STUDYING PEOPLE, ORGANIZATIONS, AND THE WEB WITH STATISTICAL TEXT MODELS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Daniel Ramage
December 2011

Preface

From social networks to academic publications, information technologies have enabled companies, organizations, and governments to collect huge datasets about the world. Many of these datasets have major textual components organized by human-applied labels or tags, promising to improve our understanding of large scale topical and social phenomena through the words people write. Doing so requires tools that can discover and quantify word usage patterns that are interpretable, trustworthy, and flexible. In particular, the discovered patterns should exploit the implicit domain knowledge embodied in tags, labels, or other categories of interest, when they are available, and lend themselves to visual exploration and interpretation.

This dissertation presents studies of topical structure of the tagged web, social language in microblogs, and innovation in academia through statistical analyses of text. Several new probabilistic topic models of metadata-enriched document collections are introduced, facilitating domain specific studies of words associated with tags, emoticons, library subject codes, and other human-provided labels. I find that tags improve high-level clustering of web pages; that language on Twitter can be quantified with respect to its role as substance, status, social, or style; and that interdisciplinary research consistently uses language that looks like academia's future. These results are evaluated both quantitatively, with gold standard and task driven metrics, and qualitatively with visualizations of the textual patterns discovered by the models.

Acknowledgments

This dissertation would not have been possible without the guidance of my advisor, Chris Manning, and my reading committee, Dan McFarland and Dan Jurafsky. I am also indebted to the invaluable contributions of my many talented collaborators, some of whose contributions are reflected here. Thanks in particular to David Hall, Ramesh Nallapati, Paul Heymann, Hector Garcia-Molina, Jeff Heer, Evan Rosen, Susan Dumais, as well others in the Stanford NLP group and AI lab. Jason Chuang and Dan Liebling deserve special mention because their visualization code generated some of screenshots you'll see within. My thanks also to NDSEG, IARDA AQUAINT, MSR, the Stanford University President's Office through IRiSS, and NSF CDI grant for supporting this research. To Janet and my family: thank you for your unwavering support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Information retrieval and web search technologies have been tremendously successful at indexing large text collections. But today, so many traces of society are indexed that the kinds of questions we want to ask go beyond the capabilities of systems designed to retrieve individual documents. For example, how are individuals' social roles reflected in the language they use in communicating with friends? How should we design social networks in light of this? What environments foster innovative ideas to take hold in academia or corporations? How do these ideas spread, and what are the implications for resource allocation?

Answers to high level questions like these can only be made by informed human judgment supported by computational tools to help collate and summarize relevant textual data. Text is uniquely suited to shed light on such questions because it is available in great quantity, is semantically richer than demographic information or network links, and is accessible to computational analysis. From financial statements to aircraft design, people depend on empirical statistics and computational models to develop informed judgment when numerical data is available. But how can we use the data in text collections for insight into social science questions about the nature and behavior of people, organizations, and ideas?

Traditional approaches from the humanities and social sciences develop the researcher's domain expertise directly: for a small enough collection, a researcher can simply read the text. Existing domain experts can be consulted to provide context.

Field studies can be run to give a researcher first-hand experience. Such approaches, however, simply do not scale to document collections numbering in the millions. Gaining insight into large collections requires the help of computational tools to aide researchers seeking meaningful insight.

General-purpose text analysis tools have been developed over the last decades, designed to discover and quantify patterns in text. These have become indispensable to text mining practitioners, whose goal is to develop quantitative, numerical measures of patterns in text collections that can be trusted and meaningfully interpreted. These tools include automatic classification and clustering techniques [87, 84], latent topic models that discover weighted sets of words that tend to co-occur across documents [16, 48], and custom gloss lists of words designed to measure specific phenomena [105, 104]. The challenge of text mining is translating these broadly applicable techniques to specific studies of phenomena in the world.

Recent research has taken first steps in combining general purpose text analysis tools with in-depth domain expertise to develop novel insight into macro-scale phenomena in the world. These studies range from models of politics [101] and mood on Twitter [44] to the study of research communities [12, 50] or types of political communication [49]. What many of these approaches have in common—above and beyond the application of one or more of the text mining techniques listed above—is the extent to which they rely on domain expertise to develop and interpret the quantitative output of models. For example, consider the focused study of a single academic field: understanding the adoption of statistical methodologies in natural language processing. Hall, et al., [50] combined topic models with domain expertise to track the rise of statistical methodologies to a single workshop where influential researchers from the NLP and speech processing communities interacted for the first time. As in [12] and [49], the authors' expertise in the subject area is critical to the study's success. In particular, studies that use unsupervised topic models [50], dimensionality reduction techniques [12], or clustering algorithms [49], demand that the text mining practitioner employ expert knowledge of the domain in order to interpret the trends in context. It is the domain expertise of the authors themselves that enable a model's output to signify trends in the real world.

However, many domains of interest are effectively devoid of domain experts. For instance, if we want to study how interdisciplinarity has affected academia as a whole (as in Chapter 7), we cannot hope to develop the necessary expertise in all areas of academia. Traditional methods from sociology and history of science based on literature reviews and expert interviews are overwhelmed by the number of combinations of fields to pursue and documents to read. But without the kind of domain expertise demonstrated in [12, 50, 49], purely unsupervised data-driven methods cannot be meaningfully interpreted. As the scale of data or scope of questions expands, we need a methodology based upon the data we have that can be interpreted with only minimal domain knowledge.

The approach I take in this dissertation is to leverage the implicit expertise lurking in the data: human-provided labels. There is no shortage of human-provided labels on many modern text collections. Curated databases, such as the PhD dissertations analyzed for in Chapter 7, often contain standardized classification codes maintained by taxonomic experts. On sites like Wikipedia, we might use category page links as a de-facto consensus vocabulary of many volunteer editors. And even in open-domain text collections, we can often find rich user-generated *tags*, free-form text annotations designed to aid human information seeking. Tags are applied to web pages (through sites like Delicious[1] and StumbleUpon[2]) and found in user-generated content on microblogs like Twitter.[3] Each kind of label space super-imposes a human-interpretable organization upon a slice of the world's electronic text, and each has the potential to act as a proxy form of domain expertise.

But how can label spaces be used as proxies for domain experts? The approach I take in this dissertation is to move away from purely unsupervised textual analysis to statistical models of *labeled* text collections. Much information exists in the implicit associations between words in documents and the label spaces people use to organize those documents. Consider a dissertation categorized as both Genetics and Computer Science. From only this one document, we could not hope to discern which words

[1] http://delicious.com/

[2] http://stumbleupon.com/

[3] http://twitter.com/ — Roughly 11% of posts in Twitter's November 2009 spritzer feed contain a hashtag.

are from Genetics and which are from Computer Science. But by looking at the distribution of words and labels across the entire collection, we might discover that words such as "genome" and "sequence" are statistically more likely to occur together in Genetics documents, whereas terms like "algorithm" and "complexity" are more likely to occur in Computer Science. I build upon this intuition in developing several statistical text models throughout this dissertation.

Not every text mining model will support meaningful quantitative insight shedding light on questions like those posed above. I identify three properties a model must exhibit in order to succeed: trustworthiness, interpretability, and flexibility. A model is *trustworthy* if its output leads to reasonable conclusions across a variety of conditions. A model is *interpretable* if its output has meaning that can be communicated externally without undue reliance on the model's internal state. Finally, a model is *flexible* if it can accommodate supervised input from domain experts or text mining practitioners without demanding that every potential pattern of interest be fully labeled.

This dissertation progressively develops models designed to have the properties of trustworthiness, interpretability, and flexibility. Any model of text that ignores human-provided labels is at a disadvantage in its trustworthiness. So in Chapter 3, I present a novel topic model called Multi-Multinomial Latent Dirichlet Allocation (MM-LDA) that incorporates human provided labels to improve the quality of latent topics. This chapter, based on work first published in [113], uses tags as a source of information analogous to words for a high-level organizational task of web page clustering. While MM-LDA outperforms traditional LDA at clustering, its latent topic assumption is an obstacle for the model's interpretability. The next model in Chapter 4, based on work first published in [112], is designed to avert this shortcoming. Labeled LDA (L-LDA), changes the relationship of labels to topics, constraining each topic to align with exactly one label. I demonstrate the model's interpretability advantages over latent topics and its competitiveness at a tag prediction task. However, the model's strong one-to-one assumption of labels to topics limits its flexibility with regard to modeling unlabeled patterns in the data. Chapter 5, based on work first published in [114], introduces Partially Labeled Dirichlet Allocation (PLDA) and the

Partially Labeled Dirichlet Process (PLDP). These models introduce new kinds of modeling flexibility: PLDA and PLDP are hybrid "partially-supervised" models that generalize both supervised techniques such as L-LDA and unsupervised techniques like LDA. They learn latent sub-topics of labels while still allowing for the presence of unlabeled, corpus-wide background topics.

The broader goal of this dissertation is to advance our ability to use text as a lens for studying human social systems. Chapter 5 of this dissertation acts as bridge between the statistical models of the preceding chapter and the research methodologies demonstrated in the case studies of Chapters 6 and 7. The methodologies speak to three major categories of questions that social scientists often explore: those about *people*, *organizations* and *ideas*. These categories are reflected in the literature published by practitioners of the other major computational approach to large scale social systems: social network analysis [39]. Social network analysis techniques examine networks of formal variables like "X communicates with Y" or "X published with Y," i.e. networks concerning people and the organizations in which those people participate [67]. They also consider variables like "paper A cites paper B" or "web page A links to page B," i.e. networks connecting ideas [20]. While I do not directly compare to these techniques in this dissertation, the case studies in Chapters 6 and 7 illustrate the richer characterizations that textual analysis makes possible, versus techniques limited to the presence or absence of ties.

Chapter 6, based on work first published in [111], presents a study of the largely social domain of people's communication through a popular microblogging social network. We find that some interesting patterns of language use can be characterized with known labels (emoticons, hashtags, etc.) and other patterns can be discovered automatically without labeling. In Chapter 7, I study organizations and ideas in an in-depth look at interdisciplinarity in academia: while science does have a social side, it is the organizational structure of universities and departments, as well as the ideas themselves, that best characterize intellectual output. We study this output through the lens of one million PhD dissertation abstracts filed since 1980. Taken together, these studies offer substantial coverage of the kinds of questions that computational social science can and should study, and we demonstrate how aspects of both can be

expressed in terms of the general labeled text modeling framework developed in the preceding chapters.

This dissertation develops four broadly applicable computational models of text collections in the presence of human-interpretable metadata in Chapters 3, 4, and 5. The central theme of these models is to exploit the human-interpretable labels to improve the trustworthiness, interpretability, and flexibility of the models. The studies of social and intellectual phenomena in Chapters 6 and 7 show how these methods can inform our analyses of semi-structured text collections at multiple scales: from a birds-eye view of patterns in the data, to how groups, individuals, and documents fit into these patterns, all the way down to the meaning and usage of individual words. I conclude with some reflections on the future of interpretable text mining in Chapter 8: how can we build upon the methods developed here to better study the world through the text people write.

Chapter 2

Preliminaries

Computational models of meta-data enriched text, such as those developed and applied in this dissertation, must first address a question of representation. What computational framework can adequately describe the meaning of words and their role in a discourse or document collection? The simplest answer to this question is the bag-of-words (BoW) assumption: the meaning of a collection of words is taken as the histogram of the counts of its words. This assumption should sound crazy. We know that the composition and context of words cannot be divorced from their counts if meaning is to be retained. Linguist Zellig Harris argued as much in one of the first usages of the phrase "bag of words" in 1954 [52]: "And this stock of combinations of elements becomes a factor in the way later choices are made ... for language is not merely a bag of words but a tool with particular properties which have been fashioned in the course of its use."

Indeed, the field of Natural Language Processing since Harris has developed numerous computational approaches to language representation that are more nuanced than a bag of words. Statistical language models [46] use multi-word n-grams as linguistic context and are a part of modern state of the art machine translation [70] and speech recognition [110] systems. Formal models of syntax such as PCFGs [86], HPSGs [106], LFGs [34], and dependency grammars [72] have found roles in natural language understanding tasks, from textual entailment [33] to paraphrase discovery [80].

Nonetheless, models of text based on the bag-of-words assumption have shown lasting value in natural language processing and related fields due to a perhaps not-too-surprising fact. *Which* words are in a text is highly representative of *what* the discourse is about, even if the structure of the discourse is destroyed. For example, a text about Shakespeare will use the bard's name as well as words like poem, iambic, and comedy far more often than will articles about the structure of DNA or professional sports. As a result, models based on the BoW assumption can be effective at retrieving [131, 120], discovering [90], classifying [77], and clustering [125] documents by content area. In general, the BoW assumption works poorly for fine-grained analysis of linguistic content within documents, but is surprisingly effective at describing the large-scale organizational structure of a document collection. The large-scale corpus analysis questions that this dissertation addresses are instances of the latter.

In the remainder of this chapter, I introduce the concepts and notation for two popular classes of models that often make the BoW assumption: vector space models and statistical models of text. Vector space models (Section 2.1) almost always start with the BoW assumption and are widely used as feature vectors for information retrieval and machine learning. Bayesian statistical models of text do not always make the BoW assumption, but two popular models do: the naive Bayes classifier (Section 2.2.1) and statistical topic models (Section 2.2.2). The VSM, naive Bayes, and statistical topic models will all be revisited in later chapters.

2.1 Vector space models of text

Despite the limitations of the BoW assumption, the Vector Space Model (VSM) of word and document meaning is a widely used technology and a core component of modern web search engines. The VSM was introduced by Salton and McGill for use in the System for the Mechanical Analysis and Retrieval of Text (SMART) [120], a pioneering Information Retrieval system built in the 1960s. The intended use of the VSM was as a basis for measuring the similarity of document pairs and of query-document pairs, where queries are treated as short documents. The intuition is simple: if we

count the number of occurrences of each term into its own dimension of a vector—and then weight these dimensions appropriately—we can quantify the similarity of document pairs by the similarity of their vectors. Several techniques exist to efficiently compute vector similarities, and many of these are quite effective when applied to document vectors. In essence, the VSM transforms the challenge of language similarity into straightforward vector algebra. Document retrieval, SMART's original goal, is achieved by returning the closest documents to a target query in the vector space.

The VSM's influence is not limited to information retrieval: it has proven to be a powerful representation for text clustering and classification. In text clustering, the goal is to automatically discover groups of related documents based on their similarities. The discovered clusters can illustrate high-level structure in a document collection. In text classification, the goal is more targeted: predict a label for a given document, such as *spam* versus *non-spam* email messages. The prediction is based on training examples where the class label is known. In both cases, algorithms that use VSM vectors as features are strong baselines that can outperform more complex models. I will return to text clustering and classification in Chapters 3 and 4, respectively.

One class of clustering algorithms warrants specific mention: those based on dimensionality reduction. Once we represent a document as a vector in a vector space, it is natural to represent a collection of documents as a matrix $D \in \mathbb{R}^{N \times V}$ where N is the number of documents and V is the size of the vocabulary. The entry $D_{d,v}$ is a weighted count of the number of times term v occurs in document d. (Counts in the VSM are often re-weighted, such as by down-weighting terms common in many documents via the tf-idf [120] weighting scheme, which we will return to in 3.2.1.) We can reduce the dimensionality of this matrix by making use of the singular value decomposition (SVD), a standard technique in linear algebra [45] first developed in the 1960s. When applied to the term-document matrix, it is known as Latent Semantic Indexing (LSI) or Latent Semantic Analysis (LSA) [37] because of its ability to derive latent usages of words that tend to co-occur in many documents and to group documents based on the usage of these word clusters.

LSA represents the contents of D as the product of three matrices $D \approx U\Sigma V^T$:

the document-topic matrix $U \in \mathbb{R}^{N \times K}$, a diagonal weight matrix $\Sigma \in \mathbb{R}^{K \times K}$, and a topic-term matrix $V \in \mathbb{R}^{V \times K}$. U represents how much each document uses each topic K and V represents how much each topic uses each word. Often, the reconstruction of D as the product of these matrices is very close to the original D when using only a small number of topics K.[1] For example, a document-specific mixture of some 300 topics may be a reasonable summary of a much larger term space with a size in the tens or hundreds of thousands.

The singular value decomposition assumes that the values in the input matrix are normally distributed [85, p. 565]. Counts in the term-document matrix most definitely are not—even after common re-weighting schemes like tf-idf—resulting in a mismatch of the mathematical technique of SVD with its linguistic application LSI. An implication of this assumption is that elements of the returned decomposition may be negative: a document may assign negative counts to some topics, and topics may assign negative counts to some words. In practice, this often occurs and leads to interpretation difficulties. Later, probabilistic Latent Semantic Indexing (pLSI) [60] was introduced to restrict the document-topic matrix and topic-term matrix to non-negative values that sum to one: i.e., to probability distributions over topics for each document and over words for each topic. pLSI is effectively an application of non-negative matrix factorization [76] to the term-document matrix. I will return to pLSI in Section 2.2.2, where I describe probabilistic topic models that do make explicit generative distributional assumptions about the nature of the input data, in contrast to pLSI. Chapters 3 and 4 revisit the VSM in more detail.

2.2 Statistical Models of Text

The influence of the bag-of-words assumption is not limited to the VSM. Many statistical models of text take the assumption literally as a statement of conditional independences among random variables. One large class of such models are Bayesian

[1] This is accomplished by setting smaller singular values in Σ to 0. Technically, this is known as "reduced SVD"—see [129, p. 17].

graphical models that represent the words in a document collection as observed random variables $\vec{w}$ drawn from some distribution(s) represented by unobserved random variables and parameters.[2] In these models, the bag of words assumption can be concretely instantiated to mean that the probability of a given word $w_{d,i}$ at position i in document d is independent of the other words in the document given some parameters, such as which topics the document participates in. Common notation describing variables in statistical text models as included in this dissertation are shown in Table 2.1.

In contrast to the VSM, statistical text modeling techniques ask us not to think of words as simply data for counting or manipulation, but rather as *evidence*. The words in the document collection are the observable output of some random process whose general form we assume but whose parameters we do not know. The words we observe suggest the values of these parameters. Formally, we can estimate the model's parameters by picking the values that maximize the likelihood of the observed words under our model.

Thinking of words as evidence instead of data opens up a wide range of well founded probabilistic modeling approaches. One particularly large class of statistical text models are those that make a *generative process assumption*. These models describe the origin of the observed words with a simple narrative: we first instantiate corpus-wide general random variables (such as the likelihood of spam vs non-spam), then pick values for more specific random variables (such as the particular distribution over words for spam and non-spam), and then finally generate the word variables from these more general random variables (select words from the chosen spam or non-spam distribution). All generative models are simplistic approximations of the real process by which documents are generated—a human author's efforts—but even surprisingly naive assumptions can effectively address some applications. I will describe two such

[2]The difference between unobserved random variables and model parameters is often subtle: by convention, if an unknown value has a prior and shares a structural position with observed random variables, we call it an unobserved random variable; else we call it a parameter. In general, we know the value of neither in advance. In many specific models such as LDA, the distinction is more relevant in that we set the values of parameters explicitly (or tune them with either held out data or a maximum likelihood technique) but we must necessarily infer or integrate out the values of hidden random variables as part of learning and inference.

$\mathbb{V}$	Vocabulary indexed by $v \in 1 \dots V$
$\mathbb{D}$	Documents indexed by $d \in 1 \dots D$
N_d	Length of document d
$w_{d,i}$	Word in $\mathbb{V}$ at word position $i \in 1 \dots N_d$

Table 2.1: Common notation for graphical models of text.

cases in the sections that follow: the Naive Bayes text classifier in Section 2.2.1, analogous to algorithms like logistic regression that can classify documents in the VSM, followed by Latent Dirichlet Allocation in Section 2.2.2, a probabilistic topic model that adds extensible probabilistic semantics to LSA/LSI.

2.2.1 Naive Bayes

The simplest widely used generative process for text classification is the multinomial naive Bayes event model [87]. Naive Bayes has been extremely influential in spam email detection since 2002 [47] and is still used in several popular desktop email applications as of 2011. One of the main reasons for the model's continuing popularity is that it is an effective [94], tweakable [121] text classifier. And, in its most basic form, naive Bayes serves as a favorite baseline for more powerful machine learning models in classification papers.

Naive Bayes assumes that each document is generated by some label l from a space of labels $\mathbb{L}$ of size L (e.g. $\{$*spam*, *non-spam*$\}$ with size 2). The labels are not necessarily equally likely, so the probability of picking a label l is assumed to come from a multinomial probability distribution π over labels. Each label is associated with a multinomial distribution β_l over words in the vocabulary $\mathbb{V}$. Like in the VSM, these multinomial distributions can be represented numerically as a vector whose length is the size of the vocabulary. However, unlike the VSM, the elements of β_l are constrained to be non-negative and sum to one. A document d is generated by first picking a label l from π (and a document length N_d) and then drawing N_d words from β_l.

Term draws in Naive Bayes are repeated without respect to ordering and so are a concrete manifestation of the BoW assumption. In particular, the probability of any

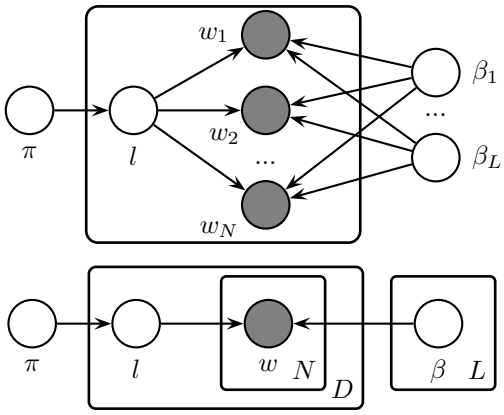

Figure 2.1: The multinomial naive Bayes event model represented as a standard Bayesian graphical model for a single document (top) and as a plate diagram for D documents (bottom). Plates (rounded boxes) represent repetition of indexed variables.

$\mathbb{L}$	Labels indexed by $l \in 1 \ldots L$
l_d	Class label for document d
π	Prior distribution over labels $\mathbb{L}$
β_l	Per-label distribution over words $\mathbb{V}$

Table 2.2: Summary of variables used in Naive Bayes, in addition to those in Table 2.1.

two words occurring in a given document are conditionally independent given their class label. Formally, the conditional independence assumption allows the probability of the observed word sequence $P(\vec{w_d}|l_d, \vec{\beta}) = P(\vec{w_d}|\beta_{l_d}) = P(w_{d,1}, \ldots, w_{d,N_d}|\beta_{l_d})$ to be factorized as simply $\prod_{i=1}^{N_d} P(w_{d,i}|\beta_{l_d})$. Using Bayes rules to incorporate the probability of picking our particular label l_d results in a final probability of a document's observed words as simply $P(l_d|\pi) \cdot P(\vec{w}_d|l_d, \vec{\beta}) = \pi_l \cdot \prod_{i=1}^{N_d} \beta_{l,w_{d,i}}$.

During training, the values of π and $\vec{\beta}$ are estimated to maximize this likelihood. During testing, the value of l_d that maximizes the likelihood of an individual document is the label that is chosen to describe that document. In practice, the optimal estimates for π and $\vec{\beta}$ can be evaluated by simply counting the fraction of documents with each label (for π) and the fraction of words within each label (for $\vec{\beta}$). A derivation of these rules can be found in [87].

Figure 2.1 shows the Bayesian graphical model representation of the Naive Bayes

generative story described above, with notation in Table 2.2. In the top half, relationship between the variables used to generate a particular document d (inside the box) is shown with respect to the model parameters (outside the box). Each word is assigned its own random variable $w_1 \dots w_n$ which are shaded to indicate that these variables are observed. The document's label l is considered observed during training but unobserved during classification. Each w is dependent on β_l and hence depends on both the value of l and the value of the $\beta_{1..L}$ variables. In the bottom half, multiple instantiations of the same variable type are collapsed using *plate notation*: the contents of each box are repeated by the number of times written in the bottom-right corner of the plate. In practice, additional hyperparameters are often included on the values of π and $\beta_{1..L}$ to allow the model to better fit the data [87].

2.2.2 Latent Dirichlet Allocation

While the VSM and Naive Bayes have been around since the 1960s, it wasn't until 2003 that a fully generative account for modeling text content with *unsupervised* topics was presented in the form of Latent Dirichlet Allocation (LDA) [16]. LDA is a generative model of text that is based on the BoW assumption and an event model similar to that of the multinomial naive Bayes classifier. However, LDA is an unsupervised algorithm that does not assume the presence of any labels. Instead, LDA assumes the presence of K latent topics, each of which is associated with a multinomial distribution over words β_k. Each document has its own mixture of topics θ_d, a document-specific multinomial over topics drawn from a Dirichlet prior α. Each word $w_{d,i}$ in the document is generated by first selecting a topic $z_{d,i}$ from θ_d and then a word from $\beta_{z_{d,i}}$. Because the topics are latent—only the values of the words $w_{d,i}$ are observed—practical difficulties in working out learning and inference in the model contributed to the long gap between LDA and the earlier generation of supervised models like naive Bayes.

The development of LDA can be traced through pLSI to LSA[60]. Like these earlier dimensionality reduction techniques, LDA learns how much each document likes each topic and how much each topic likes each word. LDA's major contribution

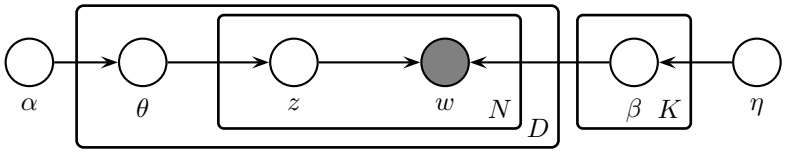

Figure 2.2: Bayesian graphical model for Latent Dirichlet Allocation.

is that, unlike earlier models, it provides fully generative probabilistic semantics to the generation of the corpus, and therefore opens itself up to extensions, customizations, and assumption modifications that are straightforward to express in the language of probabilistic graphical models. As a result, LDA has proven a fertile basis for the development of new models—e.g. [74, 14, 35, 63, 144, 95]—by a widely distributed community of researchers. The models presented in Chapters 3, 4, and 5 can be seen as part of this tradition.

Figure 2.2 shows the Bayesian graphical model for LDA and Table 2.3 makes its generative process explicit with variables described in Table 2.4. Unlike Naive Bayes, inferring the best possible values of the hidden parameters θ and β is computationally intractable. However, approximate inference techniques such as variational inference (as in [16]) and Gibbs sampling (as in [48]) are effective at estimating the values of θ and β from only the values of the observed words $\vec{w}$. Efficient [107], online [59], and distributed [5] inference for LDA has been studied explicitly. A friendly introduction to the mathematics of LDA can be found in [57], and a deeper exploration of the relationship between smoothing parameters and inference techniques in [4].

2.3 Summary

The BoW assumption is, at its core, a simplifying assumption about the nature of language that enables efficient computation on textual data. However, the value of the assumption bears out in its usefulness. A wide variety of high performing models of text are based on the BoW assumption and succeed at tasks from spam classification to topic discovery. The reason for the success of these models is a simple observation about language: large-scale thematic patterns of word usage are often captured at

1. For each topic $k \in \{1, \ldots, K\}$:
   (a) Generate $\beta_k = (\beta_{k,1}, \ldots, \beta_{k,V})^T \sim \text{Dir}(\cdot|\eta)$
2. For each document d:
   (a) Generate $\theta_d = (\theta_{d,1}, \ldots, \theta_{d,K})^T \sim \text{Dir}(\cdot|\alpha)$
   (b) For each i in $\{1 \ldots N_d\}$:
      i. Generate $z_i \in \{1 \ldots K\} \sim \text{Mult}(\cdot|\theta^{(d)})$
      ii. Generate $w_i \in \{1 \ldots V\} \sim \text{Mult}(\cdot|\beta_{z_i})$

Table 2.3: Generative process for Latent Dirichlet Allocation.

$\mathbb{K}$	Set of hidden topics indexed by $k \in 1..K$
θ_d	Per-document distribution over topics $\mathbb{K}$
β_k	Per-topic distribution over vocabulary $\mathbb{V}$
α	Dirichlet hyperparameter for $\theta_{1...D}$
η	Dirichlet hyperparameter for $\beta_{1...K}$
$z_{d,i}$	Latent topic assignment at word position i in document d

Table 2.4: Summary of variables used in LDA in addition to those in Table 2.1.

least as well by *word choice* as by sentence structure. As the BoW assumption is relaxed in various ways—from n-gram language models to syntactic parsing—the recovered knowledge can be made more fine-grained for tasks that require structure, like machine translation or entity extraction. Yet thematic organization is a reasonable fit for the BoW's assumptions, and many interesting challenges remain in thematic scope. This dissertation explores some of them.

From a modeling perspective, this dissertation builds upon and unifies topic models like LDA as well as supervised classification based on multinomial naive Bayes. In Chapter 3, I introduce a simple extension of LDA that enables effective simultaneous modeling of tags and words. In Chapter 4, I introduce Labeled LDA, which extends naive Bayes to multi-label classification by borrowing probabilistic machinery from LDA. Then, Chapter 5 introduces Partially Labeled Dirichlet Allocation, which unifies Labeled LDA and LDA into a coherent generative model of multi-labeled text that can simultaneously uncover latent topics associated with each document and latent word distributions associated with each label. These models are applied to challenges in mining and understanding the contents of large-scale real-world text datasets in Chapters 6 and 7.

Chapter 3

Clustering the Tagged Web

The web's content covers every niche of human interest. If we are to understand the structure and dynamics of the web, we need better tools for discovering high-level structure and patterns across multiple web pages. Automatic document clustering is one such mechanism: its goal is to find coherent groupings of web pages based on the words of those web pages and related signals. An effective web document clustering can, for example, tell us that pages around a particular set of domains are skewed toward a particular set of interests, what terms web authors use to describe those interests, and how both may have changed over time.

This chapter considers the task of web page clustering in the presence of tags. Tags are open-domain labels that human readers apply to web pages on social bookmarking websites. Sites like Delicious and StumbleUpon collected hundreds of thousands of keyword annotations per day (in 2008 [58]), and many of the highest quality pages are quickly tagged many times by many users. These tags are an explicit set of keywords users have found appropriate for categorizing documents within their own filing systems. Thus, tags promise to expose the domain knowledge embodied in each user's personal indexing vocabulary.

While the larger focus of this dissertation is on tools for exploration and discovery, it is worth noting that web document clustering is an interesting task in its own

This chapter draws from group work published as "Clustering the Tagged Web" in WSDM 2009 by D. Ramage, P. Heymann, C.D. Manning, and H. Garcia-Molina. [113]

right. Indeed, it has shown promise for improving several aspects of the standard information retrieval paradigm. Clustering has long been recognized as having the potential to improve search results in document retrieval [131, 133, 56] via document retrieval using topic-driven language models [82, 136], search result clustering [142]; alternative cluster-driven user interfaces [32]; and improved information presentation for browsing [90]. Others have argued that tags hold promise for ranked retrieval, [8, 62, 139, 143]. We do not explore these applications here, but rather focus on how tags can be used to improve the quality of learned clusters across a variety of models and conditions.

In more detail, we focus in on how best to exploit user-generated tags as a complementary data source to page text and anchor text for improving automatic clustering of web pages. We explore the use of tags in 1) K-means clustering in an extended vector space model that includes tags as well as page text and 2) a generative clustering algorithm, Multi-Multinomial Latent Dirichlet Allocation (MM-LDA) that jointly models text and tags. MM-LDA is an illustration of the first of three properties for successful text mining models identified in Chapter 1: trustworthiness. MM-LDA simultaneously models the words on a web page, the anchor text surrounding links to that page elsewhere on the web, and tags applied to the page on Delicious. Proper incorporation of these additional inputs improves the model's ability to discover topics that align with human similarity judgments across a variety of conditions versus both LDA and k-means. Specifically, we evaluate K-means, LDA, and MM-LDA by comparing their output to an established web directory, finding that the naive inclusion of tagging data improves cluster quality versus page text alone, but a more principled inclusion can substantially improve the quality of all models with a statistically significant absolute F-score increase of 4%. The generative model outperforms K-means with another 8% F-score increase. Improvements are found even several levels deep into the web directory hierarchy, demonstrating how tags can improve model trustworthiness in a variety of conditions.

