

# Content Modeling for Social Media Text

by

Christina Sauper

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

June 2012

© Massachusetts Institute of Technology 2012. All rights reserved.

Author .....  
Department of Electrical Engineering and Computer Science  
May 14, 2012

Certified by .....  
Regina Barzilay  
Associate Professor, Electrical Engineering and Computer Science  
Thesis Supervisor

Accepted by .....  
Leslie A. Kolodziejcki  
Chairman, Department Committee on Graduate Theses



# Content Modeling for Social Media Text

by

Christina Sauper

Submitted to the Department of Electrical Engineering and Computer Science  
on May 14, 2012, in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy

## Abstract

This thesis focuses on machine learning methods for extracting information from user-generated content. Instances of this data such as product and restaurant reviews have become increasingly valuable and influential in daily decision making. In this work, I consider a range of extraction tasks such as sentiment analysis and aspect-based review aggregation. These tasks have been well studied in the context of newswire documents, but the informal and colloquial nature of social media poses significant new challenges.

The key idea behind our approach is to automatically induce the content structure of individual documents given a large, noisy collection of user-generated content. This structure enables us to model the connection between individual documents and effectively aggregate their content. The models I propose demonstrate that content structure can be utilized at both document and phrase level to aid in standard text analysis tasks. At the document level, I capture this idea by joining the original task features with global contextual information. The coupling of the content model and the task-specific model allows the two components to mutually influence each other during learning. At the phrase level, I utilize a generative Bayesian topic model where a set of properties and corresponding attribute tendencies are represented as hidden variables. The model explains how the observed text arises from the latent variables, thereby connecting text fragments with corresponding properties and attributes.

Thesis Supervisor: Regina Barzilay

Title: Associate Professor, Electrical Engineering and Computer Science



## Acknowledgments

I have been extremely fortunate throughout this journey to have the support of many wonderful people. It is only with their help and encouragement that this thesis has come into existence.

First and foremost, I would like to thank my advisor, Regina Barzilay. She has been an incredible source of knowledge and inspiration; she has a sharp insight about interesting directions of research, and she is a strong role model in a field in which women are still a minority. Her high standards and excellent guidance have shaped my work into something I can be truly proud of. My thesis committee members, Peter Szolovits and James Glass, have also provided invaluable discussion as to the considerations of this work applied to their respective fields, analysis of medical data and speech processing.

Working with my coauthor, Aria Haghighi, was an intensely focused and amazing time; he has been a great friend and collaborator to me, and I will certainly miss the late-night coding with Tosci's ice cream. I am very grateful to Alan Woolf, Michelle Zeager, Bill Long, and the other members of the Fairwitness group for helping to make my introduction to the medical world as smooth as possible.

My fellow students at MIT are a diverse and fantastic group of people: my group-mates from the Natural Language Processing group, including Ted Benson, Branavan, Harr Chen, Jacob Eisenstein, Yoong Keok Lee, Tahira Naseem, and Ben Snyder; my officemates, including David Sontag and Paresh Malalur; and my friends from GSB and other areas of CSAIL. They have always been there for good company and stimulating discussions, both related to research and not. I am thankful to all, and I wish them the best for their future endeavors.

Finally, I dedicate this thesis to my family. They have stood by me through good times and bad, and it is through their loving support and their neverending faith in me that I have made it this far.



---

# Contents

---

<b>1</b>	<b>Introduction</b>	<b>17</b>
1.1	Modeling content structure for text analysis tasks . . . . .	23
1.2	Modeling relation structure for informative aggregation . . . . .	25
1.3	Contributions . . . . .	27
1.4	Outline . . . . .	28
<b>2</b>	<b>Modeling content structure for text analysis tasks</b>	<b>31</b>
2.1	Introduction . . . . .	32
2.2	Related work . . . . .	34
2.2.1	Task-specific models for relevance . . . . .	34
2.2.2	Topic modeling . . . . .	35
2.2.3	Discourse models of document structure . . . . .	38
2.3	Model . . . . .	39
2.3.1	Multi-aspect phrase extraction . . . . .	40
2.3.1.1	Problem Formulation . . . . .	40
2.3.1.2	Model overview . . . . .	42
2.3.1.3	Learning . . . . .	44
2.3.1.4	Inference . . . . .	46

2.3.1.5	Leveraging unannotated data . . . . .	47
2.3.2	Multi-aspect sentiment analysis . . . . .	48
2.3.2.1	Problem Formulation . . . . .	48
2.3.2.2	Model . . . . .	49
2.3.2.3	Learning . . . . .	49
2.3.2.4	Inference . . . . .	50
2.4	Data sets . . . . .	50
2.4.1	Multi-aspect sentiment analysis . . . . .	51
2.4.2	Multi-aspect summarization . . . . .	53
2.5	Experiments . . . . .	55
2.5.1	Multi-aspect phrase extraction . . . . .	56
2.5.1.1	Baselines . . . . .	56
2.5.1.2	Evaluation metrics . . . . .	57
2.5.1.3	Results . . . . .	58
2.5.2	Multi-aspect sentiment analysis . . . . .	61
2.5.2.1	Baselines . . . . .	62
2.5.2.2	Evaluation metrics . . . . .	62
2.5.2.3	Results . . . . .	62
2.6	Conclusion . . . . .	63
<b>3</b>	<b>Modeling relation structure for informative aggregation</b>	<b>65</b>
3.1	Introduction . . . . .	66
3.2	Related work . . . . .	71
3.2.1	Single-aspect sentiment analysis . . . . .	71
3.2.2	Aspect-based sentiment analysis . . . . .	72
3.2.2.1	Data-mining and fixed-aspect techniques for sentiment analysis . . . . .	72
3.2.2.2	Multi-document summarization and its application to sentiment analysis . . . . .	74
3.2.2.3	Probabilistic topic modeling for sentiment analysis . . . . .	76



3.3	Problem formulation . . . . .	79
3.3.1	Model components . . . . .	79
3.3.2	Problem setup . . . . .	81
3.4	Model . . . . .	82
3.4.1	General formulation . . . . .	82
3.4.2	Model extensions . . . . .	88
3.5	Inference . . . . .	90
3.5.1	Inference for model extensions . . . . .	96
3.6	Experiments . . . . .	98
3.6.1	Joint identification of aspect and sentiment . . . . .	99
3.6.1.1	Data set . . . . .	99
3.6.1.2	Domain challenges and modeling techniques . . . . .	100
3.6.1.3	Cluster prediction . . . . .	101
3.6.1.4	Sentiment analysis . . . . .	103
3.6.1.5	Per-word labeling accuracy . . . . .	106
3.6.2	Aspect identification with shared aspects . . . . .	109
3.6.2.1	Data set . . . . .	110
3.6.2.2	Domain challenges and modeling techniques . . . . .	110
3.6.2.3	Cluster prediction . . . . .	111
3.7	Conclusion . . . . .	113
<b>4</b>	<b>Conclusion</b>	<b>115</b>
4.1	Future Work . . . . .	116
<b>A</b>	<b>Condensr</b>	<b>119</b>
A.1	System implementation . . . . .	119
A.2	Examples . . . . .	126



---

## List of Figures

---

1-1	A comparison of potential content models for summarization and sentiment analysis . . . . .	19
(a)	A content model for summarization focused on identifying topics in text. . . . .	19
(b)	A content model for sentiment analysis focused on identifying sentiment-bearing sentences in text. . . . .	19
1-2	Three reviews about the same restaurant discussing different aspects and opinions. . . . .	21
1-3	An excerpt from a DVD review demonstrating ambiguity which can be resolved through knowledge of structure. . . . .	22
1-4	Sample text analysis tasks to which document structure is introduced.	24
(a)	Example system input and output on the multi-aspect sentiment analysis task . . . . .	24
(b)	Example system input and output on the multi-aspect phrase extraction domain . . . . .	24
1-5	An example of the desired input and output of our system in the restaurant domain, both at corpus-level and at snippet-level. . . . .	26
(a)	Example of system input and output. . . . .	26

(b)	Ideal per-word labeling for several snippets. . . . .	26
2-1	An excerpt from a DVD review illustrating ambiguous aspect of a sentiment word. . . . .	32
2-2	A graphical depiction of our model for sequence labeling tasks, as described in Section 2.3. . . . .	41
2-3	A graphical depiction of the generative process for a labeled document at training time, as described in Section 2.3. . . . .	43
2-4	Sample labeled text from the multi-aspect sentiment corpus. . . . .	51
2-5	Sample labeled text from the three multi-aspect summarization corpora.	52
(a)	Sample labeled text from Amazon HDTV reviews . . . . .	52
(b)	Sample labeled text from Yelp restaurant reviews . . . . .	52
(c)	Sample labeled text from medical summaries . . . . .	52
2-6	Annotation procedure for the multi-aspect phrase labeling task. . . .	55
2-7	Results for multi-aspect phrase extraction on the Amazon corpus using the complete annotated set with varying amounts of additional unlabeled data. . . . .	59
2-8	Results for multi-aspect phrase extraction on the Amazon corpus using half of the annotated training documents with varying amounts of additional unlabeled data. . . . .	60
2-9	Screenshot of CONDENSr, our demo system. . . . .	61
3-1	An example of the desired input and output of our system in the restaurant domain. . . . .	68
3-2	Example clusters of restaurant review snippets generated by a lexical clustering algorithm. . . . .	69
3-3	Labeled model components from the example in Figure 3-1. . . . .	80
3-4	A summary of the generative model presented in Section 3.4.1. . . . .	84
3-5	A graphical description of the model presented in Section 3.4.1. . . .	85
3-6	A graphical description of the model with shared aspects presented in Section 3.4.2. . . . .	91

3-7	The mean-field variational algorithm used during learning and inference procedure, as described in Section 3.5. . . . .	92
3-8	Variational inference update steps. . . . .	94
	(a) Inference procedure for snippet aspect, $Z_A^{i,j}$ . . . . .	94
	(b) Inference procedure for snippet value, $Z_V^{i,j}$ . . . . .	94
	(c) Inference procedure for word topic, $Z_W^{i,j,w}$ . . . . .	94
3-9	Example snippets from the review data set. . . . .	101
3-10	Training curve of DISCRIMINATIVE baseline as number of training examples increases. . . . .	106
3-11	An illustration of the tree expansion procedure for value words. . . . .	108
3-12	Example snippets from the medical data set. . . . .	111
A-1	Map interface on CONDENSr. . . . .	120
A-2	Browsing interface on CONDENSr. . . . .	122
A-3	Several complete tooltips from a variety of restaurants on CONDENSr. . . . .	128
	(a) The Beehive . . . . .	128
	(b) The Anaheim White House . . . . .	129
	(c) Herbivore . . . . .	130
	(d) Luigi’s D’italia . . . . .	131
	(e) Valencia Pizza & Pasta . . . . .	132
	(f) Grendel’s Den Restaurant & Bar . . . . .	133
	(g) The Helmand . . . . .	134



---

## List of Tables

---

2.1	A summary of notation for the multi-aspect phrase extraction task. . .	42
2.2	A summary of data set statistics. . . . .	50
2.3	Results for multi-aspect phrase extraction on the Yelp corpus. . . . .	57
2.4	Results for multi-aspect phrase extraction on the medical corpus. . .	57
2.5	Results for multi-aspect phrase extraction on the Amazon corpus. . .	58
2.6	Error rate on multi-aspect sentiment ranking. . . . .	62
3.1	Mathematical notation introduced in this chapter. . . . .	83
3.2	Seed words used by the model for the restaurant corpus. . . . .	100
3.3	Results for the cluster prediction task on both domains, using the MUC metric. . . . .	103
3.4	Sentiment prediction accuracy of our model compared to the DISCRIM- INATIVE and SEED baselines. . . . .	105
3.5	Correct annotation of a set of phrases containing elements which may be confusing. These phrases are used to test annotators before they are accepted to annotate the actual test data. . . . .	107
3.6	Per-word labeling precision and recall of our model compared to the TAGS-SMALL and TAGS-FULL baselines. . . . .	109

3.7	Examples of high-precision, low-recall aspect word labeling by our full model. . . . .	109
3.8	Results for the cluster prediction task on both domains, using the MUC metric. . . . .	112
A.1	The full set of terms and locations used for CONDENSr, as well as the final count of restaurants for each major area. . . . .	124
	(a) Set of locations, grouped by major metropolitan area. . . . .	124
	(b) Set of search terms. . . . .	124
	(c) Number of restaurants for each area. . . . .	124



# CHAPTER 1

---

## Introduction

---

Across the Internet, there is a growing collection of user-generated text data in many domains: product and restaurant reviews, personal blogs, stories about events, and many more. The sheer amount of data can be overwhelming for users who may want a quick summary of product features or advice on dish selection at a restaurant. The data found in social media is quite different than traditional formal news text, and therefore it poses several new challenges: First, there is a notable lack of structure in social media text. While professional or formal writing has a generally well-defined layout with a logical flow of topics through paragraphs and sentences, social media is far more disorganized. Second, social media text is fraught with typos and novel words. Inventions such as *OMG* or *awesometastic* are easily understandable in context to human readers; however, there is no way to anticipate every novel word in the training data for supervised tasks. Third, the text often requires situational context for successful interpretation. Rather than including a full explanation with background information, authors of social media text often assume a significant amount of knowledge. Addressing these challenges is crucial for the success of text analysis applications.

One way to approach this problem is to induce a representation of document structure useful for the task at hand. For example, consider the text in Figure 1-1. As an example application, to summarize this text, it is crucial that a high-quality summary have good coverage of the information discussed. One approach would be to rank sentences in terms of their ‘goodness’ for a final summary, then select the top sentences; however, this does not guarantee that we get a summary with good coverage of the topics. By utilizing a content model, we can learn a structure based on the underlying topic of the text (Figure 1-1a); e.g., we can learn that the first sentence discusses the quality of the *food*, the second part mentions their opinion of *ambiance*, and so on. Then, we can select sentences from each area to ensure the completeness of our summary. Another possible application would be to predict the sentiment of the text, one approach would be to use a standard binary classifier (positive vs. negative) over words in the document; however, because only the subjective sentences are relevant to this distinction, the objective sentences simply introduce noise. As in the summarization case, we can leverage the document content to improve the performance on the task; however, in this case, a different representation will be helpful. Instead, we can learn a model of structure designed to identify sentiment-bearing sentences (Figure 1-1b), then apply our sentiment analysis to those sentences only. This will eliminate the noise from objective sentences. As we examine more tasks, we find that each text analysis task receives benefit from a different model of document structure.

These types of content models have been well-studied for a variety of discourse tasks, such as sentence ordering and extractive summarization. It is well-known that even basic models of document structure can boost the performance of these tasks. However, even for discourse tasks, there are many different potential models of document structure to choose from. For summarization, one approach focuses on identifying topics, as mentioned above. Alternatively, we could use one of the traditional discourse models. For example, Rhetorical Structure Theory (RST) [52] focuses on describing the organization of text in terms of key discourse relations, such as *Causality* to describe a cause and effect relationship. When combined, these

<sup>1</sup>OK food. <sup>2</sup>Nice atmosphere. <sup>2</sup>I would go mostly for the ambiance.  
 We had an early dinner after returning from Logan airport. <sup>3</sup>The staff was friendly and seated us early.  
<sup>1</sup>We had the crispy meatballs and salad for appetizers. <sup>1</sup>The salad was fresh and tasty, but the meatballs lacked taste. <sup>1</sup>My girlfriend had seared sea scallops, which also lacked taste though they were cooked right. <sup>1</sup>I had seafood risotto. <sup>1</sup>The taste was (surprise!) bland...but there was plenty of squid and good chuck of lobster.  
<sup>1</sup>Overall, food was unimpressive. <sup>2</sup>The best thing about this restaurant is the ambiance and hotel decor, as it is situated within Hotel Marlowe, which has a chic lobby. It beats dining at the Cheesecake Factory.

(a) A content model for summarization focusing on identifying topics in text. Sentences describing food are colored orange and labeled with 1, describing atmosphere are colored green and labeled with 2, and describing service are colored purple and labeled with 3.

OK food. Nice atmosphere. I would go mostly for the ambiance.  
 We had an early dinner after returning from Logan airport. The staff was friendly and seated us early.  
 We had the crispy meatballs and salad for appetizers. The salad was fresh and tasty, but the meatballs lacked taste. My girlfriend had seared sea scallops, which also lacked taste though they were cooked right. I had seafood risotto. The taste was (surprise!) bland...but there was plenty of squid and good chuck of lobster.  
Overall, food was unimpressive. The best thing about this restaurant is the ambiance and hotel decor, as it is situated within Hotel Marlowe, which has a chic lobby. It beats dining at the Cheesecake Factory.

(b) A content model for sentiment analysis focused on identifying sentiment-bearing sentences in text. Sentiment-bearing sentences are colored blue.

Figure 1-1: A comparison of potential content models for summarization and sentiment analysis. Note that the two content models are incompatible; some information relevant to one model is completely irrelevant to the other. For example, several sentences about food are purely descriptive, rather than sentiment-bearing.

relations form a tree describing the hierarchy of content organization in a document, which can then be used to distinguish critical pieces of information from the auxiliary ones. Additionally, the concept of Entity Grids [1] can be leveraged when coherence is a driving factor in the task, such as sentence ordering or the evaluation of summary coherence. Each of these models of structure is beneficial for a variety of tasks, but there is no one standard model of content or discourse structure [84]. When compared to a field like syntax, where structure is well-defined and there is little disagreement as to what constitutes a valid syntactic model, the number of possible choices for modeling content structure is overwhelming. There is a gap between the structure provided by these models and the needs of the applications we would like to address.

To address the challenges of these tasks on social media text, this dissertation explores two main hypotheses:

*Learning document structure in an application-specific manner improves end task performance.* It is difficult to predefine what structure may be beneficial for a given application; however, we demonstrate that it is possible to learn an appropriate content model automatically. By jointly learning a content model in an unsupervised fashion with a traditional supervised task, we can specifically tailor the content model to boost task performance. This formulation allows us to capture the relevant structure that exists in loosely-structured documents such as those in social media without having to pre-specify what format that structure should take.

*Modeling the structure of relations in text allows informative aggregation across multiple documents.* We define text relations as consisting of an aspect (e.g., a property or main discussion point) and a value (e.g., sentiment or other information directly tied to the aspect). With this definition, we can design a flexible model for effective minimally-supervised content aggregation able to discover specific, fine-grained aspects and their respective values. Rather than creating a pipeline model as in previous work, our intuition is that learning aspect and value jointly and leveraging information from the entire data set improves the performance of both aspect selection and value identification.

OK food. Nice atmosphere. I would go mostly for the ambiance.

We had an early dinner after returning from Logan airport. The staff was friendly and seated us early.

We had the crispy meatballs and salad for appetizers. The salad was fresh and tasty, but the meatballs lacked taste. My girlfriend had seared sea scallops, which also lacked taste though they were cooked right. I had seafood risotto. The taste was (surprise!) bland...but there was plenty of squid and good chuck of lobster.

Overall, food was unimpressive. The best thing about this restaurant is the ambiance and hotel decor, as it is situated within Hotel Marlowe, which has a chic lobby. It beats dining at the Cheesecake Factory.

(a)

Let's start with the restaurant itself. It's inside Hotel Marlowe near the Galleria. Dimly lit for ambiance with rustic dark wood tables and a flameless candle. It's different...can't decide whether I like it or not though.

Service was average IMO...

To start if off we got an order of Arancini, a couple orders of Fried Squid and Lobster Bisque. The waiter brought out a dish that looked like 3 meatballs with tomato sauce...I thought to myself "we ordered meatballs??" That was the Arancini. It was alright...crispy, almost crunchy on the outside and the inside was...like a meatball.

For entree - I got the Signature Steak Frites. Let me start off with the fries which were very good. Thin cut and crispy...similar to Burger King fries (this is a compliment) except it doesn't leave an aftertaste of lard in your mouth. The rest of the dish not so much...

Other entrees at my table were the Shrimp & Clam Linguini (linguini were either made chewy or it was slightly undercooked), Lobster Risotto (pretty good), Pan Seared Scallops (scallops were good but only had about 5 smallish pieces in the dish), Pan Roasted Cod (didn't try it).

(b)

A much cooler choice than other restaurants in the area, chains like Cheesecake Factory and PF Changs are just so blah. If you're looking for someplace American but kind of new, different, and trendy, go to Bambara!

Very nice ambiance. Very friendly service. Menu is somewhat small, but specialized, everything is fresh and seems like it is ready to order. I got the veggie burger and it was seriously the BEST VEGGIE BURGER I've ever had. I also got a margarita, and it was deliciously strong!

I would totally come again for the veggie burger. Yum!

(c)

Figure 1-2: Three reviews about the same restaurant discussing different aspects and opinions. Note that although there is a large difference in structure and content between reviews, there is also significant overlap; e.g., all three reviews discuss the ambiance, service, and various dishes, though they disagree on some aspects.

While not particularly aggressive or immersive in any way, this mix is still expertly orchestrated. . . . Dialogue is clean and clear and never flushed out by other elements. For a light-hearted quirky picture, Anchor Bay has done a **great** job.

Figure 1-3: An excerpt from a DVD review.<sup>1</sup> Given the highlighted sentence alone, we cannot determine what part of the product the sentiment word *great* refers to. Through the use of a content model, we can identify that this section of the review refers to audio quality.

As a first motivating example, consider the tasks of multi-aspect phrase extraction and informative aggregation on review text from Figure 1-2. While these reviews do not follow a rigid format like formal text (e.g., newspaper articles), we can still distinguish structure to the underlying content. There are pieces of each review which talk about food, service, and ambiance, and we see a similar pattern across other restaurant reviews. We can leverage this underlying content structure of the document to assist in the task of multi-aspect phrase extraction; e.g., if a particular phrase is salient and appears in the section of the document discussing food, it is likely food-related as well and should be extracted. We can then design a model for informative aggregation which identifies fine-grained aspects and their sentiment rating based on these extracted phrases; for example, in this data set, we would want to know that the arancini and scallops are generally reviewed negatively, the veggie burger is rated positively, and the lobster risotto has mixed opinion. In order to successfully complete both tasks, we must understand the underlying structure of both overall document content and relations in the text.

A further example can be seen in Figure 1-3. The highlighted sentence is definitely a positive one; however, there is nothing in that sentence alone to distinguish what the positive sentiment refers to. Through the use of a content model, we can discover that this section of the review is discussing audio quality. In both examples, recovering the overall document structure can be the key to successfully distinguishing the signal from the noise. For social media text, it is crucial that we can do so in a flexible fashion, able to adapt to many different tasks which differ greatly in their relevant representation of content.

These two ideas—a generalized method of incorporating content structure into text analysis tasks and a formulation for informative data aggregation—are the main technical contribution of this thesis. I develop these ideas in the context of several text analysis tasks: multi-aspect phrase extraction, multi-aspect sentiment analysis, and informative aggregation. Below, I summarize these approaches and their application to each task.

## 1.1 Modeling content structure for text analysis tasks

Our first task investigates the benefit of content structure for two standard text analysis tasks, namely multi-aspect phrase extraction and multi-aspect sentiment analysis. For the phrase extraction task, we would like to extract phrases for each of several pre-specified aspects; for example, aspects may be *food*, *service*, and *ambiance* in the restaurant domain. For the sentiment analysis task, we would like to rate each aspect (e.g., *movie*, *sound quality*, *picture quality*, and *packaging* for DVD reviews) on a scale from 1 (worst) to 10 (best). Figure 1-4 illustrates the input and output of both tasks.

Both of these tasks are well-studied; however, they are traditionally approached either with no information about overall document structure or with limited relevance judgments to limit the scope of each decision; e.g., using only sentences containing opinions to determine document sentiment. The goal of this work, therefore, is to first describe a flexible framework for incorporation of document structure in a variety of tasks, and second to evaluate the effects of information about content structure on task performance.

We introduce a joint modeling framework which is sufficiently general to express both multi-aspect sentiment analysis and multi-aspect phrase extraction, as well as other analysis tasks. This framework consists of a content model which defines latent variables to represent the content structure coupled with a traditional task-specific model (e.g., linear-chain CRF or linear regression). By combining the models in this

---

<sup>1</sup>Retrieved from <http://bluray.ign.com/articles/107/1079490p2.html>.

<b>Movie</b>	This collection certainly offers some nostalgic fun, but at the end of the day, the shows themselves, for the most part, just don't hold up.	(5)
<b>Video</b>	Regardless, this is a fairly solid presentation, but it's obvious there was room for improvement.	(7)
<b>Audio</b>	Bass is still robust and powerful. Fans should be pleased with this presentation.	(8)
<b>Extras</b>	The deleted scenes were quite lengthy, but only shelled out a few extra laughs.	(4)

(a) Example system input and output on the multi-aspect sentiment analysis task. Each document contains several paragraphs discussing each of four aspects, where the paragraph aspect labels are not given. The text analysis task is to induce a numeric score for each aspect, from 1 (worst) to 10 (best).

<p>...<sup>A</sup>Casual, romantic, french farmhouse inspired.... This is a farm-to-table restaurant with <sup>F</sup>fresh and local ingredients....<sup>S</sup>Our waiter was a true professional, proud of the restaurant and its reputation....</p>	<p><b>A</b> = Atmosphere  <b>F</b> = Food  <b>S</b> = Service  <b>V</b> = Value  <b>O</b> = Overall</p>
<p>We had the six course tasting menu, and it was paced very well. ... the <sup>F</sup>quality of the entree and dessert (phenomenal!) ...<sup>S</sup>wonderful wine advice from our waiter (he suggested a <sup>V</sup>surprisingly affordable wine...). Sure <sup>V</sup>I winced a little at the price tag, but <sup>O</sup>I can't wait to go back.</p>	

(b) Example system input and output on the multi-aspect phrase extraction domain. Each document contains a free-form social media review which may discuss several desired aspects in addition to irrelevant topics. The text analysis task is to identify phrases which discuss any of several pre-defined aspects.

Figure 1-4: Examples of text analysis tasks to which we add document structure; specifically, multi-aspect sentiment analysis and multi-aspect phrase extraction. Using a consistent framework, we add a content model to each task which is learned jointly with the task parameters. The quality of the content model—and therefore, the task performance—can be increased through the addition of unlabeled training data.



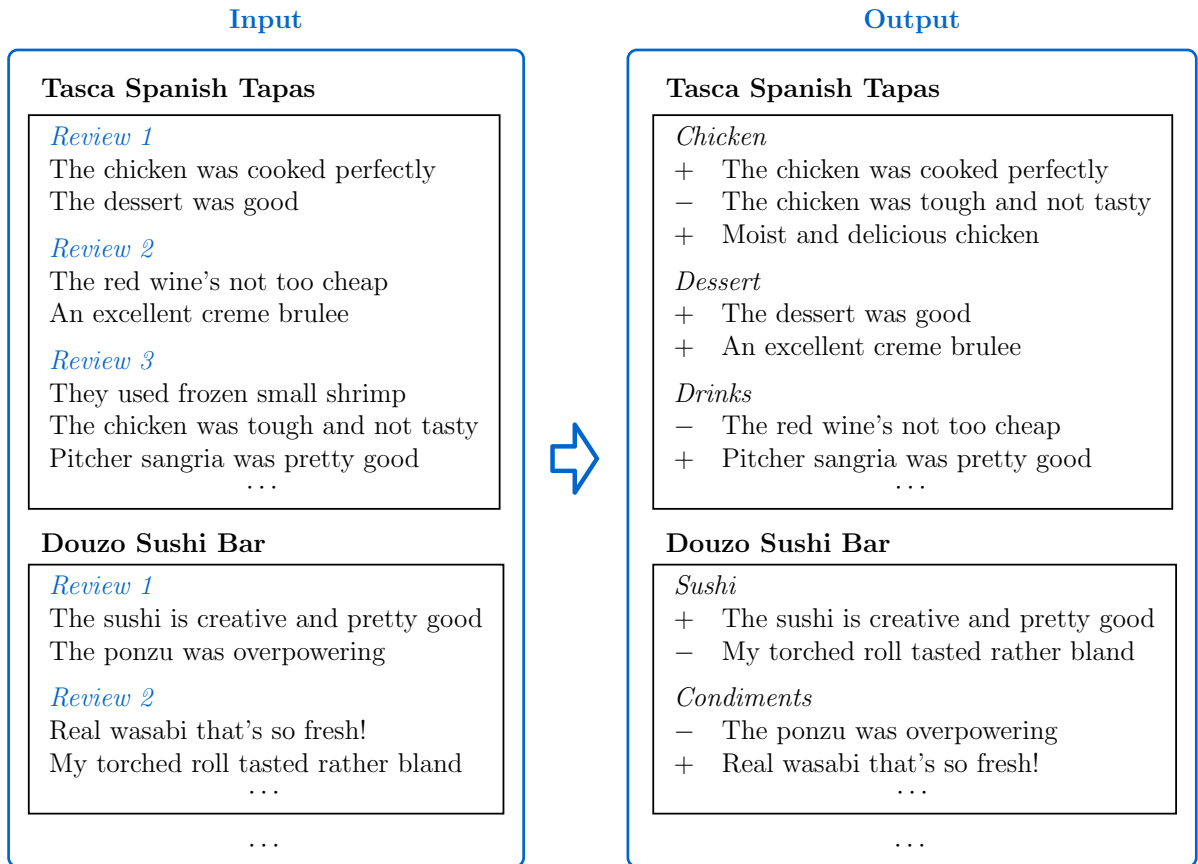
fashion, we allow them to mutually influence each other during learning. Through feedback from the task-specific model, the content model becomes better-suited for the task, while the information about structure from the content model improves the performance of the task-specific model. The combined model can be learned efficiently using a novel EM-based joint training algorithm.

We assume that the task-specific model is given annotated training data standard for the task, but the content model is learned without any annotations. Instead, we show that the model can leverage a large volume of unlabeled data to increase the quality of the content model, which in turn increases the task performance as well.

