the given keyphrase. We attempt to reduce this noise by removing from the positive examples all sentences that have no word overlap with the given keyphrase. A keyphrase is supported by a document if that keyphrase's classifier

[8]In general, SVMs have the additional advantage of being able to incorporate arbitrary features, but for the sake of comparison we restrict ourselves to using the same features across all methods.

returns a positive prediction.[9]

Lines 2-5 of Tables 4.6 and 4.7 present these results, using both gold annotations and the original authors' annotations for testing. Our model outperforms these three baselines in all evaluations with strong statistical significance.

The *keyphrase in text* baseline fares poorly: its F-score is below the random baseline in three of the four evaluations. As expected, the recall of this baseline is usually low because it requires keyphrases to appear verbatim in the text. The precision is somewhat better, but the presence of a significant number of false positives indicates that the presence of a keyphrase in the text is not necessarily a reliable indicator of the associated semantic property.

Interestingly, one domain in which *keyphrase in text* does perform well is digital cameras. We believe that this is because of the prevalence of specific technical terms in the keyphrases used in this domain, such as "zoom" and "battery life." Such technical terms are also frequently used in the review text, making the recall of *keyphrase in text* substantially higher in this domain than in the other evaluations.

The *keyphrase classifier* baseline outperforms the *random* and *keyphrase in text* baselines, but still achieves consistently lower performance than our model in all four evaluations. Notably, the performance of *heuristic keyphrase classifier* is worse than *keyphrase classifier* except in one case. This alludes to the difficulty of removing the noise inherent in the document text.

Overall, these results indicate that methods which learn and predict keyphrases without accounting for their intrinsic hidden structure are insufficient for optimal property prediction. This leads us toward extending the present baselines with clustering information.

It is important to assess the consistency of the evaluation based on free-text annotations (Table 4.7) with the evaluation that uses expert annotations (Table 4.6). While the absolute scores on the expert annotations dataset are lower than the scores with free-text annotations, the ordering of performance between the various automatic methods is the same across the two evaluation scenarios. This consistency is

[9]We thank a reviewer for suggesting this baseline.

maintained in the rest of our experiments as well, indicating that for the purpose of relative comparison between the different automatic methods, our method of evaluating with free-text annotations is a reasonable proxy for evaluation on expert-generated annotations.

Comparison against Clustering-based Approaches

The previous section demonstrates that our model outperforms baselines that do not account for the paraphrase structure of keyphrases. We now ask whether it is possible to enhance the baselines' performance by augmenting them with the keyphrase clustering induced by our model. Specifically, we introduce three more systems, none of which are "true" baselines, since they all use information inferred by our model.

- *Model cluster in text:* A keyphrase is supported by a document if it or any of its paraphrases appears in the text. Paraphrasing is based on our model's clustering of the keyphrases. The use of paraphrasing information enhances recall at the potential cost of precision, depending on the quality of the clustering. For example, assuming "healthy" and "great nutrition" are clustered together, the presence of "healthy" in the text would also indicate support for "great nutrition," and vice versa.

- *Model cluster classifier:* A separate discriminative classifier is trained for each cluster of keyphrases. Positive examples are documents that are labeled by the author with any keyphrase from the cluster; all other documents are negative examples. All keyphrases of a cluster are supported by a document if that cluster's classifier returns a positive prediction. Keyphrase clustering is based on our model. As with *keyphrase classifier*, we use support vector machines trained on word count features, and we tune the prediction thresholds for each individual cluster on a development set.

  Another perspective on *model cluster classifier* is that it augments the simplistic text modeling portion of our model with a discriminative classifier. Discriminative training is often considered to be more powerful than equivalent generative

approaches [91], leading us to expect a high level of performance from this system.

- *Heuristic model cluster classifier:* This method is similar to *model cluster classifier* above, but with additional heuristics used to reduce the noise inherent in the training data. Positive example documents may contain text unrelated to the given cluster. To reduce this noise, sentences that have no word overlap with any of the cluster's keyphrases are removed. All keyphrases of a cluster are supported by a document if that cluster's classifier returns a positive prediction. Keyphrase clustering is based on our model.

Lines 6-8 of Tables 4.6 and 4.7 present results for these methods. As expected, using a clustering of keyphrases with the baseline methods substantially improves their recall, with low impact on precision. *Model cluster in text* invariably outperforms *keyphrase in text* — the recall of *keyphrase in text* is improved by the addition of clustering information, though precision is worse in some cases. This phenomenon holds even in the cameras domain, where *keyphrase in text* already performs well. However, our model still significantly outperforms *model cluster in text* in all evaluations.

Adding clustering information to the classifier baseline results in performance that is sometimes better than our model's. This result is not surprising, because *model cluster classifier* gains the benefit of our model's robust clustering while learning a more sophisticated classifier for assigning properties to texts. The resulting combined system is more complex than our model by itself, but has the potential to yield better performance. On the other hand, using a simple heuristic to reduce the noise present in the training data consistently hurts the performance of the classifier, possibly due to the reduction in the amount of training data.

Overall, the enhanced performance of these methods, in contrast to the keyphrase baselines, is aligned with previous observations in entailment research [41], confirming that paraphrasing information contributes greatly to improved performance in semantic inference tasks.