3.1 Problem Statement

Our goal is to determine how tagging data can best be used to improve web document clustering. However, clustering algorithms are difficult to evaluate. Manual evaluations of cluster quality are time consuming and usually not well suited for comparing across many different algorithms or settings [53]. Several previous studies instead use an automated evaluation metric based on comparing an algorithm's output with a hierarchical web directory [125, 102]. Such evaluations are driven by the intuition that web directories, by their construction, embody a "consensus clustering" agreed upon by many people as a coherent grouping of web documents. Hence, better clusters are generated by algorithms whose output more closely agrees with a web directory. Here, we utilize a web directory as a gold standard so that we can draw quantitative conclusions about how to best incorporate tagging data in an automatic web clustering system.

We define the *web document clustering task* as follows:

1. Given a set of documents with both words and tags (defined in Section 3.1.4), partition the documents into groups (*clusters*) using a candidate *clustering algorithm* (defined in Section 3.1.1).

2. Create a gold standard (defined in Section 3.1.2) to compare against by utilizing a web directory.

3. Compare the groups produced by the clustering algorithm to the gold standard groups in the web directory, using an evaluation metric (defined in Section 3.1.3).

This setup gives us scores according to our evaluation metric that allow us to compare candidate clustering algorithms. We do not assert that the gold standard is the best way to organize the web—indeed there are many relevant groupings in a social bookmarking website necessarily lost in any coarser clustering. However, we argue that the algorithm which is best at the *web document clustering task* is the best algorithm for incorporating tagging data for clustering.

3.1.1 Clustering Algorithm

A web document clustering algorithm partitions a set of web documents into groups of similar documents. We call the groups of similar documents *clusters*. In this chapter, we look at a series of clustering algorithms, each of which has the following input and output:

Input A target number of clusters K, and a set of documents numbered $1, \ldots, D$. Each document consists of a bag of words from a word vocabulary W and a bag of tags from a tag vocabulary T.

Output An assignment of documents to clusters. The assignment is represented as a mapping from each document to a particular cluster $z \in 1, \ldots, K$.

This setup is similar to a standard document clustering task, except each document has tags as well as words.

Two notable decisions are implicit in our clustering algorithm definition. First, many clustering algorithms make *soft* rather than *hard* assignments. With hard assignments, every document is a member of one and only one cluster. Soft assignments allow for degrees of membership and membership in multiple clusters. For algorithms that output soft assignments, we map the soft assignments to hard assignments by selecting the single most likely cluster for that document. Secondly, our output is a flat set of clusters. In this chapter, we focus on flat (non-hierarchical) clustering algorithms rather than hierarchical clustering algorithms. The former tend to be $O(kn)$ while the latter tend to be $O(n^2)$ or $O(n^3)$ (see Zamir and Etzioni [141] for a broader discussion in the context of the web). Since our goal is to scale to huge document collections, we focus on flat clustering.

In our experiments, we look at two broad families of clustering algorithms. The first family is based on the vector space model (VSM), and specifically the K-means algorithm. K-means has the advantage of being simple to understand, efficient, and standard. The second family is based on a probabilistic model, and specifically derived from LDA. LDA-derived models have the potential to better model the data, though they may be more complicated to implement and slower (though not asymptotically).

3.1.2 Gold standard: Open Directory Project

We derive gold standard clusters from the Open Directory Project (ODP) [1]. ODP is a free, user-maintained hierarchical web directory. Each node in the ODP hierarchy has a label (e.g., "Arts" or "Python") and a set of associated documents.[1] To derive a gold standard clustering from ODP, we first choose a node in the hierarchy: the root node (the default for our experiments), or "Programming Languages" and "Society" (for Section 3.4.2). We then treat each child and its descendants as a cluster. For example, say two children of the root node are "Arts" and "Business." Two of our clusters would then correspond to all documents associated with the "Arts" node and its descendants and all documents associated with the "Business" node and its descendants, respectively.

A gold standard clustering using ODP is thus defined by a particular node's K' children. When we give the clustering algorithm a value K, this is equal to the K' children of the selected node. In general, the best performing value of K will not be K'. This heuristic is adopted to simplify the parameter space and could be replaced by one of several means of parameter selection, including cross-validation on a development set. We sometimes use the labels in the hierarchy to refer to a cluster, but these labels are not used by the algorithms. When referring to the clusters derived from the gold standard, we will sometimes call these clusters *classes* rather than *clusters*. This is in order to help differentiate clusters generated by a candidate clustering algorithm and the clusters derived from the gold standard. It is also worth noting that the algorithms we consider are unsupervised and are therefore applicable to any collection of tagged documents (as opposed to documents which conform to the categories in ODP).

3.1.3 Cluster-F1 evaluation metric

We chose to compare the generated clusters with the clustering derived from ODP by using the F1 cluster evaluation measure [84]. Like the traditional F1 score in

[1] Documents can be associated with multiple nodes in the hierarchy, but this happens very rarely in our data. When we have to choose whether a document is attached to one node or another, we break ties randomly.

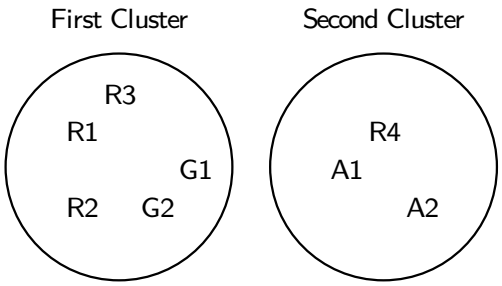

Figure 3.1: An example of clustering.

classification evaluation, the F1 cluster evaluation measure is the harmonic mean of precision and recall, where precision and recall here are computed over pairs of documents for which two label assignments either agree or disagree.

3.1.3.1 Example

Consider the example clustering shown in Figure 3.1. Two clusters are shown, and each document is denoted by its class in ODP: A for "Arts," G for "Games," R for "Recreation." A2 (for example) denotes a document which is in the ODP class "Arts" that the clustering algorithm has decided is in the second cluster.

We think of pairs of documents as being either the same class or differing classes (according to our gold standard, ODP), and we think of the clustering algorithm as predicting whether any given pair has the same or differing cluster. The clustering in Figure 3.1 has predicted that $(A1, A2) \rightarrow$ *same cluster* and that $(R2, R4) \rightarrow$ *different cluster*. If we enumerate all of the $\binom{n}{2} = 28$ pairs of documents in Figure 3.1, we get four cases:

True Positives (TP) The clustering algorithm placed the two documents in the pair into the same cluster, and our gold standard (ODP) has them in the same class. For example, $(R1, R3)$. There are 5 true positives.

False Positives (FP) The clustering algorithm placed the two documents in the pair into the same cluster, but our gold standard (ODP) has them in differing classes. For example, $(R1, G2)$. There are 8 false positives.

True Negatives (TN) The clustering algorithm placed the two documents in the pair into differing clusters, and our gold standard (ODP) has them in differing classes. For example, $(R2, A1)$. There are 12 true negatives.

False Negatives (FN) The clustering algorithm placed the two documents in the pair into differing clusters, and our gold standard (ODP) has them in the same class. For example, $(R2, R4)$. There are 3 false negatives.

We then calculate precision as $\frac{TP}{TP+FP} = \frac{5}{13}$, calculate recall as $\frac{TP}{TP+FN} = \frac{5}{8}$, precision $= \frac{TP}{TP+FP} = \frac{5}{13}$ recall $= \frac{TP}{TP+FN} = \frac{5}{8}$ and F1 as: $\frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}} \approx 0.476$.

3.1.3.2 Notes on F1

We selected F1 because it is widely understood and balances the need to place similar documents together while keeping dissimilar documents apart. We experimented with several other cluster evaluation metrics, including the Rand index [115], and information theoretic measures such as normalized mutual information [124] and variation of information [92], finding the results to be consistent across measures.

F1 is a robust metric appropriate for our choice to provide the value K to our clustering algorithms (see Section 3.1.1). In particular, having the number of clusters K' in the gold-standard as input K does not ease the task of placing similar documents together while keeping dissimilar documents apart. Indeed, there may be many small, specific groupings of the top-level ODP categories—more than the 16 top-level subcategories—which a clustering algorithm would be forced to conflate. These conflations come at the expense of introducing false positives, possibly lowering the F1 score.

Because the clustering algorithms we consider are randomized, their output can vary between runs. To assign a stable F1 score to a particular algorithm, we report the mean F1 score across 10 runs of the algorithm with identical parameters but varying random initialization. In our experiments, we report statistical significance where appropriate. When we refer to a change in F1 score as significant, we mean that the variation between the underlying runs for two algorithms is significant at the 5% level by a two-sample t-test.

3.1.4 Dataset

Our tagged document collection is a subset of the Stanford Tag Crawl Dataset [58]. The Tag Crawl consists of one contiguous month of the *recent* feed on Delicious, a popular social bookmarking website, collected starting May 25th 2007. Each post on the recent feed is the result of a user associating a URL with one or more short text strings, such as *web* or *recipe*. Aggregating across posts, we recovered a dataset of 2,549,282 unique URLs. For many URLs, the dataset also includes a crawl of the page text and backlink page text.

To evaluate the quality of clusterings of the Tag Crawl dataset, we limited consideration to only a subset of 62,406 documents that is also present in ODP. Because these pages were all tagged by a user within the last year, they include some of the most recent and relevant pages in the directory. We discarded URLs in ODP's top-level "Regional" category, as its organizational structure is largely based on the geographical region pertaining to the site. Of the remaining documents, only 15,230 were in English and had their page text crawled as part of the Tag Crawl dataset. The documents are distributed as in Table 3.1. The documents were further divided into a 2,000 document development set for parameter tuning and a 13,230 document test set for evaluating the final configurations reported here.

In our discussion, we differentiate between *types* and *tokens*. A word or tag *token* is an instance of a term being observed either in or annotated to a document, respectively. A word or tag *type* is a single unique term that is observed or annotated to at least one document in the collection, respectively. For example, a document with the text "the fuzzy dog pet the other fuzzy dog" and the tags ("dog", "fuzzy", "fuzzy") has eight word tokens, five word types, two tag types and three tag tokens.

Each document in the intersection of Delicious and ODP is represented as two sets of term occurrence counts—one for words and another for tags. Words were extracted from the Tag Crawl dataset and were tokenized with the Stanford Penn Treebank tokenizer, a fairly sophisticated finite state tokenizer. During processing, all word tokens appearing less frequently than the 10 millionth most common distinct word type were dropped as a first-cut term selection criterion [81, 140] as well as for reasons of computational efficiency. On average, a document contains 425 distinct

ODP Name	#Docs	Top Tags by PMI
Adult	36	blog illustration art erotica sex
Arts	1446	lost recipes knitting music art
Business	908	accounting business lockpicking agency
Computers	5361	web css tools software programming
Games	291	un rpg fallout game games
Health	434	parenting medicine healthcare medical
Home	654	recipes blog cooking coffee food
Kids	669	illusions anatomy kids illusion copyright
News	373	system-unfiled daily cnn media news
Recreation	411	humor vacation hotels reviews travel
Reference	1325	education reference time research dictionary
Science	1574	space dreams psychology astronomy science
Shopping	310	custom ecommerce shop t-shirts shopping
Society	1852	buddhism christian politics religion bible
Sports	146	sport cycling nfl football sports
World	756	speed bandwidth google speedtest maps

Table 3.1: Intersection of ODP with the Stanford 2007 Tag Crawl dataset. The "regional" category has been elided.

word types and 1,218 word tokens. The tag occurrence counts make up the other data of each document. The complete set of tags was crawled from Delicious for each document without additional processing, yielding an average of 131 distinct tag types and 1,307 tag tokens out of a tag vocabulary of 484,499 unique tags (including many non-English tags). Because these documents in the ODP intersection tend to be generally useful websites, they tend to be more heavily tagged than most URLs in Delicious [58].

3.2 K-means for words and tags

In this section, we examine how tagging data can be exploited by the K-means [84] algorithm, a simple to implement and highly scalable clustering algorithm that assumes the same vector space model as traditional ranked retrieval. K-means clusters documents into one of K groups by iteratively re-assigning each document to its nearest cluster. The distance of a document to a cluster is defined as the distance

of that document to the centroid of the documents currently assigned to that cluster [84]. Distance is the cosine distance implied by the standard vector space model: all documents are vectors in a real-valued space whose dimensionality is the size of the vocabulary and where the sum of the squares of each document vector's elements is equal to 1. Our implementation initializes each cluster with 10 randomly chosen documents in the collection.

A key question in clustering tagged web documents using K-means is how to model the documents in the VSM. We examine five ways to model a document with a bag of words B_w and a bag of tags B_t as a vector V:

Words Only In step one, V is defined as $\langle w_1, w_2, \ldots w_{|W|} \rangle$ where w_j is the weight assigned to word j (based on some function f_w of the frequency of words in W and/or B_w). For example, w_j can be the number of times word j occurs in B_w (term frequency or tf weighting). In step two, V is l_2-normalized so that $||V||_2 = 1$.

Tags Only Analogous to words only, except we use the bag of tags B_t rather than the bag of words B_w and the tag vocabulary T rather than the word vocabulary W in step one.

Words + Tags If we define V_w to be the words only vector, above, and V_t to be the tags only vector, above, then the *Words+Tags* vector $V_{w+t} = \langle \sqrt{\frac{1}{2}} V_w, \sqrt{\frac{1}{2}} V_t \rangle$. In other words, we concatenate the two l_2-normalized vectors, giving words and tags equal weight. The intuition underlying this choice is that tags provide an alternative information channel that can and should be counted separately and weighted independently of any word observations.

Tags as Words Times n Analogous to words only, except in step one, instead of B_w we use $B_w \cup (B_t \times n)$. In other words, we combine the two bags, but we treat each term in the tag bag B_t as n terms. Instead of W we use $W \cup T$ as our vocabulary. For example, a document that has the word "computer" once and the tag "computer" twice would be represented as the word "computer" three times under the *Tags as Words Times 1* model, and five times under the *Tags*

as Words Times 2 model. This representation is sometimes used for titles in text categorization [31].

Tags as New Words We treat tags simply as additional (different) words. V is defined as: $\langle w_1, w_2, \ldots w_{|W|}, w_{|W|+1}, w_{|W|+2}, \ldots w_{|W|+|T|} \rangle$ where w_j is the weight assigned to word j for $j \leq |W|$ or the weight assigned to tag $j-|W|$ for $j > |W|$. This is equivalent to pretending that all words are of the form *word#computer* and all tags are words of the form *tag#computer*. Then V is l_2-normalized.

These options do not cover the entire space of possibilities. However, we believe they represent the most likely and common scenarios, and give an indication of what representations are most useful. Nonetheless, it should be noted that one could optimize the relative weight given to words versus tags to maximize per-task performance.

In addition to deciding to model words or tags or both, we also need to answer the following questions:

1. How should the weights be assigned? Should more popular tags be weighted less strongly than rare tags? (Discussed in Section 3.2.1.)

2. How should we combine the words and tags of a document in the vector space model? Which of the vector representations presented above is most appropriate? (Discussed in Section 3.2.2.)

3. In the VSM, do tags help in clustering? (Discussed in Sections 3.2.1 and 3.2.2.)

3.2.1 Term weighting in the VSM

In this subsection, we study the first question above: how should the weights be assigned? We study this question for the first three document models (*Words Only*, *Tags Only*, and *Words+Tags*). In particular, we consider two common weighting functions: raw term frequency (tf) and tf-idf. In computing term frequency, each dimension of the vector is set in proportion to the number of occurrences of the corresponding term (a word or tag) within the document. For tf-idf, each dimension is the term frequency down-weighted by the log of the ratio of the total number of

	tf	tf-idf
Words	.131	.152
Tags	.201	.154
Words+Tags	.209	.168

Table 3.2: F-scores of the vector space model with pre-normalization on the 2000 document development collection (higher is better). Rows correspond to features given to the K-means model and columns present the weighting normalization function used.

documents to the number of documents containing that term. For the *Words+Tags* scheme, we did not bias the weights in favor of words or tags (we normalized the combined vector with no preference towards either words or tags).

Table 3.2 demonstrates the impact of tf versus tf-idf weighting on the K-means F1 score for 2,000 documents set aside for this analysis. Note that K-means on *Words+Tags* significantly outperforms K-means on words alone under both term frequency and tf-idf. And the best performing model—term frequency weighting on *Words+Tags*—significantly outperforms tf-idf weighting on *Words+Tags*. However, the performance difference of term frequency on both *Words+Tags* does not significantly outperform the clustering on tags alone. As in the analysis of Haveliwala et al., [53], we believe that tf-idf weighting performs poorly in this task because it over-emphasizes the rarest terms, which tend not to be shared by enough documents to enable meaningful cluster reconstruction.

The results of this initial experiment suggest that term frequency weighting is an effective and simple means of assigning weights in our document vectors. We next address the more fundamental modeling questions of how to combine words and tags, using term frequency to assign weights to each vector element.

3.2.2 Combining words and tags in the VSM

Which of the five ways to model a document presented at the beginning of this section work best in the VSM? Table 3.3 shows the averaged results of ten runs of our best weighting (tf weighting) on the 13,230 documents not used for selecting the term weighting scheme. The *Words* and *Words+Tags* score are similar to the numbers in

	K-means
Words	.139
Tags as Words ×1	.158
Tags as Words ×2	.176
Tags as New Words	.154
Words+Tags	.225

Table 3.3: F-scores for K-means clustering (tf) with several means of combining words and tags on the full test collection. All scores are averaged across 10 runs. All differences are significant except Tags as Words ×1 versus Tags as New Words.

Table 3.2—their difference reflects the change in dataset between the two experiments. The inclusion of tags as words improves every condition over baseline, but all are significantly outperformed by the *Words+Tags* model. This suggests convincingly that tags are a qualitatively different type of content than "just more words" as has been suggested recently [11]. By simply normalizing the tag dimensions independently from the word dimensions of the underlying document vectors, K-means can very effectively incorporate tagging data as an independent information channel.

3.3 Generative topic models

In the previous section, we saw the large impact to be had by appropriately including tagging information in K-means. This result affirms the notion *trustworthiness* presented in Chapter 1: models of text that ignore label information are disadvantaged when it comes to matching human judgment. In this section, we begin to develop a more refined latent variable model of text based on LDA—as discussed in Section 2.2.2 or [16]—that makes use of human-provided labels in the extensible probabilistic semantics provided by the statistical text modeling approach (Section 2.2). The clustering model we develop here has explicit probabilistic semantics appropriate for modeling the nature of words and tags as independent sets of observations, for the purpose of improving LDA's trustworthiness. We ask three questions about the LDA-derived model:

1. Can we do better than LDA by creating a model (defined in Sections 3.3.1 and

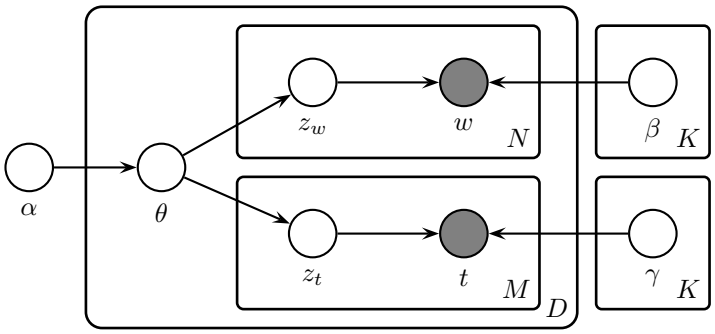

Figure 3.2: Graphical model of MM-LDA.

3.3.2) that explicitly accounts for tags and words as separate annotations of a document? (Discussed in Section 3.3.3.)

2. Do the same weighting and normalization choices from the VSM (Section 3.2) hold for generative models like LDA-derived models, or do they differ?[2] (Discussed in Section 3.3.3.)

3. Do LDA-derived models better describe the data and hence perform better on the tagged web document clustering task than clustering algorithms based on VSM? (Discussed in Section 3.3.4.)

3.3.1 MM-LDA Generative Model

In the context of tagging data, we extend LDA to jointly account for words and tags as distinct sets of observations. Our model takes its inspiration from a similar model for text and images proposed by Blei and Jordan [17]. We call our algorithm Multi-Multinomial LDA. The best way to describe MM-LDA is to outline the process it assumes has generated the dataset. We then maximize the likelihood of the data with respect to that process's parameters to reconstruct each document's cluster association probabilities as well as the probability of each word and tag per cluster. MM-LDA generates a collection of tagged documents from K topics by the process below and shown in Figure3.2:

[2]Note that term weights have no natural interpretation in a conventional LDA-derived model, so we only compare methods of combining tags and words.

1. For each topic $k \in 1 \dots K$, draw a multinomial distribution β_k of size $|W|$ from a symmetric Dirichlet distribution with parameter η_w. Each β_k represents the probability of seeing all word types given topic k.

2. Similarly, draw a multinomial γ_k of size $|T|$ from a symmetric Dirichlet with parameter η_t to represent the probability of seeing all tag types given topic k.

3. For each document $i \in 1 \dots D$ in the collection, draw a multinomial θ_i of size $|K|$ from a Dirichlet distribution with parameter α. Each θ_i represents the probability of a word in that document having been drawn from topic i.

4. For each word index $j \in 1 \dots N_i$ in document i:

   (a) Draw a topic $z_j \in 1 \dots K$ from θ_i.

   (b) Draw a word $w_j \in 1 \dots |W|$ from β_{z_j}.

5. For each tag index $j \in 1 \dots M_i$ in document i:

   (a) Draw a topic $z_j \in 1 \dots K$ from θ_i.

   (b) Draw a tag $t_j \in 1 \dots |T|$ from γ_{z_j}.

Steps one, three, and four, in isolation, are equivalent to standard LDA. In step two, we construct distributions of tags per topic analogously to the construction of the word distributions per topic. In the final step, we sample a topic for each tag in the same way sampling a topic for each word.

3.3.2 Learning MM-LDA Parameters

One of several approaches can be used to learn the parameters $\beta_k, \gamma_k, \theta_i$. Variational inference [17] and Gibbs sampling [48] are two general techniques that have been used to learn the analogous parameters in LDA. We chose to extend a Gibbs sampling algorithm like the one analyzed by Wei and Croft [136] because its running time is asymptotically competitive with that of K-means. The algorithm is as follows: we iterate repeatedly through the documents in random order. For each word (and then

for each tag) in random order, we resample a single topic z_j based on the current topic probabilities for that document and the probability of each proposed cluster assignment having generated the observed word. The Dirichlet prior parameters η and α effectively become pseudocount smoothing on the β and θ distributions, respectively, which we do not resample. This process repeats until convergence of the model's perplexity—a measure of its confusion on its input data—or earlier if a maximum of 100 iterations is reached. On the development set of 2000 documents, our LDA implementation runs in about 22 minutes whereas K-means runs in about 6 minutes. This 4:1 ratio holds up for the larger data sets as well.

We tested a wide range of smoothing parameters α, η_w, η_t for the MM-LDA model over 10 runs on the 2000 document validation set. We found that the model was fairly insensitive to the chosen values, except if the word or tag smoothing parameter was substantially smaller than the topic smoothing parameter (less than $\frac{2}{3}$ the other parameter). We chose 0.7 for the smoothing parameter for the word, tag, and topic distributions and used this value throughout.

3.3.3 Combining words and tags with MM-LDA

Does modeling words and tags separately improve performance in MM-LDA over a standard LDA model? Just as renormalizing the tag and word vector components separately improved K-means performance, the inclusion of tags as an alternative type of observation allows MM-LDA to flexibly model the tags and words that co-occur in the dataset. As an alternative, we could have employed a standard LDA model and added tags directly as words (*Tags as Words* $\times 1$); added them as words with multiplicity two (*Tags as Words* $\times 2$); or added them into an expanded region of the word feature space (*Tags as New Words*). By contrast, MM-LDA (*Tags+Words*) keeps distinct multinomial distributions for the occurrence of tags and words under a particular topic. Table 3.4 presents F-scores of LDA and MM-LDA under these model variations.

MM-LDA's *Words+Tags* model significantly outperforms all other configurations. Interestingly, the addition of tags to the word vectors decreases the performance of

	(MM-)LDA
Words	.260
Tags as Words×1	.213
Tags as Words ×2	.198
Tags as New Words	.216
Words+Tags	.307

Table 3.4: F-scores for (MM-)LDA across different tag feature modeling choices.

	(MM-)LDA	K-means
Words	.260	.139
Tags	.270	.219
Words+Tags	**.307**	.225

Table 3.5: F-scores for (MM-)LDA and K-means on 13,320 documents. Including tags improves both models significantly versus words alone. MM-LDA (bold) significantly outperforms all other conditions.

the algorithm relative to words alone. We believe this decrease is due in part to the very different distributional statistics observed for words versus tags. In particular, for our dataset there tend to be about 4 times as many word types as tag types and yet a similar number of tokens for each. When combined, the word multinomials for many topics may become disproportionately peaked around common tags at the expense of flexibility in modeling either.

3.3.4 Comparing K-Means and MM-LDA

How does the probabilistic model of MM-LDA perform compared to the VSM of K-means? In this section, we compare MM-LDA to K-means quantitatively and qualitatively.

3.3.4.1 Quantitative Comparison

We clustered documents using the K-means and LDA models on the 13,320 document test collection under three conditions: just *Words*, just *Tags*, or jointly *Words+Tags*.

Tag-Augmented K-means

	tags	*words*
1	linux security php opensource vpn unix	linux ircd php beware kernel exe
2	games go game sports firefox gaming	dmg munsey ballparks suppes racer game
3	music research finance audio mp3 lyrics	music research redirect nottingham meta
4	news business newspaper politics media	v business leadership d news j
5	politics activism travel movies law	aquaculture terrapass geothermal
6	science physics biology astronomy space	science wildman foraging collembola
7	css python javascript programming xml	squeakland sql coq css python flash
8	food recipes cooking shopping tea recipe	recipes food cooking recipe stylist tea
9	blog blogs fashion design art politics	flf blog comments posted my beuys
10	education art college university school	learning gsapp students education school
11	health medical healthcare medicine solar	health napkin cafepress.com medical care
12	java programming development compiler	java c programming goto code language
13	software windows opensource mac	software windows mac download os
14	dictionary reference language bible	dictionary english words syw dictionaries
15	internet dns search seo google web	internet shutdown sportsbook epra kbs
16	history library books literature libraries	library tarot peopling ursula guin

Multi-Multinomial LDA (MM-LDA)

	tags	*words*
1	web2.0 tools online editor photo office	icons uml powerpoint lucid dreams
2	guitar scanner chemistry military	grub outlook bittorrent rendering
3	health medical medicine healthcare	exe health openpkg okino dll polytrans
4	bible christian space astronomy religion	gaelic bible nt bone scottish english
5	politics activism environment copyright	war shall power prisoners their article
6	social community web2.0 humor fun funny	press f prompt messages ignoring each
7	reference science education research art	science research information university
8	java database programming development	java sql mysql schizophrenia testing test
9	dictionary language english reference	english writing dictionary spanish words
10	travel search maps google reference map	search deadline call flf conference paper
11	time clock timezones world train md5	quantum thu pfb am pm mf
12	food recipes cooking business shopping	my food tea wine me recipes
13	news blog music blogs technology system	comments blog he posted news pm
14	programming software webdesign web css	you can if or not use
15	photography photo compression zip	flash camera eos light e-ttl units
16	mac apple osx games unicode game	dmg u x mac b v

Table 3.6: Highest scoring tags and words from clusters generated by K-means (above) and MM-LDA (below) from one run of the 2000 document development set. The K-means terms are selected by top tf-idf and the MM-LDA terms are selected by highest interest value.

For K-means, we used tf weighting, which includes the best performing model for K-means, *Words+Tags*. Table3.5 shows that the inclusion of tagging data significantly improves the performance of MM-LDA versus tags or words alone. The improvement from moving from Words to Words+Tags was significant for both models. In contrast to K-means, LDA's improvement from *Tags* to *Words+Tags* was also significant. MM-LDA's *Words+Tags* model is significantly better than all other models. From this we conclude that, under some conditions, MM-LDA is better able to exploit the complementary information in the word and tag channels.

3.3.4.2 Qualitative Comparison

Qualitatively, both K-means and MM-LDA learn coherent clusters in the document collections, as demonstrated by the top scoring words and tags associated with each cluster in Table3.6. In addition to associating documents to topics, each algorithm outputs per-cluster affinities to words and to tags. When analyzing the generated affinities, it is important to take into account the underlying model assumed by each algorithm. K-means operates in document vector space, so we extract its top-scoring words and tags per cluster by selecting those terms with the highest tf-idf weighted score. By contrast, MM-LDA outputs multinomial probability distributions, which tend to be highly peaked and inappropriate for tf-idf weighting. For MM-LDA, we select a term t for cluster c if it has one of the highest values of *interest*, defined as be $p(t|c) - p(t)$. The interest operator balances the desire to select terms that have high absolute probability in their cluster with low probability overall.

3.4 Further studies

Lastly, we consider two questions independent of the clustering algorithm family that provide insight into the model's ability to align with surrogate human similarity scores across a variety of conditions. Specifically:

1. Does the addition of anchor text to regular plain text make tags redundant? Do our algorithms that take into account tags still outperform anchor text +

plain text together? (Discussed in Section 3.4.1.)

2. If we look at multiple levels of specificity of clusters, for example, clustering programming language documents rather than clustering general documents, does tagging data help? More or less? (Discussed in Section 3.4.2.)

3.4.1 Tags are different than anchor text

Do the advantages of tagging data hold up in the presence of anchor text? Anchor text — the text of and around incoming web hyperlinks — has helped in some tasks that use web document corpora like web search [40] and text classification [42]. Like tags, anchors act as free-form document annotations provided by a third party. For each URL in the tag crawl dataset, we extracted words within 15 tokens of hyperlinks to that URL in up to 60 pages returned by a Google API backlink query. This window size was consistent with the best results for anchor text window size for similarity search found in [53].

We experimented with two means of combining page text, anchor text, and tags. *Anchors as Words* adds all words in the extracted anchor text windows to each document's word vector analogously to the *Tags as Words* model in Section3.2. *Words+Anchors* weights anchor text words separately from the document words, like the *Words+Tags* model. The results of these model variants on the top-level ODP clustering task, as well as when Tags are added as an independent information channel to each of them, are presented in Table3.7.

We found that both MM-LDA and K-means gain from the inclusion of tagging data as compared to clustering on *Anchors as Words* or *Words+Anchors* alone. However, the results from the inclusion of anchor text are mixed. While performance of LDA improved when anchors were added as new words (*Anchors as Words*), K-means performance was slightly depressed because of the vector space model's sensitivity to the weights of the now-noisier terms. Neither model did well with *Anchors+Words*, reflecting the difficulty of extracting a high quality anchor text signal for text clustering, especially from a relatively small web crawl. We believe that these numbers might be improved by down-weighting anchor words as a function of their distance from the

	MM-LDA	K-means
Words	.260	.139
Anchors as Words	.270	.120
(Anchors as Words)+Tags	.281	.214
Words+Anchors	.248	.128
Words+Anchors+Tags	**.306**	.224

Table 3.7: Inclusion of tags in (MM-)LDA and K-means increases F1 score on the test collection even in the presence of anchor text.