## 1.2 Modeling relation structure for informative aggregation

Our second task is to provide a mechanism for effective minimally-supervised content aggregation. Specifically, for a given data set, we would like to identify a set of fine-grained aspects representing the main points of discussion and any additional information relevant to those points. For example, in the domain of restaurant reviews shown in Figure 1-5, our goal is to identify the relevant aspects for each restaurant, such as *chicken*, *dessert*, and *drinks* for *Tasca Spanish Tapas* and to find the associated sentiment values; in this case, generally positive for *dessert* and somewhat mixed for *chicken* and *drinks*.

While there has been prior work on multi-aspect sentiment analysis [38, 64, 11, 71, 81], this problem formulation poses several unique challenges. Previous work has often relied on supervised approaches or pre-defined aspects [11, 71, 74, 80]; however, in our setup, we cannot predict a priori what fine-grained aspects may be important. Therefore, we must define aspects dynamically based on the data. Text from social media outlets often contains many spelling errors and novel words (e.g., the use of *delish* as a replacement for *delicious*), so the model must be able to generalize and predict how to deal with unseen words. Additionally, in order to effectively aggregate both aspects and values, it is crucial for the model to distinguish whether each word should be treated as an aspect word or a value word.



(a) Example system input, consisting of review snippets across many reviews for each of the restaurants in the corpus. Example system output, showing several dynamically selected aspects with the correspondingly labeled snippets, in addition to sentiment value labeling for each snippet.

The noodles and the meat were actually +pretty good.

I +recommend the chicken noodle pho.

The noodles were -soggy.

The chicken pho was also +good.

(b) Ideal per-word labeling for several snippets from the aspect *pho*. Aspect words and positive and negative value words are bolded and underlined; aspect words are blue with a straight underline, positive sentiment value words are green with a wavy underline and marked with a plus (+), and negative sentiment value words are red with a wavy underline and marked with a minus (-).

Figure 1-5: An example of our informative aggregation task in the restaurant domain. The input consists of a collection of review snippets for several restaurants. The output is an aggregation of snippets by aspect (e.g., *chicken* and *dessert*) along with an associated sentiment for each snippet. Words in the output are labeled according to which distribution they are drawn from; aspect words from one of the dynamically selected aspect distributions and value words from either positive or negative sentiment value distributions.

To address these challenges, we propose a generative Bayesian topic model where the set of aspects and corresponding values are represented as hidden variables. Each piece of text input is assumed to contain one aspect and one value, and each word in the text is assumed to be generated from the corresponding aspect distribution, the corresponding value distribution, or a general background distribution. By modeling transitions, the system can identify common patterns of these word distributions; for example, in restaurant review text it is common to have a value word followed by an aspect word, as in *great pizza*. In addition to the factors within a piece of input text, our model also incorporates several factors across the text describing each individual entity, leveraging the intuition from multi-document summarization that important text should be repeated many times within the input documents.

By utilizing a combination of these pieces of information, our model is able to jointly induce both the set of relevant aspects and the corresponding sentiment values, improving performance of both aspect identification and sentiment analysis.

## 1.3 Contributions

The main contributions of this thesis are threefold:

- **Modeling content structure to improve analysis tasks** I demonstrate the benefit of modeling content structure for tasks which traditionally have not included this information. It is critical to know what form the content model should take, as each task requires a different set of information about content. To accomplish this, I introduce a framework which can automatically induce the correct model of content structure for any particular analysis task by jointly learning the parameters for an unsupervised content model with those of the supervised task-specific model.
- **Modeling the structure of text relations for better data aggregation** I present a minimally-supervised model for informative data aggregation which leverages the structure of relations in text. Rather than splitting data aggre-

gation into a pipeline of steps as in prior work, this model provides a joint formulation that combines aspect identification (i.e., determining the most relevant subjects in text) with value (e.g., sentiment value or other aspect associated with the text).

- **Systems for automatic review summarization** I create a complete end-to-end system capable of analyzing social media review text and producing a meaningful aggregation over all reviews for each restaurant. This system first performs multi-aspect phrase extraction designed with the generalized model of content structure. Then, the resulting phrases are aggregated using the model of text relations. To put them together in a meaningful way, I present a demo which integrates the resulting snippets with Google Maps, allowing effective review summarization for users.

## 1.4 Outline

The remainder of thesis proceeds as follows:

- **Chapter 2** examines generalized content modeling for text analysis applications. To do this, I introduce a flexible framework for coupling a general content model with an existing task-specific model. Through joint learning, this formulation can improve the relevance of the content model for the task at hand. I demonstrate the benefits of including content structure and of learning jointly through experiments with two different task-specific models, linear regression and a linear-chain CRF. In each case, task performance increases with a higher-quality content model.
- **Chapter 3** focuses on the task of content aggregation across a corpus through leveraging the structure of relations in text. These relations are defined as consisting of aspects and their corresponding values, and their structure in text can be modeled with an easily-extensible Bayesian graphical model. I perform experiments on two domains, namely restaurant reviews from Yelp and medical

summary text, and I demonstrate empirical benefit on relevant tasks such as aspect identification and sentiment analysis.

- **Chapter 4** summarizes the work presented in this thesis and discusses opportunities for future work, such as introducing more complex models of structure and aggregating data across time.



---

### Modeling content structure for text analysis tasks

---

In this chapter, I consider the task of generalized content modeling for text analysis tasks. We introduce a flexible framework for incorporating an unsupervised document-level model of document structure with traditional approaches for analysis tasks, such as linear chain conditional random fields for text extraction and linear regression for assigning numeric scores. The parameters of the model are learned efficiently using a joint approach in order to tailor the content model to the needs of the analysis task. We empirically demonstrate that content modeling is useful for these text analysis tasks, that joint learning does improve the quality of the learned content model in many cases, and that additional unlabeled data can be used to boost the quality of the content model.

The remainder of this chapter proceeds as follows: In Section 2.1, we motivate this problem and give a high-level overview of our solution. In Section 2.2, we describe how our approach relates to previous work in content modeling. We provide a problem formulation in Section 2.3.1.1 followed by a description of our model in Section 2.3.1.2. We provide a formal description of the learning and inference procedures

---

Code is available at [http://groups.csail.mit.edu/rbg/code/content\\_structure/](http://groups.csail.mit.edu/rbg/code/content_structure/).

<b>Audio</b>	While not particularly aggressive or immersive in any way, this mix is still expertly orchestrated. . . . Dialogue is clean and clear and never flushed out by other elements. <b>For a light-hearted quirky picture, Anchor Bay has done a <u>great</u> job.</b>	(8)
<b>Video</b>	While certainly not a glossy production by any means, the transfer disappoints quite a bit. This is a pretty flat presentation with surprisingly soft visuals and flat color design.	(6)

Figure 2-1: An excerpt from a DVD review. Note that the sentiment word *great* in the highlighted sentence is an indicator of positive sentiment; however, the aspect that it describes is ambiguous in isolation.

in Sections 2.3.1.3 and 2.3.1.4, respectively. To close the description of our model, we discuss methods for incorporating additional unlabeled data in Section 2.3.1.5 and for generalizing to other tasks in Section 2.3.2. For the practical application of our work, we present our four data sets in Section 2.4 and our experimental setup and results in Section 2.5, including work on both mutli-aspect phrase extraction (Section 2.5.1) and multi-aspect sentiment analysis (Section 2.5.2). Finally, we conclude with an analysis of the benefits of this model and directions for future work in Section 2.6.

## 2.1 Introduction

Leveraging document structure significantly benefits many text analysis applications, such as information extraction and sentiment analysis. As a motivating example, consider determining the sentiment of several aspects of a product, based on the review excerpt shown in Figure 2-1. In this task, we would like to assign separate sentiment ratings to each aspect — in this example, the aspects of audio and video quality. While the “great” is a strong indicator of positive sentiment, the sentence in which it appears does not specify the aspect to which it relates. Resolving this ambiguity requires information about global document structure.

A central challenge in utilizing such information lies in finding a relevant representation of content structure for a specific text analysis task. For instance, when performing single-aspect sentiment analysis, the most relevant aspect of content structure



is whether a given sentence is objective or subjective [59]. In a multi-aspect setting, however, information about the sentence topic is required to determine the aspect to which a sentiment-bearing word relates [74]. As we can see from even these closely related applications, the content structure representation should be intimately tied to a specific text analysis task.

In this work, we present an approach in which a content model is learned jointly with a text analysis task. We assume annotated training data for the analysis task itself, but we learn the content model from raw, unannotated text. Our approach is implemented in a discriminative framework using latent variables to represent facets of content structure. In this framework, the original task features (e.g., lexical ones) are conjoined with latent variables to enrich the features with global contextual information. For example, in Table 2-1, the feature associated with the word “great” should contribute most strongly to the sentiment of the *audio* aspect when it is augmented with a relevant topic indicator.

The coupling of the content model and the task-specific model allows the two components to mutually influence each other during learning. The content model leverages unannotated data to improve the performance of the task-specific model, while the task-specific model provides feedback to improve the relevance of the content model. The combined model can be learned effectively using a novel EM-based method for joint training. Because the content model is learned in an unsupervised fashion, we can additionally improve its quality through the addition of more raw text. This is especially beneficial for applications to text analysis tasks on social media, where annotations are expensive to acquire but we often have a large volume of raw text available.

We evaluate our approach on two complementary text analysis tasks. Our first task is a multi-aspect sentiment analysis task, where a system predicts the aspect-specific sentiment ratings [74]. Second, we consider a multi-aspect extractive summarization task in which a system extracts key properties for a pre-specified set of aspects. On both tasks, our method for incorporating content structure consistently outperforms structure-agnostic counterparts. Moreover, jointly learning content and

task parameters yields additional gains over independently learned models.

## 2.2 Related work

Prior research has demonstrated the benefits of content models for discourse-level tasks; however, the applications considered in this chapter are typically developed without discourse information, focusing instead on sentence-level relations. In this section, we first describe work on leveraging document structure in terms of task-specific relevance models, such as identifying subjective sentences for sentiment analysis (Section 2.2.1). Then, we discuss approaches for topic modeling, both as a standalone task and as a component of models for discourse tasks (Section 2.2.2). Finally, we describe traditional discourse models and the difficulties in applying them for the analysis tasks we introduce in our work (Section 2.2.3).

### 2.2.1 Task-specific models for relevance

Outside of discourse, research has focused on modeling document structure in terms of relevance of specific pieces of text for particular applications. For example, these models may focus on determining which pieces of text may be relevant for summarization [4, 63] or identifying subjective sentences for sentiment analysis [59, 18, 75]. These models are generally not concerned about the *overall* structure of the document; instead, they focus on the task-specific relevance of smaller pieces, such as sentences or paragraphs.

One direction of work focuses on relationships between pieces of a document, through their content and the structure of the document [4, 59]. For example, Berger and Mittal [4] explore relevance for query-focused summarization, in which there are several summaries for each of several documents in a corpus and the goal is to select the most relevant summary of the most relevant document. Because both queries and summaries are quite short, there is a problem of extreme sparsity if they are matched directly. To compensate, they use a probabilistic model smoothed with a series of backoff distributions over related summaries, the entire document, all documents, and

the corpus as a whole. Using similar intuition on a very different task, Pang and Lee [59] use a min-cut formulation to identify subjective sentences within a document as a step to improve document-wide sentiment analysis. Specifically, they utilize not only individual sentence judgments, but also those of related sentences. In their experiments, they use sentence proximity to define these association scores.

An alternative to this direction is to leverage linguistic resources such as WordNet or polarity lexicons as a basis for determining relevance [63, 18, 75]. Patwardhan and Riloff [63] design an approach for sentence relevance in information extraction, using a set of seed extraction patterns combined with a semantic category parser. Using an SVM, they are able to expand the set of good extraction patterns, i.e., those which extract relevant text. Choi and Cardie [18] and Somasundaran et al. [75] utilize a combination of linguistic resources and discourse information in order to make judgments of relationships between sentences and words. Choi and Cardie [18] use a compositional model which aggregates information such as sentiment and negation from several lexicons. Somasundaran et al. [75] focus linking words through discourse relations.

In each of these approaches, the focus is on finding relevance and relationships within a document in a specific task-oriented approach, rather than finding a global view of content. In our work, we would also like to approach information extraction and sentiment analysis; however, rather than creating unrelated models of content specific to each, our goal is to introduce an easily-extensible framework which can be used for both tasks.

## 2.2.2 Topic modeling

Traditionally, many discourse-level tasks such as sentence ordering [2, 27], extractive summarization [34, 79, 58, 2, 23, 36], and text segmentation [15] have been developed using content modeling techniques to identify topics within documents. Since these tasks are inherently tied to document structure, a content model is essential to performing them successfully. In addition to this work in discourse-level tasks, topic modeling has been studied as a standalone task [37, 8]. Several divergent techniques

have emerged for topic modeling for these tasks. Here, we explore a subset of these approaches and explore their potential for generalization to additional text analysis tasks.

The first topic modeling techniques focused on the statistical distributions of words across pieces of a document. For example, Latent Semantic Analysis (LSA) was introduced by Deerwester et al. [25] as a means of indexing the content of documents and first applied to document summarization by Gong [34]. Through singular value decomposition, they obtain sets of words which generally appear in similar contexts, which they identify as topics. Then, they select the sentence from the document which has the highest index value for each topic. To extend this work, Steinberger and Jeek [79] and Murray et al. [58] proposed alternate selection algorithms which prioritize sentences which have high scores all-around and allow selection of multiple sentences per topic, respectively.

Rather than learning topics in isolation, it is also possible to model their presence throughout a document. In order to capture the probability of transitioning from one topic to the next and express the latent topic of individual sentences, some work has utilized Hidden Markov Models (HMMs) [2, 32, 27]. Barzilay and Lee [2] introduced this approach and applied it for the tasks of sentence ordering and extraction-based summarization on a corpus of news articles. Their approach first initializes a set of topics using clustering techniques similar to earlier work [39, 30]. Then, parameters of the HMM are estimated via an EM-like approach. For our approach, we define a similar HMM for our content model, as it is straightforward to implement and analyze, and it effectively provides an overview of content in the document. Differing from previous work, rather than clustering sentences to initialize the topics in the document, we find it sufficient to initialize to a near-uniform distribution.

Content modeling has also been combined with work on discourse coherence to improve the performance of natural language generation [77, 27]. Soricut and Marcu [77] address a document coherence task by introducing a generic framework that integrates the global HMM content model of Barzilay and Lee [2] with the coherence models such as the entity-based model of Barzilay and Lapata [1] and a novel word-

based model. By incorporating all of these models into a single log-linear model, they are able to boost performance on the document coherence task. Elsner et al. [27] also studied the incorporation of local and global features from Barzilay and Lapata [1] and Barzilay and Lee [2]; however rather than combining these in a log-linear model, they learn them jointly using a non-parametric HMM. While these models are designed for very different applications than the ones we present, they illustrate that content structure can be a powerful source of information for certain text analysis tasks. These combination models are highly specialized to particular tasks, but our goal is to introduce a generalized, flexible framework suitable for almost any task.

Following from the work on LSA, Hofmann [37] presented Probabilistic Latent Semantic Analysis (PLSA). Specifically, PLSA aims to capture the intuition that documents are formed from a mixture of several topics. One limiting factor in this work is that it is not a proper generative model; it can only assign probability to documents which are included in the test set. Additionally, the large parameter space can lead to overfitting [65]. To overcome these issues, Blei et al. [8] introduced latent Dirichlet allocation (LDA), a generative probabilistic topic model which treats topic mixture weights as hidden variables, rather than parameters linked explicitly to the training set. Under this model, each word is generated either from one of the hidden topics or from a background word distribution.

LDA is easily extensible, and there are several notable variations which are relevant to our work. For example, Blei and McAuliffe [7] extend LDA to include a response variable for each document. This response variable could represent anything from the category of the document to the overall rating in a review. Similarly, Haghghi and Vanderwende [36] present two modifications of LDA designed for summarization tasks. The first distinguishes between document-specific and corpus-wide content information using an approach similar to Daumé III and Marcu [23]. The second additionally identifies specific sub-topics, such as *finance* and *merchandise* of a specific movie. While these models improve performance overall on their respective tasks, this type of architecture does not permit the usage of standard discriminative models which condition freely on textual features. One of the goals in our work is to create a

framework sufficiently general to incorporate existing discriminative methods for text analysis with as few restrictions as possible.

Building on both LDA and traditional models for ordering, Chen et al. [15] model not only which topics are present in a document but also topic ordering in the document with respect to the “canonical” ordering. Specifically, they model topics at the paragraph level and constrain that each topic must occur in at most one contiguous block. Distance to the canonical ordering is measured using the Generalized Mallows Model [29]. This formulation allows an effective means of topic identification and segmentation for data sets which follow certain assumptions; namely, that topics appear in single contiguous blocks and that there exists some canonical ordering. In our work, these assumptions do not necessarily hold, and we would like to avoid making additional assumptions in order to keep the model as general as possible. However, it would be possible to substitute this model instead of our chosen HMM on a data set and task for which these assumptions hold.

### 2.2.3 Discourse models of document structure

Besides topic modeling, research in document structure for discourse tasks has explored linguistically-inspired discourse relations. Rhetorical Structure Theory [52] is a key method for describing the organization of text in terms of discourse relations between pieces of text. Relations are defined between an essential piece of text, the *nucleus*, and additional information, *satellites*. For example, the relation *Evidence* indicates that one piece of text is intended to increase the reader’s belief of the nucleus. These discourse relations can be recovered from text through sentence-level discourse parsing models [76] trained with using the RST Discourse Treebank [13]. Information about these relations benefits tasks such as sentence compression, where the goal is to reduce the number of words by removing nonessential information [78]. After identifying relations within a sentence, those which are not critical to its meaning can be removed.

Discourse relations have also been used for summarization tasks [53, 6, 57]. Marcu [53] demonstrates that rhetorical relations of RST can be utilized to identify the most

salient pieces of text by selecting the nuclei and internal nodes closest to the roots. In the social media domain, rhetorical relations have been investigated for query-focused summarization of blogs [57]. Specifically, they define schemata for different types of queries (*comparative*, *suggestion*, and *reason*) which define the types of relations which should appear in the resulting summary. Then, with the discourse parser of Soricut and Marcu [76], they are able to identify predicates to fill in the appropriate schema.

In a similar line of work, Marcu and Echihabi [54] introduce a method for recognizing Rhetorical-Semantic Relations (RSRs), discourse relations comparable to those of RST. To accomplish this in a mostly-unsupervised fashion, they define cue phrases and patterns (e.g., *but* for the CONTRAST relation), then use these to build a training corpus. Blair-goldensohn and Mckeown [6] demonstrate that RSRs, specifically relations for *cause* and *contrast* can be integrated with existing multi-document summarization methods to improve performance.

While this body of work provides compelling evidence for inclusion of discourse relations for discourse tasks, it is not clear how to incorporate these relations for text analysis tasks like those we define in this paper. Additionally, extraction of these relations may be difficult, as it would require either that we can acquire labeled discourse data to train a traditional parser or that there are standard discourse patterns to extract from social media text in the style of Marcu and Echihabi [54].

## 2.3 Model

In this section, we present a generalized framework for incorporating content structure with standard models for text analysis tasks. First, in Section 2.3.1, we describe the full model using the example of multi-aspect phrase extraction, where the content model is an HMM over sentences and the task-specific model is a linear chain CRF over words in each sentence. Next, in Section 2.3.2 we present the generalization to another task, multi-aspect sentiment analysis. For the multi-aspect analysis task, the content model is an HMM over paragraphs and the task-specific model consists of independent linear regression models for each aspect sentiment rating.

## 2.3.1 Multi-aspect phrase extraction

Our first task is multi-aspect phrase extraction. In this task, our goal is to identify phrases in text which correspond to each of several pre-defined aspects; for instance, *it took 30 minutes for our waiter to come* would be labeled as *service*. We implement a supervised task-specific model with a linear-chain conditional random field (CRF), where each word in the document receives a label as being part of either one of the pre-defined aspects or the background text. We also include a large volume of unlabeled data to train our content model, an HMM over sentences in the document.

### 2.3.1.1 Problem Formulation

Our approach assumes that at training time we have a collection of labeled documents  $\mathcal{D}_L$ , each consisting of the document text  $\mathbf{s}$  and true task-specific labeling  $\mathbf{y}^*$ . For this task,  $\mathbf{y}^*$  consists of sequence labels (e.g., *value* or *service*) for the tokens of a document. Specifically, the document text  $\mathbf{s}$  is composed of sentences  $s_1, \dots, s_n$  and the labelings  $\mathbf{y}^*$  consists of corresponding label sequences  $y_1, \dots, y_n$ .<sup>1</sup> As is common in related work, we model each  $y_i$  using a CRF which conditions on the observed document text.

For this approach, we additionally assume a content model, which we fix in this work to be the document-level HMM as used in Barzilay and Lee [2]. In this content model, each sentence  $s_i$  is associated with a hidden topic variable  $T_i$  which generates the words of the sentence. We will use  $\mathbf{T} = (T_1, \dots, T_n)$  to refer to the hidden topic sequence for a document. We fix the number of topics to a pre-specified constant  $K$ . This content model is trained on the complete labeled data set  $\mathcal{D}_L$ , as well as any available collection of unlabeled documents  $\mathcal{D}_U$  from the same domain.

A summary of the notation used in this section is presented in Table 2.1.

---

<sup>1</sup>Note that each  $y_i$  is a label sequence across the words in  $s_i$ , rather than an individual label.



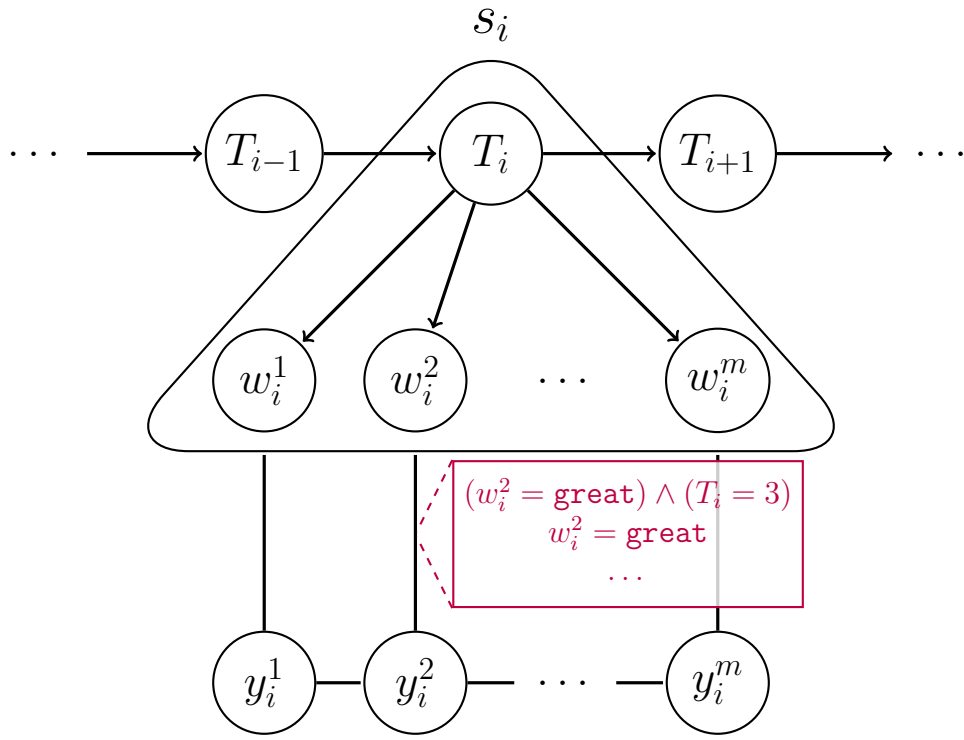


Figure 2-2: A graphical depiction of our model for sequence labeling tasks. The  $T_i$  variable represents the content model topic for the  $i$ th sentence  $s_i$ . The words of  $s_i$ ,  $(w_i^1, \dots, w_i^m)$ , each have a task label  $(y_i^1, \dots, y_i^m)$ . Note that each token label has an undirected edge to a factor containing the words of the current sentence,  $s_i$  as well as the topic of the current sentence  $T_i$ .

Data set	
$\mathcal{D}_L$	Set of labeled documents
$\mathcal{D}_U$	Set of unlabeled documents
$\mathbf{s}$	Document text
$s_i$	Sentence $i$ of the document
$w_i^j$	Word $j$ of sentence $i$ of the document
Content component	
$\mathbf{T}$	Hidden topic sequence of a document
$T_i$	Topic corresponding to $s_i$
$K$	Pre-specified number of topics
$\theta$	Parameters for the content model
Task component	
$\mathbf{y}$	Task label sequences
$y_i$	Task label sequence corresponding to $s_i$
$y_i^j$	Task label corresponding to $w_i^j$
$\phi$	Parameters for the task model

Table 2.1: A summary of notation for the multi-aspect phrase extraction task, divided by function. Note that “sentence” here can be exchanged for a different unit of text; for example, in the multi-aspect sentiment task, the division is instead by paragraph.

### 2.3.1.2 Model overview

Our model, depicted in Figure 2-2, proceeds as follows: First the document-level HMM generates a hidden content topic sequence  $\mathbf{T}$  for the sentences of a document. This content component is parametrized by  $\theta$  and decomposes in the standard HMM fashion:

$$P_{\theta}(\mathbf{s}, \mathbf{T}) = \prod_{i=1}^n P_{\theta}(T_i | T_{i-1}) \prod_{w \in s_i} P_{\theta}(w | T_i) \quad (2.1)$$

For this formulation, we implement a background topic distribution which is shared between all topics. Each word in a sentence may be drawn from either the topic-specific distribution or the shared background distribution. This hierarchical emission model is intended to capture domain-specific stop words.

The label sequences for each sentence in the document are independently modeled

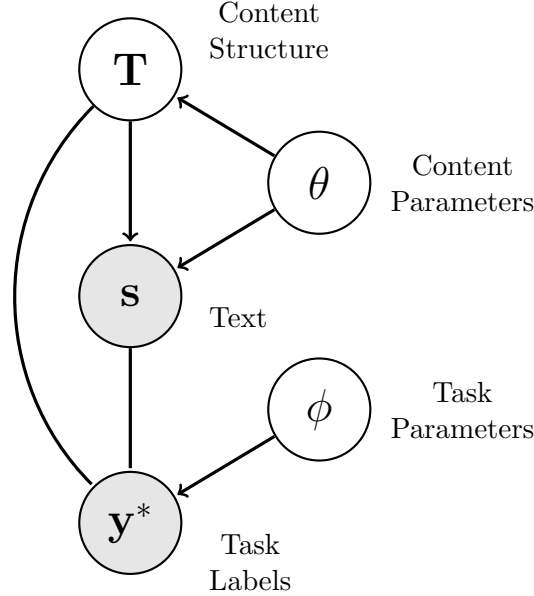


Figure 2-3: A graphical depiction of the generative process for a labeled document at training time (See Section 2.3); shaded nodes indicate variables which are observed at training time. First the latent underlying content structure  $\mathbf{T}$  is drawn. Then, the document text  $\mathbf{s}$  is drawn conditioned on the content structure utilizing content parameters  $\theta$ . Finally, the observed task labels for the document are modeled given  $\mathbf{s}$  and  $\mathbf{T}$  using the task parameters  $\phi$ . Note that the arrows for the task labels are undirected since they are modeled discriminatively.

as CRFs which condition on both the sentence features and the sentence topic:

$$P_{\phi}(\mathbf{y}|\mathbf{s}, \mathbf{T}) = \prod_{i=1}^n P_{\phi}(y_i|s_i, T_i) \quad (2.2)$$

Each sentence CRF is parametrized by  $\phi$  and takes the standard form:

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

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

Allowing the CRF to condition on the sentence topic  $T_i$  permits predictions to be more sensitive to content. For instance, using the example from Table 2-1, we could have a feature that indicates the word “great” conjoined with the segment topic (see

Figure 2-2). These topic-specific features serve to disambiguate word usage.

This joint process, depicted graphically in Figure 2-3, is summarized as:

$$P(\mathbf{T}, \mathbf{s}, \mathbf{y}^*) = P_\theta(\mathbf{T}, \mathbf{s})P_\phi(\mathbf{y}^*|\mathbf{s}, \mathbf{T}) \quad (2.3)$$

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

### 2.3.1.3 Learning

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

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

We use the standard Expectation Maximization (EM) algorithm to optimize this objective.

**E-Step** The E-Step in EM requires computing the posterior distribution over latent variables. In this model, the only latent variables are the sentence topics  $\mathbf{T}$ . To compute this term, we utilize the decomposition in Equation (2.3) and rearrange

HMM and CRF terms to obtain:

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

We note that this expression takes the same form as the document-level HMM, except that in addition to emitting the words of a sentence, we also have an observation associated with the sentence sequence labeling. We treat each  $P_\phi(y_i^* | s_i, T_i)$  as part of the node potential associated with the document-level HMM. We utilize the Forward-Backward algorithm as one would with the document-level HMM in isolation [66], except that each node potential incorporates this CRF term.

**M-Step** We perform separate M-Steps for content and task parameters. The M-Step for the content parameters is identical to the document-level HMM content model: topic emission and transition distributions are updated with expected counts derived from E-Step topic posteriors.

The M-Step for the task parameters does not have a closed-form solution. Recall that in the M-Step, we maximize the log probability of all random variables given expectations of latent variables. Using the decomposition in Equation (2.3), it is clear that the only component of the joint labeled document probability which relies upon the task parameters is  $\log P_\phi(\mathbf{y}^* | \mathbf{s}, \mathbf{T})$ . Thus for the M-Step, it is sufficient to optimize the following with respect to  $\phi$ :

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

The first equality follows from the decomposition of the task component into independent CRFs (see Equation (2.2)). Optimizing this objective is equivalent to a weighted version of the conditional likelihood objective used to train the CRF in isolation. An intuitive explanation of this process is that there are multiple CRF instances, one for each possible hidden topic  $T$ . Each utilizes different content features to explain the sentence sequence labeling. These instances are weighted according to the posterior over  $T$  obtained during the E-Step. While this objective is non-convex due to the summation over  $T$ , we can still optimize it using any gradient-based optimization solver; in our experiments, we used the LBFGS algorithm [47].