The Impact of Paraphrasing Quality

The previous section demonstrates one of the central claims of our work: accounting for paraphrase structure yields substantial improvements in semantic inference when using noisy keyphrase annotations. A second key aspect of our research is the idea that clustering quality benefits from tying the clusters to hidden topics in the document text. We evaluate this claim by comparing our model's clustering against an independent clustering baseline. We also compare against a "gold standard" clustering produced by expert human annotators. To test the impact of these clustering methods, we substitute the model's inferred clustering with each alternative and examine how the resulting semantic inferences change. This comparison is performed for the semantic inference mechanism of our model, as well as for the *model cluster in text*, *model cluster classifier* and *heuristic model cluster classifier* baselines.

To add a "gold standard" clustering to our model, we replace the hidden variables that correspond to keyphrase clusters with observed values that are set according to the gold standard clustering.[10] The only parameters that are trained are those for modeling text. This model variation, *gold cluster model*, predicts properties using the same inference mechanism as the original model. The baseline variations *gold cluster in text*, *gold cluster classifier* and *heuristic gold cluster classifier* are likewise derived by substituting the automatically computed clustering with gold standard clusters.

An additional clustering is obtained using only the keyphrase similarity information. Specifically, we modify our original model so that it learns the keyphrase clustering in isolation from the text, and only then learns the property language models. In this framework, the keyphrase clustering is entirely independent of the review text, because the text modeling is learned with the keyphrase clustering fixed. We refer to this modification of the model as *independent cluster model*. Because our model treats the document text as a mixture of latent topics, this is reminiscent of models such as supervised latent Dirichlet allocation (sLDA) [21], with the labels acquired by performing a clustering across keyphrases as a preprocessing step. As in the previous

[10]The gold standard clustering was created as part of the evaluation procedure described in Section 4.6.1.

Clustering	Restaurants	Cell Phones	Digital Cameras
Model clusters	**0.914**	**0.876**	**0.945**
Independent clusters	0.892	0.759	0.921

Table 4.8: Rand Index scores of our semantic properties model's clusters, learned from keyphrases and text jointly, compared against clusters learned only from keyphrase similarity. Evaluation of cluster quality is based on the gold standard clustering.

experiment, we introduce three new baseline variations — *independent cluster in text*, *independent cluster classifier* and *heuristic independent cluster classifier*.

Lines 9-16 of Tables 4.6 and 4.7 present the results of these experiments. The *gold cluster model* produces F-scores comparable to our original model, providing strong evidence that the clustering induced by our model is of sufficient quality for semantic inference. The application of the expert-generated clustering to the baselines (lines 10, 11 and 12) yields less consistent results, but overall this evaluation provides little reason to believe that performance would be substantially improved by obtaining a clustering that was closer to the gold standard.

The *independent cluster model* consistently reduces performance with respect to the full joint model, supporting our hypothesis that joint learning gives rise to better prediction. The independent clustering baselines, *independent cluster in text*, *independent cluster classifier* and *heuristic independent cluster classifier* (lines 14 to 16), are also worse than their counterparts that use the model clustering (lines 6 to 8). This observation leads us to conclude that while the expert-annotated clustering does not always improve results, the independent clustering always degrades them. This supports our view that joint learning of clustering and text models is an important prerequisite for better property prediction.

Another way of assessing the quality of each automatically-obtained keyphrase clustering is to quantify its similarity to the clustering produced by the expert annotators. For this purpose we use the *Rand Index* [111], a measure of cluster similarity. This measure varies from zero to one, with higher scores indicating greater similarity. Table 4.8 shows the Rand Index scores for our model's full joint clustering, as well as the clustering obtained from *independent cluster model*. In every domain, joint inference produces an overall clustering that improves upon the keyphrase-similarity-only

approach. These scores again confirm that joint inference across keyphrases and document text produces a better clustering than considering features of the keyphrases alone.

4.7 Multiple-Document Experiments

Our last experiment examines the multi-document summarization capability of our system. We study our model's ability to aggregate properties across a set of reviews, compared to baselines that aggregate by directly using the free-text annotations.

4.7.1 Data and Setup

We selected 50 restaurants, with five user-written reviews for each restaurant. Ten annotators were asked to annotate the reviews for five restaurants each, comprising 25 reviews per annotator. They used the same six salient properties and the same annotation guidelines as in the previous restaurant annotation experiment (see Section 4.2). In constructing the ground truth, we label properties that are supported in at least three of the five reviews.

We make property predictions on the same set of reviews with our model and the baselines presented below. For the automatic methods, we register a prediction if the system judges the property to be supported on at least two of the five reviews.[11] The recall, precision, and F-score are computed over these aggregate predictions, against the six salient properties marked by annotators.

4.7.2 Aggregation Approaches

In this evaluation, we run the trained version of our model as described in Section 4.6.1. Note that keyphrases are not provided to our model, though they are provided to the baselines.

[11]When three corroborating reviews are required, the baseline systems produce very few positive predictions, leading to poor recall. Results for this setting are presented in Appendix B.