URL or exploiting more advanced term weighting techniques as in [53]. However, even under such transformations, we argue that the inclusion of tagging data would still improve cluster quality.

3.4.2 Clustering more specific subtrees

Does the impact of tags depend on the specificity of the clustering? Clustering the top-level ODP subtrees is a difficult task because many coherent subtopics exist for each top-level ODP category. We believe real-world applications may benefit from clustering either a wide variety of documents, as in the top-level ODP clustering task, or documents that are focused, such as those returned by a search query.

To investigate the applicability of tag-based clustering for more specific document collections, we selected two representative ODP subtrees that each had a substantial number of documents in our dataset. The *Programming Languages* subcategory is the set of documents labeled with a subcategory of ODP's Top/Programming/Languages category. The gold-standard labels for this subset of 1,094 documents are: Java, PHP, Python, C++, JavaScript, Perl, Lisp, Ruby, and C. Documents in this subset tend to share many specific terms related to programming (e.g. words such as *loop*, and *compile*), so clustering this subcategory is not unlike clustering some types of search results.

The *Social Sciences* subcategory (SS) is the set of documents labeled with a subcategory of ODP's Top/Society tree. The 1,590 documents in this subset are each labeled as one of: Issues, Religion & Spirituality, People, Politics, History, Law, or Philosophy. This collection represents a diverse set of topics unified by a common

		(MM-)LDA	K-means
Programming Languages	Words	.288	.189
	Tags	.463	.567
	Words+Tags	.297	.556
Social Sciences	Words	.300	.196
	Tags	.310	.307
	Words+Tags	.302	.308

Table 3.8: F-scores for (MM-)LDA and K-means on two representative ODP subtrees. For these tasks, clustering on tags alone can outperform alternatives that use word information.

theme with many overlapping terms, but in a broader vocabulary space than the PL subset.

Both clustering algorithms performed at least as well in these ODP subsections as they did for the directory as a whole, as shown in Table 3.8. Tags appear to be better indicators than words in isolation, and, indeed they are so much better that jointly modeling tags and words can actually depress performance. This surprising phenomenon stems in part from the fact that users tend to tag pages at a level of specificity appropriate for their own information needs, which often correspond to the types of distinctions made within ODP subsections. For example, within the Java subcategory of "Programming Languages", the most common tag, "java," covers $488/660 = 73.9\%$ of pages. By contrast, in the top-level "Computers" subcategory, the most common tag "software" covers only $2562/11894 = 21.5\%$ of pages. Because the size of the tag vocabulary within these ODP subsections is substantially reduced from the full tag vocabulary, a higher proportion of the remaining tags are direct indicators of sub-category membership than in the top-level clustering task. We believe that the extra signal present in the words plays a lesser role and, indeed, can reduce the quality of the overall clustering. This factor applies to both models, even when K-means outperforms LDA, as on the Programming Languages cluster, where a smaller set of focused tags plays to the strengths of the vector space model's independence assumptions.

3.5 Related work

The impact of social bookmarking data has been explored in several other contexts within information retrieval and the web, including in ranked retrieval e.g., [8, 58, 62, 139] and analysis of blogs [23, 55]. Others have used tags in some clustering contexts, such as Begelman et al. [10] who conclude that clustering of tags should be used in tagging systems, for example, to find semantically related tags.

In modeling, the most closely related work to ours is Zhou et al.'s paper [143], which (like ours) looks at the potential to generatively model social annotation to improve information retrieval. That work's evaluation focuses on a specific, promising, application of improving language model based information retrieval. As a result, it produces evidence that good generative models for social annotation can in fact have a positive impact on ranked result quality for language model based information retrieval systems. Our work uses a more general evaluation metric, similarity to a gold standard (inspired by Haveliwala et al. [53]), and further assumes that search engines have access to anchor text. We believe our more general evaluation metric may make our results more applicable to the broader group of applications outlined in Section 3 while still making them convincingly applicable to language model based information retrieval, due to Zhou et al.'s work. Lastly, our MM-LDA generative model is more directly descended from Blei et al.'s work on annotation [17] than is the model in [143], which we hope makes our work more applicable to the popular current area of image retrieval with tags (see, for example, [6, 132, 116]).

A host of applications have grown out of the ability to classify web pages into web directories, including topic-sensitive web link analysis [54] and focused crawling [27]. Our work is related to this work in that we use ODP as a gold standard for our evaluation. However, it is different in that our goal is not to predict ODP classes (for which we might use a supervised method) or to create a hierarchy similar to ODP (for which we might use hierarchical clustering) but rather to improve information retrieval through clustering.

3.6 Discussion

Many of the newest and most-relevant parts of the web are constantly being tagged on large social bookmarking websites. In this work we have also found that many pages of interest are often those with the most tags. In fact, the pages that are informative and relevant enough to be in both the tag crawl dataset and in ODP have, on average, as many tag annotations as words. And because tagging happens more quickly than links for new content, tags promise to become an increasingly important signal for ranking new pages of high static quality. The baseline clustering algorithms extended in this work are themselves high-performers on traditional document clustering tasks. By exploiting tagging data when available, these techniques promise to improve web document clustering in general, and especially so for the most relevant parts of the web.

As a final note, it is perhaps worth contrasting modern tagging with other types of indexing vocabularies. The traditional comparison in the field has been between controlled indexing languages—characterized by a specific indexing vocabulary structured in advance—and full text indexing, in which the documents themselves provide all indexing terms. In some ways, tagging sits between these two extremes. As in a controlled indexing language, human beings select the terms in a tagging system that characterize a document well. Tags therefore have a level of semantic precision that full text indexing lacks. Yet in other ways, tagging is more like free text indexing in that there is no predefined vocabulary or hierarchy, tags can freely have multiple meanings, and different tags can be used for the same topic. Tagging is like free text indexing in some other important respects, as well: with ample tagging data, tags have frequency counts, just like words, and the range of tags applied to popular documents is more exhaustive than what is typical in a controlled vocabulary. In sum, we can at least hope that because tags represent human semantic classification, tagging has the potential to improve the precision of searches as well as the quality of inferred document clusters, while the exhaustiveness of tagging means that the technique will avoid the biggest limitation of traditional use of controlled indexing vocabularies.

3.7 Conclusion

This chapter has demonstrated that social tagging data provides a useful source of information for the general problem of web page clustering. We have shown that tagging data improves the performance of two automatic clustering algorithms when compared to clustering on page text alone. A simple modification to the widely used K-means algorithm enables it to better exploit the inclusion of tagging data. A novel algorithm—MM-LDA, an extension to LDA for use with parallel sets of observations—makes even better use of the complementary similarity information held in a document's words and tags on a general web clustering task. MM-LDA improves upon LDA's *trustworthiness*: incorporating human labels allows the model to discover clusters that better align with surrogate human similarity judgments.

Although we have shown in this chapter how human tagging data is a useful source of information that can be effectively exploited to improve a model's trustworthiness, we haven't yet demonstrated a way to exploit those annotations to make the discovered topics more interpretable to human beings. Indeed, the topics we see in Table 3.6 seem sensible, but our ability to interpret these topics faces the same limitations facing any latent topic model. In the next chapter, we move from models that simply use human-provided labels as an information channel to models that explicitly account for the alignment between words and labels, presenting an interpretability advantage that translates into several qualitative advantages.

Chapter 4

Credit attribution with labeled topic models

Modern multi-label document collections reflect the fact that documents are often about more than one thing—for example, a news story about a highway transportation bill might naturally be filed under both *transportation* and *politics*, with neither category acting as a clear subset of the other. Similarly, a single web page in Delicious might be annotated with tags as diverse as *arts*, *physics*, *alaska*, and *beauty*. However, these labels do not apply with equal specificity across the whole of each document. The reason the article is labeled both *transportation* and *politics*, for instance, may be because it contains some words indicative of politics ("senator" and "committee") as well others related to transportation ("road" and "highway"). The mere presence of both labels should not be construed by a model of multi-labeled text collections as evidence that every word counts equally to each label.

We call the challenge of discovering the latent alignments between a document's labels and words the *credit attribution* problem. One promising approach to the credit attribution problem lies in the machinery of latent topics models such LDA [16], described in Section 2.2.2, and MM-LDA described in Section 3.3. These models

This chapter draws from group work published as "Labeled LDA: A supervised topic model for credit attribution in multi-label corpora" in EMNLP 2009 by D. Ramage, D. Hall, R. Nallapati, and C.D. Manning. [112]

represent each document as a mixture of unsupervised topics and explicitly assume that every word is generated from one underlying topic. As a result, latent topic models are not directly applicable to the credit attribution problem because they leave no mechanism to distinctively associate the learned topics with the observed labels. This is true even in MM-LDA, which incorporates tags to improve the trustworthiness of the model's clustering decisions, but is, nonetheless, a latent topic model.

This chapter presents *Labeled LDA* (L-LDA), a generative model for multiply labeled corpora that marries the multi-label supervision common to modern text datasets with the word-assignment ambiguity resolution of the LDA family of models. In contrast to latent topic models, L-LDA associates each label with one topic in direct correspondence. As a result, L-LDA speaks to the second desirable property of text mining models we identify in Chapter 1: *interpretability*. The topics learned by L-LDA can be interpreted directly as models of each label. In this chapter, we show that L-LDA is a natural extension of both LDA (by incorporating supervision) and Multinomial Naive Bayes (by incorporating a mixture model). We demonstrate that L-LDA can go a long way toward solving the credit attribution problem in multiply labeled documents with improved interpretability over LDA in corpus visualization (Section 4.4). And we show that L-LDA's ability to perform credit attribution enables it to greatly outperform support vector machines in a tag-driven snippet extraction task on tagged web pages (Section4.5). Finally, despite its generative semantics, we show that Labeled LDA is competitive with a strong baseline discriminative classifier when used a multi-label classifier on two classification tasks (Section4.6).

4.1 Related work

Several modifications of LDA incorporating supervision have been proposed in the literature. Two such models, Supervised LDA [14] and DiscLDA [74] are inappropriate for multiply labeled corpora because they limit a document to being associated with only a single label. Supervised LDA posits that a label is generated from each document's empirical topic mixture distribution. DiscLDA associates a single categorical label variable with each document and associates a topic mixture with each label.

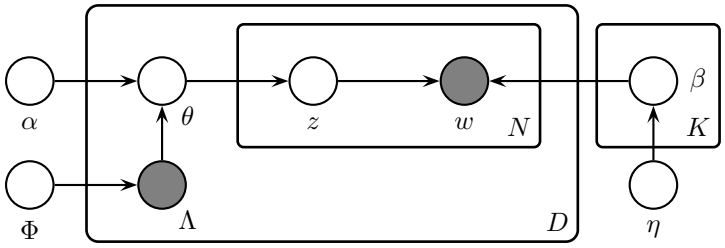

Figure 4.1: Graphical model of Labeled LDA: unlike standard LDA, both the label set $\mathbf{\Lambda}$ as well as the topic prior $\boldsymbol{\alpha}$ influence the topic mixture $\boldsymbol{\theta}$.

The MM-LDA model introduced in the previous chapter, is not constrained to one label per document because it models each document as a bag of words with a bag of labels, with topics for each observation drawn from a shared topic distribution. But, like other latent topic models, MM-LDA's learned topics do not correspond directly with the label set. Consequently, these models fall short as a solution to the credit attribution problem and are difficult to interpret.

4.2 Labeled LDA

Labeled LDA is a probabilistic graphical model that describes a process for generating a labeled document collection. Like Latent Dirichlet Allocation, Labeled LDA models each document as a mixture of underlying topics and generates each word from one topic. Unlike LDA, L-LDA incorporates supervision by simply constraining the topic model to use only those topics that correspond to a document's (observed) label set. The model description follows.

Let each document d be represented by a tuple consisting of a list of word indices $\mathbf{w}^{(d)} = (w_1, \ldots, w_{N_d})$ and a list of binary topic presence/absence indicators $\mathbf{\Lambda}^{(d)} = (l_1, \ldots, l_K)$ where each $w_i \in \{1, \ldots, V\}$ and each $l_k \in \{0, 1\}$. Here N_d is the document length, V is the vocabulary size and K the total number of unique labels in the corpus.

We set the number of topics in Labeled LDA to be the number of unique labels K in the corpus. The generative process for the algorithm is found in Table 4.1. Steps 1 and 2—drawing the multinomial topic distributions over vocabulary $\boldsymbol{\beta}_k$ for each topic k, from a Dirichlet prior $\boldsymbol{\eta}$—remain the same as for traditional LDA (see [16], page

1. For each topic $k \in \{1, \ldots, K\}$:
   (a) Generate $\boldsymbol{\beta}_k = (\beta_{k,1}, \ldots, \beta_{k,V})^T \sim \text{Dir}(\cdot|\boldsymbol{\eta})$
2. For each document d:
   (a) For each topic $k \in \{1, \ldots, K\}$
      i. Generate $\Lambda_k^{(d)} \in \{0, 1\} \sim \text{Bernoulli}(\cdot|\Phi_k)$
   (b) Generate $\boldsymbol{\alpha}^{(d)} = L^{(d)} \times \boldsymbol{\alpha}$
   (c) Generate $\boldsymbol{\theta}^{(d)} = (\theta_{l_1}, \ldots, \theta_{l_{M_d}})^T \sim \text{Dir}(\cdot|\boldsymbol{\alpha}^{(d)})$
   (d) For each i in $\{1, \ldots, N_d\}$:
      i. Generate $z_i \in \{\lambda_1^{(d)}, \ldots, \lambda_{M_d}^{(d)}\} \sim \text{Mult}(\cdot|\boldsymbol{\theta}^{(d)})$
      ii. Generate $w_i \in \{1, \ldots, V\} \sim \text{Mult}(\cdot|\boldsymbol{\beta}_{z_i})$

Table 4.1: Generative process for Labeled LDA: $\boldsymbol{\beta}_k$ is a vector consisting of the parameters of the multinomial distribution corresponding to the k^{th} topic, $\boldsymbol{\alpha}$ are the parameters of the Dirichlet topic prior and $\boldsymbol{\eta}$ are the parameters of the word prior, while Φ_k is the label prior for topic k. For the meaning of the projection matrix $L^{(d)}$, please refer to Equation 4.1.

4). The traditional LDA model then draws a multinomial mixture distribution $\boldsymbol{\theta}^{(d)}$ over all K topics, for each document d, from a Dirichlet prior $\boldsymbol{\alpha}$. However, we would like to restrict $\boldsymbol{\theta}^{(d)}$ to be defined only over the topics that correspond to its labels $\boldsymbol{\Lambda}^{(d)}$. Since the word-topic assignments z_i (see step 9 in Table 4.1) are drawn from this distribution, this restriction ensures that all the topic assignments are limited to the document's labels.

Towards this objective, we first generate the document's labels $\boldsymbol{\Lambda}^{(d)}$ using a coin toss for each topic k, with a labeling prior probability Φ_k, as shown in step 5. Next, we define the vector of document's labels to be $\boldsymbol{\lambda}^{(d)} = \{k|\Lambda_k^{(d)} = 1\}$. This allows us to define a document-specific label projection matrix $L^{(d)}$ of size $M_d \times K$ for each document d, where $M_d = |\boldsymbol{\lambda}^{(d)}|$, as follows:

For each row $i \in \{1, \ldots, M_d\}$ and column $j \in \{1, \ldots, K\}$:

$$L_{ij}^{(d)} = \begin{cases} 1 & \text{if } \lambda_i^{(d)} = j \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

In other words, the i^{th} row of $L^{(d)}$ has an entry of 1 in column j if and only if the i^{th} document label $\lambda_i^{(d)}$ is equal to the topic j, and zero otherwise. As the name indicates, we use the $L^{(d)}$ matrix to project the parameter vector of the Dirichlet topic prior $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)^T$ to a lower dimensional vector $\boldsymbol{\alpha}^{(d)}$ as follows:

$$\boldsymbol{\alpha}^{(d)} = L^{(d)} \times \boldsymbol{\alpha} = (\alpha_{\lambda_1^{(d)}}, \ldots, \alpha_{\lambda_{M_d}^{(d)}})^T \tag{4.2}$$

The dimensions of the projected vector correspond to the topics represented by the labels of the document. For example, suppose $K = 4$ and that a document d has labels given by $\boldsymbol{\Lambda}^{(d)} = \{0, 1, 1, 0\}$ which implies $\boldsymbol{\lambda}^{(d)} = \{2, 3\}$, then $L^{(d)}$ would be:

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Then, $\boldsymbol{\theta}^{(d)}$ is drawn from a Dirichlet distribution with parameters $\boldsymbol{\alpha}^{(d)} = L^{(d)} \times \boldsymbol{\alpha} = (\alpha_2, \alpha_3)^T$ (i.e., with the Dirichlet restricted to the topics 2 and 3).

This fulfills our requirement that the document's topics are restricted to its own

labels. The projection step constitutes the deterministic step 6 in Table 4.1. The remaining part of the model from steps 7 through 10 are the same as for regular LDA.

The dependency of $\boldsymbol{\theta}$ on both $\boldsymbol{\alpha}$ and $\boldsymbol{\Lambda}$ is indicated by directed edges from $\boldsymbol{\Lambda}$ and $\boldsymbol{\alpha}$ to $\boldsymbol{\theta}$ in the plate notation in Figure 4.1. This is the only additional dependency we introduce in LDA's representation—compare with Figure 2.2 in Section 2.2.2.

4.2.1 Learning and inference

In most applications discussed in this chapter, we will assume that the documents are multiply tagged with human labels, both at learning and inference time.

When the labels $\boldsymbol{\Lambda}^{(d)}$ of the document are observed, the labeling prior Φ is d-separated from the rest of the model given $\boldsymbol{\Lambda}^{(d)}$. Hence the model is same as traditional LDA, except the constraint that the topic prior $\boldsymbol{\alpha}^{(d)}$ is now restricted to the set of labeled topics $\boldsymbol{\lambda}^{(d)}$. Therefore, we can use collapsed Gibbs sampling [48] for training where the sampling probability for a topic for position i in a document d in Labeled LDA is given by:

$$P(z_i = j|\mathbf{z}_{-i}) \propto \frac{n^{w_i}_{-i,j} + \eta_{w_i}}{n^{(\cdot)}_{-i,j} + \boldsymbol{\eta}^T\mathbf{1}} \times \frac{n^{(d)}_{-i,j} + \alpha_j}{n^{(d)}_{-i,\cdot} + \boldsymbol{\alpha}^T\mathbf{1}} \tag{4.3}$$

where $n^{w_i}_{-i,j}$ is the count of word w_i in topic j, that does not include the current assignment z_i, a missing subscript or superscript (e.g. $n^{(\cdot)}_{-i,j}$)) indicates a summation over that dimension, and $\mathbf{1}$ is a vector of 1's of appropriate dimension.

Although the equation above looks exactly the same as that of LDA, we have an important distinction in that, the target topic j is restricted to belong to the set of labels, i.e., $j \in \boldsymbol{\lambda}^{(d)}$.

Once the topic multinomials $\boldsymbol{\beta}$ are learned from the training set, one can perform inference on any new labeled test document using Gibbs sampling restricted to its tags, to determine its per-word label assignments $\mathbf{z}$. In addition, one can also compute its posterior distribution $\boldsymbol{\theta}$ over topics by appropriately normalizing the topic assignments $\mathbf{z}$.

It should now be apparent how this model addresses some of the problems in multi-labeled corpora that we highlighted in Section 4. For example, since there is a one-to-one correspondence between the labels and topics, the model can display automatic topical summaries for each label k in terms of the topic-specific distribution $\boldsymbol{\beta}_k$. Similarly, since the model assigns a label z_i to each word w_i in the document d automatically, we can now extract portions of the document relevant to each label k (it would be all words $w_i \in \mathbf{w}^{(d)}$ such that $z_i = k$). In addition, we can use the topic distribution $\boldsymbol{\theta}^{(d)}$ to rank the user specified labels in the order of their relevance to the document, thereby also eliminating spurious ones if necessary.

Finally, we note that other less restrictive variants of the proposed L-LDA model are possible. For example, one could consider a version that allows topics that do not correspond to the label set of a given document with a small probability, or one that allows a common background topic in all documents. We did implement these variants in our preliminary experiments, but they did not yield better performance than L-LDA in the tasks we considered here. We return to the idea of latent background topics in Chapter 5, along with other extensions and relaxations of the L-LDA model.

4.2.2 Relationship to Naive Bayes

The derivation of the algorithm so far has focused on its relationship to LDA. However, Labeled LDA can also be seen as an extension of the event model of a traditional Multinomial Naive Bayes classifier [87] by the introduction of a mixture model. In this section, we develop the analogy as another way to understand L-LDA from a supervised perspective.

Consider the case where no document in the collection is assigned two or more labels. Now for a particular document d with label l_d, Labeled LDA draws each word's topic variable z_i from a multinomial constrained to the document's label set, i.e. $z_i = l_d$ for each word position i in the document. During learning, the Gibbs sampler will assign each z_i to l_d while incrementing $\beta_{l_d}(w_i)$, effectively counting the occurrences of each word type in documents labeled with l_d. Thus in the singly labeled document case, the probability of each document under Labeled LDA is equal to the probability

of the document under the Multinomial Naive Bayes event model trained on those same document instances. Unlike the Multinomial Naive Bayes classifier, Labeled LDA does not encode a decision boundary for unlabeled documents by comparing $P(\mathbf{w}^{(d)}|l_d)$ to $P(\mathbf{w}^{(d)}|\neg l_d)$, although we discuss using Labeled LDA for multi-label classification in Section4.6.

Labeled LDA's similarity to Naive Bayes ends with the introduction of a second label to any document. In a traditional one-versus-rest Multinomial Naive Bayes model, a separate classifier for each label would be trained on all documents with that label, so each word can contribute a count of 1 to every observed label's word distribution. By contrast, Labeled LDA assumes that each document is a mixture of underlying topics, so the count mass of single word instance must instead be distributed over the document's observed labels.

4.3 Credit attribution within tagged documents

Social bookmarking websites contain millions of tags describing many of the web's most popular and useful pages. However, not all tags are uniformly appropriate at all places within a document. In the sections that follow, we examine mechanisms by which Labeled LDA's credit assignment mechanism can be utilized to help support browsing and summarizing tagged document collections.

To create a consistent dataset for experimenting with our model, we selected 20 tags of medium to high frequency from a collection of documents dataset crawled from *delicious*, a popular social bookmarking website [58]. From that larger dataset, we selected uniformly at random four thousand documents that contained at least one of the 20 tags, and then filtered each document's tag set by removing tags not present in our tag set. After filtering, the resulting corpus averaged 781 non-stop words per document, with each document having 4 distinct tags on average. In contrast to many existing text datasets, our tagged corpus is highly multiply labeled: almost 90% of of the documents have more than one tag. (For comparison, less than one third of the news documents in the popular RCV1-v2 collection of newswire are multiply labeled). We will refer to this collection of data as the delicious tag dataset.

4.4 Topic visualization

A first question we ask of Labeled LDA is how its topics compare with those learned by traditional LDA on the same collection of documents. We ran our implementations of Labeled LDA and LDA on the delicious corpus described above. Both are based on the standard collapsed Gibbs sampler, with the constraints for Labeled LDA implemented as in Section 4.2.

Figure4.2 shows the top words associated with 20 topics learned by Labeled LDA and 20 topics learned by unsupervised LDA on the delicious document collection. Labeled LDA's topics are directly named with the tag that corresponds to each topic, an improvement over standard practice of inferring the topic name by inspection [91]. The topics learned by the unsupervised variant were matched to a Labeled LDA topic highest cosine similarity.

The topics selected are representative: compared to Labeled LDA, unmodified LDA allocates many topics for describing the largest parts of the corpus and underrepresents tags that are less uncommon: of the 20 topics learned, LDA learned multiple topics mapping to each of five tags (*web*, *culture*, and *computer*, *reference*, and *politics*, all of which were common in the dataset) and learned no topics that aligned with six tags (*books*, *english*, *science*, *history*, *grammar*, *java*, and *philosophy*, which were rarer).

4.4.1 Tagged document visualization

In addition to providing automatic summaries of the words best associated with each tag in the corpus, Labeled LDA's credit attribution mechanism can be used to augment the view of a single document with rich contextual information about the document's tags.

Figure4.3 shows one web document from the collection, a page describing a guide to writing English prose. The 10 most common tags for that document are *writing, reference, english, grammar, style, language, books, book, strunk*, and *education*, the first eight of which were included in our set of 20 tags. In the figure, each word that has high posterior probability from one tag has been annotated with that tag.

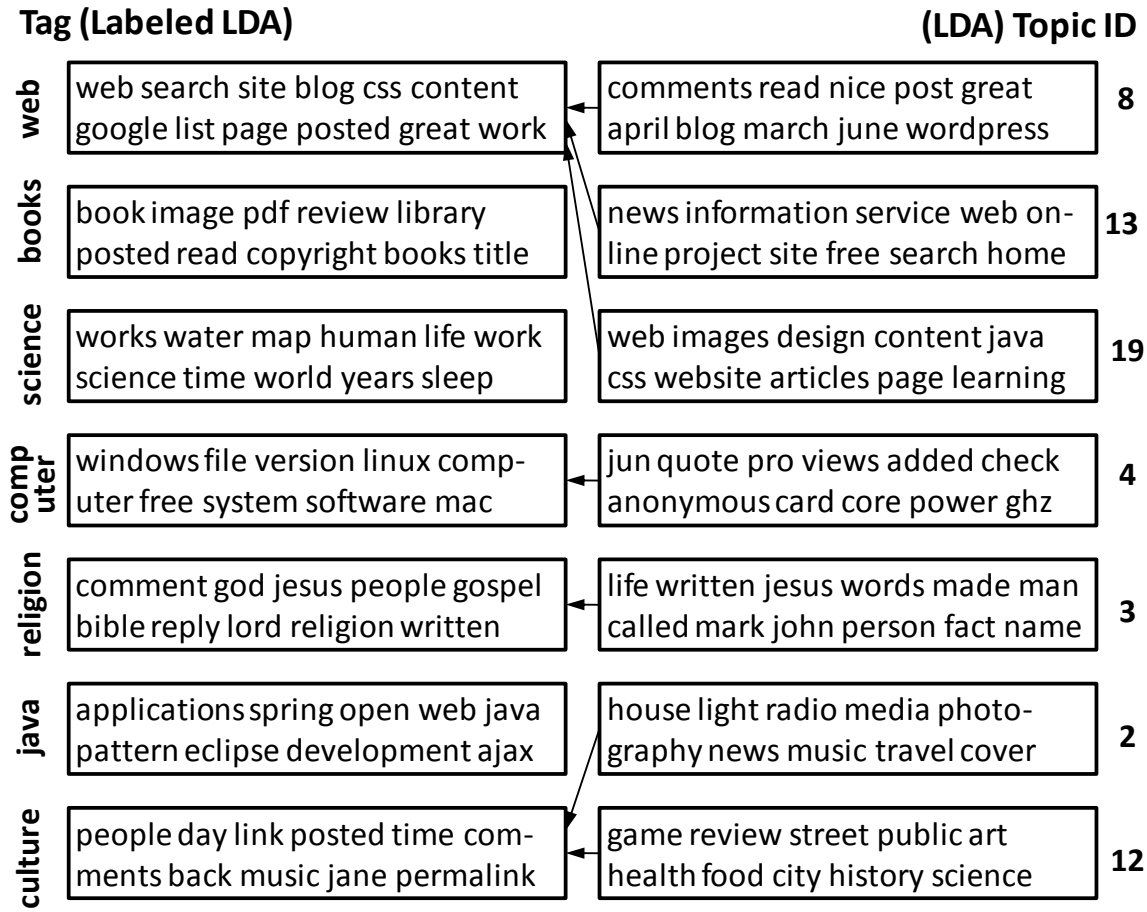

Figure 4.2: Comparison of some of the 20 topics learned on delicious by Labeled LDA (left) and traditional LDA (right), with representative words for each topic shown in the boxes. Labeled LDA's topics are named by their associated tag. Arrows from right-to-left show the mapping of LDA topics to the closest Labeled LDA topic by cosine similarity. Tags not shown are: *design, education, english, grammar, history, internet, language philosophy, politics, programming, reference, style, writing.*

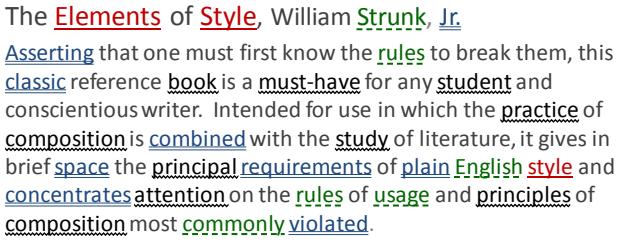

Figure 4.3: Example document with important words annotated with four of the page's tags as learned by Labeled LDA. Red (single underline) is *style*, green (dashed underline) *grammar*, blue (double underline) *reference*, and black (jagged underline) *education*.

The red words come from the *style* tag, green from the *grammar* tag, blue from the *reference* tag, and black from the *education* tag. In this case, the model does very well at assigning individual words to the tags that, subjectively, seem to strongly imply the presence of that tag on this page. A more polished rendering could add subtle visual cues about which parts of a page are most appropriate for a particular set of tags.

4.5 Snippet extraction

Another natural application of Labeled LDA's credit assignment mechanism is as a means of selecting snippets of a document that best describe its contents from the perspective of a particular tag. Consider again the document in Figure 4.3. Intuitively, if this document were shown to a user interested in the tag *grammar*, the most appropriate snippet of words might prefer to contain the phrase "rules of usage," whereas a user interested in the term *style* might prefer the title "Elements of Style."

To quantitatively evaluate Labeled LDA's performance at this task, we constructed a set of 29 recently tagged documents from delicious that were labeled with two or more tags from the 20 tag subset, resulting in a total of 149 (document,tag) pairs. For each pair, we extracted a 15-word window with the highest tag-specific score from the document. Two systems were used to score each window: Labeled LDA and a collection of one-vs-rest SVMs trained for each tag in the system. L-LDA

books
L-LDA this classic reference book is a must-have for any student and conscientious writer. Intended for
SVM the rules of usage and principles of composition most commonly violated. Search: CONTENTS Bibliographic

language
L-LDA the beginning of a sentence must refer to the grammatical subject 8. Divide words at
SVM combined with the study of literature, it gives in brief space the principal requirements of

grammar
L-LDA requirements of plain English style and concentrates attention on the rules of usage and principles of
SVM them, this classic reference book is a must-have for any student and conscientious writer.

Figure 4.4: Representative snippets extracted by L-LDA and tag-specific SVMs for the web page shown in Figure 4.3.

scored each window as the expected probability that the tag had generated each word. For SVMs, each window was taken as its own document and scored using the tag-specific SVM's un-thresholded scoring function, taking the window with the most positive score. While a complete solution to the tag-specific snippet extraction problem might be more informed by better linguistic features (such as phrase boundaries), this experimental setup suffices to evaluate both kinds of models for their ability to appropriately assign words to underlying labels.

Figure 4.3 shows some example snippets output by our system for this document.

Model	Best Snippet	Unanimous
L-LDA	**72 / 149**	24 / 51
SVM	21 / 149	2 / 51

Table 4.2: Human judgments of tag-specific snippet quality as extracted by L-LDA and SVM. The center column is the number of document-tag pairs for which a system's snippet was judged superior. The right column is the number of snippets for which all three annotators were in complete agreement (numerator) in the subset of document scored by all three annotators (denominator).

Note that while SVMs did manage to select snippets that were vaguely on topic, Labeled LDA's outputs are generally of superior subjective quality. To quantify this intuition, three human annotators rated each pair of snippets. The outputs were randomly labeled as "System A" or "System B," and the annotators were asked to judge which system generated a better tag-specific document subset. The judges were also allowed to select neither system if there was no clear winner. The results are summarized in Table 4.2.