#### 2.3.1.4 Inference

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

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

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

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

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

At inference time, we select the single labeling which minimizes the expected loss

with respect to model posterior over label sequences:

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

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

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

This minimum risk criterion has been widely used in NLP applications such as parsing [35] and machine translation [26]. Note that the above formulation differs from the standard CRF due to the latent topic variables. Otherwise the inference task could be accomplished by directly obtaining posteriors over each  $y^j$  state using the Forward-Backwards algorithm on the sentence CRF.

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

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

Once we obtain expected losses of each token prediction, we compute the minimum risk sequence labeling by running the Viterbi algorithm. The potential for each position and prediction is given by the negative expected loss. The maximal scoring sequence according to these potentials minimizes the expected risk.

### 2.3.1.5 Leveraging unannotated data

As mentioned previously, our model allows us to incorporate unlabeled documents, denoted  $\mathcal{D}_U$ , during learning to improve the content model. For each unlabeled document, we observe only the document text  $\mathbf{s}$ , which we assume is drawn from the same

content model as our labeled documents. The objective presented in Section 2.3.1.3 took into account labeled documents only; here we supplement this objective by capturing the likelihood of unlabeled documents according to the content model:

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

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

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

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

## 2.3.2 Multi-aspect sentiment analysis

Our second task is multi-aspect sentiment analysis, where the goal is to identify numeric scores (1-10, where 1 is worst and 10 is best) for each of several pre-defined aspects given an unsegmented review document. We use the same general model framework as for multi-aspect phrase extraction; in this case, the task-specific model is an independent linear regression model for each aspect which utilizes the unigrams from the entire document. Because the documents in our corpus for this domain are expert-written and very focused, our content model is an HMM over paragraphs in the document, rather than sentences.

### 2.3.2.1 Problem Formulation

For this task, the target  $y$  consists of numeric sentiment rankings  $(y_1, \dots, y_K)$  for each of  $K$  predefined aspects. For the task specific model, we define an independent linear regression for each aspect over all words in the document. Note that this structure is



not localized to any region of the document; instead, words from the entire document influence each aspect. For the content model, each paragraph  $p_i$  is associated with a hidden topic  $T_i$ . The topic sequence  $\mathbf{T} = (T_1, \dots, T_n)$  is defined as an HMM. The number of possible topics  $K$  is predefined. As in the phrase extraction case, the task-specific model is learned in a supervised fashion, while the content model is learned in an unsupervised way.

### 2.3.2.2 Model

The model follows the same general procedure as the phrase extraction model. As before, the document-level HMM generates a hidden content topic sequence  $\mathbf{T}$ , this time over paragraphs of the document. Each linear regression model is specific to a particular aspect we would like to rate and independent from other linear regression models. We assume that each numeric score is a linear combination of features  $f(\cdot)$  with weights  $\phi$ . To assess whether we have predicted correctly, we consider the probability of error, which we assume to be a normal distribution:

$$\begin{aligned} P(\mathbf{y}|\mathbf{T}, \mathbf{s}) &= \mathcal{N}(\mathbf{y} - f(\mathbf{T}, \mathbf{s}; \phi)) \\ &= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\mathbf{y} - f(\mathbf{T}, \mathbf{s}; \phi))^2} \end{aligned}$$

### 2.3.2.3 Learning

Using this formulation, the model structure still decomposes as in Figure 2-3; however, the details of learning are changed. Because the task label (aspect sentiment ratings) is not localized to any region of the document, all content model variables influence the target response. Conditioned on the target label, all topic variables become correlated. Thus when learning, the E-Step requires computing a posterior over paragraph topic tuples  $\mathbf{T}$ :

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

In our experiments, we compute this posterior exactly by enumerating  $\mathbf{T}$  tuples, since the number of sentences and possible topics is relatively small. If summation

	# Documents			Document Sizes	
	Labeled		Unlabeled	Avg	Avg
	Train	Test		# Words	# Sents
Multi-aspect sentiment	600	65	—	1,027	20.5
Multi-aspect summarization					
Amazon	35	24	12,684	214	11.7
Yelp	48	48	33,015	178	11.2
Medical	47	47	206	544	38.2

Table 2.2: This table summarizes the size of each corpus. In each case, the unlabeled texts of both labeled and unlabeled documents are used for training the content model, while only the labeled training corpus is used to train the task model. Note that the entire data set for the multi-aspect sentiment analysis task is labeled.

is intractable, the posterior may be approximated using variational techniques [5], which is applicable to a broad range of potential applications.

#### 2.3.2.4 Inference

At inference time, we must predict the numeric score  $y$  for each aspect based on the learned parameters and a predicted topic sequence  $\mathbf{T}$ . To do this, we define a quadratic loss function based on the sum of squared errors:

$$L(y, \hat{y}) = \sum_j \frac{1}{2}(y^j - \hat{y}^j)^2$$

We can minimize this loss with gradient descent using the LBFGS algorithm [47].

## 2.4 Data sets

We perform experiments on four data sets: DVD reviews, Amazon HDTV reviews, restaurant reviews from Yelp, and medical patient visit summaries. In this section, we describe each data set and present their respective challenges.

<b>Movie</b>	This collection certainly offers some nostalgic fun, but at the end of the day, the shows themselves, for the most part, just don't hold up.	(5)
<b>Video</b>	Regardless, this is a fairly solid presentation, but it's obvious there was room for improvement.	(7)
<b>Audio</b>	Bass is still robust and powerful. Fans should be pleased with this presentation.	(8)
<b>Extras</b>	The deleted scenes were quite lengthy, but only shelled out a few extra laughs.	(4)

Figure 2-4: Excerpts from the multi-aspect sentiment ranking corpus, taken from IGN.com DVD reviews. Note that the actual paragraphs are longer; these are a few sentences designed to indicate the type of content. Each paragraph is labeled with a numeric score from 1-10. Paragraphs are labeled in the corpus, but these labels are not provided to the model. Text categories are explained in detail in Section 2.4.1.

### 2.4.1 Multi-aspect sentiment analysis

To evaluate our model on multi-aspect summarization, we use a data set consisting of editorial DVD reviews from the website IGN.com.<sup>2</sup> Document and sentence statistics for this corpus are shown in Table 2.2.

Each review is written by an IGN.com staff member and consists of text accompanied by ratings on a scale of 1 (worst) to 10 (best) in four categories: content, video, audio, and DVD extras. The *content* section gives a brief summary and evaluation of the movie's plot and characters, the *video* section describes any issues with video encoding or graininess, the *audio* section gives a similar account of the sound, and the *extras* section details both special features and packaging. Sample excerpts from each section are shown in Figure 2-4.

In this data set, segments corresponding to each of the aspects are clearly delineated in each document. This segmentation is not given to our full model; however, having the segmentation allows us to experiment with alternative and "gold" content models.

---

<sup>2</sup><http://dvd.ign.com/index/reviews.html>

Just bought a <b>A32 in</b> LG TV from Amazon. Fast shipping, perfect shape (same as usual) <b>Voutstanding picture</b> , <b>Ieasy hookup</b> even for a electrically challenged octogenarian. Highly recommended.	<b>A</b> = Appearance <b>V</b> = Video <b>I</b> = Inputs <b>S</b> = Sound <b>R</b> = Remote <b>M</b> = Menu <b>E</b> = Economy <b>F</b> = Features
...As far as TV speakers go, I'm <b>Spleased with the sound quality</b> . ...Having the <b>I2 HDMI slots is really nice</b> . <b>RRemote works fine</b> , no issues there. The TV <b>Mmenus are very easy to work...</b>	

(a) Sample labeled text from Amazon HDTV reviews

... <b>ACasual, romantic, french farmhouse inspired</b> ... This is a farm-to-table restaurant with <b>Ffresh and local ingredients</b> ... <b>SOur waiter was a true professional</b> , proud of the restaurant and its reputation...	<b>A</b> = Atmosphere <b>F</b> = Food <b>S</b> = Service <b>V</b> = Value <b>O</b> = Overall
We had the six course tasting menu, and it was paced very well. ...the <b>Fquality of the entree and dessert (phenomenal!)</b> ... <b>Swonderful wine advice from our waiter</b> (he suggested a <b>Vsurprisingly affordable wine</b> ...). Sure <b>VI winced a little at the price tag</b> , but <b>OI can't wait to go back</b> .	

(b) Sample labeled text from Yelp restaurant reviews

Dear Dr. Smith, I had the pleasure of seeing your patient, <b>VJohn Doe</b> on <b>VSeptember 20</b> ....He was <b>Sexposed to lead-based paint chips</b> .... John <b>Dspeaks in 2-word phrases</b> ... <b>EHEENT within normal limits</b> . <b>EHeart rate and rhythm normal</b> . The <b>Lblood lead level was 16 mcg/dl</b> , the <b>LZPP was 35/78</b> . ... Assessment: <b>ALow body burden lead poisoning</b> . Plan: John will <b>Preturn to the PEHC in one month</b> ...	<b>V</b> = Visit <b>H</b> = History <b>S</b> = Lead Source <b>D</b> = Development <b>M</b> = Medication <b>N</b> = Nutrition <b>C</b> = Countermeasures <b>E</b> = Physical exam <b>L</b> = Lab results <b>A</b> = Assessment <b>P</b> = Plan
--	---

(c) Sample labeled text from medical summaries

Figure 2-5: Excerpts from each of the multi-aspect summarization corpora (Amazon HDTV reviews, Yelp restaurant reviews, medical summary text) with labels. Note that sentences generally focus on one or two aspects. Labels for each corpus are explained in detail in Section 2.4.2.

## 2.4.2 Multi-aspect summarization

For multi-aspect summarization, we test our model’s performance on three corpora: Amazon.com HDTV reviews, Yelp.com restaurant reviews, and medical visit summaries from the Pediatric Environmental Health Clinic (PEHC) at Children’s Hospital Boston. Statistics of each corpus are shown in Table 2.2.

**Amazon HDTV reviews** The Amazon.com data set consists of the text from user-provided HDTV reviews. While there is additional information available such as star rating for a few areas, we do not include this as part of the data set. To eliminate noisy reviews, we only retain documents that have been rated “helpful” by the users of the site; we also remove reviews which are abnormally short or long. Sample labeled excerpts are shown in Figure 2-5a.

There is a wide range of review content. Some reviewers focus on very technical aspects of the TV such as the underlying display technology and particular settings to maximize the picture quality, while others provide lay opinions without technical detail or discuss extraneous information such as their experience with the delivery company. To cover this variety, we define eight labels: remote (remote control), menu (on-screen menu and adjustments), inputs (connectors for external devices such as HDMI), economy (price and value), video (picture quality), sound (quality of internal speakers), appearance (opinions and physical description), and features (any additional content such as built-in weather and games).

**Yelp restaurant reviews** The Yelp.com data set contains user-authored restaurant reviews from the Boston area. As in the Amazon corpus, we retain only the text of the reviews, and we eliminate any reviews which are abnormally short or long. Sample labeled excerpts are shown in Figure 2-5b.

As in the Amazon corpus, reviews may contain unrelated information, such as a story of the reviewer’s evening as a whole; however, most reviews touch on a few main points. To label this information, we borrow aspects which have been used in previous work, e.g., Snyder and Barzilay [74]. Specifically, we define five labels:

food (any aspect of food or drink), atmosphere (decor, music, etc.), value (price or economy), service (wait for a table, server personality), and overall (general comments about the restaurant as a whole).

**Medical visit summaries** The medical summary data set consists of dictated summaries of patient visits from the Pediatric Environmental Health Clinic (PEHC) at Children’s Hospital Boston, specializing in lead poisoning. In the standard medical work flow, these are dictated – often as a letter – to a professional medical transcription service by a doctor at PEHC at the conclusion of a visit, then sent to the patient’s primary care physician. Here, we work with the transcribed version of the documents. As in the previous corpora, we eliminate those which are abnormally long or short. Sample labeled excerpts are shown in Figure 2-5c.

Because these documents are part of the patient’s official medical record, they are extremely focused, touching on many of the same important points with little extraneous information. By consultation with the doctors, we defined 11 labels on this set: general visit information (name, age, primary doctor), medical history, medications, developmental status (ability to speak and understand), nutritional status, lead source (paint chips, lead pans), lead countermeasures (hand washing, official inspections, etc.), physical exam, lab results, assessment (final diagnosis), and plan (future labs, medications, follow-ups).

**Annotation** Each data set was manually annotated with aspect labels. Annotators are provided with a full document and instructions containing examples of good and bad phrases. They can then highlight phrases with the mouse, then right click to select a label for the highlighted phrase, as shown in Figure 2-6. The Amazon and Yelp corpora were annotated using Mechanical Turk, which has been used for annotation in previous NLP work [73]. Since we cannot select high-quality annotators directly, we included a control document which had been previously annotated by a trusted native speaker among the documents assigned to each annotator. The work of any annotator who exhibited low agreement with the trusted annotator on the control



Figure 2-6: Annotation procedure for the multi-aspect phrase labeling task. Annotators are asked to first highlight a relevant phrase, then select a label for that phrase.

document annotation was excluded from the corpus. Because the medical corpus is confidential and requires specialized knowledge, we instead receive high-quality annotations from two doctors from PEHC.

To test task annotation agreement, we use Cohen’s Kappa [19]. On the Amazon data set, two native speakers annotated a set of four documents. The agreement between the judges was 0.54. On the Yelp data set, we simply computed the agreement between all pairs of reviewers who received the same control documents; the agreement was 0.49. On the PEHC data set, the agreement was 0.68. While these agreements are lower than traditionally desirable (0.6–0.8 indicates *significant agreement*, while 0.4–0.6 indicates *moderate agreement*), they are in line with what we expect based on other experiments in similar domains [17].

## 2.5 Experiments

We apply our approach to two text analysis tasks that stand to benefit from modeling content structure: multi-aspect phrase extraction and multi-aspect sentiment analysis. Here, we describe the experimental design, task-specific modeling adaptations, and results for each task separately.

## 2.5.1 Multi-aspect phrase extraction

The goal of this task is to extract informative phrases that identify information relevant to several predefined aspects of interest. In other words, we would like our system to both extract important phrases (e.g., *cheap food*) and label it with one of the given aspects (e.g., *value*). For concrete examples and lists of aspects for each data set, see Figures 2-5a and 2-5b. Variants of this task have been considered in review summarization in previous work [43, 9].

This task has elements of both information extraction and phrase-based summarization — the phrases we wish to extract are broader in scope than in standard template-driven IE, but at the same time, the type of selected information is restricted to the defined aspects, similar to query-based summarization. The difficulty here is that phrase selection is highly context-dependent. For instance, in TV reviews such as in Figure 2-5a, the highlighted phrase “easy to read” might refer to either the menu or the remote; broader context is required for correct labeling.

When incorporating our content model with the task-specific model, we utilize a new set of features which include all the original features as well as a copy of each feature conjoined with the content topic assignment; e.g., if the original feature is  $w_i^2 = \text{great}$  and the topic of the sentence is Topic 3, we add an additional feature  $(w_i^2 = \text{great}) \wedge (T_i = 3)$  (see Figure 2-2). We also include a feature which indicates whether a given word was most likely emitted from the underlying topic or from a background distribution.

### 2.5.1.1 Baselines

We define two baselines, both of which are simplifications of our model. In first, No Content Model (NoCM), all content features are eliminated, so that the system uses only the task-specific model with the base set of features. In the second, Independent Content Model (IndepCM), the content model is induced in isolation rather than learned jointly in the context of the underlying task. The full set of content features are used; however, they are fixed and cannot change as the task-specific model is



	$F_1$	$F_2$	Prec.	Recall
NoCM	28.8	34.8	22.4	40.3
IndepCM	37.9	43.7	31.1†*	<b>48.6†*</b>
JointCM	<b>39.2</b>	<b>44.4</b>	<b>32.9†*</b>	<b>48.6†</b>

Table 2.3: Results for multi-aspect phrase extraction on the Yelp corpus. Marked precision and recall are statistically significant with  $p < 0.05$ : \* over the previous model and † over NoCM.

	$F_1$	$F_2$	Prec.	Recall
NoCM	43.4	50.3	35.3	56.4
IndepCM	55.5	59.8	49.6†*	63.1†*
JointCM	<b>56.6</b>	<b>60.7</b>	<b>50.9†</b>	<b>63.8†</b>

Table 2.4: Results for multi-aspect phrase extraction on the medical corpus. Marked precision and recall are statistically significant with  $p < 0.05$ : \* over the previous model and † over NoCM.

learned. We refer to our full model described in Section 2.3 as the Joint Content Model (JointCM), where the content and task components are learned jointly.

### 2.5.1.2 Evaluation metrics

For this task, we measure average token precision and recall of the label assignments (Multi-label). For the Amazon corpus, we report two additional metrics: First, we present a coarser metric corresponding to unlabeled phrase extraction, which measures extraction precision and recall while ignoring labels (Binary labels). Note that in this case, the word labels are still learned in a multi-aspect setting. Second, we present the results using ROUGE [45]. To make a fair comparison between systems for ROUGE, we control for extraction length by requiring that each system predict the same number of tokens as the original labeled document.

To determine statistical significance, we perform chi-square analysis on the ROUGE scores as well as on precision and recall separately, as is commonly done in information extraction [31, 85, 28].

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

Table 2.5: Results for multi-aspect phrase extraction on the Amazon corpus. Marked ROUGE, precision, and recall are statistically significant with  $p < 0.05$ : \* over the previous model and † over NoCM.

### 2.5.1.3 Results

**Baseline Comparisons** Adding a content model significantly outperforms the NoCM baseline on all domains. The highest  $F_1$  error reduction – 23.3% – is achieved on multi-aspect phrase extraction on the medical corpus, followed by the reduction of 14.6% on Yelp multi-aspect phrase extraction, and 11.5% on Amazon multi-aspect phrase extraction. We also observe a small but consistent performance boost when comparing against the IndepCM baseline. This result supports our hypothesis about the advantages of jointly learning the content model in the context of the underlying task.

**Comparison with additional context features** One alternative to an explicit content model is to simply incorporate additional features into NoCM as a proxy for contextual information. Specifically, this can be accomplished by adding unigram features from the sentences before and after the current one.

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

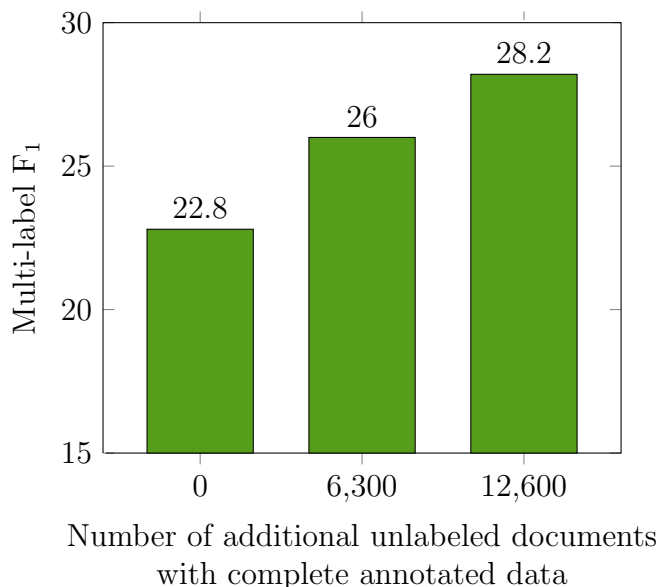


Figure 2-7: Results for multi-aspect phrase extraction on the Amazon corpus using the complete annotated set with varying amounts of additional unlabeled data. Note that because we append the unlabeled versions of the labeled data to the unlabeled set, even with 0 *additional* unlabeled documents, a content model is still learned over the set of training data.

**Impact of content model quality on task performance** We can explore the impact of content model quality on task performance by varying the number of unlabeled documents available to the system. Intuitively, the quality of the induced content model should be determined by the amount of training data, so if we significantly reduce the number of documents, we expect performance to decrease. Even in the absence of additional unlabeled documents, note that our model does still learn a basic content model over the labeled documents; however, it is likely to be highly overfitted and of low quality. As Figure 2-7 shows, performance on the multi-aspect phrase extraction task does improve as the number of unannotated documents used to learn the content model increases.

**Compensating for annotation sparsity** Manual annotation is expensive, and in many domains, there are large quantities of unannotated documents available. Therefore, one of the goals of this line of work is to reduce the need for manual annotation by using a content model which can incorporate rich contextual information. We

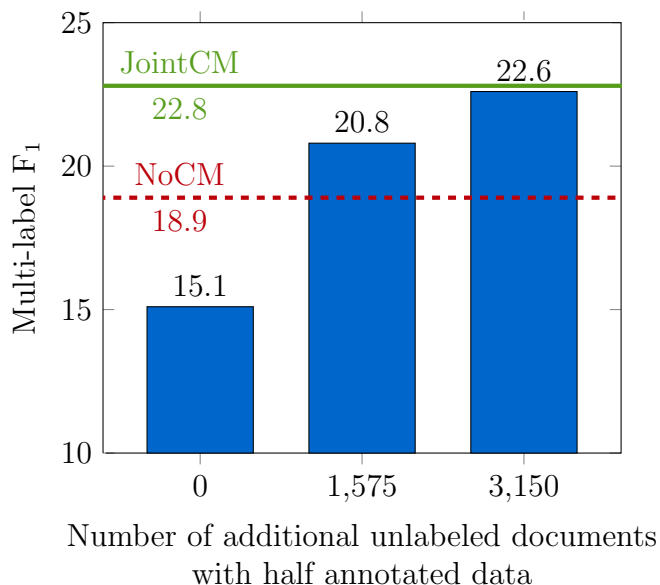


Figure 2-8: Results for multi-aspect phrase extraction on the Amazon corpus using half of the annotated training data (18 documents). The content model is trained with varying amounts of additional unlabeled data. Note that because the content model is learned over both labeled and unlabeled data, even with 0 *additional* unlabeled documents, a content model is still learned over the labeled data set. The dashed horizontal line represents NoCM with the complete annotated set, while the solid horizontal line represents JointCM with the complete annotated set and zero additional unlabeled documents (as shown in Figure 2-7).

evaluate our progress toward this goal by reducing the amount of annotated data available to the model and measuring performance at several quantities of unannotated data. As Figure 2-8 shows, the performance increase achieved by doubling the amount of annotated data can also be achieved by adding only 1,575 unlabeled documents. Additionally, by providing 3,150 unlabeled documents, the performance approaches the performance of the full model with no unlabeled documents.

**Practical evaluation** To find a qualitative understanding of the results, we created a demo system, CONDENSr, using phrases extracted from the Yelp corpus, shown in Figure 2-9. First, we use our full model (JointCM) to extract aspect-labeled phrases for each restaurant. Then, we cluster the phrases which share an aspect to attain the desired number of clusters; for this demo, we use 6 clusters and force at least 3 to be food-related. Finally, we tag them as positive or negative using a simple form of

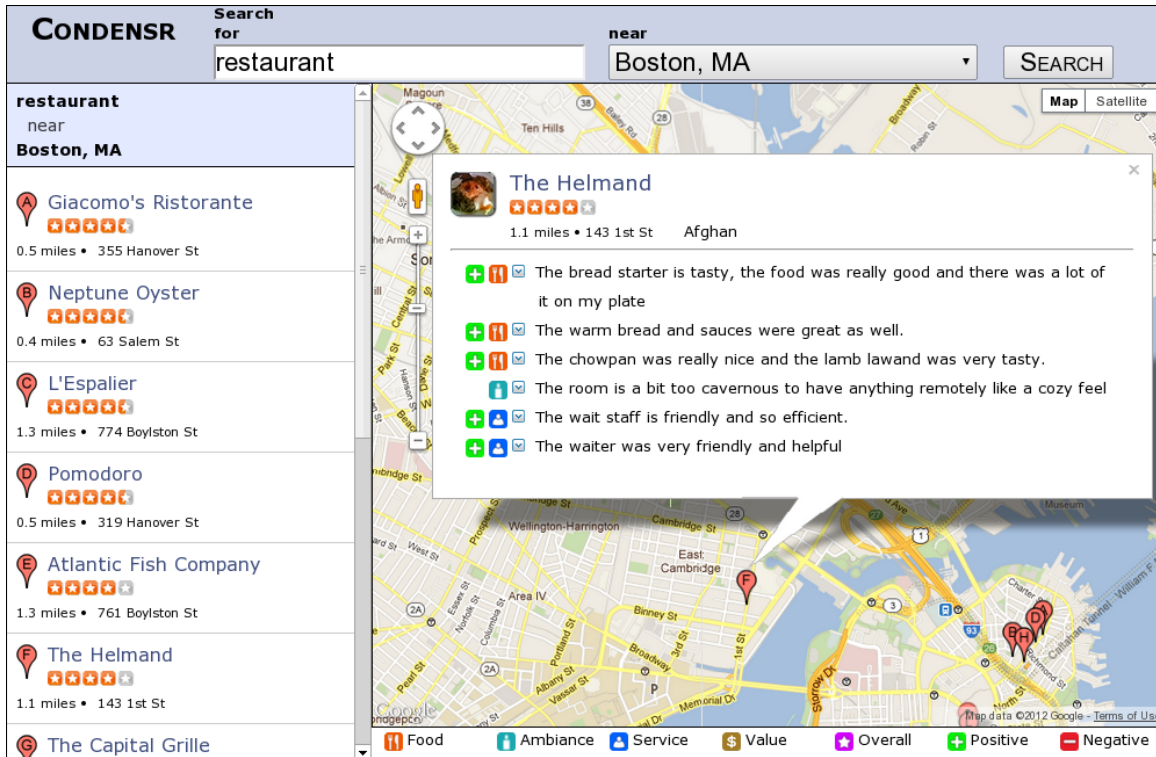


Figure 2-9: A screenshot of CONDENSr, our demo system. This system selects phrases for each of several aspects, clusters them and performs basic sentiment analysis with a set of seed words, then displays them using a searchable map interface.

sentiment analysis; specifically, we use a set of positive and negative seed words, then flip the polarity based on negation within a window of 3 words. These results are displayed on a searchable map interface which relies on Yelp for search and Google maps for navigation. This demo is currently online at <http://condensr.com>, and more details and results can be found in Appendix A.

## 2.5.2 Multi-aspect sentiment analysis

The goal of multi-aspect sentiment classification is to predict a set of numeric ranks that reflects the user satisfaction for each aspect [74]. One of the challenges in this task is to attribute sentiment-bearing words to the aspects they describe. Information about document structure has the potential to greatly reduce this ambiguity. For instance, in the example given at the beginning of this chapter in Figure 2-1, it is crucial to know that the highlighted sentence refers to *audio quality* in order to adjust

	$L_1$	$L_2$
NoCM	1.37	3.15
IndepCM	1.28 †*	2.80 †*
JointCM	<b>1.25</b> †	<b>2.65</b> †*
Gold	1.18 †*	2.48 †*

Table 2.6: The error rate on the multi-aspect sentiment ranking. We report mean  $L_1$  and  $L_2$  between system prediction and true values over all aspects. Marked results are statistically significant with  $p < 0.05$ : \* over the previous model and † over NoCM.

the score for the correct aspect.

Following standard sentiment ranking approaches [86, 60, 33, 74], we employ ordinary linear regression to independently map bag-of-words representations into predicted aspect ranks. In addition to commonly used lexical features, this set is augmented with content features as described for multi-aspect phrase extraction. For this application, we fix the number of HMM states to be equal to the predefined number of aspects.

### 2.5.2.1 Baselines

As in multi-aspect phrase extraction, we define two baselines NoCM and IndepCM to refer to using no content features and independently-learned content features, respectively. For this domain, we have gold annotations available, so we additionally evaluate the system with a gold content model.

### 2.5.2.2 Evaluation metrics

To evaluate system performance, we report the average  $L_2$  (squared difference) and  $L_1$  (absolute difference) between system predictions and the true 1-10 sentiment rating across test documents and aspects [60]. To determine statistical significance of the results, we use Student’s t-test.

### 2.5.2.3 Results

**Baseline Comparisons** As in the multi-aspect phrase extraction task, we see a significant boost of 8.75% on this task. Likewise, we observe small gains when comparing

our system against the IndepCM baseline. These results indicate that the content information learned by the system is beneficial for this task.

**Impact of content model quality on task performance** For this task, we have access to gold standard document-level content structure annotation. This affords us the ability to compare the performance of the ideal content structure, provided by the document authors, with that of the content structure that is learned automatically. As Table 2.6 shows, the manually created document structure segmentation yields the best results. However, the performance of our JointCM model is not far behind the gold standard content structure.

## 2.6 Conclusion

In this chapter, I have demonstrated the benefits of incorporating content structure into several text analysis tasks which are traditionally modeled in a local fashion. We have demonstrated on two tasks and multiple data sets that task performance improves with higher-quality content models. The improvements we observe are more pronounced on data sets which have a good amount of structure.

To facilitate this analysis, I have introduced a flexible framework for the joint learning of an unsupervised latent content model with a supervised task-specific model. This allows us to benefit from both the task-specific annotations we are able to acquire and a large volume of unannotated documents such as those available online. For domains where it is difficult or expensive to acquire task-specific annotations, such as phrase extraction, the ability to compensate with unlabeled data is especially beneficial.