Method	Recall	Prec.	F-score
Our model	0.905	0.325	**0.478**
Keyphrase aggregation	0.036	0.750	0.068 *
Model cluster aggregation	0.238	0.870	0.374 *
Gold cluster aggregation	0.226	0.826	0.355 *
Indep. cluster aggregation	0.214	0.720	0.330 *

Table 4.9: Comparison of the aggregated property predictions made by our semantic properties model and a series of baselines that use free-text annotations. The methods against which our model has significantly better results using approximate randomization are indicated with $*$ for $p \leq 0.05$.

The most obvious baseline for summarizing multiple reviews would be to directly aggregate their free-text keyphrases. These annotations are presumably representative of the review's semantic properties, and unlike the review text, keyphrases can be matched directly with each other. Our first baseline applies this notion directly:

- *Keyphrase aggregation:* A keyphrase is supported for a restaurant if at least two out of its five reviews are annotated verbatim with that keyphrase.

This simple aggregation approach has the obvious downside of requiring very strict matching between independently authored reviews. For that reason, we consider extensions to this aggregation approach that allow for annotation paraphrasing:

- *Model cluster aggregation:* A keyphrase is supported for a restaurant if at least two out of its five reviews are annotated with that keyphrase or one of its paraphrases. Paraphrasing is according to our model's inferred clustering.

- *Gold cluster aggregation:* Same as *model cluster aggregation*, but using the expert-generated clustering for paraphrasing.

- *Independent cluster aggregation:* Same as *model cluster aggregation*, but using the clustering learned only from keyphrase similarity for paraphrasing.

4.7.3 Results

Table 4.9 compares the baselines against our model. Our model outperforms all of the annotation-based baselines, despite not having access to the keyphrase annota-

tions. Notably, *keyphrase aggregation* performs very poorly, because it makes very few predictions, as a result of its requirement of exact keyphrase string match. As before, the inclusion of keyphrase clusters improves the performance of the baseline models. However, the incompleteness of the keyphrase annotations (see Section 4.2) explains why the recall scores are still low compared to our model. By incorporating document text, our model obtains dramatically improved recall, at the cost of reduced precision, ultimately yielding a significantly improved F-score.

These results demonstrate that review summarization benefits greatly from our joint model of the review text and keyphrases. Naïve approaches that consider only keyphrases yield inferior results, even when augmented with paraphrase information.

4.8 Conclusions and Future Work

In this chapter, we have shown how free-text keyphrase annotations provided by novice users can be leveraged as a training set for document-level semantic inference. Free-text annotations have the potential to vastly expand the set of training data available to developers of semantic inference systems; however, as we have shown, they suffer from lack of consistency and completeness. We overcome these problems by inducing a hidden structure of semantic properties, which correspond both to clusters of keyphrases and hidden topics in the text. Our approach takes the form of a hierarchical Bayesian model, which jointly learns from the regularities in both text and keyphrase annotations.

Our model is implemented in a system that successfully extracts semantic properties of unannotated restaurant, cell phone, and camera reviews, empirically validating our approach. Our experiments demonstrate the necessity of handling the paraphrase structure of free-text keyphrase annotations; moreover, they show that a better paraphrase structure is learned in a joint framework that also models the document text. Our approach outperforms competitive baselines for semantic property extraction from both single and multiple documents. It also permits aggregation across multiple keyphrases with different surface forms for multi-document summarization.

This work extends an actively growing literature on document topic modeling. Both topic modeling and paraphrasing posit a hidden layer that captures the relationship between disparate surface forms: in topic modeling, there is a set of latent distributions over lexical items, while paraphrasing is represented by a latent clustering over phrases. We show these two latent structures can be linked, resulting in increased robustness and semantic coherence.

We see several avenues of future work. First, our model draws substantial power from features that measure keyphrase similarity. This ability to use arbitrary similarity metrics is desirable; however, representing individual similarity scores as random variables is a compromise, as they are clearly not independent. We believe that this problem could be avoided by modeling the generation of the entire similarity matrix jointly.

A related approach would be to treat the similarity matrix across keyphrases as an indicator of covariance structure. In such a model, we would learn separate language models for each keyphrase, but keyphrases that are rated as highly similar would be constrained to induce similar language models. Such an approach might be possible in a Gaussian process framework [112].

Currently the focus of our model is to identify the semantic properties expressed in a given document, which allows us to produce a summary of those properties. However, as mentioned in Section 4.2, human authors do not give equal importance to all properties when producing a summary of pros and cons. One possible extension of this work would be to explicitly model the likelihood of each topic being annotated in a document. We might then avoid the current post-processing step that uses property-specific thresholds to compute final predictions from the model output.

Finally, we have assumed that the semantic properties themselves are unstructured. In reality, properties are related in interesting ways. For example, in review texts it would be desirable to model antonyms explicitly, *e.g.*, no restaurant review should be simultaneously labeled as having both good and bad food. Other relationships between properties, such as hierarchical structures, could also be considered. One possible way of modeling these relationships is through a model that explicitly

exploits the connections between topics, such as the *correlated topic model* of Blei and Lafferty [20].