L-LDA was judged superior by a wide margin: of the 149 judgments, L-LDA's output was selected as preferable in 72 cases, whereas SVM's was selected in only 21. The difference between these scores was highly significant ($p < .001$) by the sign test. To quantify the reliability of the judgments, 51 of the 149 document-tag pairs were labeled by all three annotators. In this group, the judgments were in substantial agreement,[1] with Fleiss' Kappa at .63.

Further analysis of the triply-annotated subset yields further evidence of L-LDA's advantage over SVM's: 33 of the 51 were tag-page pairs where L-LDA's output was picked by at least one annotator as a better snippet (although L-LDA might not have been picked by the other annotators). And of those, 24 were unanimous in that all three judges selected L-LDA's output. By contrast, only 10 of the 51 were tag-page pairs where SVMs' output was picked by at least one annotator, and of those, only 2 were selected unanimously.

4.6 Multilabeled text classification

In the preceding section we demonstrated how Labeled LDA's credit attribution mechanism enabled effective modeling within documents. In this section, we consider whether L-LDA can be adapted as an effective multi-label classifier for documents as a whole. To answer that question, we applied a modified variant of L-LDA to a

[1] Of the 15 judgments that were in contention, only two conflicted on *which* system was superior (L-LDA versus SVM); the remaining disagreements were about whether or not one of the systems was a clear winner.

multi-label document classification problem: given a training set consisting of documents with multiple labels, predict the set of labels appropriate for each document in a test set.

Multi-label classification is a well researched problem. Many modern approaches incorporate label correlations (e.g., [66], [65]). Others, like our algorithm are based on mixture models (such as [130]). However, we are aware of no methods that trade off label-specific word distributions with document-specific label distributions in quite the same way.

In Section 4.2, we discussed learning and inference when labels are observed. In the task of multilabel classification, labels are available at training time, so the learning part remains the same as discussed before. However, inferring the best set of labels for an unlabeled document at test time is more complex: it involves assessing all label assignments and returning the assignment that has the highest posterior probability. However, this is not straight-forward, since there are 2^K possible label assignments. To make matters worse, the support of $\boldsymbol{\alpha}(\Lambda^{(d)})$ is different for different label assignments. Although we are in the process of developing an efficient sampling algorithm for this inference, for the purposes of this chapter we make the simplifying assumption that the model reduces to standard LDA at inference, where the document is free to sample from any of the K topics. This is a reasonable assumption because allowing the model to explore the whole topic space for each document is similar to exploring all possible label assignments. The document's most likely labels can then be inferred by suitably thresholding its posterior probability over topics.

As a baseline, we use a set of multiple one-vs-rest SVM classifiers which is a popular and extremely competitive baseline used by most previous papers (see [66, 130] for instance). We scored each model based on Micro-F1 and Macro-F1 as our evaluation measures [77]. While the former allows larger classes to dominate its results, the latter assigns an equal weight to all classes, providing us complementary information.

Dataset	%MacroF1		%MicroF1	
	L-LDA	SVM	L-LDA	SVM
Arts	30.70(1.62)	23.23 (0.67)	39.81(1.85)	48.42 (0.45)
Business	30.81(0.75)	22.82 (1.60)	67.00(1.29)	72.15 (0.62)
Computers	27.55(1.98)	18.29 (1.53)	48.95(0.76)	61.97 (0.54)
Education	33.78(1.70)	36.03 (1.30)	41.19(1.48)	59.45 (0.56)
Entertainment	39.42(1.38)	43.22 (0.49)	47.71(0.61)	62.89 (0.50)
Health	45.36(2.00)	47.86 (1.72)	58.13(0.43)	72.21 (0.26)
Recreation	37.63(1.00)	33.77 (1.17)	43.71(0.31)	59.15 (0.71)
Society	27.32(1.24)	23.89 (0.74)	42.98(0.28)	52.29 (0.67)

Table 4.3: Averaged performance across ten runs of multi-label text classification for predicting subsets of the named Yahoo directory categories. Numbers in parentheses are standard deviations across runs. L-LDA outperforms SVMs on 5 subsets with MacroF1, but on no subsets with MicroF1.

4.6.1 Yahoo

We ran experiments on a corpus from the Yahoo directory, modeling our experimental conditions on the ones described in [65].[2] We considered documents drawn from 8 top level categories in the Yahoo directory, where each document can be placed in any number of subcategories. The results were mixed, with SVMs ahead on one measure: Labeled LDA beat SVMs on five out of eight datasets on MacroF1, but didn't win on any datasets on MicroF1. Results are presented in Table 4.3.

Because only a processed form of the documents was released, the Yahoo dataset does not lend itself well to error analysis. However, only 33% of the documents in each top-level category were applied to more than one sub-category, so the credit assignment machinery of L-LDA was unused for the majority of documents. We therefore ran an artificial second set of experiments considering only those documents that had been given more than one label in the training data. On these documents, the results were again mixed, but Labeled LDA comes out ahead. For MacroF1, L-LDA beat SVMs on four datasets, SVMs beat L-LDA on one dataset, and three were

[2]We did not carefully tune per-class thresholds of each of the one vs. rest classifiers in each model, but instead tuned only one threshold for all classifiers in each model via cross-validation on the Arts subsets. As such, our numbers were on an average 3-4% less than those reported in [65], but the methods were comparably tuned.

a statistical tie.[3] On MicroF1, L-LDA did much better than on the larger subset, outperforming on four datasets with the other four a statistical tie.

It is worth noting that the Yahoo datasets are skewed by construction to contain many documents with highly overlapping content: because each collection is within the same super-class such as "Arts", "Business", etc., each sub-categories' vocabularies will naturally overlap a great deal. L-LDA's credit attribution mechanism is most effective at partitioning semantically distinct words into their respective label vocabularies, so we expect that Labeled-LDA's performance as a text classifier would improve on collections with more semantically diverse labels.

4.6.2 Tagged web pages

We also applied our method to text classification on the delicious dataset, where the documents are naturally multiply labeled (more than 89%) and where the tags are less inherently similar than in the Yahoo subcategories. Therefore we expect Labeled LDA to do better credit assignment on this subset and consequently to show improved performance as a classifier, and indeed this is the case.

We evaluated L-LDA and multiple one-vs-rest SVMs on 4000 documents with the 20 tag subset described in Section 4.3. L-LDA and multiple one-vs-rest SVMs were trained on the first 80% of documents and evaluated on the remaining 20%, with results averaged across 10 random permutations of the dataset. The results are shown in Table 4.4. We tuned the SVMs' shared cost parameter $C(= 10.0)$ and selected raw term frequency over tf-idf weighting based on 4-fold cross-validation on 3,000 documents drawn from an independent permutation of the data. For L-LDA, we tuned the shared parameters of threshold and proportionality constants in word and topic priors. L-LDA and SVM have very similar performance on MacroF1, while L-LDA substantially outperforms on MicroF1. In both cases, L-LDA's improvement is statistically significantly by a 2-tailed paired t-test at 95% confidence.

[3]The difference between means of multiple runs were not significantly different by two-tailed paired t-test.

Model	%MacroF1	%MicroF1
L-LDA	39.85 (.989)	52.12 (.434)
SVM	39.00 (.423)	39.33 (.574)

Table 4.4: Mean performance across ten runs of multi-label text classification for predicting 20 tags on delicious data. Numbers in parentheses are standard deviations across runs. L-LDA outperforms SVMs significantly on both metrics by a 2-tailed, paired t-test at 95% confidence.

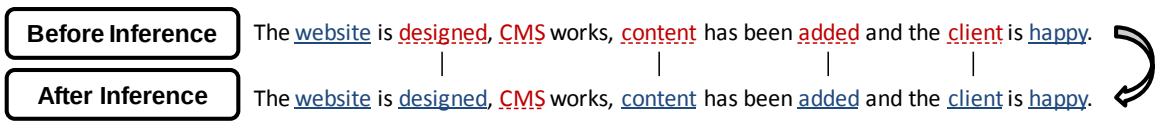

Figure 4.5: The effect of tag mixture proportions for credit assignment in a web document. Blue (single underline) words are generated from the *design* tag; red (dashed underline) from the *programming* tag. By themselves, most words used here have a higher probability in *programming* than in *design*. But because the document as a whole is more about *design* than *programming* (incorporating words not shown here), inferring the document's topic-mixture θ enables L-LDA to correctly re-assign most words.

4.7 Discussion

One of the main advantages of L-LDA on multiply labeled documents comes from the model's document-specific topic mixture θ. By explicitly modeling the importance of each label in the document, Labeled LDA can effective perform some contextual word sense disambiguation, which suggests why L-LDA can outperform SVMs on the delicious dataset.

As a concrete example, consider the excerpt of text from the Delicious dataset in Figure 4.5. The document itself has several tags, including *design* and *programming*. Initially, many of the likelihood probabilities $P(w|label)$ for the (content) words in this excerpt are higher for the label *programming* than *design*, including "content", "client", "CMS" and even "designed", while *design* has higher likelihoods for just "website" and "happy". However, after performing inference on this document using L-LDA, the inferred document probability for *design* ($P(design)$) is much higher than it is for *programming*. In fact, the higher probability for the tag more than makes up the

difference in the likelihood for all the words except "CMS" (Content Management System), so that L-LDA correctly infers that most of the words in this passage have more to do with *design* than *programming*.

The relationship between Labeled LDA and some existing models warrants more explication. In particular, the models of associating word usage with authors in [118, 88] are worth describing: the Author-Topic [118] model assumes that each author (label) has its own distribution over topics. In the Author-Topic model, you first pick an author and then pick a topic from that author's distribution over topics. So the AT model still assumes a latent topic space. Labeled LDA has no latent topic—each label *is* a topic—but does assume that the assignments of topics to words is latent. By contrast, the Author Model in that same paper directly associated each author with a distribution over words, which is much morel like L-LDA. However, the author model assumes all authors (labels) have equal weight in generating each word; Labeled LDA assumes that the labels are non-uniform with a latent mixture drawn from a Dirichlet prior. Labeled LDA is more similar in spirit to McCallum's 1999 multinomial mixture model [88] which has a similar modeling intuition: the main differences are in the intended applications (classification versus credit attribution) and in the priors – our model includes proper Dirichlet priors for the per-document mixtures over observed labels (our thetas, McCallum's lambdas) and the per-label distributions over words, whereas theirs treats these as parameters to be estimated directly with no prior or generative process. McCallum '99 also includes a background language class on each document as part of the model definition.

4.8 Conclusion

This chapter describes Labeled LDA, a model of multi-labeled corpora that addresses the credit assignment problem. The model improves upon LDA for labeled corpora by gracefully incorporating user supervision in the form of a one-to-one mapping between topics and labels, introducing newfound *interpretability* that moves beyond the capabilities of traditional latent topic models. We demonstrate the model's effectiveness on tasks related to credit attribution within documents, including document

visualizations and tag-specific snippet extraction. An approximation to Labeled LDA is also shown to be competitive with a strong baseline (multiple one vs-rest SVMs) for multi-label classification.

Because Labeled LDA is a graphical model in the LDA family, it enables a range of natural extensions for future investigation. For example, the current model does not capture correlations between labels, but such correlations might be introduced by composing Labeled LDA with newer state of the art topic models like the Correlated Topic Model [15] or the Pachinko Allocation Model [78]. And with improved inference for unsupervised $\mathbf{\Lambda}$, Labeled LDA lends itself naturally to modeling semi-supervised corpora where labels are observed for only some documents.

Labeled LDA's interpretability gains come from its association of each label with exactly one topic. However, this advantage in interpretability comes at a cost of *flexibility* in that the model is no longer suited for discovering language variation that has not been explicitly labeled. In the next chapter, we extend Labeled LDA to incorporate latent sub-structure within the provided labels, further generalizing the model and introducing new-found flexibility.

Chapter 5

Partially labeled topic models

Effective text mining tools for labeled documents should be trustworthy, interpretable, and flexible (Chapter 1). In previous chapters, we saw that labels could be used as a source of side information to make topic models more *trustworthy,* as in MM-LDA (Chapter 3). Or, by aligning each topic with some label, we could make a more *interpretable* topic model, as in Labeled LDA (Chapter 4). However, to be truly effective, these models must also be *flexible* enough to account for the latent variations in the textual patterns underlying the observed labels while still discovering unlabeled topics. Indeed, a text model with these properties could be described as *partially supervised* in that the provided document-level supervision only hints at the unlabeled relationships of interest: namely, the alignment of words, with labels, background language, or latent variations thereof. A model with these properties could help us understand and interpret the meaning of ambiguous labels in context; to uncover latent topics that are not described by any of the labels; and to discover which words should be attributed to which of a document's labels, or to none at all.

This chapter presents two new supervised generative models of labeled text that address trustworthiness, interpretability, and flexibility: Partially Labeled Dirichlet Allocation (PLDA) and the Partially Labeled Dirichlet Process (PLDP). These models generalize and unify the popular unsupervised topic model LDA (Section 2.2.2,

This chapter draws from group work published as "Partially labeled topic models for interpretable text mining" in KDD 2011 by D. Ramage, C.D. Manning, and S.T. Dumais. [114]

or [16]), the multinomial naive Bayes supervised text classifier's event model (Section 4.2.2, or [87]), and Labeled LDA [112]. Intuitively, PLDA is like other topic models in that it assumes each document's words are drawn from a document-specific mixture of latent topics, where each topic is itself represented as a distribution over words. But unlike latent topic models, PLDA assumes that each document can use only those topics that are in a topic *class* associated with one or more of the document's labels. In particular, we introduce one class (consisting of multiple topics) for each label in the label set, as well as one latent class that applies to all documents. This construction allows PLDA to discover large-scale patterns in language usage associated with each individual label, variations of linguistic usage within a label, and background topics not associated with any label. A parallel learning and inference algorithm for PLDA allows it to scale to large document collections. Our second model, PLDP, extends PLDA by incorporating a non-parametric Dirichlet process prior over each class's topic set, allowing the model to adaptively discover how many topics belong to each label, but comes at a computational cost.

In this chapter, we explore PLDA and PLDP with qualitative case studies of tagged web pages from Delicious and PhD dissertation abstracts, demonstrating improved model interpretability over traditional topic models and improved flexibility over Labeled LDA. We use the many tags present in Delicious to quantitatively demonstrate the new models' trustworthiness in its higher correlation with human relatedness scores.

5.1 Related work

Partially supervised text mining models straddle the boundary between unsupervised learning, in which models discover unmarked statistical relationships in the data, and supervised learning, which emphasizes the relationship between word features and a given space of labels for the purpose of classifying new documents. Popular unsupervised approaches like the topic models and dimensionality reduction techniques described in Chapter 2 are well suited for exploratory text analysis—e.g. [48]—but most do not account for the label space. When they do, it is usually to improve the

quality of a shared set of latent topics—such as in [74, 14, 35, 63, 144, 95]—rather than to directly model the contents of the provided labels. The choice of a purely latent topic model therefore introduces problems of interpreting what topics really mean, how they should be named, and to what extent trends based on them can be trusted in qualitative applications. These challenges ultimately stem from the nature of the topics discovered: unsupervised topics can capture broad patterns in a document collection, but they provide no guarantee about how well the learned topics align with the human provided labels. In other words, these models are flexible but not interpretable.

In contrast, supervised learning and (multi-) label prediction explicitly model the label space for the purpose of prediction (such as in [38, 65]), but tend not to discover latent sub-structure or other latent patterns. Other learning formulations exist in the space between supervised learning and unsupervised learning, most notably semi-supervised learning [30], in which the goal is to improve label classification performance by making use of unsupervised data [145]. Another, similar learning paradigm is semi-supervised clustering—such as [9, 138]—in which some supervised information is used to improve an unsupervised task. Usually this information comes in the form of human-provided pair-wise similarity/dissimilarity scores or constraints. As a result, these approaches can be used effectively for label prediction or document clustering, but do not lend themselves to more fine-grained questions about how the terms and label space interact. By contrast, the partially supervised approach pursued here is explicitly designed to improve upon the exploratory and descriptive analyses that draw practitioners to unsupervised topic models to begin with—i.e. to discover and characterize the relationships between patterns, but with the added ability to constrain those patterns to align with label classes that are meaningful to people.

Recently, researchers in the topic modeling community have begun to explore new ways of incorporating meta-data and hierarchy into their models, which is the approach to partially supervised text mining that we take here. For instance, Markov Random Topic Fields [35] and Markov Topic Models [134] both allow information about document groups to influence the learned topics. There has also been a great

amount of work on simultaneously modeling relationships among several variables, such as authors and topics in the Author-Topic model [118], tags and words in [119], and topics sentiment in [79]. All of these models assume a latent topic space that is influenced by external label information of some form. By contrast, we use topics to model the sub-structure of labels and unlabeled structure around them. Other ways to constrain and exploit topic models for text mining tasks include recent work in mining product reviews such as, Titov and McDonald [128] and later Branavan, et al. [22] who extract ratable aspects of product reviews. And recently, the Nubbi model of topics and social networks [28] introduced by Chang, et al., constrains an LDA-like topic model to learn topics that correspond to individual entities (such as heads of state in Wikipedia) and the relationships between them. Topic models that account for an extra level of topic correlation have been studied as well, with notable papers such as Blei et al.'s hierarchical topic models [13] and Li and MacCallum's Pachinko Allocation [78]. These types of models assume an extra hidden layer of abstraction that models topic-topic correlation. The label classes in this work can be seen as an analogous layer, but here they are supervised, hard assignments constraining only some topics to be active depending on a document's observed labels.

PLDA and PLDP build on Labeled LDA (Chapter 4, or [112]) and similar models such as the extension of Rubin et al. in [119]. Like PLDA and PLDP, Labeled LDA assumes that each document is annotated with a set of observed labels, and that these labels play a direct role in generating the document's words from per-label distributions over terms. However, Labeled LDA does not assume the existence of any latent topics (neither global nor within a label)—only the documents' distributions over their observed labels, as well as those labels' distributions over words, are inferred. As a result, Labeled LDA does not support latent sub-topics within a given label nor any global latent topics. In this chapter, we introduce two new models, PLDA and PLDP, that by incorporating classes of latent topics extend, generalize, and unify LDA with Labeled LDA. This simple change opens new opportunities in interpretable text mining and results in a large and surprising boost in the models' ability to correlate with human similarity judgments, as we demonstrate in 5.3.3.

5.2 Partially supervised models

In our formalization of partially supervised text mining, we are given a collection of documents $\mathbb{D}$, each containing a multi-set of words $\vec{w}_d$ from a vocabulary $\mathbb{V}$ of size V and a set of labels Λ_d from a space of labels $\mathbb{L}$. We would like to recover a set of topics Φ that fit the observed distribution of words in the multi-labeled documents, where each topic is a multinomial distribution over words $\mathbb{V}$ that tend to co-occur with each other and some label $l \in \mathbb{L}$. Latent topics that have no associated label are optionally modeled by assuming the existence of a background *latent* label $\mathbb{L}$ that is applied to all documents in the collection. In the sections below, we define PLDA and PLDP, both of which assume that the word w at position i in each document d is generated by first picking a label l from Λ_d and then a topic z from the set of topics associated with that label. Then word w is picked from the topic indexed $\Phi_{l,z}$. In this way, both PLDA and PLDP can be used for *credit attribution* of words to labels by examining the posterior probability over labels for a particular word instance. Both PLDA and PLDP generative probabilistic graphical models, and so for each we will use an approximate inference algorithm to re-construct the per-document mixtures over labels and topics, as well as the set of words associated with each label. By incorporating the latent class of topics in addition to the label classes, the model effectively forces each word to decide if it is better modeled by a broad, latent topic, or a topic that applies specifically to one of its document's labels.

5.2.1 Partially Labeled Dirichlet Allocation

Partially Labeled Dirichlet Allocation (PLDA) is a generative model for a collection of labeled documents, extending the generative story of LDA (Section 2.2.2, or [16]) to incorporate labels, and of Labeled LDA (Chapter 4, or [112]) to incorporate per-label latent topics. Formally, PLDA assumes the existence of a set of $\mathbb{L}$ labels (indexed by $1..L$), each of which has been assigned some number of topics $\mathbb{K}_l$ (indexed by $1..K_L$) and where each topic $\beta_{l,k}$ is represented as a multinomial distribution over all terms in the vocabulary $\mathbb{V}$ drawn from a symmetric Dirichlet prior η. One of these labels may optionally denote the shared global *latent* topic class, which can be interpreted

$\mathbb{L}$	Set of labels indexed by $l \in 1..L$
$\mathbb{K}_l$	set of topics assigned to label l indexed by $k \in 1..K_l$
Λ_d	Per-document assigned labels as sparse binary vector of size L
$\theta_{d,l}$	Per-document d, per-label l multinomial over topics $\mathbb{K}_l$
ψ_d	Per-document d distribution over labels $l \in \Lambda_d$
Φ	Sparse binary vector prior of Λ_d
$\beta_{l,k}$	Per-label l, per topic k distribution over vocabular $\mathbb{V}$
α	Dirichlet hyperparameter for $\theta_{1...D,1...L}$ and $\psi_{1...D,1..L}$
η	Dirichlet hyperparameter for $\beta_{1...L,1...K}$
$z_{d,i}$	Latent topic assignment at word position i in document d

Table 5.1: Summary of variables used in PLDA in addition to those in Table 2.1.

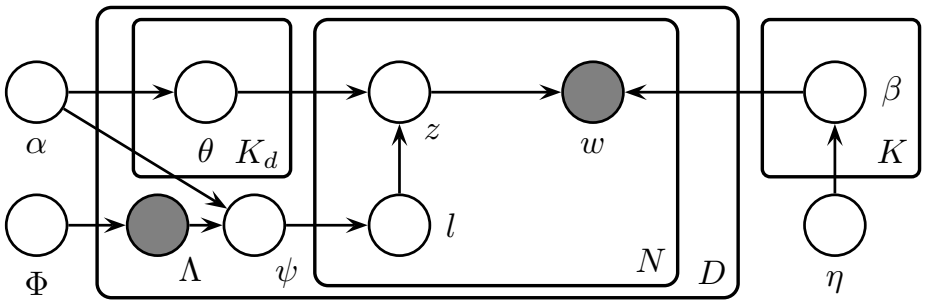

Figure 5.1: Bayesian graphical model for PLDA. Each document's words w and labels Λ are observed, with the per-document label distribution ψ, per-document-label topic distributions θ, and per-topic word distributions β taken as hidden variables. Because we assume each document's label-set Λ_d is observed, its sparse vector prior Φ is unused, included for completeness.

as a label *"latent"* present on every document d. PLDA assumes that each topic takes part in exactly one label.

Figure 5.1 shows the Bayesian graphical model for PLDA. Each document d is generated by first drawing a document-specific subset of available label classes, represented as a sparse binary vector Λ_d from a sparse binary vector prior. A document-specific mix $\theta_{d,j}$ over topics $1..K_j$ is drawn from a symmetric Dirichlet prior α for each label $j \in \Lambda_d$ present in the document. Then, a document-specific mix of observed labels ψ_d is drawn as a multinomial of size $|\Lambda_d|$ from a Dirichlet prior $\vec{\alpha_L}$, with each element $\psi_{d,j}$ corresponding to the document's probability of using label $j \in \Lambda_d$ when selecting a latent topic for each word. For derivational simplicity, we define the

element at position j of $\vec{\alpha_L}$ to be αK_j, so $\vec{\alpha_L}$ is not a free parameter. Each word w in document d is drawn from some label's topic's word distribution, i.e. it is drawn by first picking a label j from ψ_d, a topic z from $\theta_{d,l}$, and then a word w from $\beta_{l,k}$. Ultimately, this word will be picked in proportion to how much the enclosing document prefers the label l, how much that label prefers the topic z, and how much that topic prefers the word w.

We are interested in finding an efficient way to compute the joint likelihood of the observed words $\vec{w}$ with the unobserved label and topic assignments $\vec{l}$ and $\vec{z}$, $P(\vec{w}, \vec{z}, \vec{l}|\vec{\Lambda}, \alpha, \eta, \Phi) = P(\vec{w}|\vec{z}, \eta)P(\vec{z}, \vec{l}|\vec{\Lambda}, \alpha, \Phi)$. Later, we will use this joint likelihood to derive efficient updates for the parameters $\theta_{1...D,1...L}$, $\psi_{1...D,1...L}$, and $\beta_{1...K,1...V}$. First, we note that the left term $P(\vec{w}|\vec{z}, \eta) = \int_\beta P(\vec{w}|\vec{z}, \beta)P(\beta|\eta)d\beta$ is the same as for standard latent Dirichlet allocation and ultimately contributes the same terms to the full conditional as well as to the sampling formula for updating individual topic assignments $z_{d,i}$, so we use the same derivation as in e.g. [48]. Using the model's independence assumptions, we consider the joint probability of the topics and labels, $P(\vec{z}, \vec{l}|\Lambda, \alpha, \Phi) = P(\vec{z}|\vec{l}, \alpha)P(\vec{l}|\Lambda, \Phi, \alpha)$. We will examine each half of this expression in turn. First, observe that $P(\vec{z}|\vec{l}, \alpha) = \int_\theta P(\vec{z}|\vec{l}, \theta)P(\theta|\alpha)d\theta$ where:

$$\begin{aligned} P(\vec{z}|\vec{l}, \theta) &= \prod_{d=1}^{D}\prod_{i=1}^{W_d} P(z_{d,i}|l_{d,i}, \theta_{d,l_{d,i}}) \\ &= \prod_{d=1}^{D}\prod_{i=1}^{W_d} \theta_{d,l_{d,i},z_{d,i}} \\ &= \prod_{d=1}^{D}\prod_{j\in\Lambda_d}\prod_{k=1}^{K_j}(\theta_{d,j,k})^{n_{d,j,k,\cdot}} \end{aligned} \tag{5.1}$$

Here we have introduced $n_{d,j,k,t}$ as the number of occurrences of label $j \in \Lambda_d$ topic $k \in \mathbb{K}_j$ within document d as applied to term $t \in \mathbb{V}$. In this notation, we sum out counts using "$\cdot$" and select a vector of counts using ":", so for example $n_{d,j,k,\cdot}$ refers to $\sum_{t=1}^{V} n_{d,j,k,t}$ or the number of occurrences of label j and topic k in document d. Similarly, $n_{d,j,:,\cdot}$selects the vector of size K_j with the term at position k equal to $n_{d,j,k,\cdot}$, which will be used below. After multiplying by θ's Dirichlet prior

and applying the standard Dirichlet-multinomial integral, we see that $P(\vec{z}|\vec{l},\alpha) = \prod_{d=1}^{D}\prod_{j\in\Lambda_d}\frac{\Delta(n_{d,j,:,.}+\vec{\alpha})}{\Delta(\vec{\alpha})}$ making use of the notation in [57] where $\Delta(\vec{x}) = \frac{\prod_{k=1}^{\dim\vec{x}}\Gamma(x_k)}{\Gamma(\sum_{k=1}^{\dim\vec{x}}x_k)}$ and we treat $\vec{\alpha}$ as a vector of size K_j with each value equal to α. Note that because each label has its own distinct subset of topics, the topic assignment alone is sufficient to determine which label was assigned, so there is no need to represent $\vec{l}$ explicitly in order to compute $n_{d,j,:,.}$.

Now let's return to the computation of $P(\vec{l}|\vec{\Lambda},\Phi,\alpha)$, which, because Λ is considered observed, can be factorized into:

$$P(\vec{l}|\Lambda,\psi)P(\psi|\alpha,\Lambda) \;=\; \int_{\beta}\prod_{d=1}^{D}P(\psi_d|\alpha,\Lambda_d)\prod_{i=1}^{W_d}P(l_{d,i}|\Lambda_d,\psi_d)d\beta$$

By re-indexing over label types, and applying the standard Dirichlet prior and Dirichlet-multinomial integral to get our final probability:

$$P(\vec{l}|\Lambda,\vec{\alpha_L}) = \prod_{d=1}^{D}\prod_{j\in\Lambda_d}\frac{\Delta(n_{d,:,:,.}+\vec{\alpha_L})}{\Delta(\vec{\alpha_L})}$$

Because in the current setting we treat Λ_d as observed, we do not need to explicitly account for the prior term $P(\Lambda_d|\Phi)$ in this computation.[1]

Observe that the actual values of $\vec{l}$ are never used explicitly, and because every topic takes part in only a single label, we can represent the model using a Gibbs sampler tracking only the topic assignments $\vec{z}$. We do not need to allocate memory to represent which label $\vec{l}$ is assigned to each token. After combining terms, applying Bayes rule, and folding terms into the proportionality constant, the sampling update formula for assigning a new label and topic to a word token is defined as follows:

[1] In a label prediction setting we could incorporate a value such as the output of L tosses of a Φ-coin, in which case we have that $P(\Lambda_d|\Phi) = \Phi^{|\Lambda_d|}\cdot(1-\Phi)^{L-|\Lambda_d|}$.

$$
\begin{aligned}
& P(l_{d,i} = j, z_{d,i} = k | l_{\neg d,i}, z_{\neg d,i}, w_{d,i} = t; \alpha, \eta) \\
\propto \; & I[j \in \Lambda_d \wedge k \in 1..K_j] \left(\frac{n^{(\neg d,i)}_{\cdot,j,k,t} + \eta}{n^{(\neg d,i)}_{\cdot,j,k,\cdot} + V\eta} \right) \cdot \\
& \left(\frac{n^{(\neg d,i)}_{d,j,\cdot,\cdot} + (\vec{\alpha}_L)_j}{n^{(\neg d,i)}_{d,\cdot,\cdot,\cdot} + \sum_{j' \in \Lambda_d} (\vec{\alpha}_L)_{j'}} \right) \left(\frac{n^{(\neg d,i)}_{d,j,k,\cdot} + \alpha}{n^{(\neg d,i)}_{d,j,\cdot,\cdot} + K_j \alpha} \right) \\
\propto \; & I[j \in \Lambda_d \wedge k \in 1..K_j] \left(\frac{n^{(\neg d,i)}_{\cdot,j,k,t} + \eta}{n^{(\neg d,i)}_{\cdot,j,k,\cdot} + V\eta} \right) \cdot \left(n^{(\neg d,i)}_{d,j,k,\cdot} + \alpha \right)
\end{aligned} \tag{5.2}
$$

The notation $n^{(\neg d,i)}$ refers to the corresponding count excluding the current assignment of topic z and label l in document d position i. Here we have used the definition of $\vec{\alpha_L}$ at position j is αK_j, which allows the numerator in the second fraction to cancel the denominator in the last term. Because the denominator in the second fraction is independent of the topic and label assignment, it is folded into the proportionality constant. Interestingly, this sampler's update rule is like that of Latent Dirichlet Allocation [48] with the intuitive restriction that only those topics corresponding to the document's labels may be sampled.

The similarity of the model and the resulting sampling equations suggests some interesting contrasts to existing models. In particular, if we use PLDA in a purely unsupervised setting with no labels beyond the *latent* label class of k topics, the model reduces exactly to traditional LDA. At the other extreme, if every document has only a single label, if we have no *latent* topic class, and if we give each label's class a single topic, our model's per-class learning function becomes the same count and divide of terms within a class as used in the multinomial naive Bayes model [87]. Similarly, if we have no *latent* topic class, and if we give each label access to only a single topic by setting $K_l = 1$ for all labels l, then the model reduces to Labeled LDA [112]. Interestingly, Labeled LDA can be used to approximate PLDA by the construction of a synthetic label space where, for any given label l, we construct a class of labels of size K_l as labels "l-1 l-2 l-3 ... l-K_l" with all those labels are applied to every document with label l. In this case, Labeled LDA will output multiple versions

of the same label which, if symmetry is broken during initialization, may result in topics that look like our latent sub-labels in PLDA but has no theoretical guarantees as such. This construction was applied to microblogging data from Twitter by me and colleagues in [111] to good effect, seeding the development of the models in this chapter.