There are a few open avenues for future work. First, the content models that we include in this chapter are relatively simple, focusing on finding coarse topics only at sentence- or paragraph-level. A natural extension to this work is to incorporate a more sophisticated content model, either one more suited to a particular task or one which can model fine-grained topics. Because we have demonstrated the connection between

content model quality and task performance, we would expect that improvements in the content model will yield further gains in task performance.

Second, while the content and task models are learned using information from the corpus as a whole, the tasks we've examined are all focused on extracting information from single documents at test time. However, for many applications, the ability to aggregate information from multiple documents is crucial. For example, in the review domain, rather than simply extracting phrases from individual documents, we would like to extract the common topics that users mention across all reviews for a particular product.

In Chapter 3, I present a model that addresses pieces of both concerns. First, we look closer at individual sentences and phrases to find a more fine-grained representation of structure; specifically, the topics of individual words, selected from a set of dynamic topics. Second, we aggregate information across related documents to help guide the extraction process.



---

### Modeling relation structure for informative aggregation

---

In this chapter, we consider the task of data aggregation of social media review text through fine-grained aspect-based analysis. We develop a model which incorporates intuitions about the structure of text snippets in order to dynamically determine relevant aspects and their associated values (e.g., sentiment). We demonstrate that our model is able to successfully distinguish aspect and value words and leverage several sources of information such as word transitions and parts of speech in order to perform effective aggregation across the data set. We also show several model extensions which allow it to be adapted for alternate domains, such as medical summary text.

The remainder of this chapter is structured as follows: Section 3.1 motivates this approach and gives a high-level overview of our technique. Section 3.2 compares our work with previous work on both aspect identification and sentiment analysis. Section 3.3 describes our specific problem formulation and task setup more concretely. Section 3.4 presents the details of our full model and various model extensions, and Section 3.5 describes the inference procedure and the necessary adjustments for each extension. The details of both data sets, the experimental formulation, and results

---

Code is available at [http://groups.csail.mit.edu/rbg/code/content\\_attitude/](http://groups.csail.mit.edu/rbg/code/content_attitude/).

are presented in Section 3.6. We summarize our findings and consider directions for future work in Section 3.7.

## 3.1 Introduction

Online product reviews have become an increasingly valuable and influential source of information for consumers. The ability to explore a range of opinions allows consumers to both form a general opinion of a product and gather information about its positive and negative aspects (e.g., *packaging* or *battery life*). However, as more reviews are added over time, the problem of information overload gets progressively worse. For example, out of hundreds of reviews for a restaurant, most consumers will read only a handful before making a decision. In this work, our goal is to summarize a large number of reviews by discovering the most informational product aspects and their associated user sentiment.

To address this need, online retailers often use simple aggregation mechanisms to represent the spectrum of user sentiment. Many sites, such as Amazon, simply present a distribution over user-assigned star ratings, but this approach lacks any reasoning about *why* the products are given that rating. Some retailers use further breakdowns by specific predefined domain-specific aspects, such as *food*, *service*, and *atmosphere* for a restaurant. These breakdowns continue to assist in effective aggregation; however, because the aspects are predefined, they are generic to the particular domain and there is no further explanation of *why* one aspect was rated well or poorly. Instead, for truly informative aggregation, each product needs to be assigned a set of fine-grained aspects specifically tailored to that product.

The goal of our work is to provide a mechanism for effective minimally-supervised content aggregation able to discover specific, fine-grained aspects and associated values. Specifically, we represent each data set as a collection of *entities*; for instance, these can represent products in the domain of online reviews. We are interested in discovering fine-grained *aspects* of each entity (e.g., *sandwiches* or *dessert* for a restaurant). Additionally, we would like to recover a *value* associated with the aspect

(e.g., sentiment for product reviews). A summary of the input and output can be found in Figure 3-1. Our input consists of short text snippets from multiple reviews for each of several products. In the restaurant domain, as in Figure 3-1, these are restaurants. We assume that each snippet is opinion-bearing and discusses one of the aspects which are relevant for that particular product. Our output consists of a set of dynamic (i.e., not pre-specified) aspects for each product, snippets labeled with the aspect which they discuss, and sentiment values for each snippet individually and each aspect as a whole. In Figure 3-1, the aspects identified for *Tasca Spanish Tapas* include *chicken*, *dessert*, and *drinks*, and the snippets are labeled with the aspects they describe and the correct polarity.

One way to approach this problem is to treat it as a multi-class classification problem. Given a set of predefined domain-specific aspects, it would be fairly straightforward for humans to identify which aspect a particular snippet describes. However, for our task of discovering fine-grained entity-specific aspects, there is no way to know a priori which aspects may be present across the entire data set or to provide training data for each; instead, we must select the aspects dynamically. Intuitively, one potential solution is to cluster the input snippets, grouping those which are lexically similar without prior knowledge of the aspects they represent. However, without some knowledge of which words represent the aspect for a given snippet, the clusters may not align to ones useful for cross-review analysis. Consider, for example, the two clusters of restaurant review snippets shown in Figure 3-2. While both clusters share many words among their members, only the first describes a coherent aspect cluster, namely the *drinks* aspect. The snippets of the second cluster do not discuss a single product aspect, but instead share expressions of sentiment.

To successfully navigate this challenge, we must distinguish between words which indicate aspect, words which indicate sentiment, and extraneous words which do neither. For both aspect identification and sentiment analysis, it is crucial to know which words within a snippet are relevant for the task. Distinguishing them is not straightforward, however. Some work in sentiment analysis relies on a predefined lexicon or WordNet to provide some hints, but there is no way to anticipate every

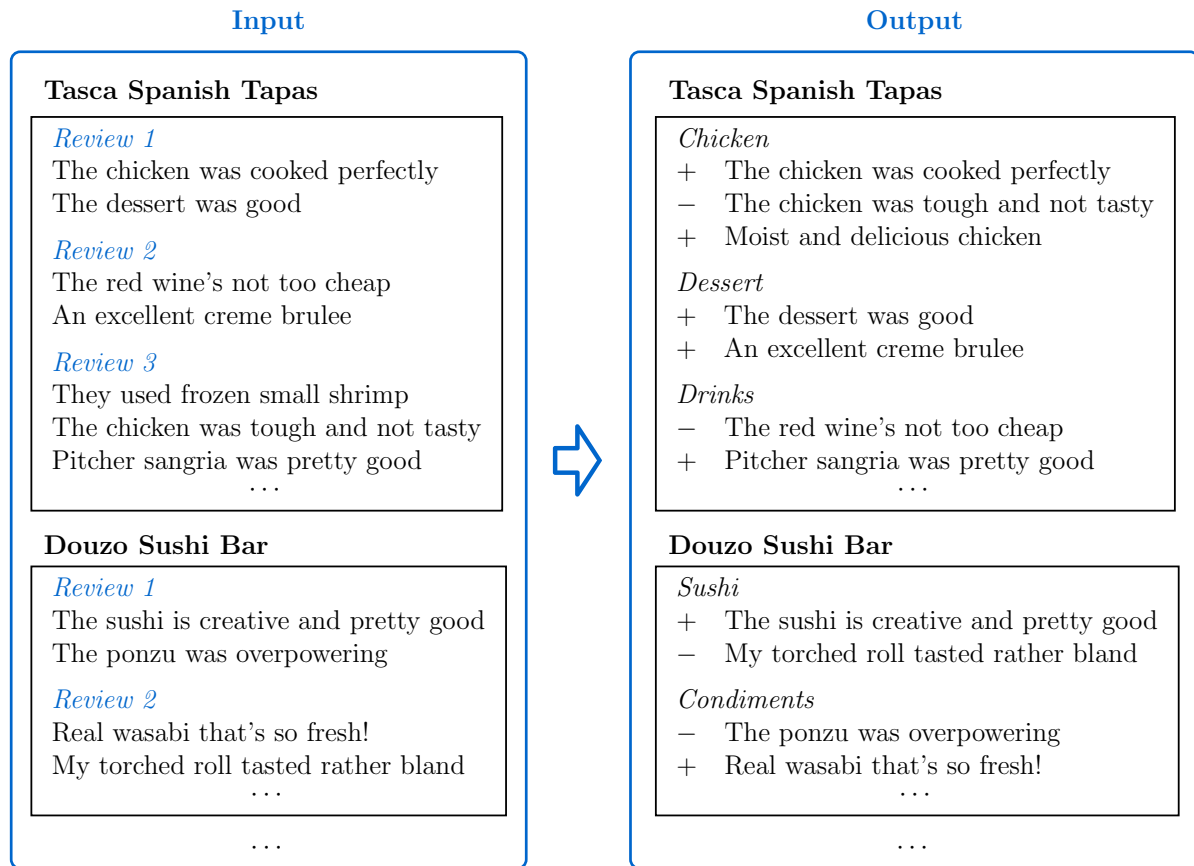


Figure 3-1: An example of the desired input and output of our system in the restaurant domain. The input consists of a collection of review snippets for several restaurants. The output is an aggregation of snippets by aspect (e.g., *chicken* and *dessert*) along with an associated sentiment for each snippet. Note that the input data is completely unannotated; the only information given is which snippets describe the same restaurant.

possible expression of aspect or sentiment, especially in user-generated data (e.g., use of slang such as “deeeeee-lish” for “delicious”). In lieu of an explicit lexicon, we can attempt to use other information as a proxy, such as part of speech; for example, aspect words are likely to be nouns, while value words are more likely to be adjectives. However, as we show later in this chapter, this additional information is again not sufficient for the tasks at hand.

Instead, we propose an approach to analyze a collection of product review snippets and jointly induce a set of learned aspects, each with a respective value (e.g., sentiment). We capture this idea using a generative Bayesian topic model where the

Coherent aspect cluster

	The <u>martinis</u> were very good.
+	The <u>drinks</u> - both <u>wine</u> and <u>martinis</u> - were tasty.
<hr/>	
	The <u>wine list</u> was pricey.
-	Their <u>wine selection</u> is horrible.

Incoherent aspect cluster

	The sushi is the <u>best I've ever had</u> .
+	<u>Best</u> paella <u>I'd ever had</u> .
	The fillet was the <u>best steak we'd ever had</u> .
	It's the <u>best soup I've ever had</u> .

Figure 3-2: Example clusters of restaurant review snippets generated by a lexical clustering algorithm; words relevant to clustering are highlighted. The first cluster represents a coherent *aspect* of the underlying product, namely the *drinks* aspect. The latter cluster simply shares a common sentiment expression and does not represent snippets discussing the same product aspect. In this work, we aim to produce the first type of aspect cluster along with the corresponding values.

set of aspects and any corresponding values are represented as hidden variables. The model takes a collection of snippets as input and explains how the observed text arises from the latent variables, thereby connecting text fragments with the corresponding aspects and values.

Specifically, we begin by defining sets of sentiment word distributions and aspect word distributions. Because we expect the types of sentiment words to be consistent across all products (e.g., any product may be labeled as “great” or “terrible”), we allow the positive and negative sentiment word distributions to be shared across all products. On the other hand, in the case of restaurant reviews and similar domains, aspect words are expected to be quite distinct between products. Therefore, we assign each product its own set of aspect word distributions. In addition to these word distributions, our model takes into account several other factors. First, we model the idea that each particular aspect of a product has some underlying quality; that is, if there are already 19 snippets praising a particular aspect, it’s likely that the 20th snippet will be positive as well. Second, we account for common patterns in language using a transition distribution between types of words. For example, it is very common to see the pattern “Value Aspect,” such as in phrases like “great pasta.”

Third, we model the distributions over parts of speech for each type of distribution. This covers the intuition that aspect words are frequently nouns, whereas value words are often adjectives. We describe each of these factors and our model as a whole in detail in Section 3.4.

This formulation provides several advantages: First, the model does not require a set of predefined aspects. Instead, it is capable of assigning latent variables to discover the appropriate aspects based on the data. Second, the joint analysis of aspect and value allows us to leverage several pieces of information to determine which words are relevant for aspect identification and which should be used for sentiment analysis, including part of speech and global or entity-specific distributions of words. Third, the Bayesian model admits an efficient mean-field variational inference procedure which can be parallelized and run quickly on even large numbers of entities and snippets.

We evaluate our approach on the domain of restaurant reviews. Specifically, we use a set of snippets automatically extracted from restaurant reviews on Yelp. This collection consists of an average of 42 snippets for each of 328 restaurants in the Boston area, representing a wide spectrum of opinions about several aspects of each restaurant. We demonstrate that our model can accurately identify clusters of review fragments that describe the same aspect, yielding 32.5% relative error reduction (9.9 absolute  $F_1$ ) over a standalone clustering baseline. We also show that the model can effectively identify snippet sentiment, with a 19.7% relative error reduction (4.3% absolute accuracy) over applicable baselines. Finally, we test the model’s ability to correctly label aspect and sentiment words, discovering that the aspect identification has high-precision, while the sentiment identification has high-recall.

Additionally, we apply a slimmed-down version of our model which focuses exclusively on aspect identification to a set of lab- and exam-related snippets from medical summaries provided by the Pediatric Environmental Health Clinic (PEHC) at Children’s Hospital Boston. These summaries represent concise overviews of the patient information at a particular visit, as relayed from the PEHC doctor to the child’s referring physician. Our model achieves 7.4% (0.7 absolute  $F_1$ ) over the standalone clustering baseline.

## 3.2 Related work

Our work falls into the area of multi-aspect sentiment analysis. In this section, we first describe approaches toward document-level and sentence-level sentiment analysis (Section 3.2.1), which provide the foundation for future work, including our own. Then, we describe three common directions of multi-aspect sentiment analysis; specifically, those which use data-mining or fixed-aspect analysis (Section 3.2.2.1), those which incorporate sentiment analysis with multi-document summarization (Section 3.2.2.2), and finally, those focused on topic modeling with additional sentiment components (Section 3.2.2.3).

### 3.2.1 Single-aspect sentiment analysis

Early sentiment analysis focused primarily on identification of coarse document-level sentiment [62, 82, 61]. Specifically, these approaches attempted to determine the overall polarity of documents. These approaches included both rule-based and machine learning approaches: Turney [82] used a rule-based method to extract potentially sentiment-bearing phrases and then compared them to the sentiment of known-polarity words, while Pang et al. [62] used discriminative methods with features such as unigrams, bigrams, part-of-speech tags, and word position information.

While document-level sentiment analysis can give us the overall view of an opinion, looking at individual sentences within the document yields a more fine-grained analysis. The work in sentence-level sentiment analysis focuses on first identifying sentiment-bearing sentences and then determining their polarity [87, 24, 42, 43, 61]. Both identification of sentiment-bearing sentences and polarity analysis can be performed through supervised classifiers [87, 24] or similarity to known text [87, 42], through measures based on distributional similarity or by using WordNet relationships.

By recognizing connections between parts of a document, sentiment analysis can be further improved [59, 55, 61]. Pang and Lee [59] leverage the relationship between sentences to improve document-level sentiment analysis. Specifically, they utilize both

the subjectivity of individual sentences and information about the strength of connection between sentences in a min cut formulation to provide better sentiment-focused summaries of text. McDonald et al. [55] examine a different connection, instead constructing a hierarchical model of sentiment between sentences and documents. Their model uses complete labeling on a subset of data to learn a generalized set of parameters which improve classification accuracy at both document-level and sentence-level.

While none of the above approaches attempt to identify aspects or analyze sentiment in an aspect-based fashion, the intuitions provide key insight into the approaches we take in our work. For example, the importance of distinguishing opinion sentences follows our own intuition about the necessity of identifying sentiment-bearing words within a snippet.

### 3.2.2 Aspect-based sentiment analysis

Following the work in single-aspect document-level and sentence-level sentiment analysis came the intuition of modeling aspect-based (also called “feature-based”) sentiment for review analysis. We can divide these approaches roughly into three types of systems based on their techniques: systems which use fixed-aspect approaches or data-mining techniques for aspect selection or sentiment analysis, systems which adapt techniques from multi-document summarization, and systems which jointly model aspect and sentiment with probabilistic topic models. Here, we examine each avenue of work with relevant examples and contrast them with our own work.

#### 3.2.2.1 Data-mining and fixed-aspect techniques for sentiment analysis

One set of approaches toward aspect-based sentiment analysis follow the traditional techniques of data mining [38, 46, 64]. These systems may operate on full documents or on snippets, and they generally require rule-based templates or additional resources such as WordNet both to identify aspects and to determine sentiment polarity. Another approach is to fix a predetermined relevant set of aspects, then focus on learning the optimal opinion assignment for these aspects [74]. Below, we summarize



each approach and compare and contrast them to our work.

One set of work relies on a combination of association mining and rule-based extraction of nouns and noun phrases for aspect identification. Hu and Liu [38], Liu et al. [46] developed a three-step system: First, initial aspects are selected by an association miner and pruned by a series of rules. Second, related opinions for each aspect are identified in a rule-based fashion using word positions, and their polarity is determined by WordNet search based on a set of seed words. Third, additional aspects are identified in a similar fashion based on position of the selected polarity words. In each of these steps, part-of-speech information provides a key role in the extraction rules. In Liu et al. [46], there is an additional component to identify *implicit* aspects in a deterministic fashion; e.g., *heavy* maps deterministically to <WEIGHT>. While their task is similar to ours and we utilize part-of-speech information as an important feature as well, we additionally leverage other distributional information to identify aspects and sentiment. Furthermore, we avoid the reliance on WordNet and predefined rule mappings in order to preserve the generality of the system. Instead, our joint modeling allows us to recover these relationships without the need for additional information.

Other approaches also rely on WordNet relationships to identify not only sentiment polarity, but also aspects, using the *parts* and *properties* of a particular product class. Popescu et al. [64] first use these relations to generate the set of aspects for a given product class (e.g., *camera*). Following that, they apply relaxation labeling for sentiment analysis. This procedure gradually expands sentiment from individual words to aspects to sentences, similar to the *Cascade* pattern mentioned in McDonald et al. [55]. Like the system of Liu et al. [46], their system requires a set of manual rules and several outside resources. While our model does require a few seed words, it does not require any manual rules or additional resources due to its joint formulation.

A separate direction of work relies on predefined aspects while focusing on improvement of sentiment analysis prediction. Snyder and Barzilay [74] define a set of aspects specific to the restaurant domain. Specifically they define an individual rating model for each aspect, plus an overall agreement model which attempts to determine

whether the resulting ratings should all agree or disagree. These models are jointly trained in a supervised fashion using an extension of the PRanking algorithm [20] to find the best overall star rating for each aspect. Our problem formulation differs significantly from their work in several dimensions: First, we desire a more refined analysis using fine-grained aspects instead of coarse predefined features. Second, we would like to use as little supervised training data as possible, rather than the supervised training required for the PRanking algorithm.

Liu and Seneff [48] also use an approach based on predefined aspects; specifically, they focus on a three-step model: First, the model uses a set of generation rules to extract related text and rewrite it as an easily-comparable paraphrase. Second, topics are generated through a clustering algorithm focused on the descriptive portions of the paraphrases. Finally, a sentiment classifier is learned through matching paraphrases with the user’s overall star rating. Liu et al. [49] extend this work to include additional fine-grained topics based on words scraped from online menus and other resources. In our approach, we would like to capture a similar type of phrase information; however, we would like to avoid reliance on predefined aspects and outside resources such as menu information.

In our work, we attempt to capture the intuitions of these approaches while reducing the need for outside resources and rule-based components. For example, rather than supplying rule-based patterns for extraction of aspect and sentiment, we instead leverage distributional patterns across the corpus to infer the relationships between words of different types. Likewise, rather than relying on WordNet relationships such as synonymy, antonymy, hyponymy, or hypernymy [38, 46, 64], we bootstrap our model from a small set of seed words.

### 3.2.2.2 Multi-document summarization and its application to sentiment analysis

Multi-document summarization techniques generally look for repetition across documents to signal important information [68, 3, 67, 51]. For aspect-based sentiment analysis, work has focused on augmenting these techniques with additional components for sentiment analysis [71, 72, 12, 41]. In general, the end goal of these ap-

proaches is the task of forming coherent text summaries using either text extraction or natural language generation. Unlike our work, many of these approaches do not explicitly identify aspects; instead, they are extracted through repeated information. Additionally, our model explicitly looks at the connection between content and sentiment, rather than treating it as a secondary computation after information has been selected.

One technique for incorporating sentiment analysis follows previous work on identification of opinion-bearing sentences. Seki et al. [71, 72] present DUC summarization systems designed to create opinion-focused summaries of task topics.<sup>1</sup> In their system, they employ a subjectivity component using a supervised SVM with lexical features, similar to those in Yu and Hatzivassiloglou [87], Dave et al. [24]. This component is used to identify subjective sentences and, in Seki et al. [72], their polarity, both in the task and in the sentences selected for the response summary. However, like previous work and unlike our task, there is no aspect-based analysis in their summarization task. It is also fully supervised, relying on a hand-annotated set of about 10,000 sentences to train the SVM.

Another line of work focuses on augmenting the summarization system with aspect selection similar to the data-mining approaches of Hu and Liu [38], rather than using single-aspect analysis. Carenini et al. [11, 12] augment the previous aspect selection with a user-defined hierarchical organization over aspects; e.g., *digital zoom* is part of the *lens*. Polarity of each aspect is assumed to be given by previous work. These aspects are then incorporated into existing summarization systems – MEAD\* sentence extraction [67] or SEA natural language generation [10] – to form final summaries. Like the work of Seki et al. [71, 72], this work does not create new techniques for aspect identification or sentiment analysis; instead, they focus on the process of integrating these sources of information with summarization systems. While the aspects produced are comparable across reviews for a particular product, the highly-supervised nature means that this approach is not feasible for a large set of products such as our corpus of reviews from many types of restaurants. Instead, we must be able to dynamically

---

<sup>1</sup>For task examples, see Dang [21, 22].

identify relevant aspects.

A final line of related work relies on the traditional summarization technique of identifying contrastive or contradictory sentences. Kim and Zhai [41] focus on generating contrastive summaries by identifying pairs of sentences which express differing opinions on a particular product feature. To do this, they define metrics of *representativeness* (coverage of opinions) and *contrastiveness* (alignment quality) using both semantic similarity with WordNet matches and word overlap. In comparison to our work, this approach follows an orthogonal goal, as we try to find the most defining aspects instead of the most contradictory ones. Additionally, while the selected pairs hint at disagreements in rating, there is no identification of how many people agree with each side or the overall rating of a particular aspect. In our work, we aim to produce both a concrete set of aspects and the user sentiment for each, whether it is unanimous or shows disagreement.

Overall, while these methods are designed to produce output summaries which focus on subjective information, they are not specifically targeted for aspect-based analysis. Instead, aspects are identified in a supervised fashion [11, 12] or are not defined at all [71, 72, 41]. In our work, it is crucial that we have dynamically-selected aspects because it is not feasible to preselect aspects in a supervised fashion.

### 3.2.2.3 Probabilistic topic modeling for sentiment analysis

The work closest to our own in the direction of aspect-based analysis focuses on the use of probabilistic topic modeling techniques for identification of aspects. These may be aggregated without specific sentiment polarity [50] or combined with additional sentiment modeling either jointly [56, 7, 81] or as a separate post-processing step [80]. Like our work, these approaches share the intuition that aspects may be represented as topics.

Several approaches focus on extraction of topics and sentiment from blog articles. In one approach, they are used as expert articles for aspect extraction in combination with a larger corpus of user reviews. Lu and Zhai [50] introduce a model with semi-supervised probabilistic latent semantic analysis (PLSA) which identifies sentiment-

bearing aspects through segmentation of an expert review. Then, the model extracts compatible supporting and supplementary text for each aspect from the set of user reviews. Aspect selection is constrained as in the rule-based approaches; specifically, aspect words are required to be nouns. Our work differs from their work significantly. While we share a common goal of identifying and aggregating opinion-bearing aspects, we additionally desire to identify the polarity of opinions, a task not addressed in their work. In addition, obtaining aspects from an expert review is unnecessarily constraining; in practice, while expert reviewers may mention some key aspects, they will not mention every aspect. It is crucial to discover aspects based on the entire set of articles.

There is work in the direction of aspect identification from blog posts. For example, Mei et al. [56] use a variation on latent Dirichlet allocation (LDA) similar to our own to explicitly model both topics and sentiment, then use a hidden Markov model to discover sentiment dynamics across topic life cycles. A general sentiment polarity distribution is computed by combining distributions from several separate labeled data sets (e.g., movies, cities, etc.). However, in their work, sentiment is measured at the document-level, rather than topic-level. Additionally, the topics discovered by their model are very broad; for example, when processing the query “The Da Vinci Code”, returned topics may be labeled as *book*, *movie*, and *religion*, rather than the fine-grained aspects we desire in our model, such as those representing major characters or events. Our model expands on their work by discovering very fine-grained aspects and associating particular sentiment with each individual aspect. In addition, by tying sentiment to aspects, we are able to identify sentiment-bearing words and their associated polarities without the additional annotation required to train an external sentiment model.

Sentiment may also be combined with LDA using additional latent variables for each document in order to predict document-level sentiment. Blei and McAuliffe [7] propose a form of supervised LDA (sLDA) which incorporates an additional response variable, which can be used to represent sentiment such as the star rating of a movie. They can then jointly model the documents and responses in order to find the latent

topics which best predict the response variables for future unlabeled documents. This work is significantly different from our work, as it is supervised and does not predict in a multi-aspect framework.

Building on these approaches comes work in fine-grained aspect identification with sentiment analysis. Titov and McDonald [81, 80] introduce a multi-grain unsupervised topic model, specifically built as an extension to LDA. This technique yields a mixture of global and local topics. Word distributions for all topics (both global and local) are drawn at the global level, however; unlike our model. The consequence of this is that topics are very easy to compare across all products in the corpus; however, the topics are more general and less dynamic than we hope to achieve. One consequence of defining global topics is difficulty in finding relevant topics for every product when there is little overlap. For example, in the case of restaurant reviews, Italian restaurants should have a completely different set of aspects than Indian restaurants. Of course, if these factors were known, it would be possible to run the algorithm separately on each subset of restaurants, but these distinctions are not immediately clear a priori. For sentiment analysis, the PRanking algorithm of Snyder and Barzilay [74] is incorporated in two ways: In Titov and McDonald [80], the PRanking algorithm is trained in a pipeline fashion after all topics are generated, while in Titov and McDonald [81], it is incorporated into the model during inference in a joint formulation. However, in both cases, as in the original algorithm, the set of aspects is fixed – each of the aspects corresponds to a fixed set of topics found by the model. Additionally, the learning problem is supervised. Because of the fixed aspects, necessary additional supervision, and global topic distribution, this model formulation is not sufficient for our problem domain, which requires very fine-grained aspects.

All of these approaches have structural similarity to the work we present, as they are variations on LDA. None, however, has the same intent as our model. Mei et al. [56] model aspect and sentiment jointly; however their aspects are very vague, and they treat sentiment at the document level rather than the aspect level. Likewise, Titov and McDonald [80, 81] model “fine-grained” aspects, but they are still coarser

than the aspects we require, as their distributions are shared globally. Finally, Lu and Zhai [50], Blei and McAuliffe [7], Titov and McDonald [80, 81] require supervised annotation or a supervised expert review that we do not have. We attempt to solve each of these issues with our joint formulation in order to proceed with minimal supervision and discover truly fine-grained aspects.

### 3.3 Problem formulation

Before explaining the model details, we describe the random variables and abstractions of our model, as well as some intuitions and assumptions.<sup>2</sup> A visual explanation of model components is shown in Figure 3-3. We present complete details and the generative story in Section 3.4.

#### 3.3.1 Model components

**Entity** An entity represents a single object which is described in the review. In the restaurant domain, these represent individual restaurants, such as *Tasca Spanish Tapas*, *Douzo Sushi Bar*, and *Outback Steakhouse*.

**Snippet** A snippet is a user-generated short sequence of words describing an entity. These snippets can be provided by the user as is (for example, in a “quick reaction” box) or extracted from complete reviews through a phrase extraction system such as the one from Sauper et al. [70]. We assume that each snippet contains at most one single aspect (e.g., *pizza*) and one single value type (e.g., *positive*). In the restaurant domain, this corresponds to giving an opinion about one particular dish or category of dishes. Examples from the restaurant domain include *Their pasta dishes are perfection itself*, **they had fantastic drinks**, and *the lasagna rustica was cooked perfectly*.

---

<sup>2</sup>Here, we explain our complete model with value selection for sentiment in the restaurant domain. For the simplified case in the medical domain where we would like to use only aspects, we may simply ignore the value-related components of the model.