Chapter 5

Conclusions

In this thesis, we showed how multiple granularities of semantic structure can be learned in an unsupervised fashion by intelligently exploiting regularities in in-domain documents. Across each of our tasks, we took advantage of regularities that previous work did not fully exploit, and found that modeling such regularities with appropriate prior and posterior constraints results in improved performance. Here we summarize our key contributions and experimental findings on each task.

- **Content Modeling**: We presented an approach to content modeling that exploits regularities in both word distributions and topic organizations of documents within a single domain. The topic assignments that we produce are encouraged to form consistent orderings thanks to the application of the Generalized Mallows Model, a flexible yet tractable distribution over discrete permutations. This stands in contrast to previous approaches that have typically made Markovian assumptions between adjacent discourse units in a text. Inference in this model is performed via a collapsed Gibbs sampling algorithm that uses a slice sampling subcomponent to estimate Mallows model parameters.

  Empirically, we applied our content model to the tasks of aligning paragraphs between documents, segmenting text within documents, and ordering new in-domain documents. On each of these tasks we found that the permutation-based approach yields improved performance compared to state-of-the-art baselines.

Furthermore, we found that the Generalized Mallows Model in particular is an appropriate modeling choice for the ordering component of content structure, as using more or less strongly constrained permutation models eroded performance on most tasks.

- **Relation Discovery**: The approach we proposed for relation discovery leverages regularities at the lexicographic, syntactic, and document structure levels. The relation phrases that it discovers are encouraged to cohere at each of these layers of linguistic phenomena. For lexicography, document position, and local syntactic properties, this coherence is a direct result of the generative process. The model posterior is estimated with variational inference, but with a twist — we apply declarative constraints during inference through posterior regularization, allowing the model to institute more global biases, particularly on syntax, that are difficult to express in the generative process itself.

  Our model's evaluation showed that our approach is better able to recover relation structure, as evaluated on both the token and sentence level, compared to several alternatives. We demonstrated that the declarative constraint approach is particularly crucial to the success of the model, as removing any of the constraint sets degraded performance drastically. Finally, we showed that, at the sentence level, our model's performance is competitive even compared to techniques that had access to supervised training data.

- **Semantic Property Prediction**: The model we introduced for predicting semantic properties relies on a form of noisy "supervision" often available alongside the raw text, specifically free-text annotations written by the document authors themselves. By leveraging regularities in these annotations, as well as the text itself, our model is able to predict multiple domain-specific properties. To account for the inconsistency and incompleteness of the annotations, we proposed a joint model that simultaneously finds a hidden clustering over annotations and distributions over text using one set of latent variables. Inference is conducted using an efficient Gibbs sampler.

The experiments on our semantic property model were conducted in both single- and multi-document scenarios. Our single-document experiments showed that the model is able to predict more precise properties and annotation clusters compared to a series of alternative approaches and simpler model variants. Our multi-document experiments demonstrated the applicability of this approach to multi-document summarization. Finally, we deployed the PRÉCIS browser online, providing an easy interface to explore hundreds of thousands of reviews over tens of thousands of products.

5.1 Future Work

At the end of each individual chapter we alluded to natural extensions of the task-specific models. Here, we describe directions of future work that relate generally to the ideas for in-domain semantic analysis explored in this thesis.

- **Joint Modeling of Multiple Granularities** The models we presented in this thesis operate in isolation to induce structure at different granularities of text. An intriguing area of future work is to explore how these kinds of structures can be learned *jointly* in a single framework. Such a joint model could ensure that the induced relations, content structure, and semantic properties are all consistent with one another. In supervised contexts, joint modeling of multiple tasks has improved performance for extraction tasks [119] as well as syntactic parsing [50]. In our setup, joint modeling could require, for example, that a paragraph whose topic is about *battery* should discuss mostly relations specific to properties of the battery, such as wattage and life.

  These kinds of consistency requirements have the potential to produce better analyses at each level of granularity. Conversely, however, the search space of possible structures grows exponentially with the number of tasks to be performed jointly. Being able to effectively search this space is a key technical hurdle for successful joint learning.

- **Semi-supervised Semantic Analysis** As we hinted at with the relation dis-

covery work, having a small number of annotated examples can be beneficial for performance. Due to their generative formulations, the models we have proposed can all utilize data in a semi-supervised manner by simply incorporating them as observed variables. However, as we saw with the relation discovery work, this does not always lead to significant performance gains compared to discriminative supervised approaches. There are a number of reasons for this performance discrepancy that we touched upon in Chapter 3. Finding how to mitigate these limitations would allow us to design one unified model that operates in a range of supervision regimes, and is a compelling area of future work.

- **Hierarchical Semantic Structures** Finally, the kinds of semantic structure we examined in this thesis are relatively shallow, *i.e.,* the hidden structures are flat. An interesting extension is to consider hierarchical hidden structures. For content models, this would mean finding nested topics in text, such as *rail*, *road*, and *air* within *transportation* in a collection of cities articles. Relation types can also be nested to form full trees, in the vein of deep semantic parsers [104, 138]. Documents also exhibit multiple granularities of properties; within a *good service* property of a restaurant review domain, there may be aspects of *good host*, *good bartenders*, and *good waitstaff*. A review can certainly praise one aspect while deriding another, but they are typically correlated, which can serve as a strong prior for learning multi-level properties. Learning deeper representations allow for a better understanding of the semantics of the text, but introduces significant complexity to the inference problem. Future work will have to address this challenge to successfully learn such deep structures.