Learning and Inference

An efficient Gibbs sampling algorithm can be developed for estimating the hidden parameters in PLDA based on the collapsed sampling formula in Equation 5.2. Efficient computation of the counts n can be done by keeping histograms over the number of times each term has been associated with each topic within each document and how often each topic has been associated with each term. The Gibbs sampler simply loops over the corpus, re-assigning topic assignment variables z and updating the corresponding histograms. However, Gibbs sampling is inherently sequential and we would like this model scale to the size of modern web collections, so we developed a parallelizable learning and inference algorithm for PLDA based on the CVB0 variational approximation to the LDA objective as described in [4]. For each word at position i in each post d, the algorithm stores a distribution $\gamma_{d,i}$ over the likelihood that each label and topic generated that word in that document using the normalized probabilities from the Gibbs sampling update formula in Equation 5.2. These distributions are then summed into fractional counts of how often each word is paired with each topic and label globally, denoted $\#_{j,k,w}$, and how often each label appears in an each document, denoted $\#_{d,j,k}$. The algorithm alternates between assigning values to $\gamma_{d,i,j,k}$ and then summing assignments in a counts phase. The update equations are listed below. Initially, we use small random values to initialize $\#_{j,k,w}$ and $\#_{d,j,k}$.

Assign:

$$\gamma_{d,i,j,k} \quad \propto \quad I[j \in \Lambda_d, k \in 1..K_j] \cdot \frac{\#_{j,k,w} - \gamma_{d,i,j,k} + \eta}{\#_{j,k} - \gamma_{d,i,j,k} + W\eta} \cdot (\#_{d,j,k} - \gamma_{d,i,j,k} + \alpha)$$

Count:

$$\begin{aligned} \#_{d,j,k} &= \sum_i \gamma_{d,i,j,k} \\ \#_{j,k,w} &= \sum_{d,i} \gamma_{d,i,j,k} \cdot I[w_{d,i} = w] \\ \#_{j,k} &= \sum_w \#_{j,k,w} \end{aligned}$$

The references to $\gamma_{d,i,j,k}$ on the right side of the proportionality in the assignment phase refer to the value at the previous iteration. This formulation allows for a data-parallel implementation, by distributing documents across a cluster of compute nodes. Assignments are done in parallel on all nodes based on the previous counts $\#_{d,j,k}$, $\#_{j,k,w}$ and $\#_{j,k}$ (initially small random values). The resulting assignments $\gamma_{d,i,j,k}$ are then summed in parallel across all compute nodes in a tree sum, before being distributed to all compute nodes for a new assignments phase. The process repeats until convergence. Like in [4], we find that the CVB0 learning and inference algorithm converges more quickly than the Gibbs sampler to a solution of comparable quality. In practice, we find that this algorithm scales to very large datasets—experiments on a corpus of one million PhD dissertation abstracts resulted in models that trained in less than a day on a cluster of twelve 4-core machines.

5.2.2 Partially Labeled Dirichlet Process

PLDA provides a great deal of flexibility in effectively defining the space of latent topics to learn both within labels and in a common latent space. Unfortunately, PLDA introduces an important new parameter for each label, K_l, representing the number of topics available within each label's topic class. Fortunately, non-parametric statistical techniques can help estimate an appropriate size for each per-label topic set automatically. In particular, we replace PLDA's per-label topic mixture θ_l with a Dirichlet process mixture model [98], which can be seen as the infinite limit of the finite mixture of topics per label used in PLDA. Formally, PLDP assumes a generative process similar to PLDA, with a multi-set of words $\vec{w}_d$ for each document and an

observed set of labels Λ_d. Like in PLDA, each word $w_{d,i}$ has an associated label variable $l_{d,i}$ and topic variable $z_{d,i}$. Here, the label $l_{d,i}$ is drawn from a document-specific multinomial over labels, which for efficiency we assume is drawn from a symmetric Dirichlet prior with parameter α. To generate a topic assignment $z_{d,i}$, PLDP picks an existing topic within label $l_{d,i}$ for word $w_{d,i}$ in proportion to how often it is used, or generates a new topic with held-out mass parameter α (the same as the Dirichlet prior for the document-specific multinomial over labels).

The word $w_{d,i}$, is then generated according to the topic distribution $\phi_{l_{d,i}z_{d,i}}$ as in PLDA. The Gibbs sampling formula for updating the joint label and topic assignment $l_{d,i}$ and $z_{d,i}$ in PLDP is:
$P(l_{d,i} = j, z_{d,i} = k | l_{\neg d,i}, z_{\neg d,i}, w_{d,i} = t; \alpha, \eta)$

$$\begin{aligned} &\propto \quad I[j \in \Lambda_d] \cdot \left(\frac{n^{(\neg d,i)}_{\cdot,j,k,t} + \eta}{n^{(\neg d,i)}_{\cdot,j,k,\cdot} + V\eta} \right) \left(\frac{n^{(\neg d,i)}_{d,j,\cdot,\cdot} + \alpha}{n^{(\neg d,i)}_{d,\cdot,\cdot,\cdot} + \alpha|\Lambda_d|} \right) \\ &\quad \cdot \begin{cases} \frac{n^{(\neg d,i)}_{d,j,k,\cdot}}{n^{(\neg d,i)}_{d,j,\cdot,\cdot} + \alpha} & \text{for } k \text{ existing} \\ \frac{\alpha}{n^{(\neg d,i)}_{d,j,\cdot,\cdot} + \alpha} & \text{for } k \text{ new} \end{cases} \end{aligned}$$

$$\propto \quad I[j \in \Lambda_d] \cdot \left(\frac{n^{(\neg d,i)}_{\cdot,j,k,t} + \eta}{n^{(\neg d,i)}_{\cdot,j,k,\cdot} + V\eta} \right) \cdot \begin{cases} n^{(\neg d,i)}_{d,j,k,\cdot} & \text{for } k \text{ existing} \\ \alpha & \text{for } k \text{ new} \end{cases} \tag{5.3}$$

As in the Gibbs expression for PLDA in Equation 5.2, we cancel the numerator in the second fraction with the denominator in both versions of the final term. Again, the denominator in the second fraction is independent of label and topic assignments, so it is folded into the proportionality constant. The Gibbs re-assignment parameters in Equation 5.3, paired with data structures updated to reflect the appropriate counts of interest at reassignment, can be used to create an efficient Gibbs sampling algorithm for the Partially Labeled Dirichlet Process. Unfortunately, the embedded Dirichlet process mixture model complicates the parallelizability of learning and inference in

this model.

It is worth noting that PLDP's embedding of the Dirichlet Process is, in some ways, an even more natural fit than in standard topic modeling applications such as the Hierarchical Dirichlet Process [127]. HDPs and related models discover a global set of latent topics within a corpus as a function of both the concentration parameter α and the corpus being analyzed. So for a known corpus of interest, text mining practioners still have a single parameter to choose—instead of picking the number of topics, they pick a concentration parameter. In practice, this is often no easier than picking the number of topics directly. In contrast, for PLDP, a single DP concentration parameter α selects the number of topics for each label in $\mathbb{L}$, effectively reducing the number of model parameters related to topic cardinality from $|\mathbb{L}|$ to one, α.

5.3 Case studies

We illustrate applications of PLDA and PLDP to partially supervised text mining tasks on two kinds of labeled corpora with very different distributional properties: PhD dissertation abstracts annotated with subject code designations and tagged web pages from Delicious. Our PhD dissertation dataset contains over 1 million United States PhD dissertation abstracts from the ProQuest UMI database[2] since 1980, averaging 2.08 subject codes out of a controlled subject code vocabulary of 259 common codes representing curated by ProQuest staff. These subject codes correspond to high-level field designations such as biochemistry, public administration, cultural anthropology, etc. Each document contains 179 non-stop words, corresponding to about two paragraphs of text from each abstract. Our Delicious dataset is a subset of 3,200 popular, heavily tagged documents from the Stanford Tag Crawl Dataset [58] collected in the Summer of 2007, with an average length of 1263 words from a word vocabulary of 321,062 terms, and an average of 122.1 distinct tags out of a vocabulary of 344,540 tags.

These datasets have very different distributional statistics, both in terms of the

[2] http://www.proquest.com/en-US/products/dissertations/

underlying texts and the label spaces. The Delicious documents are longer and have high overlap in common tags, whereas the dissertations tend to be shorter and carefully filed in a small number of subjects. Where not otherwise specified, we used fixed hyperparameters of 0.1 for α and η. In the following subsections, we examine these datasets from the partially supervised text mining perspective, finding that, despite their differences, both datasets can be effectively modeled. Because of the size of the dissertation dataset in the case study below, we focus on qualitative results that can be achieved through our parallelized PLDA model. Because of the smaller size of the Delicious data, we use the Delicious case study to quantify our intuitions about the model's ability to approach text mining challenges and compare PLDA with PLDP.

5.3.1 PhD Dissertation Abstracts

Traditional digital libraries often annotate documents with a controlled vocabulary maintained by domain experts to ease indexing, searching, and browsing. While these collections represent a shrinking fraction of all the world's electronic text, they do contain some of the most focused and important content within a limited domain. One such collection is the UMI database of PhD dissertation abstracts maintained by ProQuest. We collected 1,023,084 PhD dissertation abstracts from the Proquest UMI database filed by US students since 1980 from any of 151 schools classified as research-intensive by the Carnegie Foundation since 1994.[3] This data set is discussed and analyzed in detail in Chapter 7. In this section, we present an initial qualitative exploration of the UMI dissertations as a way to demonstrate PLDA's effectiveness on a dataset with contrasting characteristics to Delicious.

While the subject codes in our data cover the full range of academic fields, they are not evenly distributed in usage, reflecting real differences in field sizes. Indeed, the most common subject code in our dataset (electrical engineering) has 44,551 instances, whereas the least common (african literature) has only 1,041. Models like PLDA are a natural fit for analyzing these controlled-vocabulary document collections due to their ability to model both the text content in terms of latent usages of the known indexing

[3] http://classifications.carnegiefoundation.org/resources/

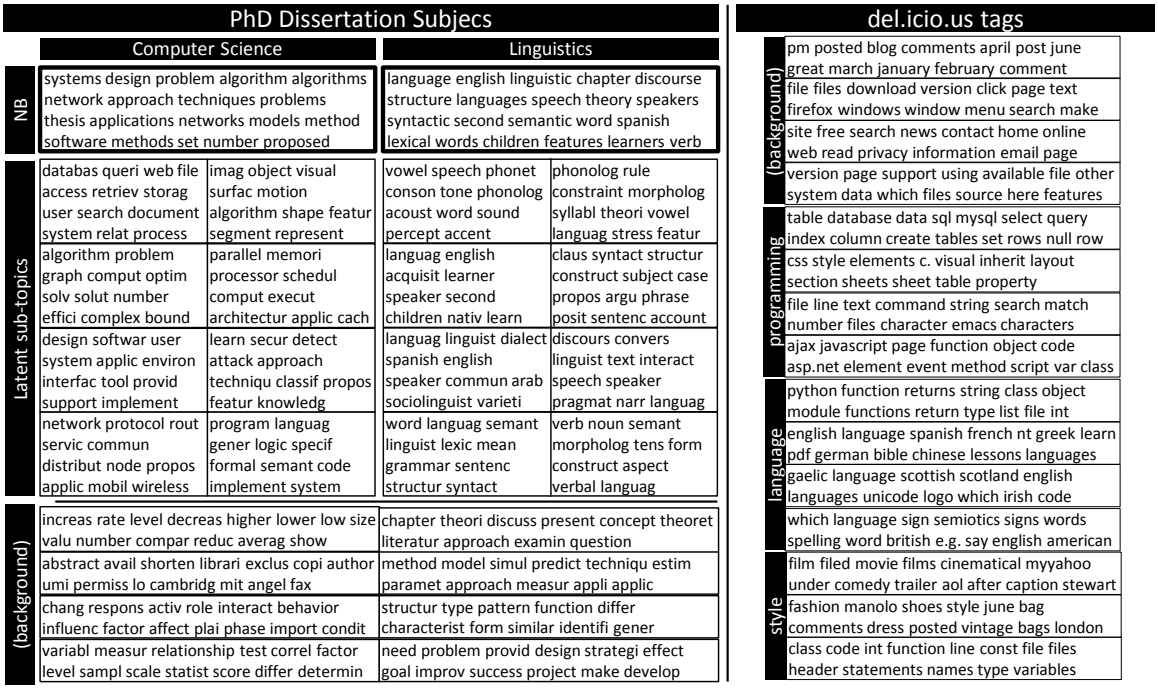

Figure 5.2: PLDA output on dissertation abstracts (left) and Delicious tags (right). Computer Science and Linguistics are two subject codes. "NB" (upper, left) refers to the naive Bayes term estimates associated with each respective code, contrasted with the latent topics learned within each. The "(background)" class (for Delicious in upper right, dissertations in lower-left) is the latent topic class shared by all documents in the respective collection.

vocabulary. By contrast, latent topics on this dataset collapse distinctions between small fields (folding them into a single topic) and overly emphasize the importance of larger ones, just based on the amount of support in the data. For example, one run of LDA on this dataset using 100 latent topics dedicated topics to fields in propotion to their prevalence in the data: electrical engineering was assigned three topics, whereas african literature was split between one topic related to all forms of race culture in America ("american, black, white, ethnic, african") and another on all forms of literature ("literari, novel, narr, text, writer"). By seeing which subject codes appeared in each topic, we can see that these two topics are themselves dominated by larger subjects: anthropology and political science for the former and modern and classical literature for the latter. This result is reasonable from the perspective of how much support there is for topics in the dataset. But by conflating smaller subject codes into a single topic, we lose the ability to describe topic dimensions in terms of the known, human interpretable objects of study (fields) while simultaneously losing all latent sub-structure within each field.

As a modeling alternative, we could train an independent topic model on all dissertations in each subject code. However, almost all dissertations have more than one subject code, with 2.08 on average and a maximum of 15. As a result, many words in the corpus will be double counted whereas PLDA can determine attribute each word in each dissertation to the appropriate subject code's latent topics. More concretely, using PLDA as a modeling framework allows for the automatic construction of shared latent background topics that pull common words found in most abstracts out of the per-field latent topics. The background topics in PLDA are explicitly labeled as background topics by the model so the practitioner does not need to manually sort the content topics from the background topics as they would for each subject code's independently trained topic model. Examples of these latent topics are shown in Figure 5.2 along with latent sub-topics discovered for several disciplines. Due to the size of the dataset, we used a distributed implementation of PLDA to learn a model with eight global latent background topics and eight latent topics per subject area, resulting in a total of 2,080 latent topics. The results shown are representative of the quality of discovered topics across all academic disciplines. Note that the major

distinctions within each subject code roughly correspond to the broad areas of study within computer science and linguistics. The latent topics capture shared common structure in PhD dissertations,[4] including basic things such as variables that increase or decrease, rates of change, and structural starting points about needs, problems, and goals.

Although this section is merely descriptive, we hope it serves to illustrate the practical impact that having human-interpretable topic dimensions can bring in a text mining context to text mining practitioners and computational social scientists in particular. In the next section, we examine content from the social bookmarking website Delicious, and use that dataset's abundance of tags as the basis for extrinsic comparison between models.

5.3.2 Tagged web pages

Users of social bookmarking websites like Delicious bookmark the pages they encounter with single word tags [26]. In contrast to more traditional supervised learning problems, user-generated tags are not predetermined nor applied uniformly to all items. For example, the tag *language* on Delicious might be applied to web pages about human languages or programming languages. We call these variations in usage of the same tag *sub-tags*. The right half of Figure 5.2 summarizes some of the types of trends discovered within each tag on Delicious. The model was run on a randomly selected 3,200 tagged web pages from [58], using 20 tags hand-selected to be relatively common but also broad in scope: reference, design, programming, internet, computer, web, java, writing, english, grammar, style, language, books, education, philosophy, politics, religion, science, history and culture. We used five latent topics and five topics for each tag. Qualitatively, the figure illustrates the model's ability to discover meaningful sub-tags, even some with a common meaning.

Because the model was trained on only a subset of all tags, we can use the remaining tags as a form of extrinsic model evaluation for computing the correlation of our model's output with a surrogate human relatedness judgment. Such an evaluation is

[4]Note that common stopwords and very rare words in the corpus were removed before training.

Table 5.2: HTJS within a tag (left) and within sub-tags (right). % change is relative to the .0183 score for randomly selected documents.

	Docs by tag		Docs by sub-tag	
Tag	**HTJS**	**Change**	**HTJS**	**Change**
books	.0254	39%	.1292	605%
computer	.0362	97%	.1609	777%
culture	.0259	41%	.0780	326%
design	.0269	47%	.0510	178%
education	.0206	12%	.1784	873%
english	.0263	44%	.0531	189%
language	.0314	71%	.1996	989%
style	.0290	58%	.2244	1124%
Overall	**.0273**	**49%**	**.1191**	**550%**

preferable to the standard perplexity-based evaluations common in topic modeling, which have been shown to disagree with human judgments of topic quality, such as in [29]. Here, we refer to the tags not explicitly modeled as held-out tags. In our experiments, most tags are held-out (128 / 132 per document, on average). Because two related documents are more likely to be tagged the same way, overlap between their held-out tags is a natural surrogate gold-standard metric for those pages' relatedness. Formally, we measure the relatedness of a pair of documents d_1 and d_2 as their *held-out tag Jaccard score* (HTJS), defined to be the Jaccard coefficient of overlap in their held out tag sets, $G(d_1)$ and $G(d_2)$, respectively: $HTJS(d_1, d_2) = \frac{|G(d_1) \cap G(d_2)|}{|G(d_1) \cup G(d_2)|}$. To measure the average relatedness within a group of documents, we randomly select k pairs of distinct documents from within the group, with replacement. Here we set $k = 500$, finding little deviation in a set's scores across different random initializations and finding no significant impact from increasing k.

HTJS is a sensible basis for evaluating the effectiveness of our model at capturing latent sub-structure in the data. We computed HTJS on a random subset of all the documents in our dataset, finding the average score to be 0.0183, showing relatively little overlap in tags of randomly chosen pages, as expected. We expect that pairs of documents that are both tagged with t will have higher held-out tag similarity than the baseline, and indeed, documents tagged with *computer* (which is not a held-out tag) have an average HTJS score of .0362, a 97% increase over the set of all documents.

The center columns in Figure 5.2 show the improvement in HTJS scores from some of the 20 modeled tags in the dataset. On average, grouping by tag increases HTJS scores by 49%, in line with our expectation that knowing the document's tag tells us something about its other tags. We can further utilize HTJS to quantify our model's ability to isolate coherent sub-tags within a tag. The HTJS score for sub-tag s of tag t is computed on all documents labeled with tag t that use sub-tag s with at least as much probability mass as the sum of the other sub-tags s' of t. For example, the HTJS score of the documents using tag computer's first sub-tag ("security news may version update network mac") scores as high as 0.312, improving the HTJS score of just knowing computer by another 31%. The right-most columns in Figure 5.2 report the HTJS score averaged across all sub-tags of the tag named in the left-most column. Not all documents tagged with t will necessarily participate in one of these subsets, as not all documents will be guaranteed to be strongly biased toward one sub-tag. The large improvements (550% relative to the baseline and 336% relative to the single tag) shown in Figure 5.2 demonstrates PLDA's ability to model coherent sub-usages of tags.

5.3.3 Model comparison by HTJS Correlation

In this section we use HTJS to compare PLDA and PLDP to several strong baselines. Better performing models should have better agreement with HTJS similarity scores across a wide range of document pairs. We quantify this intuition with Pearson's correlation coefficient: for any given model, we compute the correlation of similarity scores generated by the model with HTJS scores over 5,000 randomly selected document pairs. Higher correlations mean that the similarity score implied by the model better aligns with our surrogate human judgments.

Figure 5.3 shows the correlation of PLDP, PLDA, LDA, Labeled LDA, and tf-idf cosine similarity with HTJS scores as the total number of latent topics changes. The way we compute similarity scores depends on the model form: the partially supervised models introduced here, like other topic models, project documents into a lower dimensional topic space through their per-document topic loadings. In the case

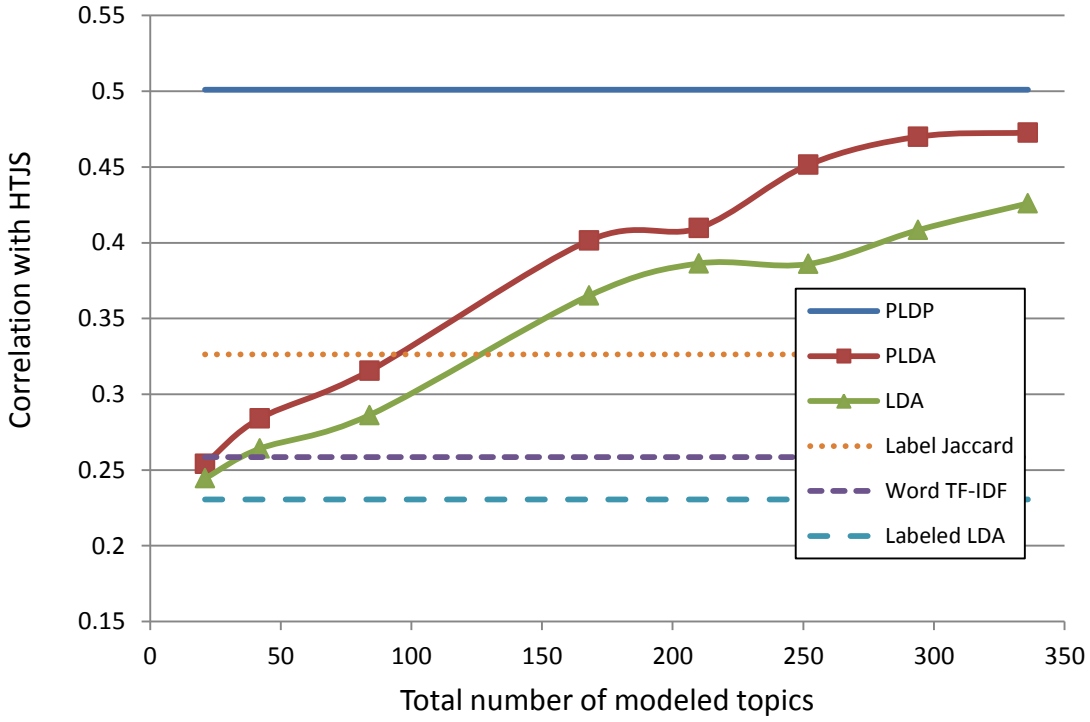

Figure 5.3: Correlation with HTJS scores for varying numbers of topic dimensions (PLDA, LDA) or as decided by model form (PLDP, Labeled LDA). Higher is better.

of standard topic models such as LDA, this loading is just the per-document topic distribution θ, which we compare using cosine similarity.[5] For PLDA and PLDP, we take a document's "θ" to be the concatenation of the documents' topic loading on all labels (even those not present in the document), resulting in a vector that is dense for topics corresponding to the document's labels and zero elsewhere.

We also included two baselines: tf-idf cosine simliarity (in word space) and the Jaccard score of the modeled (i.e. not held-out) tags. For all models, we used fixed hyperparameters of $\alpha = .01$ and $\eta = .01$. Along the x-axis is the total number of latent topics used by PLDA (varying the number of topics allocated per class from 1 to 16) and of LDA. Labeled LDA has a horizontal line corresponding to using 20 topics, one per class (and no latent class) and performs substantially worse than the other models because of its inability to model the sub-structure of each tag. PLDP demonstrates a higher correlation with the HTJS scores across the whole dataset by adapting to the label and word distributions in the data. PLDP's embedded Dirichlet process allows it to allocate different numbers of topics to each tag as a function of its concentration parameter α. Here, our PLDP model allocated 293 topics with substantial probability mass (and several hundred more occuring with very low frequency). These topics were allocated differentially according to the frequency of each tag and the variety of ways in which it is used—most were given to the latent class and common tags such as design, politics, and internet. Only four topics were allocated to the least common tag in the dataset (*grammar*). We experimented with several values of α for PLDP, resulting in more or fewer topics, but with similar distributions of topics allocated to each tag and similar overall performance results.

5.4 Scalability

The expense of adding more label classes is directly proportional to how many documents each label participates in, and is always faster than modeling more global latent topics. Indeed, the impact of a label l's topics on running time appears only in

[5] We have found cosine similarity to be a stable and high performing metric in this context in contrast to information theoretic scores such as KL-divergence.

computing the sampling proportions in documents with $l \in \Lambda_d$. This allows PLDA models such as those trained on the PhD dissertation dataset to scale to very large topic spaces and in an appreciably shorter period of time—indeed, training our 8 topics-per-subject PLDA model on one million abstracts ran in under a day on a small cluster of multi-core computers. Training a comparable number of latent topics (2,080) on this dataset took, on average 82 times longer per iteration. Incorporating a large sparse label space, such as Twitter hashtags (described in the next chapter), has little impact on the model's running time when holding the number of global latent topics fixed: even with more than 6 times as many parameters, we found an average increase of only 5.8 seconds per iteration across a wide range of sizes of the shared latent class and no impact on the convergence rate of our parallelized PLDA implementation. Adding rare label classes is computationally inexpensive, but opens up new possibilities in flexibly modeling annotations and scales to very large datasets with sufficient computational resources.

On collections with more common labels that have a higher degree of overlap, such as Delicious, incorporating more label classes or topics per class increases the computational load, but at a rate much slower than the cost of adding more global shared latent topics, as most tags are not applied to most documents. Figure 5.4 shows the running time per iteration (in minutes) for the collapsed variational Bayes learning algorithm on roughly twelve thousand documents from Delicious as the effective number of topics increases (using the same schedule of topics as in Figure 5.3). Even though PLDA's output better fits human similarity judgments, it is substantially faster to train. We note, however, that practitioners should use models like PLDA with care, choosing the set of labels modeled and topics per label depending on the statistics of the dataset. PLDP can help by automatically determining an appropriate number of topics per label class, but its flexibility comes at the expense of speed, as the model takes about four times longer to train than the Gibbs sampler for PLDA, and does not yet have a data parallel implementation.

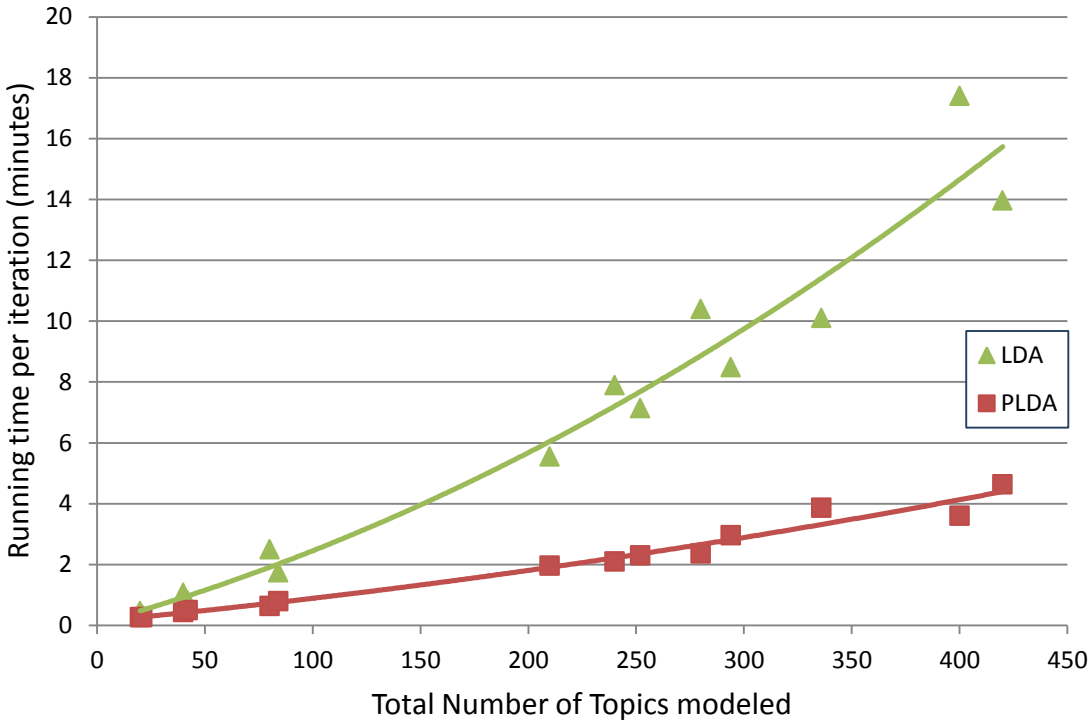

Figure 5.4: Average training time per iteration (in minutes) while varying the total number of latent topics for LDA (top) and PLDA (bottom) on tagged web pages. PLDA is substantially faster than LDA at a comparable number of topics because of the sparsity inherent to PLDA's sampling distribution.

5.5 Conclusion

This chapter introduced two topic models that incorporate label supervision in novel ways: PLDA and PLDP, which learn latent topic structure within the scope of observed, human-interpretable labels. This model represents a culmination of the modeling efforts in this dissertation toward a text mining model that is simultaneously *trustworthy* in its ability to align with a surrogate measure of human similarity, *interpretable* in that the topics it learns are associated with the labels given in the data, and *flexible* in that it can model labeled as well as unlabeled patterns.

The models introduce high-level constraints on a latent topics that cause them to align with human provided labels, essentially "filling in the details" with the use of unsupervised machine learning. The addition of these constraints improves interpretability of the resulting topics, shortens running time, and improves correlation with similarity judgments. And because these models fit into the Bayesian framework, they can be extended to incorporate other features, such as time or sequence information. Another extension could allow the labels to be treated as unobserved—to handle missing labels, for instance. Similarly, PLDA and PLDP do well with ambiguity in the label space—by uncovering latent variations of labels' usage—but do not directly model (partial-)synonymy by sharing topics across labels. Although such topic sharing would come at a computational cost (if the shared topics are discovered during inference) and would complicate the interpretation of latent topic usage, it is intuitively appealing. I believe that PLDA, PLDP, and similar future models hold promise for addressing the challenges of partially supervised learning for more interpretable text mining, where human provided labels are present but do not always align with the needs of text mining practitioners.

In the next chapters, we study two applications of these partially supervised techniques: first through an analysis of language usage in social media in Chapter 6 where some labeled and some latent patterns are modeled together, as well as a more in-depth look at the UMI dissertation database in Chapter 7.

Chapter 6

Mining and interpreting microblogs

A central theme of this dissertation is that textual data can be used as an effective lens for understanding domains of human endeavor. In this chapter, we explore traces of people's social interactions in microblog posts from Twitter by the application of the modeling techniques developed in the previous chapter. Millions of people turn to microblogging services to gather real-time news or opinion about people, things, or events of interest. These services are used for social networking, e.g., to stay in touch with friends and colleagues. And they are increasingly used as a publishing platforms for creating and consuming content from sets of users with overlapping and disparate interests. Understanding the types of content that people create and consume is critical both to our understanding of people's behavior on Twitter as well as to addressing new categories of content-driven information needs.

Microblogging services must now support information needs above and beyond their traditional roles as social networks. However, most users' interaction with Twitter is still primarily focused on their social graphs, forcing the often inappropriate conflation of "people I follow" with "stuff I want to read." Consider a hypothetical user @jane who follows user @frank because of the latter's posts about college football. However, @frank additionally uses Twitter to coordinate social arrangements with friends and occasionally posts political viewpoints. Currently, @jane has few tools to

This chapter draws from group work published as "Characterizing microblogs with topic models" in ICWSM 2010 by D. Ramage, S. Dumais, and D. Liebling. [111]

filter non-football content from @frank. In short, Twitter assumes that all posts from the people @jane follows are posts she wants to read. Similarly, @jane has a limited set of options for identifying new people to follow. She can look at lists of users in the social graph (e.g., those followed by @frank), or she can search by keyword and then browse the returned tweets' posters. However, it remains difficult to find people who are like @frank in general or—more challengingly—like @frank but with less social chatter or different political views.