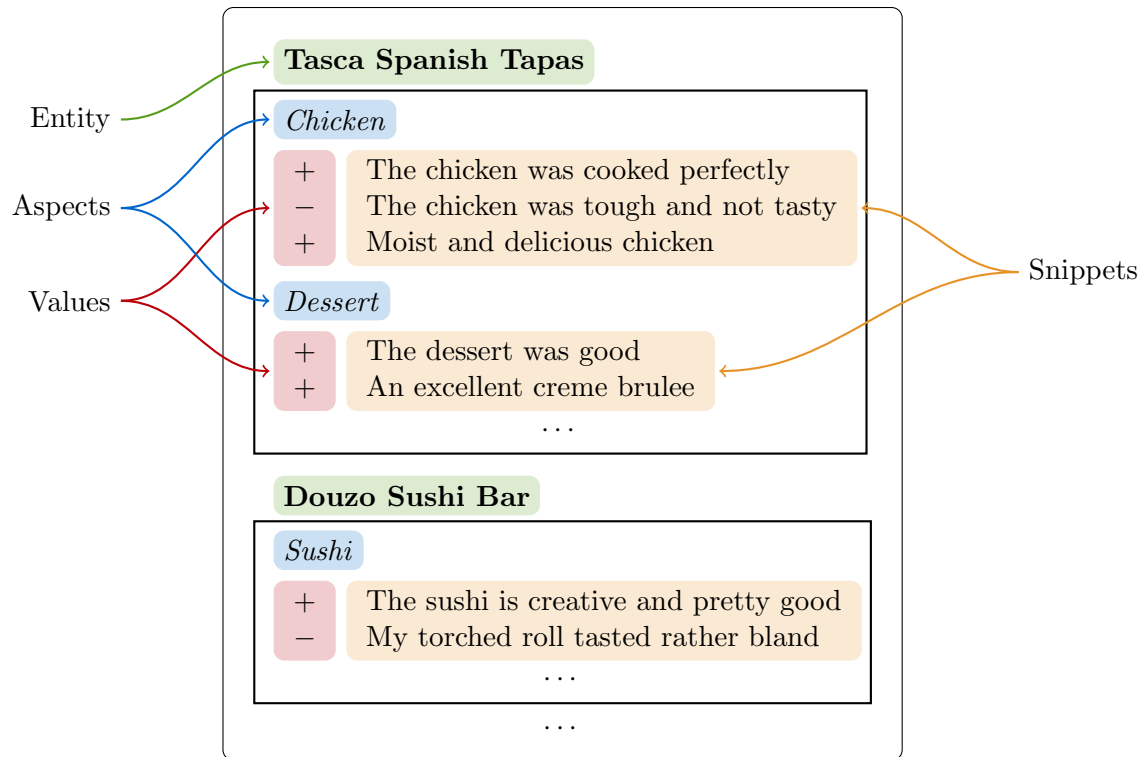


Figure 3-3: Labeled model components from the example in Figure 3-1. Note that aspects are never given explicit labels, and the ones shown here are presented purely for ease of understanding; aspects exist simply as groups of snippets which share a common subject. Also, word topics are not pictured here; a word topic (Aspect, Value, or Background) is assigned to each word in each snippet. These model components are described at high level in Section 3.3.1 and in depth in Section 3.4.

**Aspect** An aspect corresponds to one of several properties of an entity. In the restaurant domain where entities represent restaurants, aspects may correspond to individual dishes or categories of dishes, such as *pizza* or *alcoholic drinks*. For this domain, each entity has its own unique set of aspects. This allows us to model aspects at the appropriate granularity. For example, an Italian restaurant may have a *dessert* aspect which pertains to information about a variety of cakes, pies, and gelato. However, most of a bakery’s menu would fall under that same *dessert* aspect. Instead, to present a useful aspect-based summary, it would require separate aspects for each of *cakes*, *pies*, and so on. Because aspects are entity-specific rather than shared, there are no ties between restaurants which have aspects in common (e.g., most sushi restaurants will have a *sashimi* aspect); we consider this a point for potential future



work. Note that it is still possible to compare aspects across entities (e.g., to find the best restaurant for a *burger*) by comparing their respective word distributions.

**Value** Values represent the information associated with an aspect. In the review domain, the two value types represent positive and negative sentiment respectively. In general, it is possible to use value to represent other distinctions; for example, in a domain where some aspects are associated with a numeric value and others are associated with a text description, each of these can be set as a value type. The intended distinctions may be encouraged by the use of seed words (see Section 3.3.2), or they may be left unspecified for the model to assign whatever it finds to best fit the data. The number of value types must be prespecified; however, it is possible to use either very few or very many types.

**Word Topic** While the words in each snippet are observed, each word is associated with an underlying latent topic. The possible latent topics correspond to aspect, value, and a background topic. For example, in the review domain, the latent topic of words *great* or *terrible* would be **Value**, of words which represent entity aspects such as *pizza* would be **Aspect**, and of stop words like *is* or of in-domain white noise like *food* would be **Background**.

### 3.3.2 Problem setup

In this work, we assume that the snippet words are always observed, and the correlation between snippets and entities is known (i.e., we know which entity a given snippet describes). In addition, we assume part of speech tags for each word in each snippet. As a final source of supervision, we may optionally include small sets of seed words for a lexical distribution, in order to bias the distribution toward the intended meaning. For example, in the sentiment case, we can bias one value distribution toward positive and one toward negative.

Note that in this formulation, the relevant aspects for each restaurant are **not** observed; instead, they are represented by lexical distributions which are induced at

inference time. In the system output, aspects are represented as unlabeled clusters over snippets.<sup>3</sup> Given this formulation, the goal of this work is then to induce the latent aspect and value underlying each snippet.

## 3.4 Model

Our model has a generative formulation over all snippets in the corpus. In this section, we first describe in detail the general formulation and notation of the model, then discuss novel changes and enhancements for particular corpora types. Inference for this model will be discussed in Section 3.5. As mentioned previously, we will describe the complete model including aspect values.

### 3.4.1 General formulation

For this model, we assume a collection of all snippet words for all entities,  $\mathbf{s}$ . We use  $s^{i,j,w}$  to denote the  $w$ th word of the  $j$ th snippet of the  $i$ th entity. We also assume a fixed vocabulary of words  $W$ .

We present a summary of notation in Table 3.1, a concise summary of the model in Figure 3-4, and a model diagram in Figure 3-5. There are three levels in the model design: global distributions common to all snippets for all entities in the collection, entity-level distributions common to all snippets describing a single entity, and snippet- and word-level random variables. Here, we describe each in turn.

#### Global distributions

At the global level, we draw a set of distributions common to all entities in the corpus. These include everything shared across a domain, such as the background stop-word distribution, value types, and word topic transitions.

---

<sup>3</sup>If a label is desired, we can automatically extract one by selecting the highest probability word or words for a particular aspect. For the examples in this paper, we provide manual cluster labels for illustration purposes.

Data Set	
$\mathbf{s}$	Collection of all snippet words from all entities
$s^{i,j,w}$	$w$ th word of $j$ th snippet of $i$ th entity
$t^{i,j,w}$ *	Part-of-speech tag corresponding to $s^{i,j,w}$
$W$	Fixed vocabulary
$W_{seed_v}$	Seed words for value type $v$
Lexical Distributions	
$\theta_B$	Background word distribution
$\theta_A^{i,a}$ ( $\theta_A^a$ *)	Aspect word distribution for aspect $a$ of entity $i$
$\theta_V^v$	Value word distribution for type $v$
$\theta_I$ *	Ignored words distribution
Other Distributions	
$\Lambda$	Transition distribution over word topics
$\phi^{i,a}$ ( $\phi^a$ *)	Aspect-value multinomial for aspect $a$ of entity $i$
$\psi^i$ ( $\psi$ *)	Aspect multinomial for entity $i$
$\eta$ *	Part-of-speech tag distribution
Latent Variables	
$Z_A^{i,j}$	Aspect selected for $s^{i,j}$
$Z_V^{i,j}$	Value type selected for $s^{i,j}$
$Z_W^{i,j,w}$	Word topic ( $A, V, B, I$ *) selected for $s^{i,j,w}$
Other Notation	
$K$	Number of aspects $a$
$A$	Indicator corresponding to aspect word
$V$	Indicator corresponding to value word
$B$	Indicator corresponding to background word
$I$ *	Indicator corresponding to ignored word

Table 3.1: Notation used in this chapter. Items marked with \* relate to extensions mentioned in Section 3.4.2.

**Global Level:**

Draw background word distribution  $\theta_B \sim \text{DIRICHLET}(\lambda_B W)$

For each value type  $v$ ,

Draw value word distribution  $\theta_V^v \sim \text{DIRICHLET}(\epsilon W + \lambda_V W_{seed_v})$

**Entity Level:**

For each entity  $i$ ,

Draw aspect word distributions  $\theta_A^{i,a} \sim \text{DIRICHLET}(\lambda_A W)$  for  $a = 1, \dots, K$

Draw aspect value multinomial  $\phi^{i,a} \sim \text{DIRICHLET}(\lambda_{AV} N)$  for  $a = 1, \dots, K$

Draw aspect multinomial  $\psi^i \sim \text{DIRICHLET}(\lambda_M K)$

**Snippet Level:**

For each snippet  $j$  describing the  $i$ th entity,

Draw snippet aspect  $Z_A^{i,j} \sim \psi^i$

Draw snippet value  $Z_V^{i,j} \sim \phi^{i,Z_A^{i,j}}$

Draw sequence of word topic indicators  $Z_W^{i,j,w} \sim \Lambda | Z_W^{i,j,w-1}$

Draw snippet word given aspect  $Z_A^{i,j}$  and value  $Z_V^{i,j}$

$$s_{i,j,w} \sim \begin{cases} \theta_A^{i,Z_A^{i,j}}, & \text{when } Z_W^{i,j,w} = A \\ \theta_V^{Z_V^{i,j}}, & \text{when } Z_W^{i,j,w} = V \\ \theta_B, & \text{when } Z_W^{i,j,w} = B \end{cases}$$

Figure 3-4: A summary of our generative model presented in Section 3.4.1. We use  $\text{DIRICHLET}(\lambda W)$  to denote a finite Dirichlet prior where the hyper-parameter counts are a scalar times the unit vector of vocabulary items. For the global value word distribution, the prior hyper-parameter counts are  $\epsilon$  for all vocabulary items and  $\lambda_V$  for  $W_{seed_v}$ , the vector of vocabulary items in the set of seed words for value  $v$ .

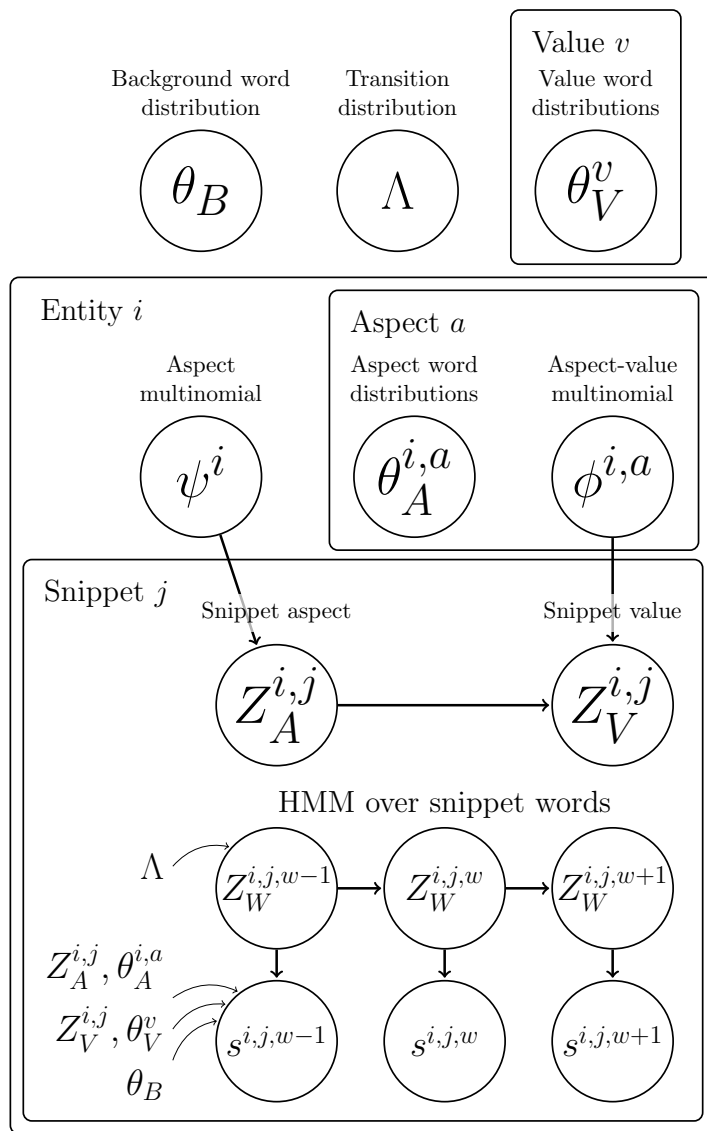


Figure 3-5: A graphical description of the model presented in Section 3.4.1. A written description of the generative process is located in Figure 3-4. Curved arrows indicate additional links which are present in the model but not drawn for readability.

**Background Distribution** A global background word distribution  $\theta_B$  is drawn to represent stop-words and in-domain white noise (e.g., “food” becomes white noise in a corpus of restaurant reviews). This distribution is drawn from a symmetric Dirichlet with concentration parameter  $\lambda_B$ .

**Value Distributions** A value word distribution  $\theta_V^v$  is drawn for each value type  $v$ . For example, in a review domain with positive and negative sentiment types, there will be a distribution over words for the positive type and one for the negative type. Seed words  $W_{seed_v}$  are given additional probability mass on the value priors for type  $v$ ; specifically, a non-seed word receives  $\epsilon$  hyperparameter, while a seed word receives  $\epsilon + \lambda_V$ .

**Transition Distribution** A transition distribution  $\Lambda$  is drawn to represent the transition probabilities of underlying word topics. For example, it may be very likely to have a **Value Aspect** transition in a review domain, which fits phrases like “great pizza.” This distribution is given a slight bias toward more helpful transitions; for example, encouraging sticky behavior by providing a small boost to self-transitions. This bias is easily overridden by data; however, it provides a useful starting point.

### Entity-specific distributions

There are naturally variations in the aspects which snippets describe and how many snippets describe each aspect. For example, a mobile device popular for long battery life will likely have more snippets describing the battery than a device which is known for its large screen. Some domains may have enormous variation in aspect vocabulary; for example, in restaurant reviews, two restaurants may not serve any of the same food items to compare. To account for these variations, we define a set of entity-specific distributions which generate both aspect vocabulary and popularity, as well as a distribution over value types for each aspect.

**Aspect Distributions** An aspect word distribution  $\theta_A^{i,a}$  is drawn for each aspect  $a$ . Each of these represents the distribution over unigrams for a particular aspect. For

example, in the domain of restaurant reviews, aspects may correspond to menu items such as *pizza*, while in reviews for cell phones, they may correspond to details such as *battery life*. Each aspect word distribution is drawn from a symmetric Dirichlet prior with hyperparameter  $\lambda_A$ .

**Aspect-Value Multinomials** Aspect-value multinomials  $\phi^{i,a}$  determine the likelihood of each value type  $v$  for the corresponding aspect  $a$ . For example, if value types represent positive and negative sentiment, this corresponds to agreement of sentiment across snippets. Likewise, if value types represent formatting such as integers, decimals, and text, each aspect generally prefers the same type of value. These multinomials are drawn from a symmetric Dirichlet prior using hyperparameter  $\lambda_{AV}$ .

**Aspect Multinomial** The aspect multinomial  $\psi^i$  controls the likelihood of each aspect being discussed in a given snippet. This encodes the intuition that certain aspects are more likely to be discussed than others for a given entity. For example, if a particular Italian restaurant is famous for their *pizza*, it is likely that the *pizza* aspect will be frequently discussed in reviews, while the *drinks* aspect may be mentioned only occasionally. The aspect multinomial will encode this as a higher likelihood for choosing *pizza* as a snippet aspect than *drinks*. This multinomial is drawn from a symmetric Dirichlet distribution with hyperparameter  $\lambda_M$ .

### Snippet- and word-specific random variables

Using the distributions described above, we can now draw random variables for each snippet to determine the aspect and value type which will be described, as well as the sequence of underlying word topics and words.

**Aspect** A single aspect  $Z_A^{i,j}$  which this snippet will describe is drawn from the aspect multinomial  $\psi^i$ . All aspect words in the snippet (e.g., *pizza* in a corpus of restaurant reviews) will be drawn from the corresponding aspect word distribution  $\theta_A^{i,Z_A^{i,j}}$ .

**Value Type** A single value type  $Z_V^{i,j}$  is drawn conditioned on the selected aspect from the corresponding aspect-value multinomial  $\phi^{i,Z_A^{i,j}}$ . All value words in the snippet (e.g., “great” in the review domain) will be drawn from the corresponding value word distribution  $\theta_V^{Z_V^{i,j}}$ .

**Word Topic Indicators** A sequence of word topic indicators  $Z_W^{i,j,1}, \dots, Z_W^{i,j,m}$  is generated using a first-order Markov model parameterized by the transition matrix  $\Lambda$ . These indicators determine which unigram distribution generates each word in the snippet. For example, if  $Z_W^{i,j,w} = B$ , the  $w$ th word of this snippet is generated from the background word distribution  $\theta_B$ .

### 3.4.2 Model extensions

There are a few optional components of the model which may improve performance for some cases. We briefly list them here, then present the necessary modifications to the model in detail for each case. Modifications to the inference procedure will be presented in Section 3.5.1. First, for corpora which contain irrelevant snippets, we may introduce an additional word distribution  $\theta_I$  and word topic **Ignore** to allow the model to ignore certain snippets or pieces of snippets altogether. Second, if it is possible to acquire part of speech tags for the snippets, using these as an extra piece of information is quite beneficial. Finally, for corpora where every entity is expected to share the same aspects, the model can be altered to use the same set of aspect distributions for all entities.

#### Ignoring snippets

When snippet data is automatically extracted, it may be noisy, and some snippets may violate our initial assumptions of having one aspect and one value. For example, we find some snippets which were mistakenly extracted that have neither aspect nor value. These extraneous snippets may be difficult to identify a priori. To compensate for this, we modify the model to allow partial or entire snippets to be ignored through



the addition of a global unigram distribution, namely the Ignore distribution  $\theta_I$ . This distribution is drawn from a symmetric Dirichlet with concentration parameter  $\lambda_I$ .

In order to successfully incorporate this distribution into our model, we must allow the word topic indicator  $Z_W^{i,j,w}$  to consider the Ignore topic. Additionally, because this distribution is intended to select whole snippets or large portions of snippets, we give a large boost to the prior of the `Ignore Ignore` sequence in the transition distribution  $\Lambda$ .

### Part-of-speech tags

Part-of-speech tags can provide valuable evidence in determining which snippet words are drawn from each distribution. For example, aspect words are often nouns, as they represent concrete properties or concepts in a domain. Likewise, in some domains, value words describe aspects and therefore tend to be expressed as numbers or adjectives.

This intuition can be directly incorporated into the model in the form of additional outputs. Specifically, we modify our HMM to produce both words and tags. Additionally, we define distributions over tags  $\eta_A^a$ ,  $\eta_V^v$ , and  $\eta_B$ , similar to the corresponding unigram distributions.

### Shared aspects

When domains are very regular, and every entity is expected to express aspects from a consistent set, it is beneficial to share aspect information across entities. For example, in a medical domain, the same general set of lab tests and physical exam categories are run on all patients. Note that this is quite unlike the restaurant review case, where each restaurant’s aspects are completely different (e.g., *pizza*, *curry*, *scones*, and so on).

Sharing aspects in this way can be accomplished by modifying the aspect distributions  $\theta_A^{i,a}$  to become global distributions  $\theta_A^a$ . Likewise, aspect-value multinomials  $\phi^{i,a}$  become shared across all entities as  $\phi^a$ . Treatment of the aspect multinomials depend on the domain properties. If the distribution over aspects is expected to be

the same across all entities, it can also be made global; however, if each individual entity is expected to exhibit variation in the number of snippets related to each aspect, they should be kept as entity-specific. For example, reviews for a set of cell phones may be expected to focus on varying parts, depending on what is most unique or problematic about those phones. A graphical description of these changes compared to the original model is shown in Figure 3-6.

### 3.5 Inference

The goal of inference in this model is to predict the aspect and value for each snippet  $i$  and product  $j$ , given the text of all observed snippets, while marginalizing out the remaining hidden parameters:

$$P(Z_A^{i,j}, Z_V^{i,j} | \mathbf{s})$$

We accomplish this task using variational inference [8]. Specifically, the goal of variational inference is to find a tractable approximation  $Q(\cdot)$  to the full posterior of the model.

$$P(\theta_B, \boldsymbol{\theta}_V, \Lambda, \boldsymbol{\theta}_A, \boldsymbol{\psi}, \boldsymbol{\phi}, \mathbf{Z} | \mathbf{s}) \approx Q(\theta_B, \boldsymbol{\theta}_V, \Lambda, \boldsymbol{\theta}_A, \boldsymbol{\psi}, \boldsymbol{\phi}, \mathbf{Z})$$

For our model, we assume a full mean-field factorization of the variational distribution, shown in Figure 3-7. This variational approximation is defined as a product of factors  $q(\cdot)$ , which are assumed to be independent. This approximation allows for tractable inference of each factor individually. To obtain the closest possible approximation, we attempt to set the  $q(\cdot)$  factors to minimize the KL divergence to the true model posterior:

$$\arg \min_{Q(\cdot)} KL(Q(\theta_B, \boldsymbol{\theta}_V, \Lambda, \boldsymbol{\theta}_A, \boldsymbol{\psi}, \boldsymbol{\phi}, \mathbf{Z}) || P(\theta_B, \boldsymbol{\theta}_V, \Lambda, \boldsymbol{\theta}_A, \boldsymbol{\psi}, \boldsymbol{\phi}, \mathbf{Z} | \mathbf{s}))$$

We optimize this objective using coordinate descent on the  $q(\cdot)$  factors. Concretely, we update each factor by optimizing the above criterion with all other factors

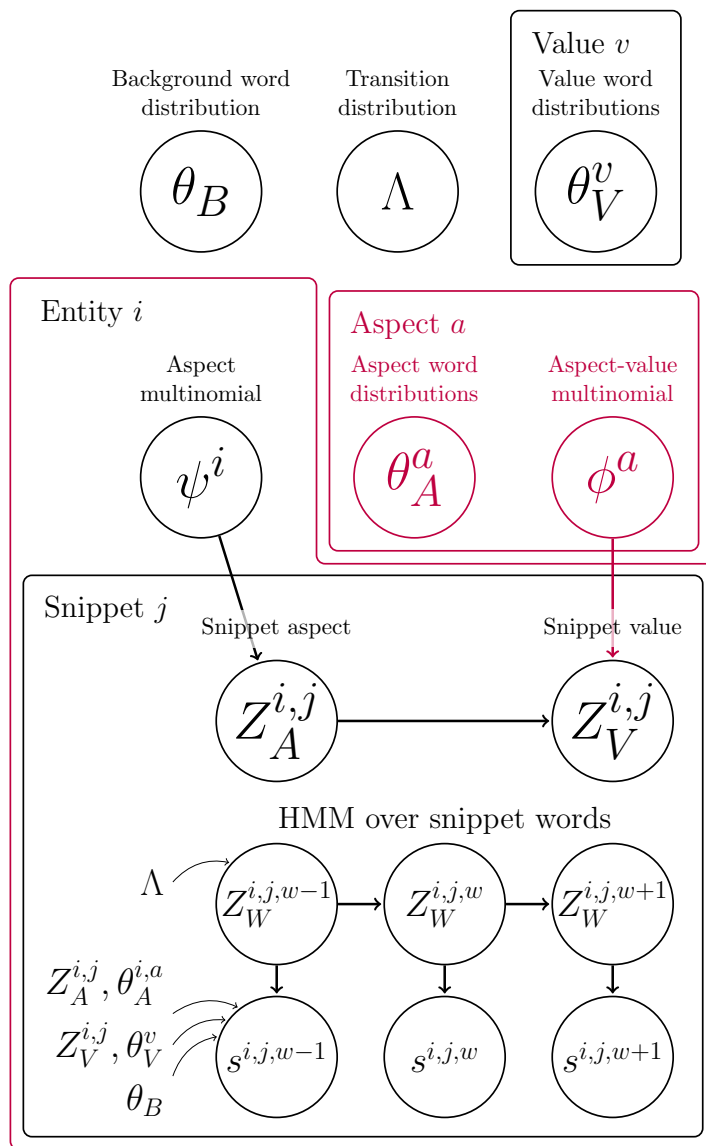


Figure 3-6: A graphical description of the model with shared aspects presented in Section 3.4.2. Note the similarities to Figure 3-5; however in this version, aspects are shared for the entire corpus, rather than being entity-specific. It would also be possible to share the aspect multinomial corpus-wide; in that case it would indicate that all entities share the same general distribution over aspects, while in this version the individual entities are allowed to have completely different distributions.

**Mean-field Factorization**

$$Q(\theta_B, \theta_V, \Lambda, \theta_A, \psi, \phi, \mathbf{Z}) = q(\theta_B) q(\Lambda) \left( \prod_{v=1}^N q(\theta_V^v) \right) \left( \prod_i q(\psi^i) \left( \prod_{a=1}^K q(\theta_A^{i,a}) q(\phi^{i,a}) \right) \left( \prod_j q(Z_V^{i,j}) q(Z_A^{i,j}) \prod_w q(Z_W^{i,j,w}) \right) \right)$$

**Snippet Aspect Indicator**

$$\log q(Z_A^{i,j} = a) \propto \mathbb{E}_{q(\psi^i)} \log \psi^i(a) + \sum_w q(Z_W^{i,j,w} = A) \mathbb{E}_{q(\theta_A^{i,a})} \log \theta_A^{i,a}(s^{i,j,w}) + \sum_{v=1}^N q(Z_V^{i,j} = v) \mathbb{E}_{q(\phi^{i,a})} \log \phi^{i,a}(v)$$

**Snippet Value Type Indicator**

$$\log q(Z_V^{i,j} = v) \propto \sum_a q(Z_A^{i,j} = a) \mathbb{E}_{q(\phi^{i,a})} \log \phi^{i,a}(v) + \sum_w q(Z_W^{i,j,w} = V) \mathbb{E}_{q(\theta_V^v)} \log \theta_V^v(s^{i,j,w})$$

**Word Topic Indicator**

$$\log q(Z_W^{i,j,w} = A) \propto \log P(Z_W = A) + \mathbb{E}_{q(\Lambda)} \log \left( \Lambda(Z_W^{i,j,w-1}, A) \Lambda(A, Z_W^{i,j,w+1}) \right) + \sum_a q(Z_A^{i,j} = a) \mathbb{E}_{q(\theta_A^{i,a})} \log \theta_A^{i,j}(s^{i,j,w})$$

$$\log q(Z_W^{i,j,w} = V) \propto \log P(Z_W = V) + \mathbb{E}_{q(\Lambda)} \log \left( \Lambda(Z_W^{i,j,w-1}, V) \Lambda(V, Z_W^{i,j,w+1}) \right) + \sum_v q(Z_V^{i,j} = v) \mathbb{E}_{q(\theta_V^v)} \log \theta_V^v(s^{i,j,w})$$

$$\log q(Z_W^{i,j,w} = B) \propto \log P(Z_W = B) + \mathbb{E}_{q(\Lambda)} \log \left( \Lambda(Z_W^{i,j,w-1}, B) \Lambda(B, Z_W^{i,j,w+1}) \right) + \mathbb{E}_{q(\theta_B)} \log \theta_B(s^{i,j,w})$$

Figure 3-7: The mean-field variational algorithm used during learning and inference to obtain posterior predictions over snippet properties and attributes, as described in Section 3.5. Mean-field inference consists of updating each of the latent variable factors as well as a straightforward update of latent parameters in round robin fashion.

fixed to their current values:

$$q(\cdot) \leftarrow \mathbb{E}_{Q/q(\cdot)} \log P(\theta_B, \boldsymbol{\theta}_V, \Lambda, \boldsymbol{\theta}_A, \boldsymbol{\psi}, \boldsymbol{\phi}, \mathbf{Z}, \mathbf{s})$$

A summary of the variational update equations is given in Figure 3-7, and a graphical representation of the involved variables for each step is presented in Figure 3-8. Here, we will present the update for each factor.

**Snippet Aspect Indicator** First, we consider the update for the snippet aspect indicator,  $Z_A^{i,j}$  (Figure 3-8a):

$$\log q(Z_A^{i,j} = a) \propto \mathbb{E}_{q(\psi^i)} \log \psi^i(a) \tag{3.1a}$$

$$+ \sum_w q(Z_W^{i,j,w} = A) \mathbb{E}_{q(\theta_A^{i,a})} \log \theta_A^{i,a}(s^{i,j,w}) \tag{3.1b}$$

$$+ \sum_{v=1}^N q(Z_V^{i,j} = v) \mathbb{E}_{q(\phi^{i,a})} \log \phi^{i,a}(v) \tag{3.1c}$$

The optimal aspect for a particular snippet depends on three factors. First, we include the likelihood of discussing each aspect  $a$  (Eqn. 3.1a). As mentioned earlier, this encodes the prior probability that some aspects are discussed more frequently than others. Second, we examine the likelihood of a particular aspect based on the words in the snippet (Eqn. 3.1b). For each word which is identified as an aspect word, we add the probability that it discusses this aspect. Third, we determine the compatibility of the chosen aspect type with the current aspect (Eqn. 3.1c). For example, if we know the value type is most likely an integer, the assigned aspect should accept integers.