Appendix A

Development and Test Set Statistics for the Semantic Properties Experiments

Table A.1 lists the semantic properties for each domain and the number of documents that are used for evaluating each of these properties. As noted in Section 4.6.1, the gold standard evaluation is complete, testing every property with each document. Conversely, the free-text evaluations for each property only use documents that are annotated with the property or its antonym — this is why the number of documents differs for each semantic property.

Domain	Property	Development documents	Test Documents
Restaurants (gold)	*All properties*	50	120
Restaurants	Good food Bad food	88	179
	Good price Bad price	31	66
	Good service Bad service	69	140
Cell Phones	Good reception Bad reception	33	67
	Good battery life Poor battery life	59	120
	Good price Bad price	28	57
Cameras	Small Large	84	168
	Good price Bad price	56	113
	Good battery life Poor battery life	51	102
	Great zoom Limited zoom	34	69

Table A.1: Breakdown by property for the development and test sets used for the evaluations in section 4.6.2.

Appendix B

Additional Multiple Review Summarization Results for the Semantic Properties Model

Table B.1 lists results of the multi-document experiment, with a variation on the aggregation — we require each automatic method to predict a property for three of five reviews to predict that property for the product, rather than two as presented in Section 4.7. For the baseline systems, this change causes a precipitous drop in recall, leading to F-score results that are substantially worse than those presented in Section 4.7.3. In contrast, the F-score for our model is consistent across both evaluations.

Method	Recall	Prec.	F-score
Our model	0.726	0.365	**0.486**
Keyphrase aggregation	0.000	0.000	0.000 $*$
Model cluster aggregation	0.024	1.000	0.047 $*$
Gold cluster aggregation	0.036	1.000	0.068 $*$
Indep. cluster aggregation	0.036	1.000	0.068 $*$

Table B.1: Comparison of the aggregated property predictions made by our semantic properties model and a series of baselines that only use free-text annotations. Aggregation requires three of five reviews to predict a property, rather than two as in Section 4.7. The methods against which our model has significantly better results using approximate randomization are indicated with $*$ for $p \leq 0.05$.

Bibliography

[1] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of DL*, 2000.

[2] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM*, 55(5):23:1–23:27, 2008.

[3] Ernst Althaus, Nikiforos Karamanis, and Alexander Koller. Computing locally coherent discourses. In *Proceedings of ACL*, 2004.

[4] Daniel Gildea andiel Dan Jurasfky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, 2002.

[5] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction from the web. In *Proceedings of IJCAI*, 2007.

[6] Michele Banko and Oren Etzioni. The tradeoffs between open and traditional relation extraction. In *Proceedings of ACL*, 2008.

[7] Frederic C. Bartlett. *Remembering: a study in experimental and social psychology.* Cambridge University Press, 1932.

[8] Regina Barzilay and Noemie Elhadad. Sentence alignment for monolingual comparable corpora. In *Proceedings of EMNLP*, 2003.

[9] Regina Barzilay, Noemie Elhadad, and Kathleen McKeown. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55, 2002.

[10] Regina Barzilay and Mirella Lapata. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34, 2008.

[11] Regina Barzilay and Lillian Lee. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of HLT/NAACL*, 2004.

[12] Regina Barzilay, Kathleen McKeown, and Michael Elhadad. Information fusion in the context of multi-document summarization. In *Proceedings of ACL*, 1999.

[13] Regina Barzilay and Kathleen R. McKeown. Extracting paraphrases from a parallel corpus. In *Proceedings of ACL*, 2001.

[14] Doug Beeferman, Adam Berger, and John D. Lafferty. Statistical models for text segmentation. *Machine Learning*, 34:177–210, 1999.

[15] Kedar Bellare and Andrew McCallum. Generalized expectation criteria for bootstrapping extractors using record-text alignment. In *Proceedings of EMNLP*, 2009.

[16] Taylor Berg-Kirkpatrick, Alexandre Bouchard-Cote, John DeNero, and Dan Klein. Painless unsupervised learning with features. In *Proceedings of HLT/NAACL*, 2010.

[17] José M. Bernardo and Adrian F.M. Smith. *Bayesian Theory.* Wiley Series in Probability and Statistics, 2000.

[18] Indrajit Bhattacharya and Lise Getoor. A latent Dirichlet model for unsupervised entity resolution. In *Proceedings of ICDM*, 2006.

[19] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[20] David M. Blei and John D. Lafferty. Correlated topic models. In *Advances in NIPS*, 2006.

[21] David M. Blei and Jon McAuliffe. Supervised topic models. In *Advances in NIPS*, 2008.

[22] David M. Blei and Pedro J. Moreno. Topic segmentation with an aspect hidden markov model. In *Proceedings of SIGIR*, 2001.

[23] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[24] Danushka Bollegala, Naoaki Okazaki, and Mitsuru Ishizuka. A bottom-up approach to sentence ordering for multi-document summarization. In *Proceedings of ACL/COLING*, 2006.

[25] Jordan Boyd-Graber, David Blei, and Xiaojin Zhu. A topic model for word sense disambiguation. In *Proceedings of EMNLP*, 2007.