The example above illustrates two of content-oriented information needs, beyond the capability of traditional network-based approaches. Content analysis on Twitter poses unique challenges: posts are short (140 characters or less) with language unlike the standard written English on which many supervised models in machine learning and NLP are trained and evaluated. Effectively modeling content on Twitter requires techniques that can readily adapt to the data at hand and require little supervision. The approach taken in this chapter makes use of the latent variable models developed in the preceding chapters. While LDA and related models have a long history of application to news articles and academic abstracts, one open question is if they will work on documents as short as Twitter posts and with text that varies greatly from the traditionally studied collections—here we find that the answer is yes.

What types of patterns can topic models discover from tweets? Section 6.2 argues from surveys and interviews that language use should be roughly categorized into four types: substance topics about events and ideas, social topics recognizing language used toward a social end, status topics denoting personal updates, and style topics that embody broader trends in language usage. Next, in the Section 6.3, we employ PLDA to map the content of the Twitter feeds into dimensions. Some of these dimensions exploit implied tweet-level labels where available, enabling models of text associated with hashtags, replies, emoticons, and the like. However, many of the most interesting patterns of language are not labeled, and can be categorized roughly into *substance*, *style*, *status*, and *social* characteristics of posts. In Section 6.4, we characterize selected Twitter users along these learned dimensions, showing that the models can provide interpretable summaries or characterizations of users' streams. Finally, Section 6.5 demonstrates the approach's effectiveness at modeling Twitter

content with a set of experiments on users' quality rankings of their own subscribed feeds.

6.1 Related work

Most of the published research about Twitter has focused on questions related to Twitter's network and community structure. For example, Krishnamurthy, et al. in [71] summarize general features of the Twitter social network such as topological and geographical properties, patterns of growth, and user behaviors. Others such as Java, et al. [64], argue from a network perspective that user activities on Twitter can be thought of as information seeking, information sharing, or as a social activity.

Less work has presented a systematic analysis of the textual content of posts on Twitter. Recent work has examined content with respect to specific Twitter conventions: @user mentions in [61] and re-tweeting, or re-posting someone else's post in [21]. Notably, Naaman, et al. [97] characterizes content on Twitter and other "Social Awareness Streams" via a manual coding of tweets into categories of varying specificity, from "Information Sharing" to "Self Promotion." Naaman, et al., extrapolate from these categories, inducing two kinds of users: "informers" that pass on non-personal information and "meformers" that mostly tweet about themselves. Others have proposed forms of content analysis on Twitter with specific focuses, such as modeling conversations [117]. Although rich with insight, these works do not present automatic methods for organizing and categorizing all Twitter posts by content, the problem we approach here.

6.2 Understanding following behavior

What needs drive following and reading behavior on Twitter, and to what extent does Twitter satisfy them? To help organize our own intuitions, we conducted in-depth structured interviews with four active Twitter users (with number of following and followed users ranging from dozens to thousands), and followed up with a web-based survey of 56 more users. We found that both the content of posts and social

factors played important roles when our interviewees decided whether to follow a user. Distilling our conversations down to their essence, we found that all those interviewed made distinctions between people worth following for the subjects they write about (substance, e.g., about a hobby or professional interest), because of some social value (social, e.g., for making plans with friends), because of (dis)interest in personal life updates from the poster (status, e.g., where someone is or what they are doing), or because of the tone or style of the posts (style, e.g., humor or wit).

To examine these intuitions in a broader context, we conducted a web-based survey cataloging reasons that underlie users' following decisions on Twitter, as determined from our interviews and other direct interaction with regular Twitter users. 56 respondents within Microsoft completed the survey during one week in November 2009. 65% were male and 75% were between the ages of 26 and 45. 67% were very active consumers of information, reading posts several times a day. 37% posted more than once per day, and 54% posted with frequency between once a day and once a month. While this sample does not represent the full range of Twitter's demographics, we believe it provides useful insight into challenges facing Twitter users more generally.

Respondents were asked how often they considered 26 reasons when making decisions about whom to follow, with most reasons falling into one of the substance, status, social and style categories identified earlier. Each respondent rated each reason on a five-point scale: "rarely," "sometimes", "about half the time," "often," to "almost always." The most common reasons for following represent a mixture of the four categories of reasons: the two most common reasons were "professional interest" and "technology" (substance). These particular substantive topics reflected the demographics of the respondents. The next most commonly used reasons were "tone of presentation" (style), "keeping up with friends" (social), "networking" (social), and "interested in personal updates" (status). Low ranked reasons included "being polite by following back" and "short-term needs (like travel info)."

Respondents were also queried about nine reasons for un-following users, i.e. removing users from their streams. We found that "too many posts in general" was the most common reason for a user to be un-followed. Other common reasons were: "too much status/personal info" (status), "too much content outside my interest set"

(substance), and "didn't like tone or style" (style). Respondents rarely un-followed for social reasons like "too many conversations with other people." The least common reason was, unsurprisingly, "not enough posts" – because such users are rarely seen by their followers simply by lack of activity. 24 users provided additional reasons for un-following: 10 mentioned spam, 8 mentioned insufficiently interesting / boring / duplicative posts, and 6 un-followed because of offensive posts (e.g. religious or political views, general tone, or about other people).

In response to an open-ended question about what an ideal interface to Twitter would do differently, survey respondents identified two main challenges related to content on Twitter, underscoring the importance of improved models of Twitter content. First, new users have difficulty discovering feeds worth subscribing to. Later, they have too much content in their feeds, and lose the most interesting/relevant posts in a stream of thousands of posts of lesser utility. Of the 45 respondents who answered this question, 16 wanted improved capabilities for filtering of their feeds by user, topic on context (e.g., "organize into topics of interest", "ignore temporarily people, tags or topics"). In addition, 11 wanted improved interfaces for following, such as organization into topics or suggestions of new users to follow (e.g. "suggestions on who to follow that have similar interests").

6.3 Modeling posts with PLDA

The information needs outlined above point to the importance of developing better models of textual content on Twitter. The approach we use here is based on latent variable topic models inspired by LDA that incorporate some supervision in the form of the implicit tweet-level label spaces. The original paper on which this chapter is based [111] used a mixture of latent and labeled topics in a synthetic label space in Labeled LDA (Chapter 4), which inspired the PLDA model in Chapter 5. This chapter re-frames the original analysis in terms of PLDA.

6.3.1 Dataset description

We trained models on data collected by crawling one week of public posts from Twitter's "spritzer" stream. This public stream's makeup is determined by Twitter and contains posts sampled from all public posts made on the site. Our collection contains 8,214,019 posts from the 17th through the 24th of November 2009 (OneWeek). Posts were processed by tokenizing on whitespace and on punctuation subject to rules designed to keep together URLs, emoticons, usernames, and hashtags. Some multi-word entity names were collapsed into single tokens (such as michael_jackson) by using a gloss lookup derived from Wikipedia and query logs. After processing, posts contained an average of 13.1 words from a vocabulary of 5,119,312 words. As an important pre-processing step, we removed the 40 most common terms in the corpus[1] and all terms appearing in fewer than 30 documents. Some experiments were conducted on just those posts from the 24th of November (OneDay), containing just over 1M posts. It is worth noting that the number of documents in both collections is substantially larger than most applications of latent variable topic models, where collections tend to be on the order of tens of thousands of documents, although those documents are usually longer.

Besides the number and types of labels used, PLDA has two parameters: we used un-tuned symmetric Dirichlet priors of .01 for η and .01 for α, which can be thought of as pseudo-count smoothing on per-label word distributions and per-post label distributions, respectively. In early experimentation with these values, we found similar qualitative results across a wide range of small positive values.

6.3.2 Model implementation and scalability

In order to scale to our test collection size—and beyond for real-time analysis of all Twitter data—our implementation must be parallelizable. We use the CVB0 variational approximation to the PLDA objective described in Section 5.2.1 based on the work in [4]. For each word at position i in each post d, the algorithm stores

[1] The most common terms are effectively a corpus-specific collection of stop-words; removing them improves running time and the subjective quality of learned topics.

a distribution $\gamma_{d,i}$ over the likelihood that each topic (associated with some label) generated that word in that document. These distributions are then converted into counts of how often each word is paired with each topic globally, denoted $\#_{lkw}$, and how often each label appears in an each document, denoted $\#_{dlk}$. The algorithm alternates between assigning values to $\gamma_{d,i,l,k}$ and then summing assignments in a counts phase. The update equations are listed below. Initially, we use small random values to initialize $\#_{lkw}$ and $\#_{dlk}$. The references to $\gamma_{d,i,l,k}$ on the right side of the proportionality in the assignment phase refer to the value at the previous iteration.

Formulating the PLDA learning problem in this way allows for a data-parallel implementation. Documents are distributed across a cluster of compute nodes. Before each assignment phase, all nodes are given a copy of the current counts $\#_{dlk}$, $\#_{lkw}$ and $\#_{lk}$. The assignments phase is done in parallel on all processors. Then, processors aggregate their local counts by summing their assignments in parallel, and then passing along the sums to higher rank nodes until the master node has the sum of all counts. This iterative process repeats for a fixed number of iterations or until the change in model parameters falls below a threshold. Our implementation does threading within compute nodes and communicates across nodes with MPI, and can complete training on the OneWeek dataset within about four days on a 24-machine cluster.

In the results presented in this chapter, the PLDA models will contain 100 or 200 dimensions (a parameter we set) that correspond to latent trends in the data (like labels "Topic 1" through "Topic K" applied to each post), and about 500 labeled dimensions (depending on the dataset) that correspond to hashtags, etc, as described in the Section 6.3.4. After describing the characteristics of these dimensions, we go on to describe how they can be used to characterize users or sets of posts in Section 6.4 and how they impact performance on two ranking tasks in Section 6.5.

6.3.3 Latent dimensions in Twitter

Before examining the types of content captured by the labels in PLDA, we first examine Twitter's latent structure, as modeled using K labels applied to every post

in the collection. These labels are incorporated so that unsupervised large-scale trends can be captured by the model. By inspection, we find that many of these learned latent dimensions can be divided into one of the four categories defined above: those about events, ideas, things, or people (substance), those related to some socially communicative end (social), those related to personal updates (status), and those indicative of broader trends of language use (style). Later, we refer to text analyses using these categories as a 4S analysis.

We manually labeled 200 latent dimensions from one run of our model on the OneDay dataset according to the 4S categories by examining the most frequent words in each dimension's term distribution. Four raters labeled each dimension as any combination of substance, status, style, social, or other—i.e. each dimension may have more than one 4S category assignment. As an example, the most frequent words in "Topic 1" are: "watching tv show watch channel youtube episode and season," which was labeled as substance. The other dimensions tended to be dominated by non-English terms, by numbers, by symbols, or by generic word classes like terms for males (him his he boy father man, etc).

Table 6.1 summarizes the number of latent dimensions associated with each category, the inter-rater agreement in labeling, and the top words in an example dimension for each category. We used Fleiss' κ to compute inter-rater agreement for each of these categories across our four judges as separate binary classification tasks. As shown in Table 1, we find fair to substantial agreement across all categories. The social category shows the lowest inter-rater agreement, which is in part because so much language usage on Twitter has some social component, regardless of whether it is also substantive, stylistic, etc. Indeed, boyd, et al. [21] report that 36% of posts mention another user, and of those roughly 86% are directed specifically to that user. As a caveat, categorizing latent dimensions in this way can be difficult for three reasons. First, the judgments (and even our categories) are inherently subjective, although we do find reasonable agreement. Second, some legitimate trends may be hidden in the lower frequency terms in each distribution. Finally, many discovered dimensions are inherently ambiguous in usage, such as some indicative linguistic styles being coupled with social intent. Nonetheless, we believe that this type of high-level summary can

Category	Fleiss' κ	Example topic
Substance 54/200	.754	obama president american america says country russia pope island failed honduras talks national george us usa
Status 30/200	.599	am still doing sleep so going tired bed awake supposed hell asleep early sleeping sleepy wondering ugh
Style 69/200	.570	haha lol :) funny :p omg hahaha yeah too yes thats ha wow cool lmao though kinda hilarious totally
Social 21/200	.370	can make help if someone tell_me them anyone use makes any sense trying explain without smile laugh
Other 47/200	.833	la el en y del los con las se por para un al es una su mais este nuevo hoy

Table 6.1: Inter-rater agreement from four raters marking 200 latent dimensions with 4S categories. Left: number of dimensions in category marked by ≥ 2 raters. Middle: Fleiss' κ showing all four categories have at least fair agreement. Right: high scoring words in an example from each category.

provide value insofar as it quantifies agreed-upon intuitions, and holds up to scrutiny when examined at the level of individual posts. In our own exploration, we found the 4S categorization corresponded to distinctions that arose commonly in the interviews, survey and content analysis and, furthermore, that there was good agreement about categorization decisions from multiple labelers.

6.3.4 Labeled dimensions in Twitter

While the latent dimensions in Twitter can help us quantify broad trends, much additional meta-data is available on every post that can help uncover specific, smaller trends. In addition to the latent dimensions discussed above, several classes of tweet-specific labels were applied to subsets of the posts. For instance, we create one label for each hashtag. A hashtag is a Twitter convention used to simplify search, indexing, and trend discovery. Users include specially designed terms that start with # into the body of each post. For example a post about a job listing might contain the term #jobs. By treating each hashtag as a label applied only to the posts that contain it, PLDA discovers which words are best associated with each hashtag. We associate one latent topic with each hashtag. Common words better described by some latent dimension tend not to be attributed to the hashtag label.

We incorporated several other types of labels into the model. Emoticon-specific labels were applied to posts that used any of a set of nine canonical emoticons: smile, frown, wink, big grin, tongue, heart, surprise, awkward, and confused. Canonical variations were collapsed: e.g. =] and :-) mapped to :). @user labels were applied to posts that addressed any user as the first word in the post, as per the Twitter convention of direct messaging. reply labels were added to any post that the Twitter API has designated as a reply, i.e. because a user clicked a reply link on another post. question labels were applied to posts that contain a question mark character. Because the emoticons, @user, reply, and question labels were relatively common, we gave each 10 latent topics to model natural variation in how each label was used. The number 10 was chosen heuristically given the relative commonality of these symbols compared to hashtags. Posts contained an average of 8.8 labels out of a label vocabulary of

158,223 distinct labels. Of those labels, the majority (158,103) were hashtags; we filtered hashtags occurring on less than 30 posts, resulting in a final set of 504 labels.

Table 6.2 shows some characteristic topics associated with each label class. Natural variation in the linguistic usage is evident: one of the excerpted smile labels is used to express gratitude and another consists of various forms of social bonding ("xoxo" means hugs and kisses). Similarly, one frown label is dedicated to feeling ill, whereas another represents frustration (mostly with computers). The specificity of these labeled dimensions hints at new directions in sentiment analysis on Twitter content. One reply label is dedicated to confirmations (thanks ok good yeah) and another represents a somewhat rowdier linguistic style (lmao yea tho wat hell). Analogous distinctions are found through the other label types. We are interested in exploring applications of isolating each of these trends, such as improved browsing interfaces for hashtag labels, better sentiment analysis using emoticon labels, and conversation and question modeling using the social labels. An open challenge in formulating this kind of model is how best to select the number of sub-labels per label type, which we plan to explore in future work.

Beyond the inherent appeal of explicitly modeling these label types, their incorporation supports our 4S analysis. For example, we know that all posts that are replies or are directed to specific users are, to some extent, social, so we can count usage of any reply or @user label as usage of the social category. Emoticons are usually indicative of a particular style and/or a social intent. Because hashtags are intended to be indexed and re-found, they might naturally be labeled as substance. Although not all labels fall cleanly into the assigned categories, the great majority of usage of each label type is appropriately categorized as listed above, enabling us to expand our 4S label space without manual annotation.

6.4 Characterizing Content on Twitter

PLDA can be used to map individual posts into learned latent and labeled dimensions, which we have grouped into 4S categories – substance status style social, either manually (for 200 latent dimensions) or by construction (for 504 labeled ones). These

Emoticons	**:)**	thanks thank much too hi following love very you're welcome guys awww appreciated ah
		love all guys tweet awesome x nice twitter your goodnight followers later y'all sweet xoxo
	:(	miss sick still feeling ill can't much today already sleep triste him baby her sooo fml
		ah working won't stupid why anymore :'(isn't suck computer isnt ahh yeah nope nothing
Social Signal	Reply	thanks i'm sure ok good will i'll try yeah cool x fine yes definitely hun yep glad xx okay
		lmao yea tho yu wat kno thats nah hell lmfao idk dont doin aint naw already ima gotta we
	@user	haha yeah that's know too oh thats cool its hahaha one funny nice though he pretty yes
	?	did how does anyone know ?? ?! get where ??? really any mean long are ever see
		?! ?? !? who wtf !! huh ??? hahaha wow ?!! ?!? right okay ??!! hahahaha eh oh knew
Hashtags	#travel	travel #traveltuesday #lp hotel #ac ac tip tips #food national air airline #deals countries #tips
	#twilight	#newmoon #twilight twilight watching edward original watch soundtrack Jacob tom_cruise
	#politics	#cnn al_gore hoax climategate fraud #postrank gop inspires policy because why new bill

Table 6.2: Example word distributions learned for various classes of labels, supplementing latent topics (not shown).

mappings can be aggregated across posts to characterize large-scale trends in Twitter as well as patterns of individual usage. Formally, a post d's usage of topic k, denoted $\theta_{d,l,k}$ is computed simply as $\#_{dlk}/|d|$. We compute an aggregate signature for any collection of posts by summing and normalizing $\#_{dlk}$ across a collection of documents, such as posts written by a user, followed by a user, the result set of a query, etc. The usage of any 4S category can be determined by summing across dimensions within that category.

By aggregating across the whole dataset , we can present a large-scale view of what people post on Twitter. At the word level, Twitter is 11% substance, 5% status, 16% style, 10% social, and 56% other. Despite the common perception to the contrary, usage of substance dimensions outnumbers status dimensions on Twitter by two to one.

Other is so common because of how our 4S categorization interacts with other kinds of common trends that on Twitter. For instance, time words and numbers are contained prominently in several topics that are labeled other. The largest source of other, however, comes from the distribution of languages on Twitter. In particular, about half of user traffic comes from non-English speaking countries,[2] and the language in which a post is written is a powerful similarity signal across posts. The model effectively segregates usage of these languages into their own dimensions, which we manually labeled as other. Only once a language has enough posts will the model have enough data to subdivide by linguistic usage.

By aggregating PLDA dimensions across recent posts from two Twitter accounts, we can visually contrast their language usage. Figure 6.1 shows a 4S analysis of 200 recent posts written by a popular celebrity (@oprah, right) and by the World Wide Web Consortium (@w3c, left). In the center, we see the ratios of these two account's usage of dimensions that fall into each 4S category, denoted as stacked vertical segments drawn to scale. Background statistics for the dataset are shown as a third stacked bar in the center, from which we can see that @w3c is highly skewed toward substance, whereas @oprah has slightly more status than average. The most

[2]While we could not find an exact statistic for the distribution of languages by post on Twitter, English-speaking countries make up about 49% of user traffic (http://www.alexa.com/siteinfo/twitter.com as of November 2010).

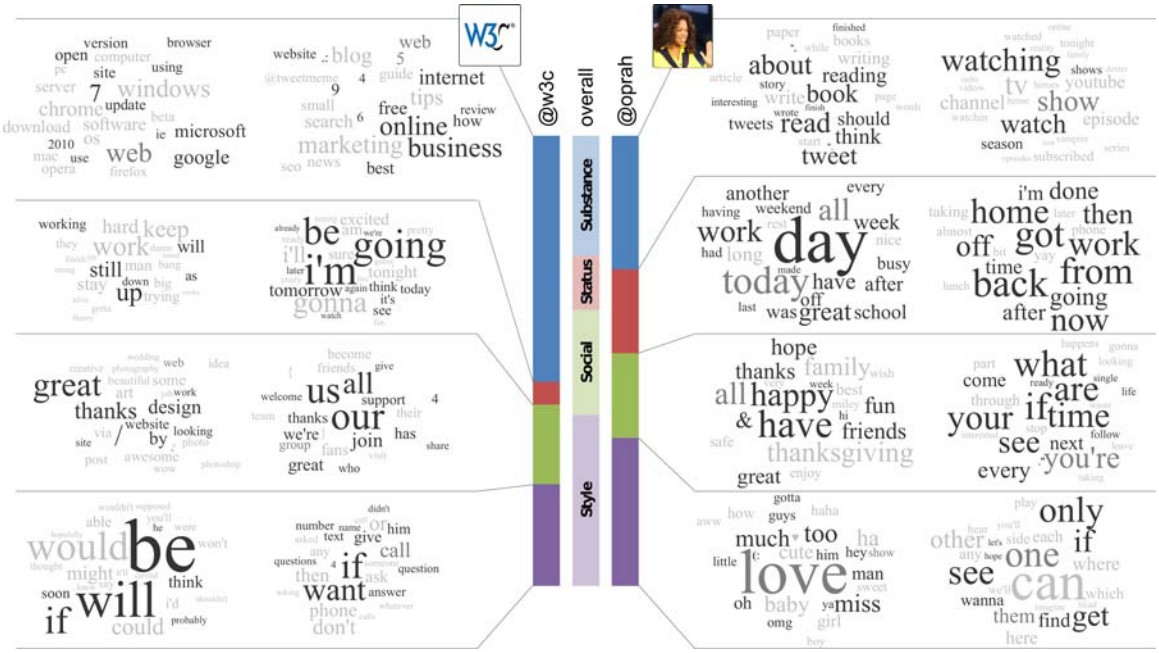

Figure 6.1: 4S analysis of two users: @w3c (left) and @oprah (right). The usage of dimensions from substance (top row), status (second), social (third), or style (bottom) categories is shown in the vertical bars, with Twitter's average usage shown in the center. Common words in selected dimensions from each category are shown as word clouds. Word size is proportional to frequency in that dimension globally, and word shade is proportional to the frequency in the user's recent tweets. Light gray words are unused in recent tweets.

common words for selected dimensions within each 4S category are shown to the left and right. The size of a word reflects how important it is in that dimension globally (i.e. in the training data), and shading depends upon how often the poster uses each word within that dimension.

Images like Figure 6.1 can be used to visually characterize and contrast users. For instance, we can see that @oprah posts about her television show (top right) and about books (adjacent in region). In particular, we see that @oprah uses the "book" dimension to talk about reading (darker) rather than writing (unshaded). Similarly, @w3c often posts about technology (top left) and the web (adjacent). Within the web topic, @w3c uses words like "internet" and "online" but not "marketing" or "seo." Socially, @w3c comes across as an open organization by using words like join, we, our and us, whereas @oprah talks to her followers (your, you're).

A scalable, interactive version of this visualization was developed and deployed to the web as *Twahpic*[3]. Two screenshots of the visualization are shown in Figure 6.2. The topic browser allows users on the web to enter a Twitter username or search query. Inference is performed on 200 returned posts matching the query using a pre-trained PLDA model, where latent topics are manually classified into 4S categories and given short descriptive titles. The posts returned by the query are shown at the left, and for each post, the fraction of its words attributed to each type of language is shown as a small adjacent bar chart. These distributions are summed, with the highest probability topics in each 4S category shown at the right. The area of each topic is proportional to the amount that it is used. The color of each topic is determined by it 4S category. From the visual aggregations alone, we can easily learn the relative prevalence of the 4S categories. In this case, we see that the Bing account (@bing) is a corporate account designed to interact with other Twitter users and engage in conversations about Bing, whereas the Microsoft Research account (@msftresearch) is more of an information-push style account that announces new results and initiatives.

To provide a context grounding our understanding in the underlying words, we show each topic box at the right with its highest probability words as a word cloud. Words in each cloud are sized in proportion to how often they are used within the

[3]http://twahpic.cloudapp.net/

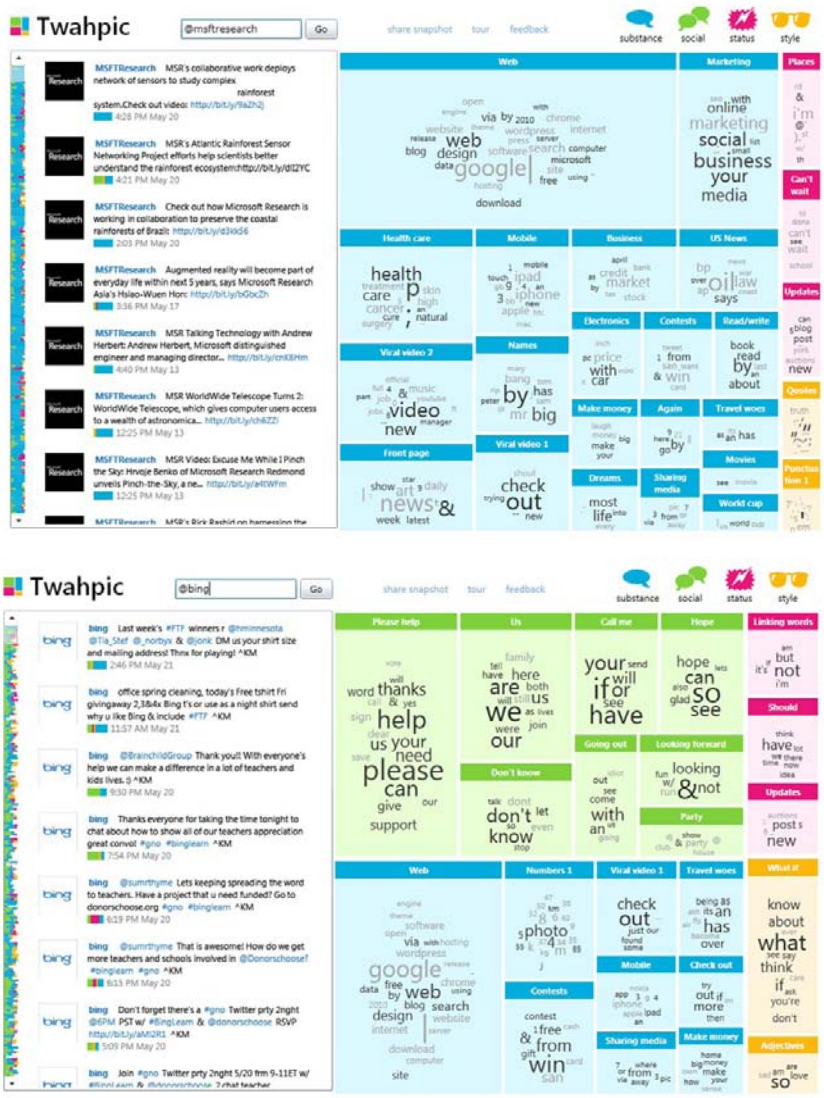

Figure 6.2: Screenshots of the interactive *Twahpic* browser's 4S analysis of two corporate Microsoft accounts: Microsoft Research and Bing. These screenshots were taken in May 2010. Note the very different aggregate distribution of *social* language (green, upper left quadrant) and *substance* language (blue, bottom left quadrant) in the two accounts. The difference stems from the alternative purposes served by the two different kinds of corporate Twitter accounts: @msftresearch is almost entirely substance, promoting new research and initiatives within the lab. @bing, on the other hand, is designed to be a customer-facing service that responds directly to other Twitter users to engage others in conversations about Microsoft's search service. Images by Dan Liebling.

topic overall and shaded according to how often each word occurs within the modeled posts at the left. The juxtaposition of size and shading allows the visualization to support a context-specific understanding of how the particular set of posts makes use of the language in each topic. For instance, we see that the Microsoft Research account (top half of Figure 6.2) uses a topic called "Web" without using the word "google" (light shading) even though "google" is commonly used by others using the "Web" topic overall.

6.5 Ranking experiments

The previous section demonstrated ways we can use PLDA with a 4S analysis to characterize sets of posts according to the model's learned dimensions. Here we examine the model from a different perspective: effectiveness at modeling Twitter content as measured by performance on two information consumption tasks. One task considers ranking posts from a person's current feed; the other is aimed at recommending new users to follow. In these experiments, we do not make use of the 4S categorization of the PLDA dimensions, instead focusing on the relative effectiveness of two representations of Twitter content: the per-post feature space defined by PLDA's per-post θ_d and standard tf-idf feature vectors built from tokenized posts. We also report the performance of a combination of these models and two baseline methods, ordering randomly and ordering by time. The PLDA model used here was a 100 latent dimension model with all labeled dimensions as described above, trained on the OneWeek dataset.

Active Twitter users within Microsoft were asked to rate the quality of posts from users they follow on a three point scale. For each participating rater, we selected up to seven posters with public feeds followed by that rater. We collected the 14 most recent posts from each poster using Twitter's public API. This collection of 7×14 posts was presented to the rater in chronological order. Each rater was asked to score the selected posts on a three point scale: 3 = "must read," 2 = "maybe worth the reading time," and 1 = "not really worth reading." 43 users completed at least 60 judgments, providing us a dataset of 4,267 judgments. Most raters in our study

Model	Mean Avg Prec	Mean Prec@1	Mean RR@1R
PLDA + tf-idf	.622	.634	.756
PLDA	.605	.537	.681
tf-idf	.608	.585	.718
Temporal	.565	.537	.678
Random	.542	.537	.670

Table 6.3: Performance on the by-rater post ranking task.

were unhappy with most posts in their feeds. The average rating was only 1.67, with a majority of posts (2,187) scored as "not really worth reading." Individual raters displayed a range of satisfaction: the median per-rater average score was 1.64, with a minimum of 1.08 and a max of 2.26.

6.5.1 By-rater post ranking task

The by-rater post ranking task models a content-driven information consumption scenario: given only a few minutes, which posts should @jane read from her feed. To evaluate this task, we split the set of judgments by rater, ordering posts chronologically. The earliest 70% of posts were taken as a training set, and the remaining were scored as a test set, with the goal of ranking the most preferred posts first. While a more involved supervised classification algorithm could be used, here we trained a simple centroid-based ranker on the positive examples (those rated as "must read" or "maybe worth the reading time") in order to compare feature spaces. Test posts were ordered by their cosine similarity to the mean feature vector of the positive examples.[4]

Table 6.3 shows the results of computing several standard IR rank evaluations (Mean Average Precision, Mean Precision @ 1, and Mean Reciprocal Rank of the first relevant item) on the resulting test sets. We compared performance for models based on raw tf-idf features computed on terms in the posts, the lower dimensional feature space of PLDA, a combination of the two, a random baseline, and a baseline based on time (the Twitter default). We observe that the tf-idf and PLDA models have similar performance, but that a weighted combination of their similarity scores

[4]For the probabilistic models, we also experimented with information theoretic measures like KL-divergence, but found them inferior to cosine similarity.

Model	Reciprocal Rank
PLDA + tf-idf	.965
PLDA	.579
tf-idf	.839
Temporal	.103
Random	.314

Table 6.4: Performance on the user recommendation task.

(18% PLDA, 82% tf-idf) outperforms all models by a substantial margin. While a full exploration of combinations of similarity models is outside the scope of this chapter, this particular mixture was picked by examining performance on a set of bootstrap samples on a fraction of our dataset; performance was fairly stable and nearly optimal across a range of values between 15% and 20% PLDA.