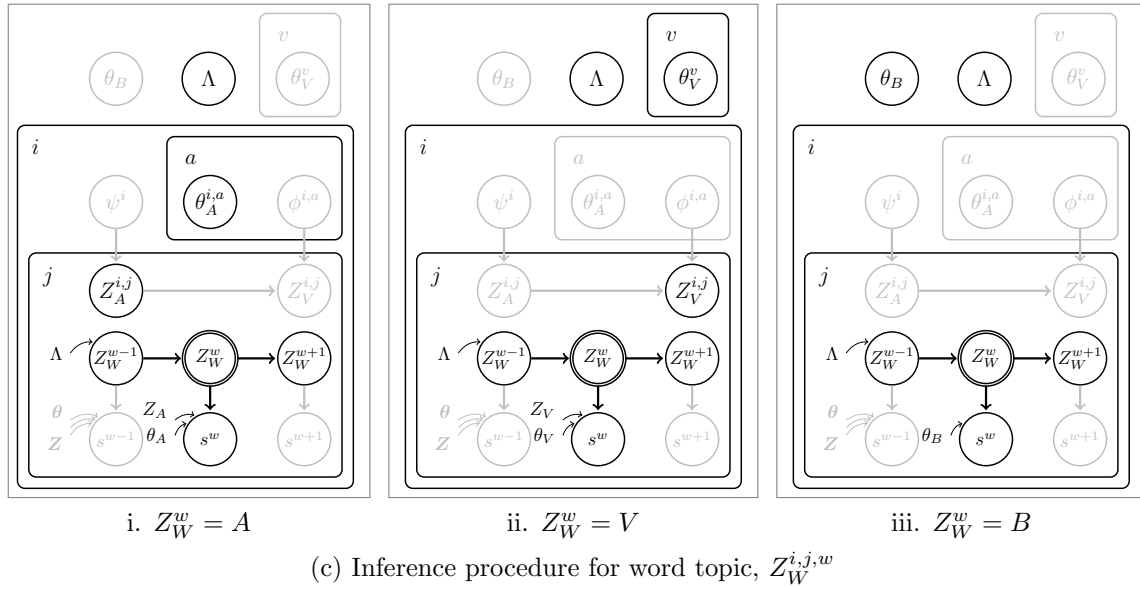
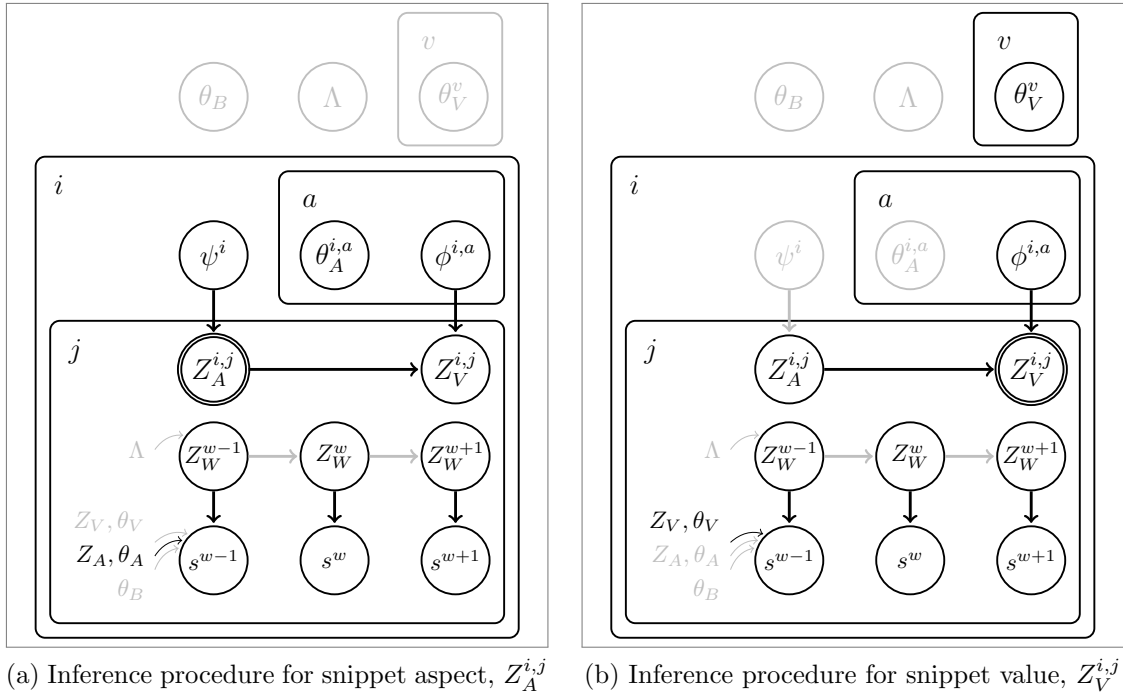


Figure 3-8: Variational inference update steps for each latent variable. The latent variable currently being updated is shown in a double circle, and the other variables relevant to the update are highlighted in black. Those variables which have no impact on the update are grayed out. Note that for snippet aspect (a) and snippet value type (b), the update takes the same form for each possible aspect or value type. However, for word topic (c), the update is not symmetric as the relevant variables are different for each possible word topic.

**Snippet Value Type Indicator** Next, we consider the update for the snippet value type indicator,  $Z_V^{i,j}$  (Figure 3-8b):

$$\log q(Z_V^{i,j} = v) \propto \sum_a q(Z_A^{i,j} = a) \mathbb{E}_{q(\phi^{i,a})} \log \phi^{i,a}(v) \quad (3.2a)$$

$$+ \sum_w q(Z_W^{i,j,w} = V) \mathbb{E}_{q(\theta_V^v)} \log \theta_V^v(s^{i,j,w}) \quad (3.2b)$$

The best value type for a snippet depends on two factors. First, like the snippet aspect indicator, we must take into consideration the compatibility between snippet aspect and value type (Eqn. 3.2a). Second, for each word identified as a value word, we include the likelihood that it comes from the given value type.

**Word Topic Indicator** Finally, we consider the update for the word topic indicators,  $Z_W^{i,j,w}$  (Figure 3-8c). Unlike the previous indicators, each possible topic has a slightly different equation, as we must marginalize over all possible aspects and value types.

$$\begin{aligned} \log q(Z_W^{i,j,w} = A) &\propto \log P(Z_W = A) + \mathbb{E}_{q(\Lambda)} \log \left( \Lambda(Z_W^{i,j,w-1}, A) \Lambda(A, Z_W^{i,j,w+1}) \right) \\ &+ \sum_a q(Z_A^{i,j} = a) \mathbb{E}_{q(\theta_A^{i,a})} \log \theta_A^{i,j}(s^{i,j,w}) \end{aligned} \quad (3.3a)$$

$$\begin{aligned} \log q(Z_W^{i,j,w} = V) &\propto \log P(Z_W = V) + \mathbb{E}_{q(\Lambda)} \log \left( \Lambda(Z_W^{i,j,w-1}, V) \Lambda(V, Z_W^{i,j,w+1}) \right) \\ &+ \sum_v q(Z_V^{i,j} = v) \mathbb{E}_{q(\theta_V^v)} \log \theta_V^v(s^{i,j,w}) \end{aligned} \quad (3.3b)$$

$$\begin{aligned} \log q(Z_W^{i,j,w} = B) &\propto \log P(Z_W = B) + \mathbb{E}_{q(\Lambda)} \log \left( \Lambda(Z_W^{i,j,w-1}, B) \Lambda(B, Z_W^{i,j,w+1}) \right) \\ &+ \mathbb{E}_{q(\theta_B)} \log \theta_B(s^{i,j,w}) \end{aligned} \quad (3.3c)$$

The update for each topic is composed of the prior probability of having that topic, transition probabilities using this topic, and the probability of the word coming from the appropriate unigram distribution, marginalized over all possibilities for snippet aspect and value indicators.

**Parameter Factors** Updates for the parameter factors under variational inference are derived through simple counts of the latent and observed variables  $Z_A$ ,  $Z_V$ ,  $Z_W$ , and  $\mathbf{s}$ :

$$\begin{aligned}\psi^i(Z_A = a) &\propto \sum_j P(Z_A^{i,j} = a) \\ \phi^{i,a}(Z_V = v) &\propto \sum_j P(Z_V^{i,j,a} = v) \\ \theta_Y(s = w) &\propto \sum_i \sum_j \sum_w P(s^{i,j,w} = w | Z_W^{i,j,w} = Y) \\ \tau(Z_W^w = X \wedge Z_W^{w+1} = Y) &\propto \sum_i \sum_j \sum_w P(Z_W^{i,j,w} = X \wedge Z_W^{i,j,w+1} = Y)\end{aligned}$$

Note that this formulation yields partial counts; if a particular snippet has aspect probability  $P(Z_A^{i,j} = a) = 0.35$ , it would contribute 0.35 count to  $\psi^i(a)$ .

**Algorithm Details** Given this set of update equations, the update procedure is straightforward. First, iterate over the corpus computing the updated values for each random variable, then do a batch update for all factors simultaneously. This update algorithm is run to convergence. In practice, convergence is achieved by the 50th iteration, so the algorithm is quite efficient.

Note that the batch update means each update is computed using the values from the previous iteration, unlike Gibbs sampling which uses updated values as it runs through the corpus. This difference allows the variational update algorithm to be parallelized, yielding a nice efficiency boost. Specifically, to parallelize the algorithm, we simply split the set of entities evenly among processors. Updates for entity-specific factors and variables are computed during the pass through the data, and updates for global factors are collected and combined at the end of each pass.

### 3.5.1 Inference for model extensions

As discussed in Section 3.4.2, we can add additional components to the model to improve performance for data with certain attributes. Here, we briefly discuss the



modifications to the inference equations for each extension.

### Ignoring snippets

The main modifications to the model for this extension are the addition of the unigram distribution  $\theta_I$  and word topic  $I$ , which can be chosen by  $Z_W$ . The update equation for  $Z_W$  is modified by the addition of the following:

$$\log q(Z_W^{i,j,w} = I) \propto \log P(Z_W = I) + \mathbb{E}_{q(\theta_I)} \log \theta_I(s^{i,j,w})$$

As in the other pieces of this equation (Eqn. 3.3), this is composed of the prior probability for the word topic  $I$  and the likelihood that this word is generated by  $\theta_I$ .

In addition, the transition distribution  $\Lambda$  must be updated to include transition probabilities for  $I*$  and  $*I$ . As mentioned earlier, the  $II$  transition receives high weight, while all other transitions to and from  $I$  receive very low weight.

### Part-of-speech tags

To add part of speech tags, the model is updated to include part-of-speech distributions  $\eta_A$ ,  $\eta_V$ , and  $\eta_B$ , one for each word topic. Note that unlike the unigram distributions  $\theta_A^{i,a}$  and  $\theta_V^v$ , the corresponding tag distributions are not dependent on snippet entity, aspect, or value. These are included and referenced in the updates for

$Z_W$  as follows:

$$\begin{aligned} \log q(Z_W^{i,j,w} = A) &\propto \log P(Z_W = A) + \mathbb{E}_{q(\Lambda)} \log \left( \Lambda(Z_W^{i,j,w-1}, A) \Lambda(A, Z_W^{i,j,w+1}) \right) \\ &\quad + \mathbb{E}_{q(\eta_A)} \log \eta_A(t^{i,j,w}) + \sum_a q(Z_A^{i,j} = a) \mathbb{E}_{q(\theta_A^a)} \log \theta_A^{i,j}(s^{i,j,w}) \end{aligned}$$

$$\begin{aligned} \log q(Z_W^{i,j,w} = V) &\propto \log P(Z_W = V) + \mathbb{E}_{q(\Lambda)} \log \left( \Lambda(Z_W^{i,j,w-1}, V) \Lambda(V, Z_W^{i,j,w+1}) \right) \\ &\quad + \mathbb{E}_{q(\eta_V)} \log \eta_V(t^{i,j,w}) + \sum_v q(Z_V^{i,j} = v) \mathbb{E}_{q(\theta_V^v)} \log \theta_V^v(s^{i,j,w}) \end{aligned}$$

$$\begin{aligned} \log q(Z_W^{i,j,w} = B) &\propto \log P(Z_W = B) + \mathbb{E}_{q(\Lambda)} \log \left( \Lambda(Z_W^{i,j,w-1}, B) \Lambda(B, Z_W^{i,j,w+1}) \right) \\ &\quad + \mathbb{E}_{q(\eta_B)} \log \eta_B(t^{i,j,w}) + \mathbb{E}_{q(\theta_B)} \log \theta_B(s^{i,j,w}) \end{aligned}$$

Here, we define  $\mathbf{t}$  as the set of all tags and  $t^{i,j,w}$  as the tag corresponding to the word  $s^{i,j,w}$ .

### Shared aspects

A global set of shared aspects is a simplification of the model in that it reduces the total number of parameters. This model redefines aspect distributions to be  $\theta_A^a$  and aspect-value multinomials to be  $\phi^a$ . Depending on domain, it may also redefine the aspect multinomial to be  $\psi$ . The resulting latent variable update equations are the same; only the parameter factor updates are changed. Rather than collecting counts over snippets describing a single entity, counts are collected across the corpus.

## 3.6 Experiments

We perform experiments on two tasks. First, we test our full model on joint prediction of both aspect and sentiment on a corpus of review data. Second, we use a simplified version of the model designed to identify aspects only on a corpus of medical summary data. These domains are structured quite differently, and therefore present very

different challenges to our model.

### 3.6.1 Joint identification of aspect and sentiment

Our first task is to test our full model by jointly predicting both aspect and sentiment on a collection of restaurant review data. Specifically, we would like to dynamically select a set of relevant aspects for each restaurant, identify the snippets which correspond to each aspect, and recover the polarity of each snippet individually and each aspect as a whole. We perform three experiments to evaluate our model’s effectiveness. First, we test the quality of learned aspects by evaluating the predicted snippet clusters. Second, we assess the quality of the polarity classification. Third, we examine per-word labeling accuracy.

#### 3.6.1.1 Data set

Our data set for this task consists of snippets selected from Yelp restaurant reviews by our previous system [70]. The system is trained to extract snippets containing short descriptions of user sentiment towards some aspect of a restaurant.<sup>4</sup> For the purpose of this experiment, we select only the snippets labeled by that system as referencing *food*. In order to ensure that there is enough data for meaningful analysis, we ignore restaurants that have fewer than 20 snippets across all reviews. There are 13,879 snippets in total, taken from 328 restaurants in and around the Boston/Cambridge area. The average snippet length is 7.8 words, and there are an average of 42.1 snippets per restaurant. We use the MXPOST tagger [69] to gather POS tags for the data. Figure 3-9 shows some example snippets.

For this domain, the value distributions are divided into positive and negative distributions. These are seeded using 42 and 33 seed words respectively. Seed words are hand-selected based on the restaurant review domain; therefore, they include domain-specific words such as *delicious* and *gross*. A complete list of seed words is included in Table 3.2.

---

<sup>4</sup>For exact training procedures, please reference that paper.

Positive				Negative		
amazing	awesome	best	delicious	average	awful	bad
delightful	divine	enjoy	excellent	bland	boring	confused
extraordinary	fantastic	fav	favorite	disappointed	disgusting	dry
flavorful	free	fresh	fun	expensive	fatty	greasy
generous	good	great	happy	gross	horrible	inedible
heaven	huge	incredible	interesting	lame	less	mediocre
inexpensive	love	nice	outstanding	meh	mushy	overcooked
perfect	phenomenal	pleasant	quality	poor	pricey	salty
recommend	rich	sleek	stellar	tacky	tasteless	terrible
stimulating	strong	tasty	tender	tiny	unappetizing	underwhelming
wonderful	yummy			uninspiring	worse	worst

Table 3.2: Seed words used by the model for the restaurant corpus, 42 positive words and 33 negative words in total. These words are manually selected for this data set.

### 3.6.1.2 Domain challenges and modeling techniques

This domain presents two challenging characteristics for our model. First, there are a wide variety of restaurants within our domain, including everything from high-end Asian fusion cuisine to greasy burger fast food places. If we were to try to represent these using a single shared set of aspects, the number of aspects required would be immense, and it would be extremely difficult for our model to make fine-grained distinctions between them. By defining aspects separately for each restaurant as mentioned in Section 3.4, we can achieve the proper granularity of aspects for each individual restaurant without an overwhelming or overlapping selection of choices. For example, the model is able to distinguish that an Italian restaurant may need only a single *dessert* aspect, while a bakery requires separate *pie*, *cake*, and *cookie* aspects.

Second, while there are usually a fairly cohesive set of words which refer to any particular aspect (e.g., the *pizza* aspect might be commonly be seen with the words *slice*, *pepperoni*, and *cheese*), there are a near-unlimited set of potential sentiment words. This is especially pronounced in the social media domain where there are many novel words used to express sentiment (e.g., *deeeeeeeelish* as a substitute for *delicious*). As mentioned in Section 3.4, the part-of-speech and transition components of the model helps to identify which unknown words are likely to be sentiment words; however, we additionally need to identify the polarity of their sentiment. To do this, we can leverage the aspect-value multinomial, which represents the likelihood of

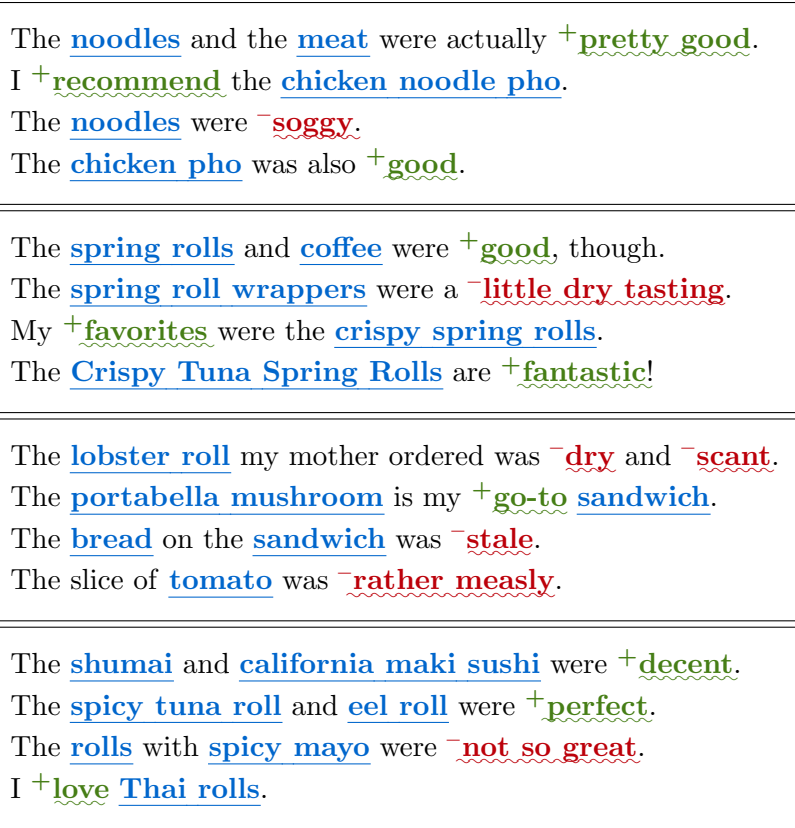


Figure 3-9: Example snippets from our data set, grouped according to aspect. Aspect words are underlined and colored blue, NEGATIVE value words are labeled - and colored red, and POSITIVE value words are labeled + and colored green. The grouping and labeling are *not* given in the data set and must be learned by the model.

positive or negative sentiment for a particular aspect. If most of the snippets about a given aspect are positive, it is likely that the word *deeeeeeeelish* represents positive sentiment as well.

### 3.6.1.3 Cluster prediction

The goal of this task is to evaluate the quality of aspect clusters; specifically the  $Z_P^{i,j}$  variable in Section 3.4. In an ideal clustering, the predicted clusters will be cohesive (i.e., all snippets predicted to discuss a given aspect are related to each other) and comprehensive (i.e., all snippets which discuss an aspect are selected as such). For example, a snippet will be assigned the aspect *pizza* if and only if that snippet mentions some aspect of pizza, such as its crust, cheese, or toppings.

**Annotation** For this experiment, we use a set of gold clusters on the complete sets of snippets from 20 restaurants, 1026 snippets in total (an average of 51.3 snippets per restaurant). Cluster annotations were provided by graduate students fluent in English. Each annotator was provided with a complete set of snippets for a particular restaurant, then asked to cluster them naturally. There were 199 clusters in total, which yields an average of 10.0 clusters per restaurant. These annotations are high-quality; the average annotator agreement is 81.9 by the MUC evaluation metric (described in detail below). Baseline systems and our full model are asked to produce 10 aspect clusters per restaurant, matching the average annotated number.

**Baseline** We use two baselines for this task, both using a clustering algorithm weighted by TF\*IDF as implemented by the publicly available CLUTO package [40],<sup>5</sup> using agglomerative clustering with the cosine similarity distance metric [14, 16].

The first baseline, CLUSTER-ALL, clusters over entire snippets in the data set. This baseline will put a strong connection between things which are lexically similar. Because our model only uses aspect words to tie together clusters, this baseline may capture correlations between words which our model does not correctly identify as aspect words.

The second baseline, CLUSTER-NOUN, works over only the nouns from the snippets. Each snippet is POS-tagged using MXPOST [69],<sup>6</sup> and any non-noun (i.e., not NN, NNS, NNP, or NNPS) words are removed. Because we expect that most aspects contain at least one noun, this acts as a proxy for the aspect identification in our model.

**Metric** We use the MUC cluster evaluation metric for this task [83]. This metric measures the number of cluster merges and splits required to recreate the gold clusters given the model’s output. Therefore, it can concisely show how accurate our clusters are as a whole. While it would be possible to artificially inflate the score by putting everything into a single cluster, the parameters on our model and the likelihood

---

<sup>5</sup>Available at <http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview>.

<sup>6</sup>Available at [http://www.inf.ed.ac.uk/resources/nlp/local\\_doc/MXPOST.html](http://www.inf.ed.ac.uk/resources/nlp/local_doc/MXPOST.html).

	Precision	Recall	F1
CLUSTER-ALL	57.3	60.1	58.7
CLUSTER-NOUN	68.6	70.5	69.5
Our model	<b>74.3</b>	<b>85.3</b>	<b>79.4</b>

Table 3.3: Results using the MUC metric on cluster prediction for the joint aspect and value identification task. While MUC has a deficiency in that putting everything into a single cluster will artificially inflate the score, all models are set to use the same number of clusters. Note that for this task, the CLUSTER-NOUN significantly outperforms the CLUSTER-ALL baseline, indicating that part of speech is a crucial piece of information for this task.

objective are such that the model prefers to use all available clusters, the same number as the baseline system.

**Results** Results for our cluster prediction task are in Table 3.3. Our model shows strong performance over each baseline, for a total error reduction of 32% over the CLUSTER-NOUN baseline and 50% over the CLUSTER-ALL baseline. The most common cause of poor cluster choices in the baseline systems is their inability to distinguish which words are relevant aspect words. For example, in the CLUSTER-ALL baseline, if many snippets use the word *delicious*, there may end up being a cluster based on that alone. The CLUSTER-NOUN baseline is able to avoid some of these pitfalls thanks to its built-in filter. It is able to avoid common value words such as adjectives and also focus on what seems to be the most concrete portion of the aspect (e.g., *blackened **chicken***); however, it still cannot make the correct distinctions where these assumptions are broken. Because our model is capable of distinguishing which words are aspect words (i.e., words relevant to clustering), it can choose clusters which make more sense overall.

#### 3.6.1.4 Sentiment analysis

We evaluate the system’s predictions of snippet sentiment using the predicted posterior over the value distributions for the snippet (i.e.,  $Z_A^{i,j}$ ). For this task, we consider the binary judgment to be simply the one with higher value in  $q(Z_A^{i,j})$  (see Section 3.5). The goal of this task is to evaluate whether our model correctly distinguishes the sen-

timent of value words.

**Annotation** For this task, we use a set of 662 randomly selected snippets from the Yelp reviews which express opinions. To get a clear result, this set specifically excludes neutral, mixed, or potentially ambiguous snippets such as *the fries were too salty but tasty* or *the blackened chicken was very spicy*, which make up about 10% of the overall data. This set is split into a training set of 550 snippets and a test set of 112 snippets, then each snippet is manually labeled POSITIVE or NEGATIVE. For one baseline, we use the set of positive and negative seed words which were manually chosen for our model, shown in Table 3.2. Note that as before, our model has access to the full corpus of unlabeled data plus the seed words, but no labeled examples.

**Baseline** We use two baselines for this task, one based on a standard discriminative classifier and one based on the seed words from our model.

The DISCRIMINATIVE baseline for this task is a standard maximum entropy discriminative binary classifier<sup>7</sup> over unigrams. Given enough snippets from enough unrelated aspects, the classifier should be able to identify that words like *great* indicate positive sentiment and those like *bad* indicate negative sentiment, while words like *chicken* are neutral and have no effect. To illustrate the effect of training size, we include results for DISCRIMINATIVE-SMALL, which uses 100 training examples, and DISCRIMINATIVE-LARGE, which uses 550 training examples.

The SEED baseline simply counts the number of words from the same positive and negative seed lists used by the model,  $V_{seed+}$  and  $V_{seed-}$ , as listed in Table 3.2. If there are more words from  $V_{seed+}$ , the snippet is labeled positive, and if there are more words from  $V_{seed-}$ , the snippet is labeled negative. If there is a tie or there are no seed words, we split the prediction. Because the seed word lists are manually selected specifically for restaurant reviews (i.e., they contain food-related sentiment words such as *delicious*), this baseline should perform well.

---

<sup>7</sup>Available at <https://github.com/lzhang10/maxent>.



	Accuracy
MAJORITY	60.7
DISCRIMINATIVE-SMALL	74.1
SEED	78.2
DISCRIMINATIVE-LARGE	80.4
Our model	<b>82.5</b>

Table 3.4: Sentiment prediction accuracy of our model compared to the DISCRIMINATIVE and SEED baselines, as well as MAJORITY representing the majority class (POSITIVE) baseline. One advantage of our system is its ability to distinguish aspect words from sentiment words in order to restrict judgment to only the relevant terms; another is the leverage that it gains from biasing unknown sentiment words to follow the polarity observed in other snippets relating to the same aspect.

**Results** The overall sentiment classification accuracy of each system are shown in Table 3.4). Our model outperforms both baselines. The obvious flaw in the SEED baseline is the inability to pre-specify every possible sentiment word. It does perform highly, due to its tailoring for the restaurant domain and good coverage of the most frequent words (e.g., *delicious*, *good*, *great*), but the performance of our model indicates that it can generalize beyond these seed words.

The DISCRIMINATIVE-LARGE outperforms the SEED baseline on this test set; however, given the smaller training set of DISCRIMINATIVE-SMALL, it performs worse. The training curve of the DISCRIMINATIVE baseline is shown in Figure 3-10. While the DISCRIMINATIVE baseline system can correctly identify the polarity of statements containing information it has seen in the past, it has two main weaknesses. First, every sentiment word must have been present in training data. For example, in our test data, *rancid* appears in a negative sentence; however, it does not appear in the training data, so the model labels the example incorrectly. This is problematic, as there is no way to find training data for every possible sentiment word, especially in social media data where novel words and typos are a frequent occurrence. Our model’s ability to generalize about the polarity of snippets describing a particular aspect allows it to predict sentiment values for words of unknown polarity. For example, if there are already several positive snippets describing a particular aspect, the system can guess that a snippet with unknown polarity will likely also be positive.

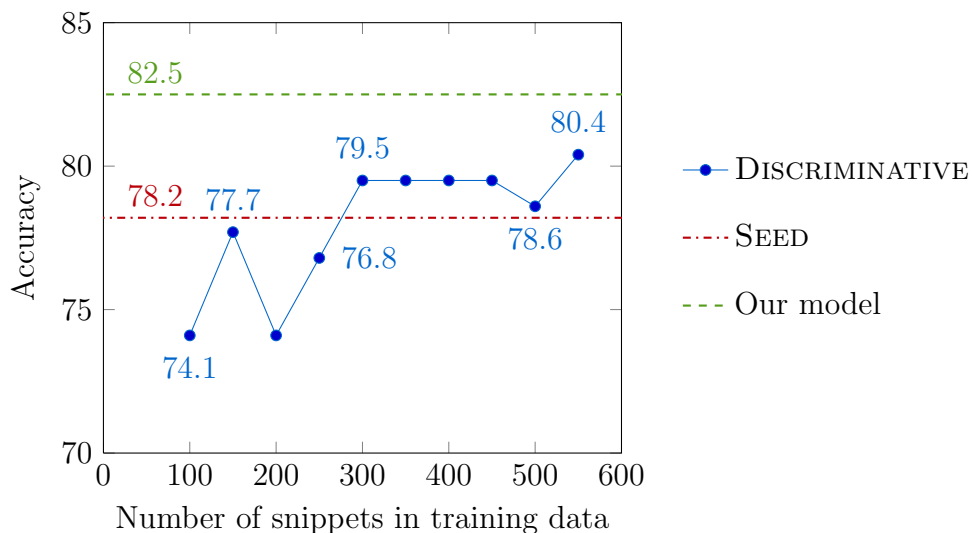


Figure 3-10: DISCRIMINATIVE baseline performance as the number of training examples increases. While performance generally increases, there are some inconsistencies. The main issue with this baseline is that it needs to see examples of words in training data before it can improve; this phenomenon can be seen at the plateau in this graph.

### 3.6.1.5 Per-word labeling accuracy

The goal of this task is to evaluate whether each word is correctly identified as an aspect word, value word, or background word. This distinction is crucial in order to achieve correctness of both clustering and sentiment analysis, so errors here may help us identify weaknesses of our model.

**Annotation** Per-word annotation is acquired from Mechanical Turk. The per-word labeling task seems difficult for some Turk annotators, so we implement a filtering procedure to ensure that only high-quality annotators are allowed to submit results. Specifically, we ask annotators to produce labels for a set of “difficult” phrases with known labels (shown in Table 3.5). Those annotators who successfully produced correct or mostly-correct annotations are allowed to access the annotation tasks containing new phrases. Each of these unknown tasks is presented to 3 annotators, and the majority label is taken for each word. In total, we test on 150 labeled phrases, for a total of 7,401 labeled words.

<p>The <u>rolls</u> also were <u>n't very well made</u> .</p> <p>The <u>pita</u> was <u>beyond dry</u> and <u>tasted like cardboard</u> !</p> <p>The Falafel King has the <u>best falafel</u> !</p> <p>The <u>rolls with spicy mayo</u> were <u>not so good</u> .</p> <p>Ordered the <u>spicy tuna</u> and <u>california roll</u> – they were <u>amazing</u> !</p>
--

Table 3.5: Correct annotation of a set of phrases containing elements which may be confusing, on which annotators are tested before they are allowed to annotate the actual test data. Aspect words are colored blue and underlined; value words are colored orange and underlined with a wavy line. Some common mistakes include: annotating *n't* as background (because it is attached to the background word *was*), annotating *cardboard* as an aspect because it is a noun, annotating *Falafel King* as an aspect because it is in subject position.