[26] Jordan Boyd-Graber and David M. Blei. Syntactic topic models. In *Advances in NIPS*, 2008.

[27] S.R.K. Branavan, Harr Chen, Jacob Eisenstein, and Regina Barzilay. Learning document-level semantic properties from free-text annotations. In *Proceedings of ACL*, 2008.

[28] S.R.K. Branavan, Harr Chen, Jacob Eisenstein, and Regina Barzilay. Learning document-level semantic properties from free-text annotations. *Journal of Artificial Intelligence Research*, 34:569–603, 2009.

[29] Sergey Brin. Extracting patterns and relations form the World-Wide Web. In *Proceedings of WebDB*, 1998.

[30] Razvan C. Bunescu and Raymond J. Mooney. Learning to extract relations from the web using minimal supervision. In *Proceedings of ACL*, 2007.

[31] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

[32] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of ACM SIGIR*, 1998.

[33] Ming-Wei Chang, Lev Ratinov, and Dan Roth. Guiding semi-supervision with constraint-driven learning. In *Proceedings of ACL*, 2007.

[34] Harr Chen, S.R.K. Branavan, Regina Barzilay, and David R. Karger. Content modeling using latent permutations. *Journal of Artificial Intelligence Research*, 36:129–163, 2009.

[35] Harr Chen, S.R.K. Branavan, Regina Barzilay, and David R. Karger. Global models of document structure using latent permutations. In *Proceedings of HLT/NAACL*, 2009.

[36] Jinxiu Chen, Dong-Hong Ji, Chew Lim Tan, and Zheng-Yu Niu. Automatic relation extraction with model order selection and discriminative label identification. In *Proceedings of IJCNLP*, 2005.

[37] Nancy Chinchor. Statistical significance of MUC-6 results. In *Proceedings of the Conference on Message Understanding*, 1995.

[38] Nancy Chinchor, David D. Lewis, and Lynette Hirschman. Evaluating message understanding systems: An analysis of the third message understanding conference (MUC-3). *Computational Linguistics*, 19(3):409–449, 1993.

[39] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[40] William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.

[41] Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL recognising textual entailment challenge. *Lecture Notes in Computer Science*, 3944:177–190, 2006.

[42] Marie-Catherine de Marneffe and Christopher D. Manning. The stanford typed dependencies representation. In *Proceedings of the COLING Workshop on Cross-framework and Cross-domain Parser Evaluation*, 2008.

[43] Gregory Druck, Gideon Mann, and Andrew McCallum. Semi-supervised learning of dependency parsers using generalized expectation criteria. In *Proceedings of ACL*, 2009.

[44] Jacob Eisenstein and Regina Barzilay. Bayesian unsupervised topic segmentation. In *Proceedings of EMNLP*, 2008.

[45] Noemie Elhadad and Kathleen R. McKeown. Towards generating patient specific summaries of medical articles. In *Proceedings of NAACL Workshop on Automatic Summarization*, pages 32–40, 2001.

[46] Micha Elsner, Joseph Austerweil, and Eugene Charniak. A unified local and global model for discourse coherence. In *Proceedings of HLT/NAACL*, 2007.

[47] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: an experimental study. *Artificial Intelligence*, 165(1):91–134, 2005.

[48] Charles J. Fillmore and Collin F. Baker. A frame approach to semantic analysis. In Bernd Heine and Heiko Narrog, editors, *Oxford Handbook of Linguistic Analysis*. Oxford University Press, January 2010.

[49] Jenny R. Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of ACL*, 2005.

[50] Jenny Rose Finkel and Christopher D. Manning. Joint parsing and named entity recognition. In *Proceedings of HLT/NAACL*, 2009.

[51] M.A. Fligner and J.S. Verducci. Distance based ranking models. *Journal of the Royal Statistical Society, Series B*, 48(3):359–369, 1986.

[52] Michel Galley, Kathleen R. McKeown, Eric Fosler-Lussier, and Hongyan Jing. Discourse segmentation of multi-party conversation. In *Proceedings of ACL*, 2003.

[53] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Texts in Statistical Science. Chapman & Hall/CRC, 2nd edition, 2004.

[54] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:609–628, 1984.

[55] Ana-Maria Giuglea and Alessandro Moschitti. Semantic role labeling via framenet, verbnet and propbank. In *Proceedings of COLING/ACL*, 2006.

[56] Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. Contextual dependencies in unsupervised word segmentation. In *Proceedings of ACL*, 2006.

[57] João Graça, Kuzman Ganchev, and Ben Taskar. Expectation maximization and posterior constraints. In *Advances in NIPS*, 2007.

[58] A. Graesser, M. Gernsbacher, and S. Goldman, editors. *Handbook of Discourse Processes.* Erlbaum, 2003.

[59] Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101:5228–5235, 2004.

[60] Thomas L. Griffiths, Mark Steyvers, David M. Blei, and Joshua B. Tenenbaum. Integrating topics and syntax. In *Advances in NIPS*, 2005.

[61] Ralph Grishman and Beth Sundheim. Message understanding conference - 6: A brief history. In *Proceedings of COLING*, 1996.