6.5.2 User recommendation task

The user recommendation task models a different content-driven information need: given posts from users I follow, recommend a new user to follow. In this task, we ignore the positive and negative per-post ratings, and simply model the centroids of posts from the rater's followed users. For each rater, we build a representation of their interests using posts from six of the posters that they follow, and hold out posts from the one remaining poster as a positive test example. As negative test examples we use 8 other posters that the rater does not follow. Models are compared by the extent to which they recommend the positive test user over the negative users. Specifically, we measure the reciprocal rank of the positive test example in the set of test posters. This measure is somewhat conservative since the rater may actually be interested in some people whom they don't currently follow, particularly because our negative test examples were drawn from within the same post ranking dataset. Because all raters work for the same company and share some interest in social networking, we expect there to be more similarity between followed users and non-followed users in this dataset than for Twitter as whole.

Table 6.4 shows the performance across the same models as the previous experiment. Here, the temporal baseline ranks users by their average post time, so users who posted more recently more often are ranked higher. In this task, tf-idf greatly outperforms PLDA alone, but the combination substantially outperforms either model individually. And more pointedly, the combination classifier returns a nearly perfect score of .96 – i.e. it ranks the actually followed user first in almost all test instances.

In both tasks, the best classifier was a weighted combination of these inputs. The weighted combination works well because PLDA and the tf-idf model capture different aspects of textual similarity. In particular, we expect PLDA to outperform tf-idf when document vectors share few terms in common because PLDA reduces the dimensionality of the word space to a much smaller label space. Conversely, we expect the tf-idf model to outperform PLDA when there are enough terms in common such that the occasionally spurious conflations in the reduced space do more harm than good. Because both of these similarity signals are informative, the weighted combination allowed the models to complement each other and outperform either model on its own.

6.6 Conclusion

This chapter has shown how content-based analysis of language use on social microblogs like Twitter can provide a rich characterization of the kinds of language used as a way of understanding people's interactions online. We have shown how these methods can support rich analyses of Twitter content at large scale through aggregate statistics of social, substance, status, and style language use. We have also shown their applicability to understanding individual users with interactive visualizations of the 4S dimensions: the models' lower dimensional feature representation can be used to characterize users by the topics and words they most commonly use. The approach effectively models important similarity information in posts, improving performance on two concrete tasks modeled after information needs: personalized feed re-ranking and user suggestion.

The effectiveness of the models points to clear directions for future work, including

temporal dynamics of learned models as well as the combination of content analysis with techniques from reputation and social network analysis. In combination, these models might enable us to answer more kinds of questions like: How much does the distribution of substance, status, social, and style change across parts of the social network? And how does each person's usage of language evolve over time? There are also clear implications for the development of applications of PLDA and similar models of Twitter content, including improvements in finding and following new users as well as filtering feeds to topics of interest.

More importantly, this chapter demonstrates how a mixture of latent and labeled topics—such as with PLDA—can support an effective computational approach to studying social questions on Twitter. The mixture of latent and labeled topics exploits user-provided labels to better understand language in context, such as the words associated with emoticons, mentions, etc. It also leaves room to discover unknown topics, which we make sense of using the 4S categories derived from traditional social sciences approaches (surveys and interviews). The result is a new kind of visualization and characterization of language use on Twitter that would not have been possible without a combination of our own domain insight (4S categories), the implicit domain expertise of the users (in emoticons, hashtags, etc.), and a modeling framework to coherently integrate these signals with the raw Twitter data. Taken as a whole, this chapter presents one case study in how to approach a text mining challenge with known unknowns (emoticons, hashtags, etc.) as well as unknown unknowns (latent topics organized into 4S categories) that combine to construct interpretable characterizations of the language of individuals and queries on a social microblogging site.

In the next chapter, we transition from the study of individuals and their communication—representing one kind of question traditionally approached in the social sciences at a smaller scale—to the study of ideas and the organizations that enable them.

Chapter 7

Academia through a textual lens

In this chapter, we illustrate how text can be used to analyze ideas and the organizational structures that support their creation through a statistical analysis of PhD dissertation abstracts spanning three decades. We examine borrowing of language across disciplines over time and the ways in which individual departments may lead or lag the rest of a field. This case study is an instance of the general themes of interpretable text mining for the social sciences outlined in Chapter 1: we demonstrate the ways that we can make use of the implicit domain expertise in a text collection to study the ideas and organizational structure represented in a large scale text collection. In particular, I analyze academia through a study of one million PhD dissertation abstracts filed in the United States over the past three decades, from 1980 to 2010. The analysis is with respect to three kinds of labels—the implicit domain expertise—embedded in each dissertation's metadata: the areas in which a dissertation is filed, the school at which it is written, and its year of publication.

Specifically, this chapter considers two aspects of phenomena in our dataset: *incorporation*, or the use of language or ideas from outside an academic area, and *leading and lagging*, or the use of language ahead or behind its time. We also consider the intersection of these phenomena, which we think of as the *returns from interdisciplinarity*.

Incorporation. Work that crosses disciplinary boundaries is heralded as a source of new ideas in the sciences, engineering, and humanities. That researchers from two

fields might come together to create something new, beyond the scope of each individual field, has held sway in the popular press and the sociological literature, and it has directed funding agencies and university initiatives [68]. While this coming together of people is symmetrical, the ideas, methods, we find in Section 7.4 that vocabulary of science has a directional flow. Using PLDA to model languages incorporated by fields across disciplinary boundaries, we document how methodological (computer science, statistics) and theoretical approaches (philosophy, mathematics) export their language to other fields more than they borrow from elsewhere. In addition, we find a split in the biological sciences between reductionist and system-level perspectives on biological phenomena. And we find a large-scale, sustained change in the humanities and social sciences driven by gender and ethnic studies.

Leading and Lagging. Every PhD dissertation is a product of its intellectual environment and a contribution to human knowledge. From studies of individual proteins to Middle English prose, the scope of each dissertation is usually quite focused. Yet as new ideas, methods, and technologies emerge, some dissertations will come to look more like academia's future than others. For example, some early work in DNA sequencing presaged the larger shift in biology toward computational methods by several years. In Section 7.5, we use the year of a dissertation's publication to analyze the extent to which each dissertation leads the future of academia as a whole.

Returns from interdisciplinarity. Finally, in Section 7.6, we consider the combination of interdisciplinarity scores with leading and lagging scores by university. We find that work that crosses disciplinary boundaries is substantially more future leaning than average dissertations across the entire time period.

7.1 Related work

Prior academic study of scholarship has primarily used traditional methods from the social sciences including literature reviews, expert interviews, and surveys [73, 75]. Most studies examine single fields [43] or compare several [69]. In aggregate, these case studies convey stories about the development and division of broader areas, such as the divide between STEM fields (science, technology, engineering, and math) and

the rest of academia [93]; the growth of reductionism in general and in microbiology in particular [24]; and the growth in climate science and gender and ethnic studies [137]. As we will see, these observations are largely confirmed in the patterns of language incorporation we document. However, detailed studies employing traditional methods have not scaled to similarly complete studies of academia writ large.

The few larger-scale studies of scholarship are based on link analysis rather than language use [36]. These methods analyze networks of formal variables describing links such as citations or co-authorship [99, 20, 18, 19]. Some even study interdisciplinarity as mixing in team science [123]. However, these network-based studies have two fundamental limitations. Conceptually, they are limited to formal linkages and miss the hidden structure of academia that arises in the myriad informal conversations and often-uncited distant readings of others' work. As a result, they lack a representation of how concepts may be borrowed across disciplines. Practically, they suffer from the limited availability of high quality, accurately disambiguated metadata that crosses field boundaries. Because of this, results are biased toward journal-heavy fields like biomedicine and leave book-heavy fields like the humanities under-represented. A focus on citation and authorship further favors fields with high output, short papers and short-term collaborations. A study of dissertation language usage provides a scientific tool that avoids the biased conventions of citation and authorship.

Recent noteworthy approaches to the study of academic scholarship through language analysis have focused on detailed studies of domain in which the authors have expertise. For instance, Blatt [12]—himself an anthropologist—finds textual evidence of the recent split in anthropology into cultural anthropology and anthropological science. Similarly, Hall, et al. [50] examine the rise of statistical methods in natural language processing by utilizing a combination of topic models and their own domain expertise. In this case, two of the three authors have independently written two of the best known textbooks in the field. Because of the unsupervised methods used, these studies are inherently limited to the authors' domain of expertise. By contrast, the approach we take uses PLDA to exploit the implicit domain knowledge of each dissertation filer and the ProQuest taxonomists tasked with filing each dissertation

into its appropriate subject codes. As a result, our results are applicable to studies of broader scale phenomena in academia as a whole.

While focused studies can tell us about the detailed history of a particular field, they do not illuminate larger scale patterns across academia as a whole. The textual analysis of the PhD dissertation database presented here does not suffer from the scale limitations or sample biases that previous approaches face. Dissertations are ideal for the study of interdisciplinarity because every academic writes one, embodying several years' effort to extend the state of the art in some field. We examine thirty-one years of dissertations published from research intensive universities in the UMI database maintained by ProQuest [109]: long enough for longitudinal analysis while ensuring high coverage of all schools and areas. To our knowledge, this work is the first large-scale study of academia through the lens of its graduating students. In contrast to other datasets, PhD dissertation abstracts reflect the entire academic output of a university. Every graduating PhD student produces a dissertation reflecting several years' effort. These tend to be of high quality: each is judged to be a new contribution to human knowledge by faculty members of the graduating institution.

7.2 Dataset description

Shortly after completion, most PhD dissertations in the United States are filed in the UMI database maintained by ProQuest. ProQuest is designated by the Library of Congress as the collection agency for published PhD dissertations in the United States [109]. We analyze a subset of 1.05 million dissertation abstracts filed between 1980 and 2010 from 157 schools. We selected schools that have been classified as research-intensive in one of three surveys of higher education conducted by the Carnegie Foundation since 1994 [89]. We examined only data until 1980 because electronic abstract records became much sparser before then.

Each dissertation contains a title, abstract, author, advisor, date, subject codes, and keywords. The abstracts contain an average of 179 words after removing common stop-words (such as "about" and "the"), removing rare terms occurring in less

than 5 documents, and collapsing term variations with a Porter stemmer [108].[1] 268 commonly used subject codes in our dataset are taken as implicit domain expertise—reflecting the knowledge both of the filer and the taxonomists at ProQuest—corresponding to relatively high-level field designations such as biochemistry, public administration, cultural anthropology, etc. These subject codes have been manually curated by ProQuest, and have been grouped by area. Some subject codes are introduced or disappear during the time span of the data, but most are stable designations over the 31 year period.[2]

Most (92%) dissertations have more than one subject, with the bulk having either two (58%) or three (24%). Unfortunately, the subject codes themselves are unevenly distributed: some areas like Physics have a rich taxonomy of subject codes (13 subject codes for 52,432 dissertations) whereas other areas like Computer Sciences contain a paucity of subject codes (only two subject codes for 41,605 dissertations). The subject codes in ProQuest are extensive, but much more fine-grained and with less clear organizational validity than the well-established basic disciplines reflected in common field designations like those in the National Research Council's 2010 report [103]. Consequently, we grouped subject codes into 69 areas based on the NRC classification, which we, in turn, group into seven broad area designations: Engineering, Physical & Mathematical Sciences, Biological Sciences, Health & Medical Sciences, Earth and Agricultural Sciences, Social Sciences, Humanities. Three more broad areas primarily oriented toward professional training—Education, Business, and Law, containing 12 areas—are not considered in the analysis below. Even after grouping subject codes into areas, we find that the average dissertation contains 1.6 areas, with nearly half (46%) still participating in more than one area and many (17%) having three or more areas designated.

[1] In general, stemming is not necessarily useful in topic modeling because synonymous variants are often correctly placed into the same topics. The reason terms are stemmed here is simply computational convenience: we are memory limited, so reducing model size—by reducing vocabulary size–allows more document data to fit each compute node.

[2] ProQuest has made changes to the subject code hierarchy over the time span of the data, creating mappings between hierarchies. Another difference by year is in the number of subject codes that can be applied while filing. The approach we take here controls for such variations (to a large degree) by filling in missing labels with PLDA inference.

The dataset has very high coverage of all dissertations published in the United States from the 157 research universities classified as research intensive by the Carnegie Foundation. For example, the dataset contains 35,942 dissertations attributed to 1994, or roughly 80% of the 45,394 PhDs granted by all United States graduate programs in that year according to the National Center for Education Statistics [122]. The National Science Foundation estimates a lower number of PhDs for the same year, at just under 40,000 [83, p. 7], but does not include tallies of all professional school PhDs. To ensure our coverage was as high as it seemed, we used data from the office of the registrar at Stanford to examine Stanford dissertations from 1993-2008: our collection of UMI records contains 8,836 Stanford dissertations, or 94% of the 9,331 PhDs granted by the university. From these statistics, we have reasonable confidence that our coverage of academia as a whole, while not exhaustive, is sufficiently extensive to extrapolate high level trends.

7.3 Methodology

Every dissertation exists in several contexts: the time it is written, the university at which it is produced, and the academic areas that delineate its boundaries. Our goal is to model each dissertation with respect to these contexts in order to infer, for example, that a given dissertation incorporates roughly 20% of its language from engineering disciplines, or that it looks more like the future of its field than the past. In both cases, we must perform two steps: *learning* and *inference*. First, we use PLDA to *learn* models of what the language of a given label space looks like (as topics) given the observed labels on each document. Then, in a second phase, we use these topics to *infer* the natural distribution of the entire label space for each document. We consider the case of learning the language associated with each academic area below.

In the learning phase, we build models of the language in each area. From only a single dissertation with two or more labels, we could not hope to discern which words belong to each. But by looking at the distribution of words and labels across the entire collection, we can learn that words such as “genome” and “sequence” are statistically more likely to occur together in Genetics & Genomics documents, whereas terms like

"algorithm" and "complexity" are better attributed to the Computer Sciences. As a result, we can determine which words in a dissertation labeled both as Computer Sciences and Genetics & Genomics are better attributed to each label (and, more specifically, what kind of CS or Genetics). We continue to use the PLDA model introduced in Chapter 5. While the disciplinary labels are known, the topics are not—the model discovers sub-disciplines. Formally, the probability of sampling a particular label and topic given an some document's observed set of labels Λ_d and Dirichlet hyper-parameters η and α is given by:

$$\begin{aligned} &P(l_{d,i} = j, z_{d,i} = k | l_{\neg d,i}, z_{\neg d,i}, w_{d,i} = t; \alpha, \eta) \\ &\propto I[j \in \Lambda_d \wedge k \in 1..K_j] \left(\frac{n^{(\neg d,i)}_{\cdot,j,k,t} + \eta}{n^{(\neg d,i)}_{\cdot,j,k,\cdot} + V\eta} \right) \cdot \left(n^{(\neg d,i)}_{d,j,k,\cdot} + \alpha \right) \end{aligned}$$

In the inference phase, we re-examine every dissertation without the restriction that its words be generated by one of the dissertation's labels: i.e. Λ_d is considered to be the full set of labels L. Instead, we allow the model to determine the optimal mixture of labels that would result in generating the words we see in each dissertation. In this way, the model can fill in missing labels by assigning high probability to a particular label that may not have been present during training. Or it can effectively remove a mis-applied label by assigning it very low probability. These scenarios do happen often in practice—for instance, spot checks of dissertations in the field of computational linguistics (where I have domain expertise) demonstrates several dissertations filed under either Computer Sciences or Linguistics (but not both) but where the model assigns high probability to both areas. The resulting per-document label distributions can be interpreted as the percentage of words in a given dissertation that can be attributed to each area of academia.

With these per-dissertation statistics computed, we can compute an aggregate statistic of how much language is borrowed by an entire area or within a given year. This is simply the expectation of the probability that any given word is assigned to the particular label j across all documents in a set of interest D^*:

$$
\begin{aligned}
& E_{d \in D^*} P(l_{d,i} = j) \\
= \; & \frac{1}{|D^*|} \sum_{d \in D^*} \frac{1}{N_d} \sum_{i=1}^{N_d} \sum_{k \in K_j} P(l_{d,i} = j, z_{d,i} = k | ...)
\end{aligned}
$$

To derive real qualitative insight into the data, we need to be sure that the statistics the model computes reflect real patterns of language usage in the world and not just artifacts of a choice of parameters or classification. To select the number of topics per model, we fit many models to the data varying the number of latent subjects per area designation, looking for areas of consistency between the models' assessment of cross-disciplinary language incorporation. Figure 7.1 shows the agreement between models based directly on subject codes and models based on areas (aggregated subject codes) as the number of latent topics per label varies. Note the high absolute correlations overall—representing the relative stability of the models learned—as well as the tendency for models trained with a sufficient number of latent topics under either labeling scheme to agree. The agreement demonstrates that the patterns of language incorporation learned are stable with respect to variations in the number of topics per label (for sufficiently many labels) as well as the granularity of the classification scheme. Where not otherwise stated, we use the 12 topics per area model because of its high consistency with the other models and its comparatively small size. The model has a total of 829 topics versus a maximum of 4,289 topics for the 16-topics-per-subject model. Learning and inference in the PLDA model can be accomplished in parallel. As a result, a model with 12 topics per label and 69 area labels can be learned in about a day on a small compute cluster. The stories we find here are consistent across models from 8 to 16 topics per label.

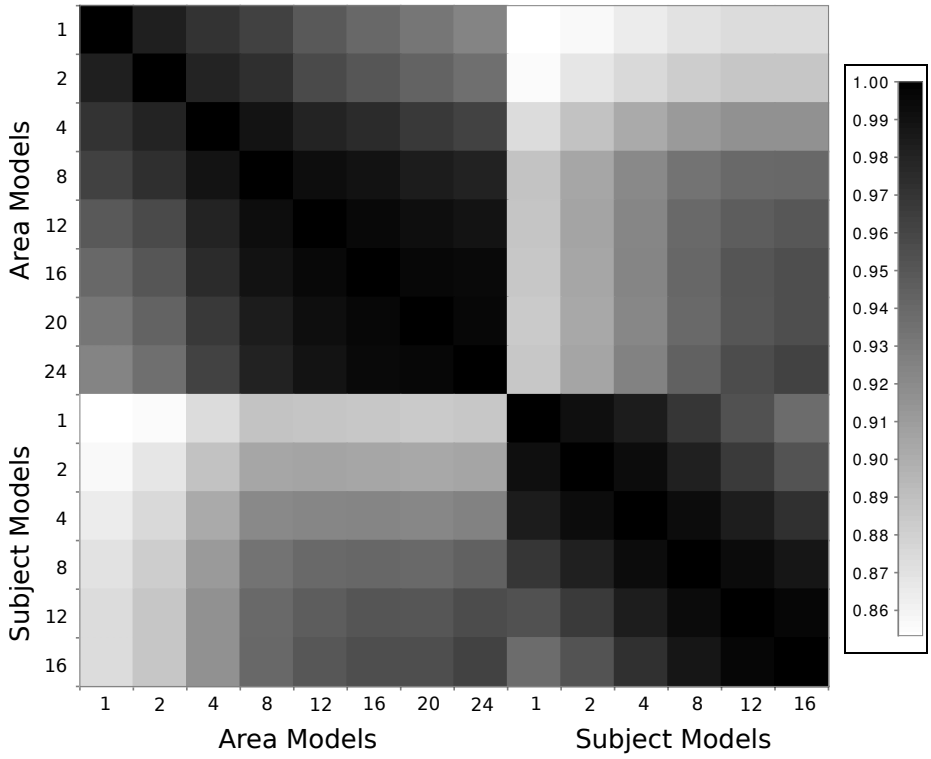

Figure 7.1: Intra-model consistency among PLDA models of academic fields. PLDA models were trained using one background topic, and either 2, 4, 8, 12, or 16 topics per *Subject* (bottom, right) or per *Area* (top, left, also including 20 and 24 topics per area). These models are compared by first computing the expected percentage of words borrowed between all pairs of *areas* in all years. For the subject codes, the percentages of each subject are summed by area to create a comparable scale. The correlation of these inter-area borrowing percentages are computed for all pairs of models, generating the plot above.

7.4 Language incorporation across disciplines

Interdisciplinary research is work that crosses disciplinary boundaries, borrowing ideas, methods, and terminology from elsewhere. Insofar as these borrowed factors are represented in the written word of a discipline, we can have hope that the model described above can quantify the extent of borrowing across these disciplinary boundaries. In this section, we examine the area model from Section 7.3 in detail.

We begin by observing that the disciplinary organization of academia is both ubiquitous and stable. Abbott, a prominent historian of academia, notes in [2, pp. 122-23] that "the departmental structure of the American university has remained largely unchanged since its creation between 1890 and 1910," with few exceptions such as a split in biology and new fields like linguistics and comparative literature. Menand [93] elaborates by arguing that the root of the current disciplinary organization lies in the professionalization of American academia at the turn of the 20th century. By putting scholars from each discipline in charge of training, hiring, and offering tenure to faculty from that discipline, universities effectively ceded control of the categories of academic inquiry to groups of scholars interested in developing and maintaining their own norms. Menand argues that the resulting strength of disciplinary organization at the university level sowed the future importance of both interdisciplinary work—work that cuts across these boundaries—as well as the rise of inherently "antidisciplinary" fields in the humanities such as gender and ethnic studies designed to challenge the traditional divisions. Indeed, in this section, we find that both the fracturing of biology (as noted by Abbott) and the rise of gender and ethnic studies (as noted by Menand) are two of the strongest statistical signals in our study of language incorporation.

We use the partially labeled topic model trained as described in Section 7.3 to induce a per-dissertation distribution over areas of academic study, and use these to study academic disciplines by proxy.[3] From here, we can begin to develop intuition into the dynamics of language incorporation in academia.

[3]Observe that the areas in which a dissertation is filed do not necessarily reflect the department of the graduating student, as the names and exact boundaries of departments do vary across universities.

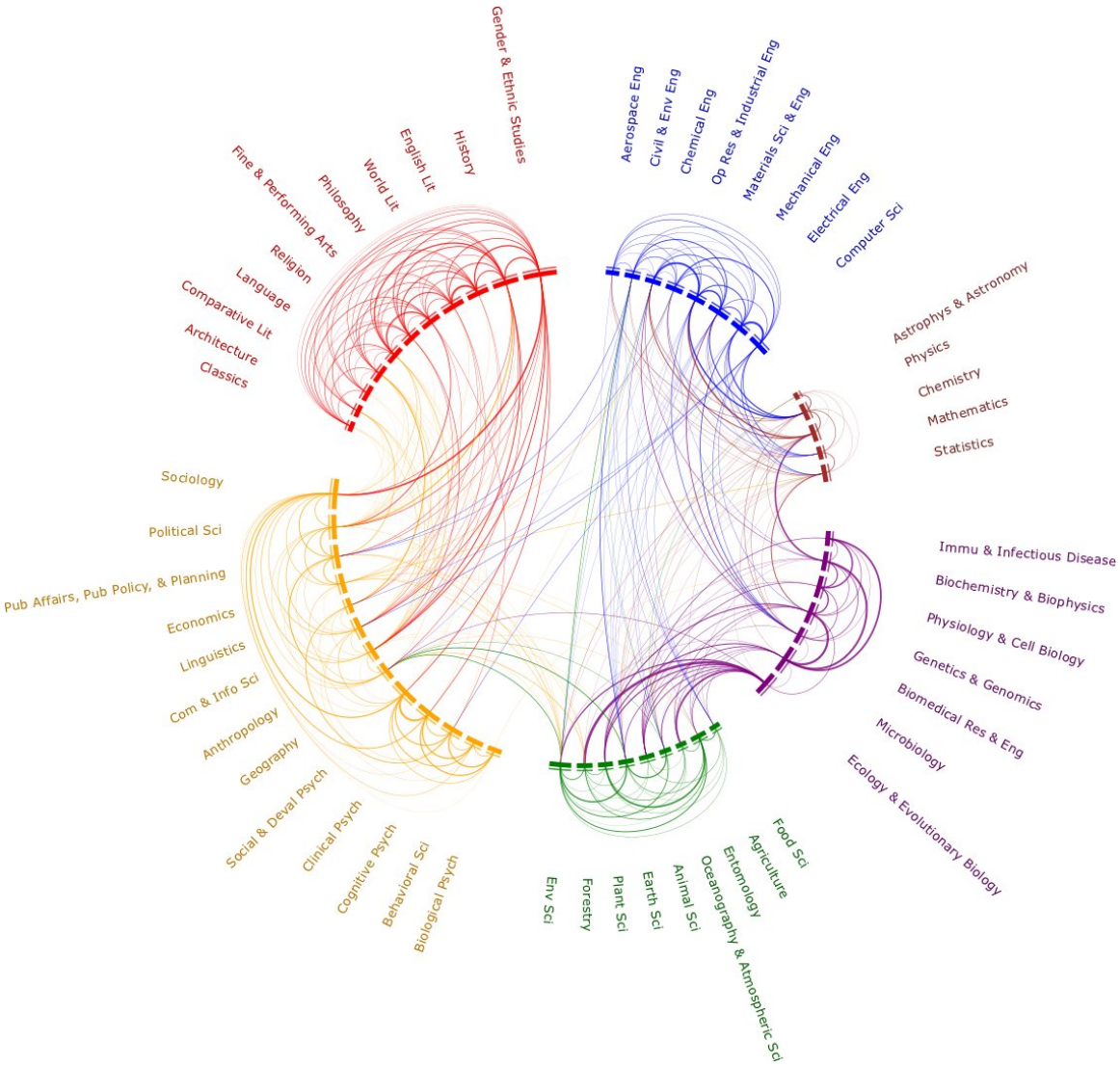

Figure 7.2: Language incorporation across all fields in academia, 2000-2010. Every area is shown on the outer ring, grouped by broad area. Clockwise from top: Engineering, Physical & Mathematical Sciences, Biological Sciences, Health Sciences, Earth & Agricultural Sciences, Social Sciences, Humanities. Arcs are drawn between areas with thickness in proportion to the total amount of language borrowed between the fields (in either direction) and with color determined by the area that sends more language. Note the extent of division between the STEM fields (right) and non-stem fields (left). Arcs between broad areas are shown inside the circle, while arcs within a broad area are shown outside the circle, in order to emphasize broad multidisciplinary influences. Earlier versions of this image by Jason Chuang.

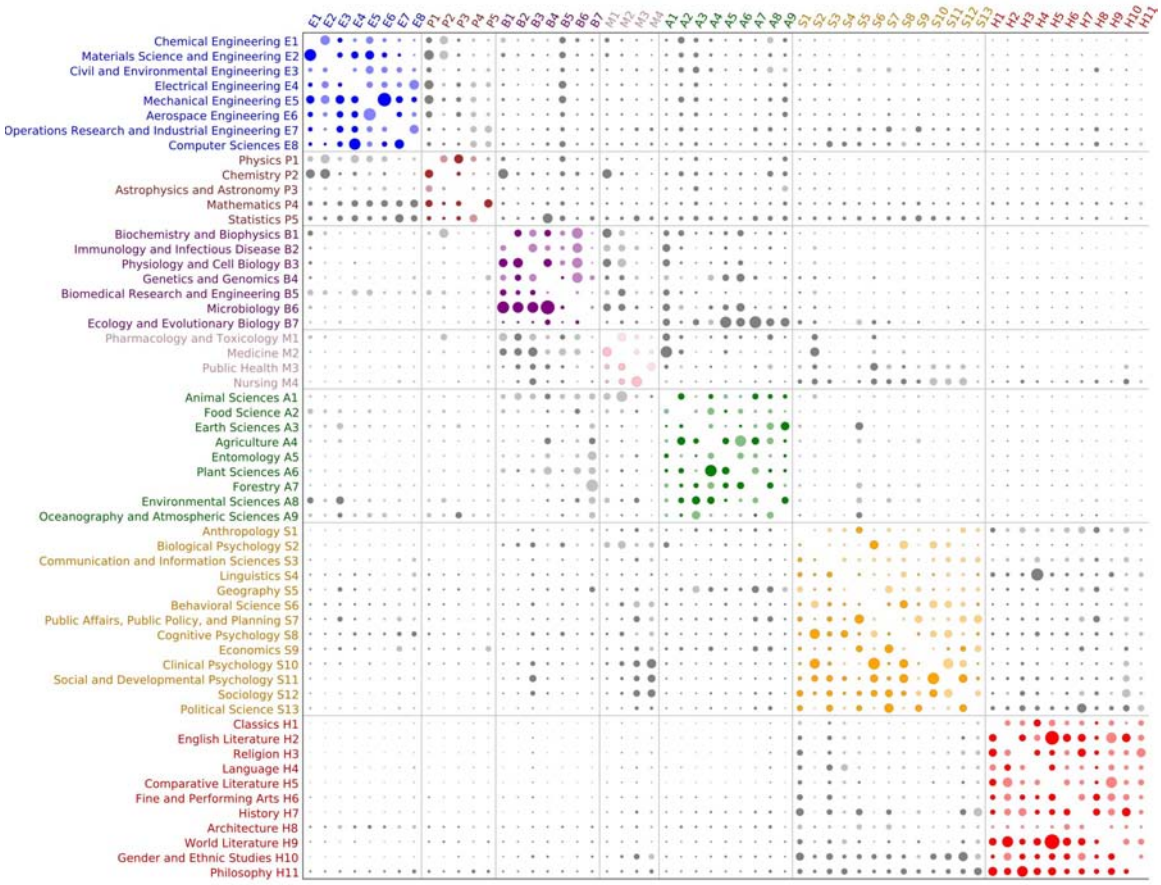

Figure 7.3: Borrowing among academic fields over all years. This image is analogous to Figure 7.2, but as a scatter matrix it makes visible the asymmetry in language incorporation. The value in a cell (i, j) is determined by how much of the language of row i is incorporated in dissertations labeled as column j. Cells are shaded when the given row sends more language to the column than it receives. Earlier versions of this image by Jason Chuang.

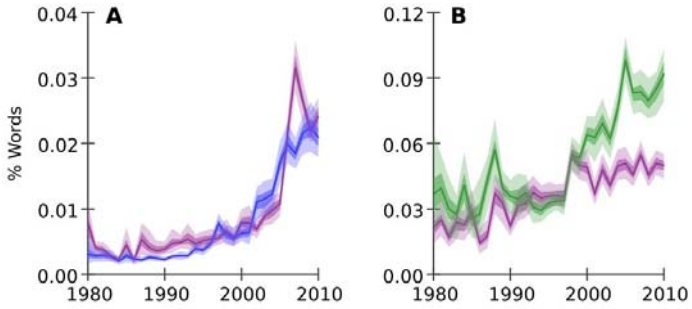

Figure 7.4: Concurent (A) and asymmetric (B) language incorporation in two pairs of fields. (A) shows the concurrent rise of computational biology (blue) and biocomputation (purple). The percentage of words in the computer science incorporated from genetics and genomics is shown in blue, while the percentage of words in genetics and genomics incorporated from computer sciences is shown in purple. Error bars are derived from bootstrap resampled estimates of the statistic (5% and quartile). B shows the asymmetrical incorporation of ecology and evolutionary biology into environmental sciences (green), versus the other way around (purple).

Figure 7.2 shows borrowing among all academic fields in years 2000-2010. Around the ring (clockwise) are the broad areas of Engineering, Physical and Mathematical Sciences, Biological Sciences, Earth and Agricultural Sciences, Social Sciences, and the Humanities. Area within each broad area are shown as bars. Each link is a measure of the extent to which language is incorporated (in either direction) and is colored by the field that sends more. This figure provides a stark visual representation of the gulf between the sciences and engineering, on the right, and the humanities and social sciences, on the left. Very few dissertations incorporate much language from across the divide. Those that do tend to be applied disciplines.

The model can discover the formation of new interdisciplinary areas. Figure 7.4A shows the uptake of terms from the Computer Sciences in dissertations from Genetics and Genomics and vice versa as a percentage of each field. While the growth in these areas' usage of each other tracks closely, Computer Sciences' usage of Genetics leads slightly in the early 2000's before Genetics' use of CS reaches a higher peak in the late 2000s.