**Baseline** The baseline for this task relies again on the intuition that part-of-speech is a useful proxy for aspect and value identification. We know that aspects usually represent concrete entities, so they are often nouns, and value words are descriptive or counting, so they are often adjectives or adverbs. Therefore, we again use the MXPOST tagger to find POS for each word in the snippet. For the main baseline, TAGS-FULL, we assign each noun (NN\*) an aspect label, and each numeral, adjective, adverb, or verb participle (CD, RB\*, JJ\*, VBG, VBN) a value label. For comparison, we also present results for a smaller tagset, SMALL-TAGS, labeling only nouns (NN\*) as aspect and adjectives (JJ\*) as values. Note that each of the tags added in the TAGS-FULL baseline are beneficial to the baseline’s score.

**Tree expansion** Because our full model and the baselines are all designed to pick out relevant individual words rather than phrases, they may not correspond well to the phrases which humans have selected as relevant. Therefore, we also evaluate on a set of expanded labels identified with parse trees from the Stanford Parser [44].<sup>8</sup> Specifically, for each non-background word, we identify the largest containing noun phrase (for both aspects and values) or adjective or adverb phrase (for aspects only) which does not also contain oppositely-labeled words. For example, in the noun phrase *blackened chicken*, if *chicken* was labeled as an aspect word and *blackened* was

<sup>8</sup>Available at <http://nlp.stanford.edu/software/lex-parser.shtml>.

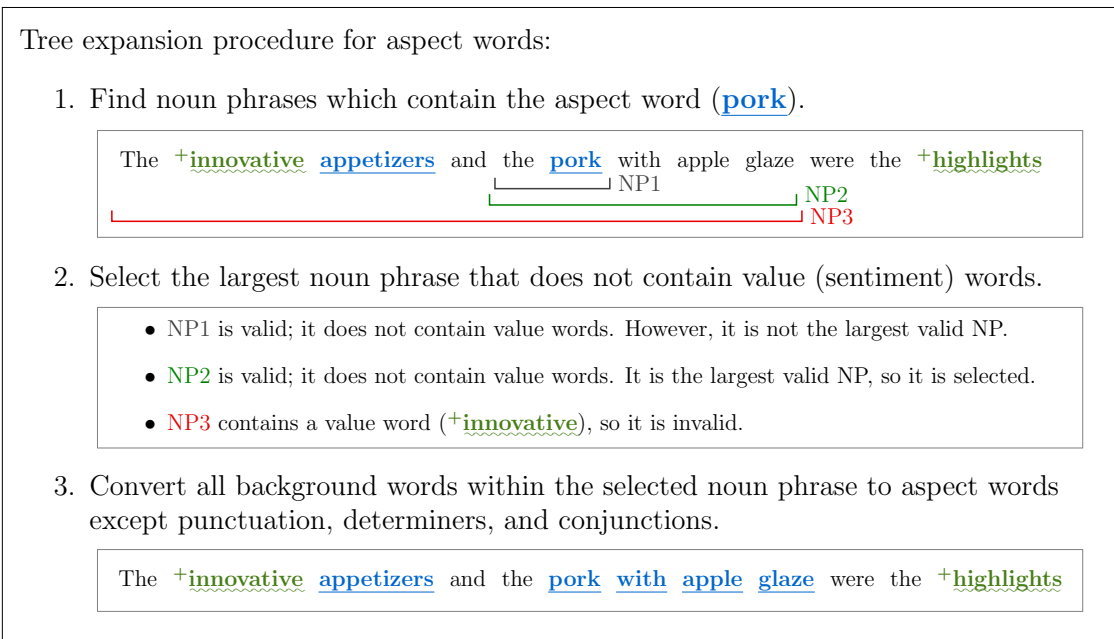


Figure 3-11: The tree expansion procedure for value words, with an example snippet. The procedure is similar for aspect words, except adjective phrases and adverb phrases are also considered for expansion.

labeled as a background word, both will now be labeled as aspect words. However, in the noun phrase *tasty chicken* where “tasty” is already labeled as a value, the label will not be changed and no further expansion will be attempted. As a final heuristic step, any punctuation, determiners, and conjunctions which would be newly labeled as aspect or value words are ignored and kept as background words. The steps of this procedure with an illustrative example are shown in Figure 3-11.

**Results** We evaluate all systems on precision and recall for aspect and value separately. Results for all systems are shown in Table 3.6. Our model without the tree expansion is highly precise at the expense of recall; however when the expansion is performed, its recall improves tremendously, especially on value words.

While this result is initially disappointing, it is possible to adjust model parameters to increase performance at this task; for example, for aspect words we could put additional mass on the prior for  $Z_W^{i,j,w} = A$  or increase the Dirichlet hyperparameter  $\lambda_A$ . However, while this increases performance on the word labeling task, it also decreases performance correspondingly on the clustering task. By examination of the

	Aspect			Value		
	Precision	Recall	F <sub>1</sub>	Precision	Recall	F <sub>1</sub>
TAGS-SMALL	79.9	79.5	<b>79.7</b>	78.5	45.0	57.2
Tree	74.0	<b>83.0</b>	78.2	<b>79.2</b>	57.4	66.5
TAGS-FULL	79.9	79.5	<b>79.7</b>	78.1	68.7	73.1
Tree	75.6	81.4	78.4	77.1	70.1	73.4
Our model	<b>85.2</b>	52.6	65.0	70.5	61.6	65.7
Tree	79.5	71.9	75.5	76.7	<b>70.9</b>	<b>73.7</b>

Table 3.6: Per-word labeling precision and recall of our model compared to the TAGS-SMALL and TAGS-FULL baselines, both with and without expansion by trees. Our model is most precise on aspect and has better recall on value. Note that in general the process of expanding labels with the tree structure increases recall at the expense of precision.

The moqueca was delicious and perfect winter food , warm , filling and hearty but not too heavy .  
The bacon wrapped almond dates were amazing but the plantains with cheese were boring .  
the artichoke and homemade pasta appetizers were great

Table 3.7: High-precision, low-recall aspect word labeling by our full model. Note that a human would likely identify complete phrases such as *bacon wrapped almond dates* and *homemade pasta appetizers*; however, the additional noise degrades performance on the clustering task.

data, this correlation is perfectly reasonable. In order to succeed at the clustering task, the model selects only the most relevant portions of the snippet as aspect words. When the entire aspect and value are identified, clustering becomes noisy. Table 3.7 shows some examples of the high-precision labeling which achieves high clustering performance.

### 3.6.2 Aspect identification with shared aspects

Our second task uses a simplified version of our model designed for aspect identification only. For this task, we use a corpus of medical visit summaries. In this domain, each summary is expected to contain similar relevant information; therefore, the set of aspects is shared corpus-wide. To evaluate our model in this formulation, we examine the predicted clusters of snippets, as in the full model.

### 3.6.2.1 Data set

Our data set for this task consists of phrases selected from dictated patient summaries at the Pediatric Environmental Health Clinic (PEHC) at Children’s Hospital Boston, specializing in treatment of children with lead poisoning. Specifically, after a patient’s office visit and lab results are completed, a PEHC doctor dictates a letter to the referring physician containing information about previous visits, current developmental and family status, in-office exam results, lab results, current diagnosis, and plan for the future.

For this experiment, we select phrases from the in-office exam and lab results sections of the summaries. Phrases are separated heuristically on commas and semicolons. There are 6198 snippets in total, taken from 271 summaries. The average snippet length is 4.5 words, and there are an average of 23 snippets per summary. As in the Yelp domain, we use the MXPOST tagger [69] to gain POS tags. Figure 3-12 shows some example snippets. For this domain, there are no values; we simply concentrate on the aspect-identification task. Unlike the restaurant domain, we use no seed words.

### 3.6.2.2 Domain challenges and modeling techniques

In contrast to the restaurant domain, the medical domain uses a single global set of aspects. These represent either individual lab tests (e.g., *lead level*, *white blood cell count*) or particular body systems (e.g., *lungs* or *cardiovascular*). Some aspects are far more common than others, and it is very uncommon for a summary to include more than one or two snippets about any given aspect. Therefore, as mentioned in Section 3.4.2, we model the aspect word distributions and the aspect multinomial as shared between all entities in the corpus.

Also in contrast to the restaurant domain, aspects are defined by words taken from the entire snippet. Rather than having aspects only associated with names of measurements (e.g., ‘weight’), units and other descriptions of measurement (e.g., ‘kilograms’) are also relevant for aspect definition. This property extends to both

He was 113 <u>cm</u> in <u>height</u> Patient's <u>height</u> was 146.5 <u>cm</u>
<u>Lungs</u> : <u>Clear bilaterally</u> to <u>auscultation</u> <u>lungs</u> were normal
<u>Heart</u> regular <u>rate</u> and <u>rhythm</u> ; no <u>murmurs</u> <u>Heart</u> normal <u>S1</u> <u>S2</u>

Figure 3-12: Example snippets from the medical data set, grouped according to aspect. Aspect words are underlined and colored blue. This grouping and labeling are *not* given in the data set and must be learned by the model.

numeric and written measurements; for example, the aspect ‘lungs’ is commonly described as ‘clear to auscultation bilaterally’. In order to achieve high performance, our model must leverage all of these clues to provide proper aspect identification when the name of the measurement is missing (e.g., “patient is 100 cm”). While part of speech will still be an important factor to model, we predict that there will be greater importance on additional parts of speech other than nouns.

Finally, our data set is noisy and contains some irrelevant snippets, such as section headings (e.g., “Physical examination and review of systems”) or extraneous information. As described in Section 3.4.2, we modify our model so that it can ignore partial or complete snippets.

### 3.6.2.3 Cluster prediction

As for joint aspect and sentiment prediction, the goal of this task is to evaluate the quality of aspect identification. Because the aspects are shared across all documents, clusters are generally much larger, and the set of annotated snippets represents only a fraction of each cluster.

**Annotation** For this experiment, we use a set of gold clusters gathered over 1,200 snippets, annotated by a doctor who is an expert in the domain from the Pediatric Environmental Health Clinic at Children’s Hospital Boston. Note that as mentioned before, clusters here are global to the domain (e.g., many patients have snippets representing *blood lead level*, and these are all grouped into one cluster). The doctor

	Precision	Recall	F1
CLUSTER-ALL	88.2	93.0	90.5
CLUSTER-NOUN	88.4	83.9	86.1
Our model	<b>89.1</b>	<b>93.4</b>	<b>91.2</b>

Table 3.8: Results using the MUC metric on cluster prediction for the aspect identification only task. Note that the CLUSTER-ALL baseline significantly outperforms CLUSTER-NOUN, the opposite of what we observe in the joint aspect and value prediction task. This is due to the dependence of aspect identification on more than just the name of a lab test, such as the units or other description of the test results, as mentioned in Section 3.6.2.2.

was asked to cluster 100 snippets at a time (spanning several patients), as clustering the entire set would have been infeasible for a human annotator. After all 12 sets of snippets were clustered, the resulting clusters were manually combined to match up similar clusters from each set. For example, the *blood lead level* cluster from the first set of 100 snippets was combined with the corresponding *blood lead level* clusters from each other set of snippets. Any cluster from this final set with fewer than 5 members was removed. In total, this yields a gold set of 30 clusters. There are 1,053 snippets total, for an average of 35.1 snippets per cluster. To match this, baseline systems and our full model are asked to produce 30 clusters across the full data set.

**Baselines & Metric** To keep these results consistent with those on the previous task, we use the same baselines and evaluation metric. Both baselines rely on a TF\*IDF-weighted clustering algorithm, specifically implemented with CLUTO package [40] using agglomerative clustering with the cosine similarity distance metric. As before, CLUSTER-ALL represents a baseline using unigrams of snippets from the entire data set, while CLUSTER-NOUN works over only the nouns from the snippets. We again use the MUC cluster evaluation metric for this task. For more details on both baselines and the evaluation metric, please see Section 3.6.1.3.

**Results** For this experiment, our system demonstrates an improvement of 7% over the CLUSTER-ALL baseline. Absolute performance is relatively high for all systems in the medical domain, indicating that the lexical clustering task is less misleading



than in the restaurant domain. It is interesting to note that unlike in the restaurant domain, the CLUSTER-ALL baseline outperforms the CLUSTER-NOUN baseline. As mentioned in Section 3.6.2.2, the medical data is notable for the relevance of the entire snippet for clustering (e.g., both ‘weight’ and ‘kilograms’ are useful to identify the *weight* aspect). Because of this property, using only nouns to cluster in the CLUSTER-NOUN baseline hurts performance significantly.

### 3.7 Conclusion

In this chapter, we have presented an approach for fine-grained content aggregation using probabilistic topic modeling techniques to discover the structure of individual text snippets. Our model is able to successfully identify clusters of snippets in a data set which discuss the same aspect of an entity as well as the associated values (e.g., sentiment). It requires no annotation, other than a small list of seed vocabulary to bias the positive and negative distributions in the proper direction.

Our results demonstrate that delving into the structure of the snippet can assist in identifying key words which are important and unique to the domain at hand. When there are values to be learned, the joint identification of aspect and value can help to improve the quality of the results. The word labeling analysis reveals that the model learns a different type of labeling for each task; specifically, a strict, high-precision labeling for the clustering task and a high-recall labeling for sentiment. This follows the intuition that it is important to identify specific main points for clustering, while in the sentiment analysis task, there may often be several descriptions or conflicting opinions presented which all need to be weighed together to determine the overall sentiment.

This model admits a fast, parallelized inference procedure. Specifically, the entire inference procedure takes roughly 15 minutes to run on the restaurant corpus and less than 5 minutes on the medical corpus. Additionally, the model is neatly extensible and adjustable to fit the particular characteristics of a given domain.

There are a few limitations of this model which can be improved with future

work: First, our model makes no attempt to explicitly model negation or other word interactions, increasing the difficulty of both aspect and sentiment analysis for our model. By performing error analysis, we find that negation is a common source of error for the sentiment analysis task. Likewise, the model can make errors when attempting to differentiate aspects such as *ice cream* and *cream cheese* which share the common aspect word *cream*.

Second, while defining aspects per-entity as in the restaurant domain has advantages in that it is possible to get a very fine-grained set of applicable aspects, it also fails to leverage some potential information in the data set. Specifically, we know that restaurants sharing the same type (e.g., Italian, Indian, Bakery, etc.) should share some common aspects; however, there are no ties between them in the current model. Likewise, even at a global level, there may be some aspects which tie in across all restaurants. A hierarchical version of this model would potentially be able to tie these together and identify different types of aspects: global (e.g., *presentation*), type-level (e.g., *pasta* for the Italian type), and restaurant-level (e.g., the restaurant's special dish).

## CHAPTER 4

---

### Conclusion

---

In this thesis, I have explored the benefits of content structure across a wide variety of text analysis tasks on social media text. This data poses several challenges to traditional techniques for NLP, including a lack of formal structure, a propensity for novel words, and a dependence on outside context. However, I have demonstrated two methods for overcoming these hurdles: in Chapter 2, a technique for leveraging an automatically-induced representation of content structure applicable to a wide variety of text analysis tasks, and in Chapter 3, a method for joint aggregation across multiple documents that relies on understanding the underlying structure of text relations.

The first of these techniques addresses an important open question from previous work—how does one select an appropriate content model for a particular task? There are many ways to define the content structure of a document (e.g., sentiment-bearing sentences, discourse relations, underlying topics), and it is important not just to have any content model, but to have a content model which is *relevant* to any particular task. With the proposed joint model, it is possible to learn the appropriate content structure automatically. The task-specific model provides information to the content model to tailor it specifically for task at hand, and the content model in turn provides

more relevant information for the task-specific model.

The second approach explores the benefit of modeling structure in the form of relations. While previous approaches focus on pipeline models or predefined aspects, we instead focus on the relationship between aspect and value. We model the interaction of aspect and value across all documents in the corpus, which allows us to incorporate intuitions from multi-document summarization, such as the salience of repeated information. Additionally, we can leverage the varying distributional properties of aspect and value across the corpus.

Through multiple tasks for each technique, we have demonstrated the importance of content structure for increasing system performance. While the details of the approaches differ, in each case, by creating a structured representation of content in the input data, we are able to compensate for some of the challenges that social media text presents.

## 4.1 Future Work

This thesis introduces several opportunities for further research, including the following:

- **Effect on formal text** While the work in this paper has been presented primarily on social media text, these methods should be applicable to formal text as well. The task of choosing a relevant content model is just as pertinent to formal text [84], and the models presented here could easily be applied to a more formal domain, such as newspaper text. The tendency of more rigid structure in these domains should lead to an increased reliance on the content modeling component.
- **Modeling of more complex structure** Our approach to learn content models automatically is still limited to a particular class of content models by the HMM structure. We have demonstrated that improving the quality of the content model (e.g., by including additional unlabeled data) does improve the task

performance; therefore, a natural extension would be to incorporate more complex model structures. For example, a hierarchical structure might allow the content model to automatically learn subtopics of a particular topic. These more complex structures should fit into our current model architecture without issues.

The same point holds with our approach to aggregation; while there is flexibility in the model to define aspects as entity-specific or being shared corpus-wide (e.g., per-restaurant vs. shared between patients), the model can potentially benefit from additional layers of structure. For example, in the restaurant domain, learning a grouping over restaurants could help to identify those with common aspects; e.g., Italian restaurants will share aspects like *pasta* and *pizza* while bakeries will share *cake* and *bread*.

- **Temporal aggregation** Beyond inducing a structure within documents, we can also think about structure across the set of documents. One of the interesting features present with social media data is that it usually includes a time and date. Rather than treating the data as a single snapshot, we can treat the documents as a sequence and consider aggregation of trends over time. Some online retailers have started producing a graph of star rating over time; this could take that to the next level, just as our current system improves on the aggregation that may be gained from a histogram alone.
- **Social media metadata** A unique property of social media is its vast array of user data. Users are typically identified by a particular unique id (username) through all of their activities on a website, they may have biographical data or interests listed, and on some sites, they are assigned a ‘reputation’ which indicates a trust level. Even individual posts are given ratings on some sites; for example, Yelp reviews are rated *useful*, *funny*, or *cool*, while other sites such as Amazon have ratings of *helpful* and *not helpful*. Through these ratings and various user data, we can ask further interesting research questions; for example, on the restaurant domain: Do locals/tourists/business executives like

this restaurant? What's the overall opinion of "trusted" reviewers, and does that differ from the general population? Can we ignore false reviews in our aggregation; i.e., ignore reviews created by businesses to boost their reputation or tarnish their competitors' reputations? How can we determine standout reviews based on user personalities (e.g., a user generally posts negative reviews of burgers and then a highly positive review of one at a particular restaurant)? These questions transfer easily to other domains of social media.

# APPENDIX A

---

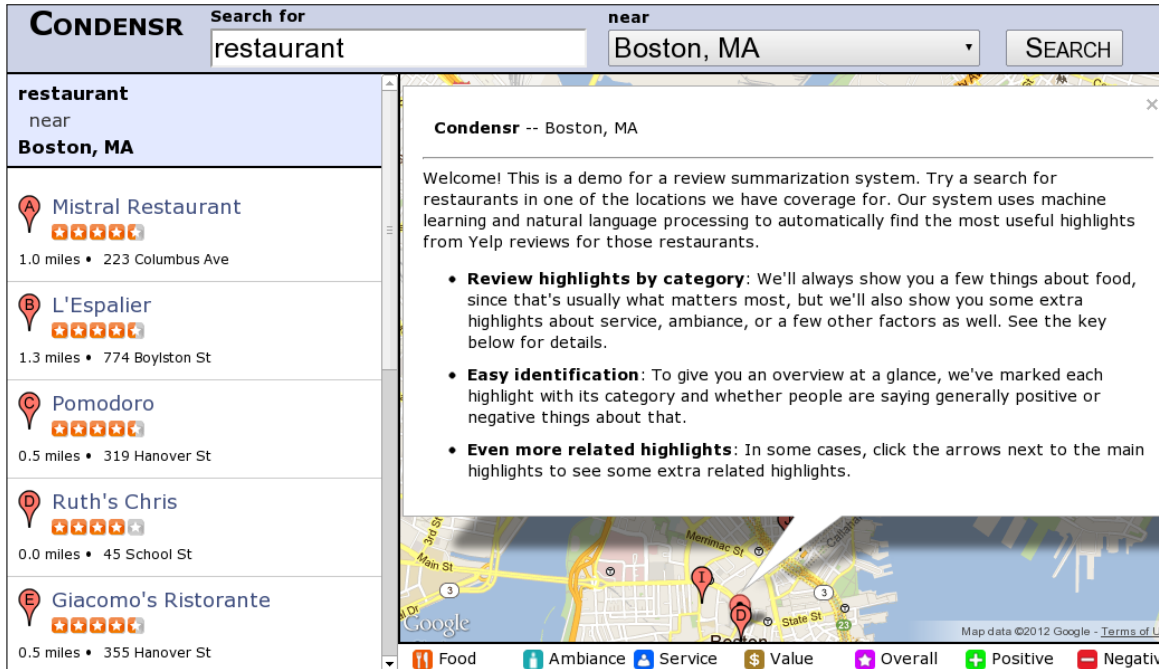
## Condensr

---

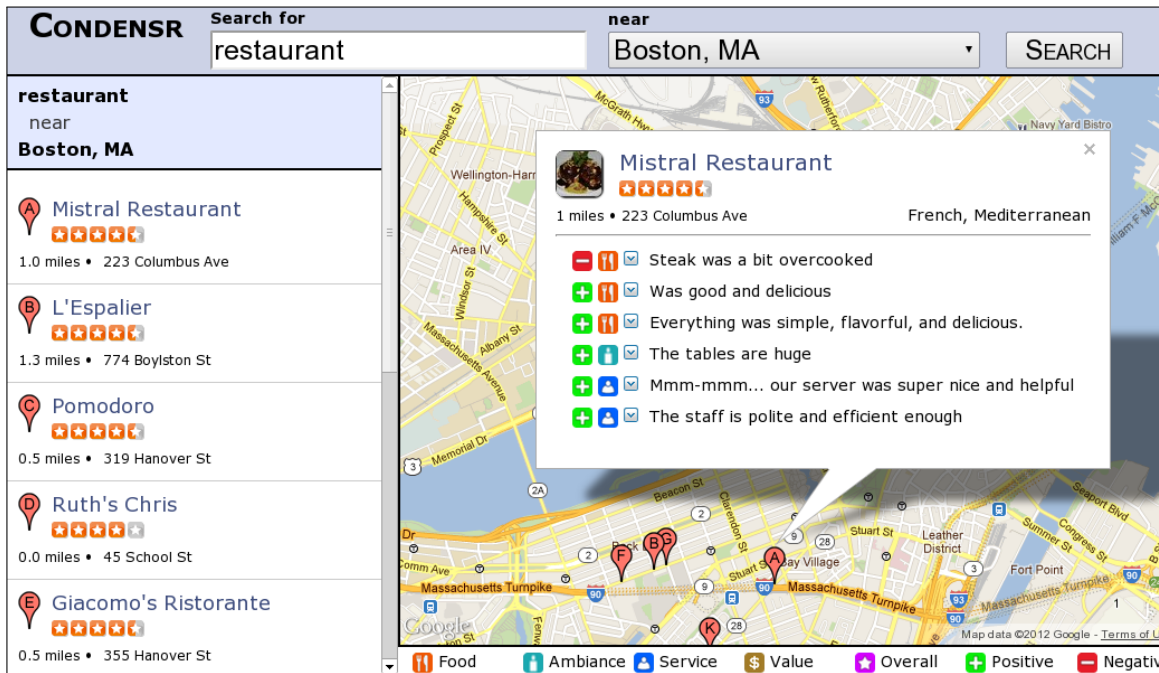
The CONDENSr demo system is designed to provide a qualitative evaluation of the results presented in Chapter 2. It incorporates the multi-aspect phrase extraction system from our work with Yelp restaurant search and a Google Maps interface. Figure A-1 demonstrates the main interface at <http://condensr.com>, and Figure A-2 shows the tooltip browsing interface at <http://condensr.com/browse>. In this chapter, I will provide details on system implementation (Section A.1) and some examples of results (Section A.2).

### A.1 System implementation

For the purpose of this demo system, all of the computation is performed in advance for a preselected set of restaurants. First, a set of restaurants is selected and their reviews are gathered. Next, a set of phrases are extracted using the multi-aspect phrase extraction system and pruned down to a small display set. Finally, the phrases for each restaurant are integrated with Google Maps and Yelp in order to produce the final demo.



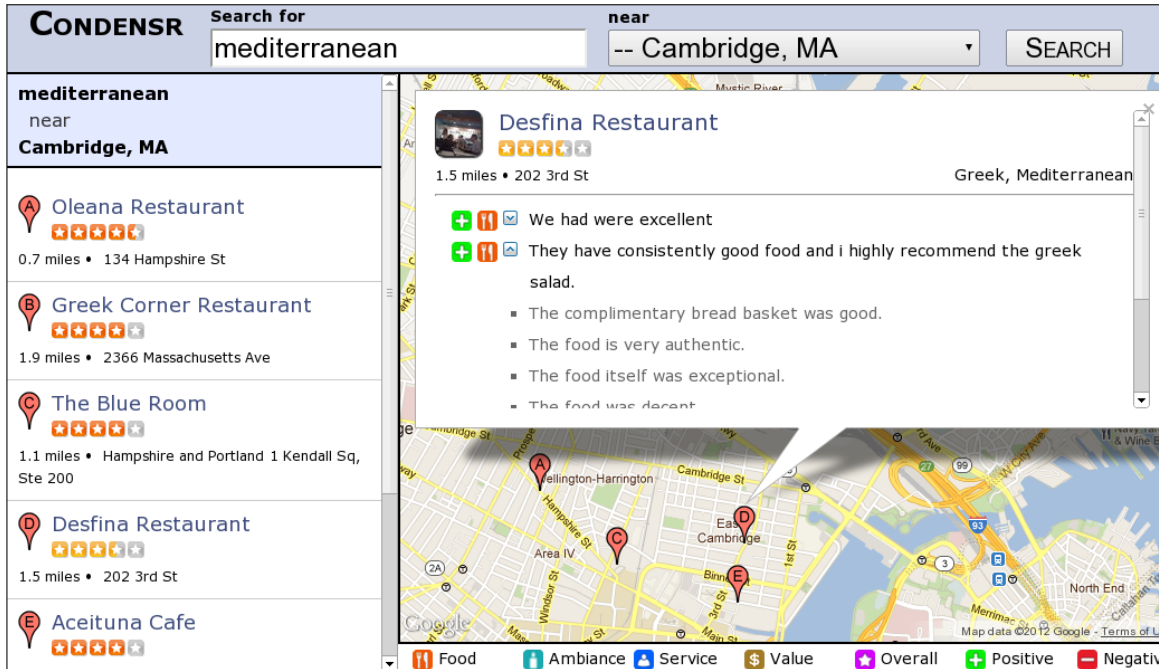
(a)



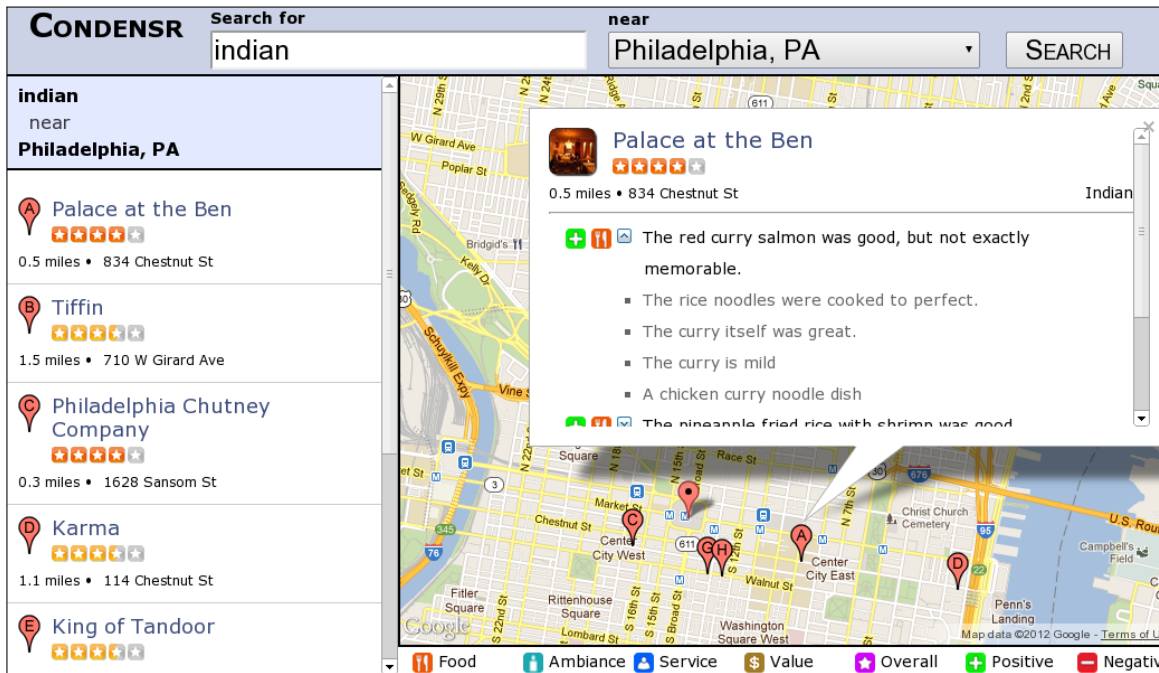
(b)

Figure A-1: Map interface on CONDENSr. Restaurants may be searched by location and search term (e.g., *restaurant* or *sandwich*). Search results are listed on the left with basic information, and the map interface functions as a standard Google Maps interface, where clicking on one of the markers either on the left or on the map itself will pull up a tooltip.





(c)



(d)


Figure A-1: Map interface on CONDENSr. Restaurants may be searched by location and search term (e.g., *restaurant* or *sandwich*). Search results are listed on the left with basic information, and the map interface functions as a standard Google Maps interface, where clicking on one of the markers either on the left or on the map itself will pull up a tooltip.