[62] Barbara J. Grosz and Candace L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.

[63] Amit Gruber, Michal Rosen-Zvi, and Yair Weiss. Hidden topic markov models. In *Proceedings of AISTATS*, 2007.

[64] M. A. K. Halliday and Ruqaiya Hasan. *Cohesion in English.* Longman, 1976.

[65] Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. Discovering relations among named entities from large corpora. In *Proceedings of ACL*, 2004.

[66] Marti Hearst. Multi-paragraph segmentation of expository text. In *Proceedings of ACL*, 1994.

[67] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of SIGKDD*, 2004.

[68] Paul D. Ji and Stephen Pulman. Sentence ordering with manifold-based classification in multi-document summarization. In *Proceedings of EMNLP*, 2006.

[69] Thorsten Joachims. *Making Large-Scale Support Vector Machine Learning Practical*, pages 169–184. MIT Press, 1999.

[70] Richard Johansson and Pierre Nugues. Extended constituent-to-dependency conversion for english. In *Proceedings of NODALIDA*, 2007.

[71] Mark Johnson. Why doesn't EM find good HMM POS-taggers? In *Proceedings of EMNLP*, 2007.

[72] Michael I. Jordan. *Learning in Graphical Models.* Adaptive Computation and Machine Learning. MIT Press, 1998.

[73] Nikiforos Karamanis, Massimo Poesio, Chris Mellish, and Jon Oberlander. Evaluating centering-based metrics of coherence for text structuring using a reliably annotated corpus. In *Proceedings of ACL*, 2004.

[74] Soo-Min Kim and Eduard Hovy. Automatic identification of pro and con reasons in online reviews. In *Proceedings of COLING/ACL*, 2006.

[75] Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. Extending VerbNet with novel verb classes. In *Proceedings of LREC*, 2006.

[76] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of ACL*, 2003.

[77] Alexandre Klementiev, Dan Roth, and Kevin Small. Unsupervised rank aggregation with distance-based models. In *Proceedings of the ICML*, 2008.

[78] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, 2001.

[79] Mirella Lapata. Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of ACL*, 2003.

[80] Mirella Lapata. Automatic evaluation of information ordering: Kendall's tau. *Computational Linguistics*, 32(4):471–484, 2006.

[81] Alberto Lavelli, Mary Elaine Califf, Fabio Ciravegna, Dayne Freitag, Claudio Giuliano, Nicholas Kushmerick, Lorenza Romano, and Neil Ireson. Evaluation of machine learning-based information extraction algorithms: criticisms and recommendations. *Language Resources and Evaluation*, 42(4):361–393, 2008.

[82] Guy Lebanon and John Lafferty. Cranking: combining rankings using conditional probability models on permutations. In *Proceedings of ICML*, 2002.

[83] Dekang Lin and Patrick Pantel. DIRT - discovery of inference rules from text. In *Proceedings of SIGKDD*, 2001.

[84] Bing Liu, Minqing Hu, and Junsheng Cheng. Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of WWW*, 2005.

[85] Yue Lu and ChengXiang Zhai. Opinion integration through semi-supervised topic modeling. In *Proceedings of WWW*, 2008.

[86] Igor Malioutov and Regina Barzilay. Minimum cut model for spoken lecture segmentation. In *Proceedings of ACL*, 2006.

[87] Colin L. Mallows. Non-null ranking models. *Biometrika*, 44:114–130, 1957.

[88] Inderjeet Mani and Eric Bloedorn. Multi-document summarization by graph search and matching. In *Proceedings of AAAI*, 1997.

[89] Gideon S. Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *Proceedings of ACL*, 2008.

[90] Erwin Marsi and Emiel Krahmer. Explorations in sentence fusion. In *Proceedings of the European Workshop on Natural Language Generation*, 2005.

[91] Andrew McCallum, Kedar Bellare, and Fernando Pereira. A conditional random field for discriminatively-trained finite-state string edit distance. In *Proceedings of UAI*, 2005.

[92] Andrew McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval Journal*, 3(2):127–163, 2000.

[93] Marina Meilă and Le Bao. Estimation and clustering with infinite rankings. In *Proceedings of UAI*, 2008.

[94] Marina Meilă and Harr Chen. Dirichlet process mixtures of generalized Mallows models. In *Proceedings of UAI*, 2010.

[95] Marina Meilă, Kapil Phadnis, Arthur Patterson, and Jeff Bilmes. Consensus ranking under the exponential model. In *Proceedings of UAI*, 2007.

[96] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL/IJCNLP*, 2009.

[97] Radford M. Neal. Slice sampling. *Annals of Statistics*, 31:705–767, 2003.

[98] Rani Nelken and Stuart M. Shieber. Towards robust context-sensitive sentence alignment for monolingual corpora. In *Proceedings of EACL*, 2006.

[99] Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of SIGIR*, 2006.

[100] Eric Noreen. *Computer-Intensive Methods for Testing Hypotheses: An Introduction.* John Wiley and Sons, 1989.

[101] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus with semantic roles. *Computational Linguistics*, 31(1):71–106, 2005.

[102] Fuchun Peng and Andrew McCallum. Accurate information extraction from research papers using ocnditional random fields. In *Proceedings of HLT/NAACL*, 2004.

[103] Lev Pevzner and Marti A. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28:19–36, 2002.

[104] Hoifung Poon and Pedro Domingos. Unsupervised semantic parsing. In *Proceedings of EMNLP*, 2009.

[105] Ana-Maria Popescu, Bao Nguyen, and Oren Etzioni. OPINE: Extracting product features and opinions from reviews. In *Proceedings of HLT/EMNLP*, pages 339–346, 2005.

[106] Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. Fast collapsed Gibbs sampling for latent Dirichlet allocation. In *Proceedings of SIGKDD*, 2008.

[107] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. The Penn discourse treebank 2.0. In *Proceedings of LREC*, 2008.

[108] Matthew Purver, Konrad Körding, Thomas L. Griffiths, and Joshua B. Tenenbaum. Unsupervised topic modelling for multi-party spoken discourse. In *Proceedings of ACL/COLING*, 2006.

[109] Dragomir R. Radev, Hongyan Jing, and Malgorzata Budzikowska. Centroid-based summarization of multiple documents: Sentence extraction, utility-based evaluation and user studies. In *Proceedings of ANLP/NAACL Summarization Workshop*, 2000.

[110] Dragomir R. Radev and Kathleen McKeown. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500, September 1998.

[111] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, December 1971.

[112] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[113] Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proceedings of ACL*, 2002.

[114] Stefan Riezler and John T. Maxwell. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005.

[115] Ellen Riloff. Automatically generating extraction patterns from untagged texts. In *Proceedings of AAAI*, 1996.

[116] Ellen Riloff and Rosie Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of AAAI*, 1999.

[117] Benjamin Rosenfeld and Ronen Feldman. Clustering for unsupervised relation identification. In *Proceedings of CIKM*, 2007.

[118] Dan Roth and Wen-tau Yih. A linear programming formulation for global inference in natural language tasks. In *Proceedings of CoNLL*, 2004.

[119] Christina Sauper, Aria Haghighi, and Regina Barzilay. Incorporating content structure into text analysis applications. In *Proceedings of EMNLP*, 2010.

[120] Deborah Schiffrin, Deborah Tannen, and Heidi E. Hamilton, editors. *The Handbook of Discourse Analysis*. Blackwell, 2001.

[121] Yusuke Shinyama and Satoshi Sekine. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of HLT/NAACL*, 2006.

[122] Benjamin Snyder and Regina Barzilay. Multiple aspect ranking using the good grief algorithm. In *Proceedings of NAACL/HLT*, 2007.

[123] Radu Soricut and Daniel Marcu. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of HLT/NAACL*, 2003.

[124] Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. An improved extraction pattern representation model for automatic IE pattern acquisition. In *Proceedings of ACL*, 2003.

[125] Ivan Titov and Ryan McDonald. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL*, pages 308–316, 2008.

[126] Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of WWW*, pages 111–120, 2008.

[127] Kristina Toutanova and Mark Johnson. A Bayesian LDA-based model for semi-supervised part-of-speech tagging. In *Advances in NIPS*, 2008.

[128] Masao Utiyama and Hitoshi Isahara. A statistical model for domain-independent text segmentation. In *Proceedings of ACL*, 2001.

[129] Paul van Mulbregt, Ira Carp, Lawrence Gillick, Steve Lowe, and Jon Yamron. Text segmentation and topic tracking on broadcast news via a hidden markov model approach. In *Proceedings of ICSLP*, 1998.

[130] Hanna M. Wallach. Topic modeling: beyond bag of words. In *Proceedings of ICML*, 2006.

[131] Michael White, Tanya Korelsky, Claire Cardie, Vincent Ng, David Pierce, and Kiri Wagstaff. Multi-document summarization via information extraction. In *Proceedings of HLT*, 2001.

[132] Alison Wray. *Formulaic Language and the Lexicon.* Cambridge University Press, Cambridge, 2002.

[133] Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of COLING*, 2000.

[134] Limin Yao, Sebastian Riedel, and Andrew McCallum. Collective cross-document relation extraction without labelled data. In *Proceedings of EMNLP*, 2010.

[135] Alexander Yates and Oren Etzioni. Unsupervised methods for determining object and relation synonyms on the web. *Journal of Artificial Intelligence Research*, 34:255–296, 2009.

[136] Alexander Yeh. More accurate tests for the statistical significance of result differences. In *Proceedings of COLING*, 2000.

[137] Annie Zaenen. Mark-up barking up the wrong tree. *Computational Linguistics*, 32(4):577–580, 2006.

[138] Luke S. Zettlemoyer and Michael Collins. Learning to map sentences to logical form: structured classification with probabilistic categorial grammars. In *Proceedings of UAI*, 2005.

[139] Min Zhang, Jian Su, Danmei Wang, Guodong Zhou, and Chew Lim Tan. Discovering relations between named entities from a large raw corpus using tree similarity-based clustering. In *Proceedings of IJCNLP*, 2005.

[140] Jun Zhu, Zaiqing Nie, Xiaojing Liu, Bo Zhang, and Ji-Rong Wen. StatSnowball: a statistical approach to extracting entity relationships. In *Proceedings of WWW*, 2009.