(a)

Figure A-2: Browsing interface on CONDENSr. All system tooltips can be navigated, and phrase clusters can be examined.


<< First      < Prev      **Page 280**      Next >      Last >>  
 Random


**Zaftigs Delicatessen**  
★ ★ ★ ★ ★

335 Harvard St Coolidge Corner Brookline, MA 02446      Breakfast & Brunch, Delis

---


- + 🍴 ☑ They serve bagel chips and a cream cheese dip which are fun and tasty.
- 🍴 ☑ Their prices are decent
- + 🍴 ☑ They all taste fresh and delicious
- + 👤 ☑ The staff is always friendly
- + 👤 ☑ The service was pretty good.
- + ★ ☑ This place is amazing


**Zahav**  
★ ★ ★ ★ ★

237 St James Pl Society Hill Philadelphia, PA 19106      Middle Eastern

---


- + 🍴 ☑ The rolls were great - i highly recommend including the spicy crunchy tuna roll
- 🍴 ☑ The rice was dry
- + 🍴 ☑ The sea urchin (uni), kept fresh in salt water, was sweet and creamy.
- 🚪 ☑ It is painted black and there are bars on the window so it feels like a claustrophobic jail cell
- 🚪 ☑ Soothingly toned-down lighting and minimalist pale wood dominated decor complete the ideal experience.
- + 👤 ☑ Service was very nice


**Zama**  
★ ★ ★ ★ ★

128 S 19th St Rittenhouse Square Philadelphia, PA 19103      Sushi Bars, Japanese

---

- + 🍴 ☑ I'd particularly recommend the beets
- + 🍴 ☑ The pizza with provelone and mini veal meatballs was amazing
- + 🍴 ☑ And boy was it good.
- + 👤 ☑ Trendy, fun and exciting
- + 👤 ☑ She was great... very nice and friendly.
- + 👤 ☑ The hostess was very helpful and polite to squeeze us in


**Zavino**

(b)

Figure A-2: Browsing interface on CONDENSr. All system tooltips can be navigated, and phrase clusters can be examined.

Boston	Los Angeles	Seattle
Boston, MA	Los Angeles, CA	Belltown, Seattle, WA
Cambridge, MA	Hollywood, CA	Bellevue, WA
Allston, MA	Encino, CA	Seattle, WA
Harvard Square	Santa Monica, CA	Redmond, WA
Somerville, MA	Anaheim, CA	
32 Vassar St (CSAIL)		
New York	Philadelphia	San Francisco
New York, NY	Philadelphia, PA	San Francisco, CA
Manhattan, NY	University of Pennsylvania	Berkeley, CA
W 34th St, NY		

(a) Set of locations, grouped by major metropolitan area

indian	thai
chinese	bar
cafe	salad
sandwich	soup
coffee	tea
pizza	vegetarian
sushi	

(b) Set of search terms

Area	# Restaurants
Boston	557
Los Angeles	720
New York	501
Philadelphia	295
San Francisco	467
Seattle	443

(c) Number of restaurants for each area

Table A.1: The full set of terms and locations used for CONDENSER, as well as the final counts of restaurants for each area. Locations are grouped by major metropolitan area.

## Restaurant selection

Restaurants are gathered by searching Yelp for the most relevant restaurants for each of several terms and locations. Specifically, we manually select a list of terms (e.g., *restaurant*, *Italian*, *cafe*) and a list of locations (e.g., *Cambridge*, *Boston*, *San Francisco*, *Berkeley*) and then find up to 20 results (the maximum returned by the Yelp API) for each term-location pair. For Boston specifically, we additionally include the restaurants from our original data set. The full set of terms and locations are given in Table A.1. For each unique restaurant in the resulting list, we scrape the full set of reviews from Yelp.

## Adaptation of the multi-aspect phrase extraction system

For this demo system, we would like to extract a very limited set of phrases for each restaurant, suitable for display on a map interface. To accomplish this, we would like to extract a set of very high-precision results, at the expense of recall if necessary. We can adjust the recall penalty mentioned in Section 2.3.1.4; specifically, by setting  $c = 0.5$  we allow the system to focus on high-precision extraction.

After running the multi-aspect phrase extraction system, the set of extracted phrases for each restaurant is still much too large to display on a map tooltip; there are an average of 43 phrases per restaurant. To prune this set further, we first separate food snippets from those with other labels; unsurprising for the restaurant domain, this is the largest label class. We then cluster the food phrases and other phrases separately based on the unigrams in the phrase and the category label, specifying 3 clusters for each set. Finally, we take the centroid of each cluster as the set of displayed phrases. To provide supporting evidence, we additionally select up to 4 phrases which are determined to be close to the centroid phrases; these phrases will be displayed on a dropdown for each of the centroid phrases.

In addition to displaying labeled phrases, we also would like to provide sentiment labels for each phrase or group of phrases. To do this, we use a straightforward sentiment classification algorithm involving a set of positive and negative seed words and negation trigger words. The algorithm counts the number of positive and negative words in the phrase, flipping any sentiment word which appears up to 3 places after a negation trigger word. If the final count falls towards positive, we label the phrase as positive; likewise, if the final count falls towards negative, we label the phrase as negative. For groups of phrases, if all phrases agree on sentiment, we display the final sentiment. Otherwise, we omit the sentiment label.

## Integration with Google Maps and Yelp

Interfacing with Google Maps and Yelp is fairly straightforward. Yelp provides a search API<sup>1</sup> with which we can identify the best results for a given search term and geographic area. We filter these results to select only those in our original set of restaurants. Google Maps also provides an API<sup>2</sup> with which we can display a map centered on a particular location, place lettered identifying markers, and specify a tooltip for each marker. We display all relevant results from Yelp on the left side, and display the map interface with popup marker tooltips on the right. Each tooltip displays the preselected set of phrases with category and sentiment labels, and for groups of phrases, there is a button to show or hide the remainder of the group.

### Improvements and scalability

At the current point in time, CONDENSER exclusively uses phrases which are generated offline using the multi-aspect phrase extraction system. In a future version of this system, results could be improved by pulling these results in real time. There are two major obstacles to this: first, the text of all reviews for each restaurant must be acquired quickly, and second, inference for the phrase extraction procedure must be efficient. While the second of these obstacles is not far away and could easily be obtained, the first would require a more direct connection to the data than the one we have available. Furthermore, beyond using the multi-aspect phrase extraction system, it would be possible to incorporate results like those from our aggregation system, which provide both more relevant aspects and a more accurate representation of sentiment.

## A.2 Examples

In this section, I present example tooltips from CONDENSER, taken from the browsing interface. When examining these tooltips, there are several clustering or sentiment

---

<sup>1</sup><http://www.yelp.com/developers/documentation/v2/overview>

<sup>2</sup><https://developers.google.com/maps/>

errors; however, as mentioned, clustering and polarity are merely heuristic postprocessing steps for display. Looking at the phrases themselves, the most common errors are cases where a good phrase is truncated at one end or the other. We also see several cases where the phrases are overly generic, such as “The food was good”. Finally, there are some entire restaurants which are mislabeled and should not appear in a set of restaurant data; these are a result of mislabeled data in our input set from Yelp.



## The Beehive

★★★★☆

541 Tremont St South End Boston, MA 02116

American (Traditional), Music Venues

---

+
ff
u
To be good

- ff The food proved to be good.
- ff The menu looked promising enough
- ff Promising enough
- ff Most of us ate eggs benedict with either ham or smoked salmon

+
ff
u
Both were so rich and delicious

- ff The fries and gravy were amazing!
- ff All very good
- ff The drinks are absolutely delicious
- ff The fried lavash chips were cold and stale

+
ff
u
The burger looked amazing and huge

- ff The wedge salad was sooo yummy.
- ff The lobster mac and cheese was soupy
- ff My friends food she got was luke warm
- ff The beehive!! i've been there 3 of the 5 nights it's been open

+
i
u
To sit downstairs, eat a good meal, and listen to some great live music

- i To be able to sit downstairs, eat a good meal, and listen to some great live music
- i The live music added to the hip vibe
- i The music was good
- i Cover for live music is worth something too

+
u
u
The waitress was very attentive and engaged

- i The atmosphere is nice
- i The people were all gorgeous, hip, artsy, studious, industrious
- u The service was even better
- i The atmosphere is warm and inviting

★
u
I'll definitely be back

- \$ Granted the drinks are a little \$ \$ \$, but it's worth it.
- \$ The drink prices are not cheap - \$ 10.50 for my cocktail
- ★ To give this place more stars
- ★ This place is definitely neat.

(a) The Beehive, <http://www.yelp.com/biz/the-beehive-boston>

Figure A-3: Several complete tooltips from a variety of restaurants on CONDENSr.





**The Anaheim White House**

887 S Anaheim Blvd Anaheim, CA 92805

★★★★★

Italian, Seafood

---

+ 👤 👤 👤 The food was tasty and very fulfilling.

- 👤 The food was exquisite
- 👤 All the food looked good but tasted pretty bland.
- 👤 The food was not great at all
- 👤 The food was not buffet style

+ 👤 👤 👤 The calamari was delicious but just slightly too rich for my taste

- 👤 The sea bass and calamari dish is delicious.
- 👤 And the souffle was just one of my favorites.
- 👤 Wife had the sea bass that she loved!
- 👤 Us a souffle for dessert

+ 👤 👤 👤 The fish was delicious, it was buttery and melted in my mouth it was delicious the cacciatore was okay.

- 👤 The chocolate banana tart was very yummy
- 👤 The presentation was good
- 👤 The salmon was flakey and pretty good
- 👤 It was exceptional

+ 👤 👤 👤 Their customer service was awesome and friendly.

- 👤 The staff was great.
- 👤 Overall the service was very good
- 👤 The service was quick, too
- 👤 The staff are eager to serve and provide the best in customer service.

+ 👤 👤 👤 The servers were swift on their feet, attentive, and friendly.

- 👤 They were very helpful and friendly.
- 👤 Our waiter exhibited great skills
- 👤 Our waiter took our cocktail and appetizer order
- 👤 To wait on us

- ★ 👤 👤 This place is bad.

- ★ This place is strange.
- ★ This is a special occasion type of place
- 👤 The ambiance of it is quite homey, kind of quiet.
- ★ This place is almost hard to spot.

(b) The Anaheim White House, <http://www.yelp.com/biz/the-anaheim-white-house-anaheim>

Figure A-3: Several complete tooltips from a variety of restaurants on CONDENSr.



## Herbivore

★
★
★
★
★

983 Valencia St Mission San Francisco, CA 94110

Vegetarian, Vegan

---

+
👤
🗨️

The portion was generous

- 👤 I highly recommend the ravioli with basil pesto
- 👤 Soy chicken shwarma.
- 👤 The half veggie burger was pretty good but the hot and sour soup or whatever the hell was horrible.
- 👤 Their vietnamese spring rolls are must

+
👤
🗨️

Their grilled vegetables was also pretty good.

- 👤 It was simple
- 👤 It was alright
- 👤 Was really big

+
👤
🗨️

Their coconut curry soup which was good but nothing memorable

- 👤 The pancakes were delicious and fluffy.
- 👤 I had the pasta primavera which was excellent
- 👤 The lemon-herb cream sauce penne pasta dish
- 👤 Love the penne pasta with lemon herb sauce!

+
👤
🗨️

The staff is super friendly too

- 👤 We arrived at the cross streets and looked around
- 👤 The waitress spilled water on me not once but thrice
- 👤 " the service is always relatively

-
👤
🗨️

The service was sluggish at best, and ironically subsequently impatient.

- 👤 The service was spotty
- 👤 Our service was sluggishly slow
- 👤 The wait was sooo not worth it


+
★
🗨️

I highly recommend this place

- ★ To try this place again
- 👤 To receive our food
- 👤 And the servers are
- 👤 The floors and walls are bare

(c) Herbivore, <http://www.yelp.com/biz/herbivore-san-francisco-2>

Figure A-3: Several complete tooltips from a variety of restaurants on CONDENSr.



## Luigi's D'italia

★
★
★
★
★

801 S State College Blvd Anaheim, CA 92806
Italian, Pizza

---

+
👤
👤

Salad dressing was great

- 👤 The salad was the same.
- 👤 The house salad dressing is also pretty tepid and irrelevant
- 👤 Food's gotta be authentic italian

+
👤
👤

My two favorite entrées are the chicken tortellini and the seafood pasta dish.

- 👤 All the pasta tasted fresh
- 👤 Calamari was a little dry but good.
- 👤 Getting the calamari appetizer
- 👤 The pizza was good/tasty

+
👤
👤

Their sauce was so incredibly tasty & tasted homemade, i absolutely loved it.

- 👤 The meat was very tender
- 👤 The dips it came with were great
- 👤 It was fantastic
- 👤 The sauce was fabulous

👤
👤

She promptly took our drinks and orders.

- 👤 To ignore our section

+
👤
👤

The owners and staff are really friendly

- 👤 Possibly are not trained well enough to implement good service
- 👤 The place was busy with people
- 👤 The service was spot on.


+
★
👤

This place a great bargain and find

- ★ This is a good restaurant.
- ★ This is the best authentic italian food around here

(d) Luigi's D'italia, <http://www.yelp.com/biz/luigis-d-italia-anaheim>

Figure A-3: Several complete tooltips from a variety of restaurants on CONDENSr.



## Valencia Pizza & Pasta

★
★
★
★
★
★

801 Valencia St Mission San Francisco, CA 94110

Pizza, Italian, Breakfast & Brunch

---

+
🍴
👤

Generally the food is consistently good

- 🍴 The food was still great
- 🍴 The food is plentiful
- 🍴 The food is comforting

+
🍴
👤

Its so good

- 🍴 It's always firm and fresh
- 🍴 It wasn't my favorite
- 🍴 It was ok
- 🍴 It's grilled to perfection

+
🍴
👤

The pork chop and steaks are fantastic

- 🍴 The sauteed veggies are always delicious.
- 🍴 The portion sizes are outrageous

+
👤
👤

Service is also friendly and efficient.

- 👤 Service was prompt, not too friendly
- 👤 Service is very friendly
- 👤 The service was decent

+
👤
👤

My service was fantastic and we had no wait at all

- 👤 The waitress ' attitude was uncalled for
- 👤 Us had already received their entrees
- 👤 The wait staff is rude


+
★
👤

This place is pretty good.

- ★ This place is awesome.
- ★ This place delivers a really great value.
- ★ This place is goooood!
- ★ This place is magic comfort food.

(e) Valencia Pizza & Pasta, <http://www.yelp.com/biz/valencia-pizza-and-pasta-san-francisco>

Figure A-3: Several complete tooltips from a variety of restaurants on CONDENSr.





## Grendel's Den Restaurant & Bar





★★★★★




89 Winthrop St Harvard Square Cambridge, MA 02138




Bars, Sandwiches




---





  **The food's not spectacular**




-  The food was not spectacular.
-  The food is just pub food
-  The food isn't the best i've ever had
-  The bar food is decent





   **The lobster roll my mother ordered was dry and scant -- she also thought it was "fishy" tasting, however**




-  It was dry
-  It was a very small portion
-  It was lighter and a little sweeter.




   **But the spinach pie is great!**




-  The homefries were excellent - whole slices of potato satueed lightly.
-  The cheese grits and spinach frittata was excellent
-  The flavors were good
-  The slice of tomato was rather measly




   **It just feels like a fun place to be- basement, fireplace, relaxed crowd, neat tables and seat arrangements**

-  The staff is warm and friendly, you seat yourself but everyone is really pleasant.
-  The place had a relaxed vibe
-  I love this place
-  The place is sort of dark and almost cavernous

   **To be helpful**


-  The tables are small
-  Us to sit wherever we wanted
-  The drinks are yum

   **Waiters are also very friendly and attentive.**

-  The service is brisk but attentive.
-  The atmosphere was nice and comfortable
-  The music was also a little too loud.

(f) Grendel's Den Restaurant & Bar, <http://www.yelp.com/biz/grendels-den-restaurant-and-bar-cambridge>

Figure A-3: Several complete tooltips from a variety of restaurants on CONDENSr.









## Grendel's Den Restaurant & Bar




89 Winthrop St Harvard Square Cambridge, MA 02138




Bars, Sandwiches




---





  **The food's not spectacular**




-  The food was not spectacular.
-  The food is just pub food
-  The food isn't the best i've ever had
-  The bar food is decent





   **The lobster roll my mother ordered was dry and scant -- she also thought it was "fishy" tasting, however**




-  It was dry
-  It was a very small portion
-  It was lighter and a little sweeter.




   **But the spinach pie is great!**




-  The homefries were excellent - whole slices of potato satueed lightly.
-  The cheese grits and spinach frittata was excellent
-  The flavors were good
-  The slice of tomato was rather measly




   **It just feels like a fun place to be- basement, fireplace, relaxed crowd, neat tables and seat arrangements**

-  The staff is warm and friendly, you seat yourself but everyone is really pleasant.
-  The place had a relaxed vibe
-  I love this place
-  The place is sort of dark and almost cavernous

   **To be helpful**

-  The tables are small
-  Us to sit wherever we wanted
-  The drinks are yum

   **Waiters are also very friendly and attentive.**

-  The service is brisk but attentive.
-  The atmosphere was nice and comfortable
-  The music was also a little too loud.

(g) The Helmand, <http://www.yelp.com/biz/the-helmand-cambridge>

Figure A-3: Several complete tooltips from a variety of restaurants on CONDENSr.

---

## Bibliography

---

- [1] Regina Barzilay and Mirella Lapata. Modeling local coherence: an entity-based approach. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 141–148, 2005.
- [2] Regina Barzilay and Lillian Lee. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of HLT-NAACL*, pages 113–120, 2004.
- [3] Regina Barzilay, Kathleen McKeown, and Michael Elhadad. Information fusion in the context of multi-document summarization. In *Proceedings of ACL*, pages 550–557, 1999.
- [4] Adam Berger and Vibhu O. Mittal. Query-relevant summarization using faqs. In *In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 294–301, 2000.
- [5] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.
- [6] Sasha Blair-goldensohn and Kathleen Mckeown. Integrating rhetorical-semantic relation models for query-focused summarization. In *In Proceedings of the Document Understanding Conference, DUC-2006*, 2006.
- [7] David M. Blei and Jon McAuliffe. Supervised topic models. In *Advances in NIPS*, pages 121–128, 2008.
- [8] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
- [9] S. R. K. Branavan, Harr Chen, Jacob Eisenstein, and Regina Barzilay. Learning document-level semantic properties from free-text annotations. *JAIR*, 34:569–603, 2009.

- [10] Giuseppe Carenini and Johanna D. Moore. Generating and evaluating evaluative arguments. *Artificial Intelligence*, 170:925–952, August 2006.
- [11] Giuseppe Carenini, Raymond T. Ng, and Ed Zwart. Extracting knowledge from evaluative text. In *Proceedings of the 3rd international conference on Knowledge capture*, K-CAP '05, pages 11–18, New York, NY, USA, 2005. ACM. ISBN 1-59593-163-5. doi: <http://doi.acm.org/10.1145/1088622.1088626>. URL <http://doi.acm.org/10.1145/1088622.1088626>.
- [12] Giuseppe Carenini, Raymond Ng, and Adam Pauls. Multi-document summarization of evaluative text. In *Proceedings of EACL*, pages 305–312, 2006.
- [13] Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. Rst discourse tree-bank, 2002. URL <http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002T07>.
- [14] Harr Chen, S. R. K. Branavan, Regina Barzilay, and David R. Karger. Global models of document structure using latent permutations. In *Proceedings of ACL/HLT*, pages 371–379, 2009.
- [15] Harr Chen, S. R. K. Branavan, Regina Barzilay, and David R. Karger. Content modeling using latent permutations. *JAIR*, 36:129–163, 2009.
- [16] Harr Chen, Edward Benson, Tahira Naseem, and Regina Barzilay. In-domain relation discovery with meta-constraints via posterior regularization. In *Proceedings of ACL*, pages 530–540, 2011.
- [17] Timothy Chklovski and Rada Mihalcea. Building a sense tagged corpus with open mind word expert. In *Proceedings of the ACL-02 workshop on Word sense disambiguation: recent successes and future directions - Volume 8*, WSD '02, pages 116–122, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118675.1118692. URL <http://dx.doi.org/10.3115/1118675.1118692>.
- [18] Yejin Choi and Claire Cardie. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the EMNLP*, pages 793–801, 2008.
- [19] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.
- [20] Koby Crammer and Yoram Singer. Pranking with ranking. In *Advances in Neural Information Processing Systems 14*, pages 641–647. MIT Press, 2001.
- [21] Hoa Trang Dang. Overview of duc 2005. In *Proceedings of the Document Understanding Conf. Wksp. 2005 (DUC 2005) at the Human Language Technology Conf./Conf. on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, 2005.



- [22] Hoa Trang Dang. Overview of duc 2006. In *Proceedings of HLT-NAACL 2006*, 2006.
- [23] Hal Daumé III and Daniel Marcu. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 305–312, Sydney, Australia, July 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220214. URL <http://www.aclweb.org/anthology/P06-1039>.
- [24] Kushal Dave, Steve Lawrence, and David M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web, WWW '03*, pages 519–528, 2003. ISBN 1-58113-680-3.
- [25] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society For Information Science*, 41(6):391–407, 1990.
- [26] John DeNero, David Chiang, and Kevin Knight. Fast consensus decoding over translation forests. In *Proceedings of the ACL/IJCNLP*, pages 567–575, 2009.
- [27] Micha Elsner, Joseph Austerweil, and Eugene Charniak. A unified local and global model for discourse coherence. In *Proceedings of the NAACL/HLT*, pages 436–443, 2007.
- [28] Jenny Rose Finkel and Christopher D. Manning. Joint parsing and named entity recognition. In *Proceedings of the NAACL*, 2009.
- [29] M A Fligner and J S Verducci. Distance based ranking models. *Journal of the Royal Statistical Society*, 48(3):359–369, 1986.
- [30] Radu Florian and David Yarowsky. Dynamic nonlocal language modeling via hierarchical topic-based adaptation. In *Proceedings of the ACL*, pages 167–174, 1999.
- [31] Dayne Freitag. Trained named entity recognition using distributional clusters. In *Proceedings of the EMNLP*, pages 262–269, 2004.
- [32] Pascale Fung and Grace Ngai. One story, one flow: Hidden markov story models for multilingual multidocument summarization. *ACM Trans. Speech Lang. Process.*, 3:1–16, July 2006.
- [33] Andrew B. Goldberg and Xiaojin Zhu. Seeing stars when there aren’t many stars: Graph-based semi-supervised learning for sentiment categorization. In *Proceedings of the NAACL/HLT Workshop on TextGraphs*, pages 45–52, 2006.

- [34] Yihong Gong. Generic text summarization using relevance measure and latent semantic analysis. In *in Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001.
- [35] Joshua Goodman. Semiring parsing. *Computational Linguistics*, 25(4):573–605, 1999.
- [36] Aria Haghighi and Lucy Vanderwende. Exploring content models for multi-document summarization. In *Proceedings of the NAACL/HLT*, pages 362–370, 2009.
- [37] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence, UAI99*, pages 289–296, 1999.
- [38] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of SIGKDD*, pages 168–177, 2004.
- [39] R. Iyer and M. Ostendorf. Modeling long distance dependence in language: Topic mixtures vs. dynamic cache models. In *IEEE Transactions on Speech and Audio Processing*, pages 236–239, 1996.
- [40] George Karypis. CLUTO a clustering toolkit. Technical Report 02-017, Dept. of Computer Science, University of Minnesota, 2002. Available at <http://www.cs.umn.edu/~cluto>.
- [41] Hyun Duk Kim and ChengXiang Zhai. Generating comparative summaries of contradictory opinions in text. In *Proceedings of CIKM*, pages 385–394, 2009.
- [42] S.M. Kim and E. Hovy. Automatic detection of opinion bearing words and sentences. In *Companion Volume to the Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP-05)*, pages 61–66, 2005.
- [43] Soo-Min Kim and Eduard Hovy. Automatic identification of pro and con reasons in online reviews. In *Proceedings of COLING/ACL*, pages 483–490, 2006.
- [44] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of ACL*, pages 423–430, 2003.
- [45] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the ACL*, pages 74–81, 2004.
- [46] Bing Liu, Minqing Hu, and Junsheng Cheng. Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of WWW*, pages 342–351, 2005.
- [47] Dong C. Liu, Jorge Nocedal, Dong C. Liu, and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

- [48] Jingjing Liu and Stephanie Seneff. Review sentiment scoring via a parse-and-paraphrase paradigm. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 161–169, Singapore, August 2009. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D/D09/D09-1017>.
- [49] Jingjing Liu, Stephanie Seneff, and Victor Zue. Dialogue-oriented review summary generation for spoken dialogue recommendation systems. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 64–72, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5. URL <http://dl.acm.org/citation.cfm?id=1857999.1858007>.
- [50] Yue Lu and ChengXiang Zhai. Opinion integration through semi-supervised topic modeling. In *Proceedings of WWW*, pages 121–130, 2008.
- [51] I. Mani. *Automatic summarization*, volume 3. John Benjamins Pub Co, 2001.
- [52] William C. Mann and Sandra A. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *TEXT*, 8(3):243–281, 1988.
- [53] Daniel Marcu. From discourse structures to text summaries. In *In Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 82–88, 1997.
- [54] Daniel Marcu and Abdessamad Echihabi. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 368–375, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073145. URL <http://dx.doi.org/10.3115/1073083.1073145>.
- [55] Ryan Mcdonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 2007.
- [56] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of WWW*, pages 171–180, 2007.
- [57] Shamima Mithun and Leila Kosseim. A hybrid approach to utilize rhetorical relations for blog summarization. In *Proceedings of TALN*, 2010.
- [58] Gabriel Murray, Steve Renals, and Jean Carletta. Extractive summarization of meeting recordings. In *in Proceedings of the 9th European Conference on Speech Communication and Technology*, pages 593–596, 2005.

- [59] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, pages 271–278, 2004.
- [60] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*, pages 115–124, 2005.
- [61] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2:1–135, January 2008. ISSN 1554-0669. doi: 10.1561/15000000011. URL <http://dl.acm.org/citation.cfm?id=1454711.1454712>.
- [62] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, 2002.
- [63] Siddharth Patwardhan and Ellen Riloff. Effective information extraction with semantic affinity patterns and relevant regions. In *Proceedings of the EMNLP/CoNLL*, pages 717–727, 2007.
- [64] 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.
- [65] Rin Popescul and Lyle H. Ungar. Probabilistic models for unified collaborative and content-based recommendation in sparsedata environments. In *Proceedings of UAI*, pages 437–444, 2001.
- [66] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, feb 1989. ISSN 0018-9219. doi: 10.1109/5.18626.
- [67] D. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: Sentence extraction, utility-based evaluation and user studies. In *Proceedings of ANLP/NAACL Summarization Workshop*, 2000.
- [68] Dragomir Radev and Kathleen McKeown. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500, September 1998.
- [69] A. Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proceedings of EMNLP*, volume 1, pages 133–142, 1996.
- [70] Christina Sauper, Aria Haghighi, and Regina Barzilay. Incorporating content structure into text analysis applications. In *Proceedings of EMNLP*, pages 377–387, 2010.

- [71] Yohei Seki, Koji Eguchi, Noriko Kanodo, and Masaki Aono. Multidocument summarization with subjectivity analysis at duc 2005. In *Proc. of the Document Understanding Conf. Wksp. 2005 (DUC 2005) at the Human Language Technology Conf. / Conf. on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, 2005.
- [72] Yohei Seki, Koji Eguchi, Noriko Kanodo, and Masaki Aono. Opinion-focused summarization and its analysis at DUC 2006. In *Proceedings of DUC*, pages 122–130, 2006.
- [73] Rion Snow, Brendan O’Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the EMNLP*, 2008.
- [74] Benjamin Snyder and Regina Barzilay. Multiple aspect ranking using the good grief algorithm. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, pages 300–307, 2007.
- [75] Swapna Somasundaran, Galileo Namata, Janyce Wiebe, and Lise Getoor. Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In *Proceedings of the EMNLP*, pages 170–179, 2009.
- [76] Radu Soricut and Daniel Marcu. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL ’03*, pages 149–156, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073475. URL <http://dx.doi.org/10.3115/1073445.1073475>.
- [77] Radu Soricut and Daniel Marcu. Discourse generation using utility-trained coherence models. In *Proceedings of the COLING/ACL on Main conference poster sessions*, COLING-ACL ’06, pages 803–810, 2006.
- [78] Caroline Sporleder and Mirella Lapata. Discourse chunking and its application to sentence compression. In *Proceedings of the HLT/EMNLP*, pages 257–264, 2005.
- [79] Josef Steinberger and Karel Jeek. Using latent semantic analysis in text summarization and summary evaluation. In *In Proc. ISIM 04*, pages 93–100, 2004.
- [80] Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of WWW*, pages 111–120, 2008.
- [81] Ivan Titov and Ryan McDonald. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL*, pages 308–316, 2008.

- [82] Peter D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02*, pages 417–424, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: <http://dx.doi.org/10.3115/1073083.1073153>. URL <http://dx.doi.org/10.3115/1073083.1073153>.
- [83] Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. A model-theoretic coreference scoring scheme. In *Proceedings of MUC*, pages 45–52, 1995.
- [84] B. Webber, M. Egg, and V. Kordoni. Discourse structure and language technology. *Natural Language Engineering*, FirstView:1–54, 2011.
- [85] Julie Weeds, David Weir, and Diana McCarthy. Characterising measures of lexical distributional similarity. In *Proceedings of the COLING*, page 1015, 2004.
- [86] Theresa Wilson, Janyce Wiebe, and Rebecca Hwa. Just how mad are you? finding strong and weak opinion clauses. In *Proceedings of the AAAI*, pages 761–769, 2004.
- [87] H. Yu and V. Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 129–136. Association for Computational Linguistics, 2003.