Furthermore, the model tells us that interdisciplinarity is *directional*, as seen in the scatter matrix of all language incorporation shown in Figure 7.3. The scatter image

clearly delineates which areas send more language to which other areas. However, these asymmetries become more interesting when temporal dynamics are considered. Figure 7.4A shows that the amount of genetics language incorporated by computer science is not equal to the amount of computer science language incorporated by genetics. Figure 7.4B shows a starker example of asymmetric influence: ecology and evolutionary biology has had a larger impact on environmental sciences than the other way around. Indeed, asymmetries abound—some areas consistently incorporate more language from other areas than vice versa. Some areas act as language organizers for a broad area, such as Ecology and Evolutionary Biology for the Earth and Agricultural Sciences or Sociology for the social sciences. We revisit these findings in more detail below.

The method discovers known histories of interdisciplinarity among intellectual disciplines, such as the recent rise in computational biology. Here we document patterns of influence across many pairs of areas. First, we find that fields play distinct roles in language production: some areas are net sources and others are net sinks of language. Then, we describe the two major patterns in multi-discipline dynamics that have occurred over the past three decades: the split in biological sciences and the rise of gender and ethnic studies. Finally, we conclude the section with an explicit measure of interdisciplinarity.

7.4.1 Disciplinary Roles in Language Production

Academic disciplines have assumed distinct roles in the production of academic language during these past 30 years. Every field repeatedly draws upon its own language over time, but they vary in how much they export their language to other fields or incorporate the language generated elsewhere. Some fields are frequently sources for new academic concepts, while others consistently borrow concepts and apply them to their own domain. We formalize this intuition as the net source score for an area a. The net source score is a sum over all other areas b, adding one if b incorporates significantly more language from a than vice versa, or subtracting one if a incorporates significantly more language from b than vice versa. Note that not every area

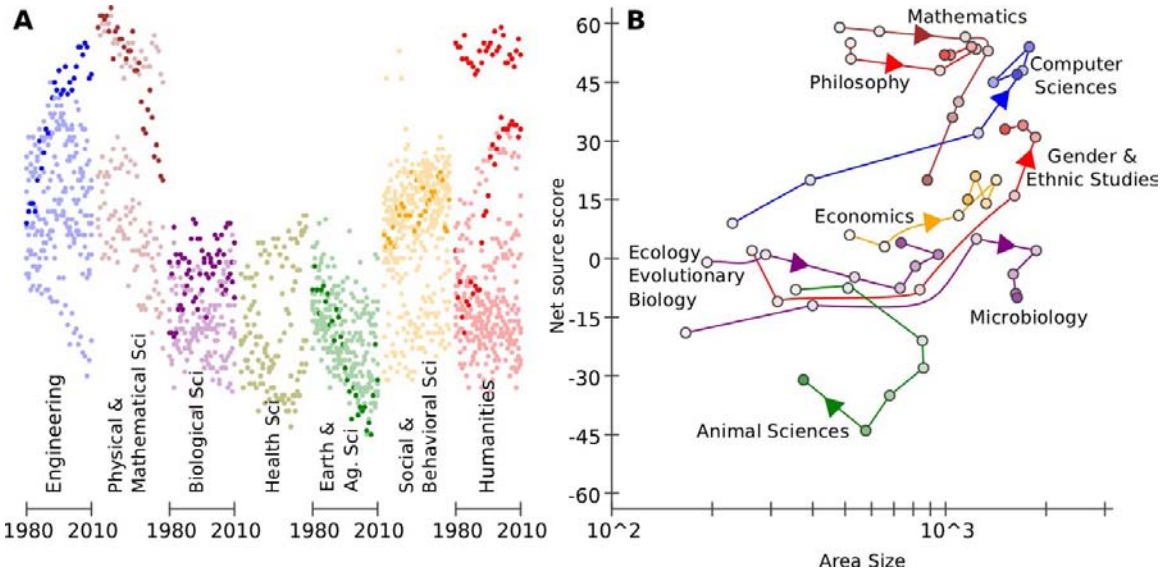

Figure 7.5: Net source score (y, shared axis) for academic areas. In A, each area's net source score is plotted over time, grouped by broad area. The highlighted areas are in detail in B where each area's size (x) is plotted versus its net source score over time (line series). In B, from top to bottom, the brown line is Mathematics; the red lines from the Humanities are Philosophy and Gender and Ethnic Studies; the blue line is Computer Sciences, the Purple lines are Ecology and Evolutionary biology and Microbiology, and the Green line is Animal Sciences. Each line represents a time series from 1980 to 2010 by 5 year increments, progressing from the lightest to darkest dot.

contributes to the sum: many pairs of fields incorporate each other's language at statistically indistinguishable rates.[4] Figure 7.5A shows the net source scores in each broad area over time. The shape of these point clouds demonstrates that some areas in Engineering[5] and the Social Sciences have gained in influence, whereas many in the Physical & Mathematical Sciences, Earth & Agricultural Sciences, and some of the Humanities have lost influence. Selected areas are shown in Figure 7.5B as trajectories of area size versus net source score over time.

Areas that export language to other fields are frequently methodological and concern abstract reasoning: e.g., Computer Science, Statistics, Mathematics and Philosophy and History. Mathematics and Philosophy have elsewhere been described as "root disciplines" [41] because their generality and abstraction anchor the development of more applied fields. We find quantitative support for this argument. Math and philosophy do clearly export more language to other fields than they borrow concepts from elsewhere. However, we also observe Computer Science and Statistics assuming a similar role, and possibly becoming more of a preferred root category in contemporary science. Methodology and machines of engineering and statistics are more often incorporated into other fields than are philosophy and mathematics. In contrast, other areas—particularly in the biological sciences—have grown much larger without gaining external influence. Others incorporate language from elsewhere substantially more than they export. These fields tend to be humanistic, applied, or topical domains like classics, languages, and the whole of earth and agricultural sciences. These topical areas tend to rely on abstract reasoning and methods to further understanding in their knowledge domains.

7.4.2 The Rise of Molecules and Machines

In the 1980s, the biological sciences were dominated by two primary modes of inquiry. On the one hand were integrative approaches to biological systems, from individual

[4] We compute statistical significance by comparing the distributions of our statistic (mean language use of some area) in 100 bootstrap samples of the dissertations in any target area.

[5] Not all areas of Engineering gain influence: the gains are driven largely by increases in Computer Science, Electrical Engineering, and Operations Research. The notable downward trajectory in Engineering in Figure 7.5A is Chemical Engineering.

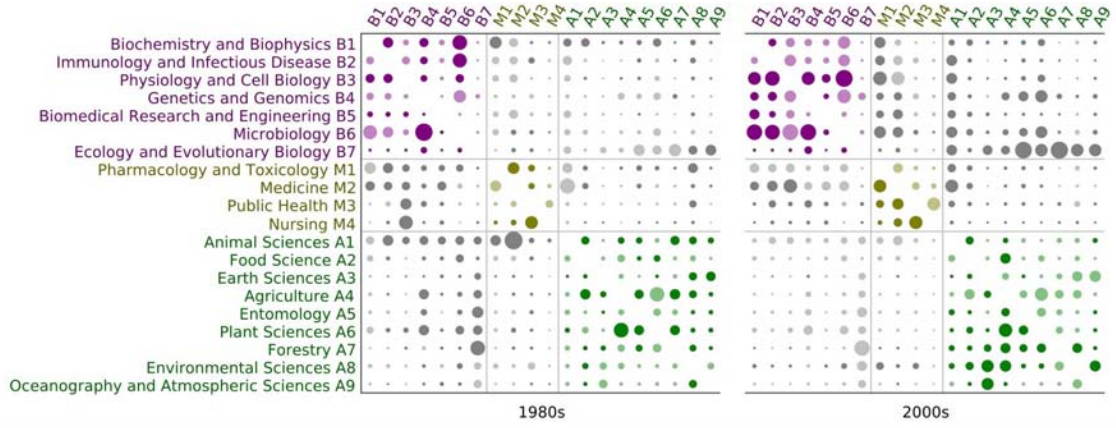

Figure 7.6: Interdisciplinary language incorporation among the Biological Sciences (purple), Health Sciences (gold), and Earth and Agricultural Sciences (green) in two time points (1980s and 2000s). The value of cell i,j represents the fraction of words in column j that were incorporated from row i. Note the tremendous increase in language in the Earth & Agricultural sciences incorporated from Ecology and Evolutionary Biology. Note also the increased influence of the Biological Sciences (as compared to the Animal Sciences) on Health Sciences.

animals all the way up to ecosystems. On the other hand were reductionist approaches that sought to understand biology from its base components, first through microbiology and later through more specific fields such as genetics and genomics as well as cell biology [24]. Indeed, the split in the field is so deep that some universities have already discussed dividing their departs into two if they have not done so already—to molecular biology and evolutionary biology.[6]

Figure 7.6 shows borrowing among all the biological sciences, health sciences, and earth & agricultural sciences in two decades: the 1980s and 2000s. What we find is that ecology & evolutionary biology—once an integral part of the biological sciences both by mass and by the extent its language was incorporated elsewhere—shrinks greatly in influence within the rest of the biological sciences. However, over the same period of time, it grows to become dominant in the earth & agricultural sciences. Simultaneously, we witness an enormous growth and blossoming of the reductionist

[6]Biology major may split. October 21, 2011. Yale Daily News. http://www.yaledailynews.com/news/2011/oct/21/biology-major-may-split/

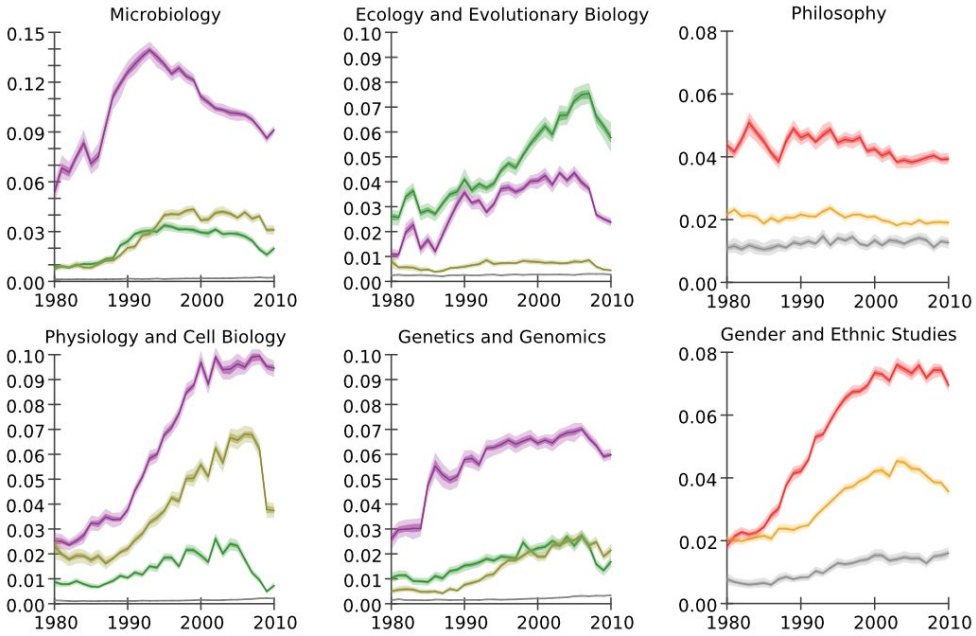

Figure 7.7: Language incorporated from each area (graph title) over time from other areas of the Biological Sciences (purple), Health Sciences (gold), Earth & Agricultural Sciences (green). Language incorporation is also shown for Humanities (red), and Social Sciences (orange) corresponding to areas described in Section 7.4.3. These graphs represent the total percentage of words in the given broad area incorporated from the named area. Usage in all other broad areas not captured by the lines in a given graph is shown in gray.

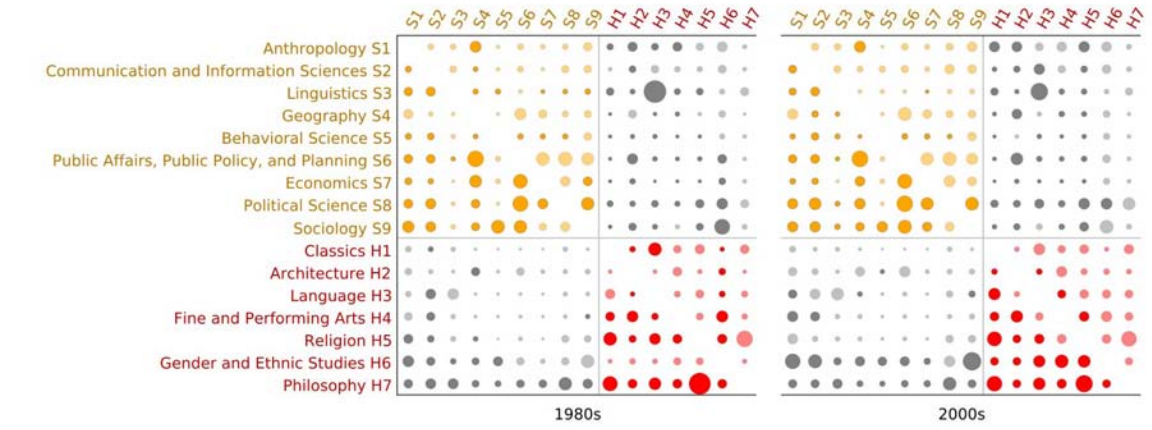

Figure 7.8: Interdisciplinary language incorporation among the Social Sciences (orange) and Humanities (red).

biological fields in the biological sciences. Figure 7.7 shows the amount of language exported by each of several areas to other areas within selected broad areas. For instance, we see that microbiology grows quickly through the 1980s before losing relative impact in biology to related reductionist approaches in the 1990s, including Genetics & Genomics as well as Physiology & Cell Biology. Molecular biology acts as an integrator of the rest of biological sciences and a growing source of language for health sciences, as well.

The ability to quantify and document these trends has implications for the way we structure university initiatives. For instance, many consider the rise of environmental studies as a whole as an independent discipline [25], but it heavily incorporates language from the studies of biological systems. In contrast, the rise in reductionist biology is driven by the technologically of miniaturization, data generation, data collection, and associated engineering methodologies. Indeed, the percentage of language in the biological sciences borrowed from engineering disciplines (as a whole) roughly doubles over the time period (to about 7%).

7.4.3 Rise of Gender and Ethnic Studies

The field of Gender and Ethnic Studies began in the 1960s before our dataset's starting year of 1980 [93, 137]. In the 1960's, the humanities and social sciences faced an intellectual crisis of legitimacy in an era of broad cultural shifts that demanded greater representation of traditionally under-represented voices and views. Our methods show that it was not until the late 1980s that Gender and Ethnic studies began to grow in influence. By the 2000s, Gender and Ethnic Studies are a primary mode of organizing thought, pervasive both throughout the humanities and social sciences, with its influence beginning to plateau in the early 2000s.

Figure 7.8 shows language borrowing among the social sciences and humanities. It shows that Gender and Ethnic studies has moved from incorporating the language of others in the 1980s to acting as an organizing force for two broad areas: both the humanities and social sciences. Contemporary with the rise in Gender and Ethnic Studies, Philosophy declines in size but does not decline in relative influence. Indeed, Philosophy remains one of the largest net sources of language in the dataset, as shown in Figure 7.5, despite the growth of Gender and Ethnic Studies. We can think of Gender and Ethnic studies as, in effect, an intellectual movement—a critique of disciplines—that got institutional support and grew [93]. Its growth demonstrates the wide and growing acceptance of identity as a construct necessary to academic inquiry into social and humanistic questions. Perhaps unsurprisingly, we find no evidence of a significant language usage of Gender and Ethnic in either Engineering or the Physical and Mathematical Sciences.

7.4.4 Interdisciplinarity

The extent to which a dissertation's language comes from disciplines other than its primary discipline can then be taken as a measure of that dissertation's interdisciplinarity. We can use the inferred per-document distribution over areas to compute a percentage of words borrowed from *outside* a given dissertation's reference area. For example, a dissertation labeled Computer Sciences that borrows 20% of its words

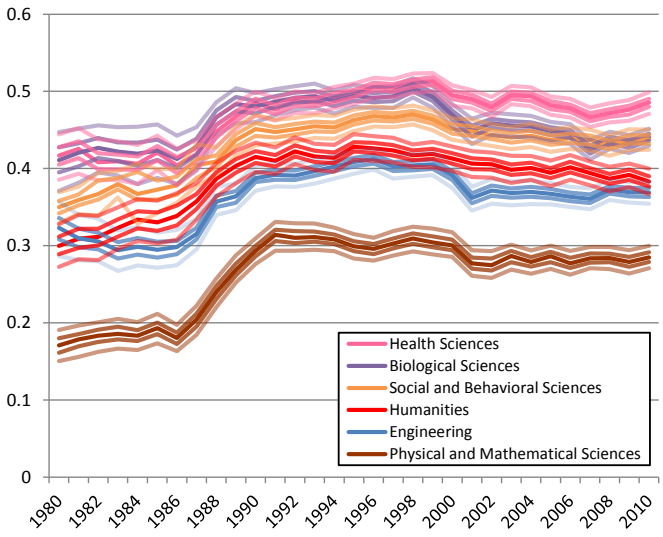

Figure 7.9: Percentage of words incorporated from outside areas, by broad area. Error bars are from bootstrap resampled estimates of the mean percentage of words drawn externally.

from Genetics & Genomics would likely be an interdisciplinary dissertation in computational biology, whereas another dissertation that is 99% computer science is likely to be a work in a core area of computer science such as theory or systems.

We formalize this definition by computing an interdisciplinarity score for any given dissertation d with respect to an observed reference label $l \in \Lambda_d$ as the expected probability mass in the document's distribution over labels that is over some threshold $\tau \in [0, 1]$ and is attributed to areas other than l. This interdisciplinarity score $I_{d,l}$ is defined as:

$$I_{d,l} = \sum_{l' \in L} \psi_{d,l'} \cdot I[l' \neq l \wedge \psi_{d,l'} > \tau]$$

Note that this formalization counts each dissertation separately as a member of each of its labeled areas and will, in general, have a different score with respect to each reference label.

Looking at the expected value of this test statistic over the whole document collection gives us a sense of the change in interdisciplinarity in academia over our time

span. Figure 7.9 shows a bootstrap resampled estimate of the mean $I_{d,l}$ for all areas grouped by (and averaged within) each broad area for $\tau = .1$. In general, all areas have seen a net rise in interdisciplinary language over the time span, although some areas show a distinct downward trend from a peak in the early 2000s. Dissertations saw an average increase in the number of subject codes applied per dissertation in the late 1980s across all areas, which is reflected in the measure of language incorporation. Note that the Physical and Mathematical Sciences are the most disciplinary, whereas the health and medical sciences borrow the most broadly. In recent years, the Social Sciences have surpassed the Biological Sciences in terms of the percentage of interdisciplinary language incorporated.

7.5 Leading and lagging

The area designations considered in the previous section are only one kind of contextual labeling for a dissertation. We also made use of the year in which each dissertation was published in order to study the dynamics of language borrowing over time. But another way we might use the year of a dissertation's publication is directly as part of a label space. If we do so, we can use the same methodology introduced in Section 7.3 to directly the measure the *temporal distribution* of any given dissertation and, by extension, whole universities or fields. Concretely, we assume that the zeitgeist of one year in academia can be represented as the relative frequencies of terms used in that year. From this assumption, we model every dissertation as a mixture of years—if this dissertation were allowed to see the future of academia and the past (relative to its date of publication), which years' words would the dissertation prefer to use?

Formally, we train a simple PLDA model with only one label per dissertation corresponding to the year in which it was published. Note that in this simplified model, the maximum likelihood words in any given year are the same as those output by the simple multinomial naive Bayes event model, as per our discussion in 5.2.1. However, after inference, the model will infer a probability of the words in each document coming from each year, i.e. $\sum_{y=1980}^{2010} \psi_{d,y} = 1$ for all dissertations. For a dissertation d published in some year $\hat{y}$, we can compute a *future*, *past*, and *present*

score for that dissertation by simply summing the elements of ψ_d that are less than, greater than, or equal to $\hat{y}$. This gives us a total number of fractional dissertations attributable to each time range. The future score for a dissertation is simply the total probability of words being drawn from years that follow the year in which the dissertation was published: $S_d^{fut} = \sum_{y=\hat{y}_d+1}^{2010} \psi_{d,y}$. The past score is the total probability of words being drawn from years that precede the dissertation: $S_d^{past} = \sum_{y=1980}^{\hat{y}_d-1} \psi_{d,y}$. The present score is the total probability of words being drawn from the year in which that dissertation was published: $S_d^{pres} = \psi_{d,\hat{y}_d}$.

To compute the future score for a set of dissertations D, we simply take the average of their future scores $S_D^{fut} = \frac{1}{|D|} \sum_{d \in D} S_d^{fut}$. We define the past and present scores for a set of dissertations analogously. Similarly, we can compute a future number as the expected number of dissertations drawn from the future for a given set of dissertations as $\#_D^{fut} = \sum_{d \in D} S_d^{fut} = |D| * S_D^{fut}$. The past number and present number are defined analogously.

All of these scores can be computed in two conditions. One is where the PLDA model is trained using *all dissertations*, i.e. it captures large scale patterns of language change in academia as a whole. The other considers only dissertations *within some area* during training, i.e. it captures changing patterns of language use within the field in which a dissertation is published. We consider both metrics below.

7.5.1 Future-leaning schools

When applied to the level of academic programs within schools, our entirely data-driven metric reconstructs program rankings similar to those of the National Research Council's 2010 report [103]. This is a strong validation of the technique as well as a surprising finding. The potential impact of data-driven techniques in the study of academic research is large. Science policy makers, university administrators, funding agencies, and prospective students all rely on many factors when deciding which academic institutions to become involved with. Organizations have stepped in to provide information to support such decision makers. From US News & World Report rankings [100] to the recently released National Research Council report on

academic institutions [103], there is great interest in generating objective benchmarks of academic institutions. Traditionally, these benchmarks have focused on the inputs associated with each institution—amount of money raised, SAT scores of incoming students, number of grant dollars and research staff, etc.—or on the reputation of those institutions as judged by their peers. Yet by their nature, academic institutions produce a great deal of output, usually in the form of the text of books, peer-reviewed publications, and dissertations. Such text-rich datasets tend to be overlooked in quantitative analysis of institutional performance because making effective, quantitative use of text is a challenging problem. In this section, we analyze these same institutions from a new perspective: scoring institutions by how much each institution looks like the future of academia, judged quantitatively from the text of each institution's PhD dissertation abstracts.

The National Research Council spent years and large sums of money developing its report on the quality of graduate programs at universities throughout the United States. The findings are based on surveys (in two ways) and partially account for uncertainty in survey responses. Each university program self-reported information including features such as the number of full time faculty and research staff, number of publications per faculty, number of minority students, and many other features. These features are then used to compose a ranking in one of two ways: S-rankings and R-rankings. The S-rankings use surveys of academics within each field to assign the relative importance of these various features to the aggregate quality of a department. An aggregate weighting function is then created, with which departments are ranked. For the R-rankings, academics are surveyed to solicit their judgments of the best universities in their field, which are used to learn weights from the input features that best predict the solicited rankings. In both cases, a final ranking is not produced: instead, the model generates many candidate rankings and reports a range of possible rankings for a given university program. The S- and R-rankings do not always agree, but often do. Nor do they always agree with the rankings reported by the US News & World Report (see, for example, a comparison in Psychology reported here [51]). Nonetheless, the rankings represent a good-faith effort at producing a standardized reference for assessing the importance of various factors in overall university rankings.

To construct a data-driven assessment of graduate research programs, we must first divide our dissertations by department. Unfortunately, the department in which a dissertation is filed is not recorded in the UMI database: only the standardized subject codes are provided. We used the area mapping manually constructed as described in Section 7.2 to stand as a proxy for the department in which a dissertation was filed. These areas were designed to match the NRC's 58 areas of study as much as possible. So as a proxy for Stanford Computer Science Department dissertations, we take any dissertation filed at Stanford University and containing either the subject code "computer science" or the subject code "artificial intelligence." Then we can compute the future score S^{fut} and future number $\#^{fut}$ of the Stanford Computer Science dissertations as well as for every other department at every other school.

To evaluate our new metric's performance, we compare to how well it predicts the NRC university rankings. As a reference point, we can see how well each input feature used in the construction of the ranking correlates with the final ranking. Figure 7.10 shows the average correlation across areas of each feature with the final R-ranking score at $P = .05$, i.e. at the lower bound of the reported rankings. Nearly identical results are found when comparing to .95 bounds and to S-rankings at either bound. The most predictive features are, unsurprisingly, features related to the academic output of the department. These are, in order, Average PhDs 2002 to 2006 (the size of the department), Awards per Allocated Faculty (a measure of external prestige), Average GRE (the quality of the incoming students), Cites per Publication (how high quality are the published works), and Publications per Allocated Faculty (how prolific are faculty members). Interestingly, a large gap in the predictive feature value follows before the next feature (Percent Faculty with Grants), before feature values' predictive scores fall off for features that just don't matter, including Percentage Female Students and Percent Female Faculty.

In the gap in predictive value lie several of our data-driven features: all those based on the size of the department ($\#^{fut}$, $\#^{present}$, and $\#^{past}$) as well as S^{fut} trained on the overall model. First let us consider the percentage scores S^{fut}, $S^{present}$, and S^{past} trained on the full dataset (overall). The NRC features are provided during training of the linear model that eventually outputs the final ranking, so the fact that

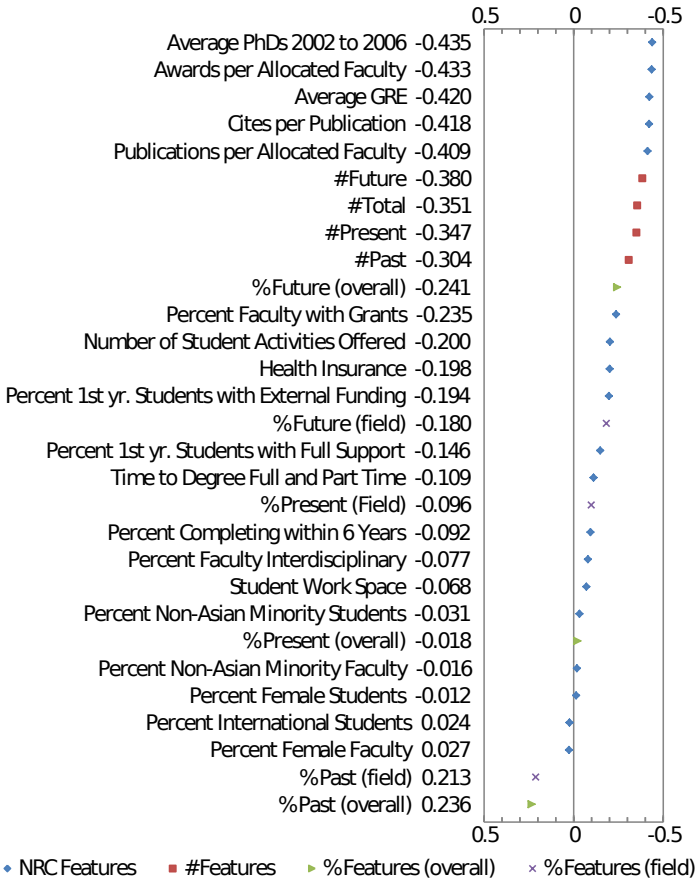

Figure 7.10: Relative predictive strength of NRC features and future scores. The correlation of each NRC input feature (blue squares) is computed with respect to the final score used for ranking departments, averaged across areas. The red squares features represent $\#^{fut}$, $\#^{present}$, and $\#^{past}$ computed on the overall model. The green triangle features represent S^{fut}, $S^{present}$, and S^{past} computed on the overall model. The gray crosses represent the same S scores computed on the per-field model.

S^{fut} (overall) correlates better than all but four of the input features is a positive and surprising finding. Similarly, $S^{present}$ (overall) has almost no correlation with the final score. This, too, is interesting—it shows us that simply being with the times is not predictive of overall department quality. Finally, we see that S^{past} (overall) is actually negative. Again, an intuitive result: those departments that do not keep moving forward are left behind in the rankings. The corresponding S (field) scores are trained on only those dissertations within each field, do not correlate as highly with the overall rankings. We will return to these scores in Section 7.5.2.

The impressive performance of the features that depend on department size ($\#^{fut}$, $\#^{present}$, and $\#^{past}$) is illuminating: size is one of the major factor in the overall determinant of department quality, as has been noted in by multiple researchers [7, 135, 126]. Indeed, Average PhDs 2002-2006 is the highest ranked feature in the NRC dataset. When we scale the corresponding S (overall) scores by the size of each department to get to the # scores, more information is included in each feature. Even the negatively correlated coefficient (S^{past}) has high overall correlation when scaled by department size because of the strength of the latter. We note also that the relative order of the three # scores preserves the properties we would hope: $\#^{fut}$ is a better predictor than simply the total number of dissertations, which in turn is a better predictor than $\#^{present}$, which itself is a better predictor than $\#^{past}$. Nonetheless, the performance of S^{fut} is heartening because it is itself uncorrelated with size.

The metrics above can be used to directly rank the universities within each field, which we compare to the NRC rankings using Kendall's tau rank correlation. The $\#^{fut}$ metric, our highest scoring, has an overall Kendall's tau correlation of .398 with NRC's R-ranking at $P = .05$. This correlation itself depends on the broad area under consideration. The metric does best in progressive, technical fields of Engineering (average of -.445), Physical and Mathematical Sciences (-.377), and Biological and Health Sciences (-.335). It falls flatter in Agricultural Sciences (-.300), Humanities (-.288), and Social and Behavioral Sciences (-0.278). This ordering tells us about each of the broad areas: not only are the Humanities and Social and Behavioral Sciences structurally disconnected from the rest of academia (Section 7.2), they have fundamentally different norms about the nature of their work. Norms in the Humanities

and Social Sciences do not reward progressive, long-term development of the language in a field to the extent that STEM fields do. We analyze fields by their future leaning scores in more detail in the next section.

7.5.2 Future-leaning areas

Let us consider the field-by-field temporal variation in more detail. Figure 7.11 shows the temporal distribution of language used by all dissertations in the dataset grouped by area and colored by broad area. The distributions shown are the temporal distributions $\psi_{d,y}$ for each dissertation, summed within each area relative to the dissertation's year of publication. So a dissertation whose language looks like the year in which it was published and the following two years would contribute a small amount (+0 +1 and +2) for its field's distribution over years.

Fields are colored by broad area. For each field, the figure shows which years' words make up what fraction of dissertations, on average. The 50th percentile is marked with a black line; the 25th and 75th percentile are marked in color; and the 5th and 95th percentile are marked in gray. On the left is the extent to which a field defines the language of academia's future overall. Some fields, such as Classics, tend to use language that looks more like academia's past, whereas fields like Computer Science tend to use language that looks more like academia's future. The left side of the figure, then, tells us a bit about the relative growth in importance of the language from each area over time.

On the right of Figure 7.11 is the distribution based on comparing every dissertation's language to the language used *in its own field* independent of the rest of academia. In other words, this model is trained only on dissertations published in each field and is otherwise identical to the left half of the figure. Whereas the left figure shows how much dissertations from a single field define future academic language generally (i.e., interdisciplinary transfer), the right shows shows how much a field is inwardly defining, or paradigmatic. Some fields that looked future leaning relative to the rest of academia (such as Computer Science) here look more toward their own past. Others that tend to look like academia's past (Classics) actually use language

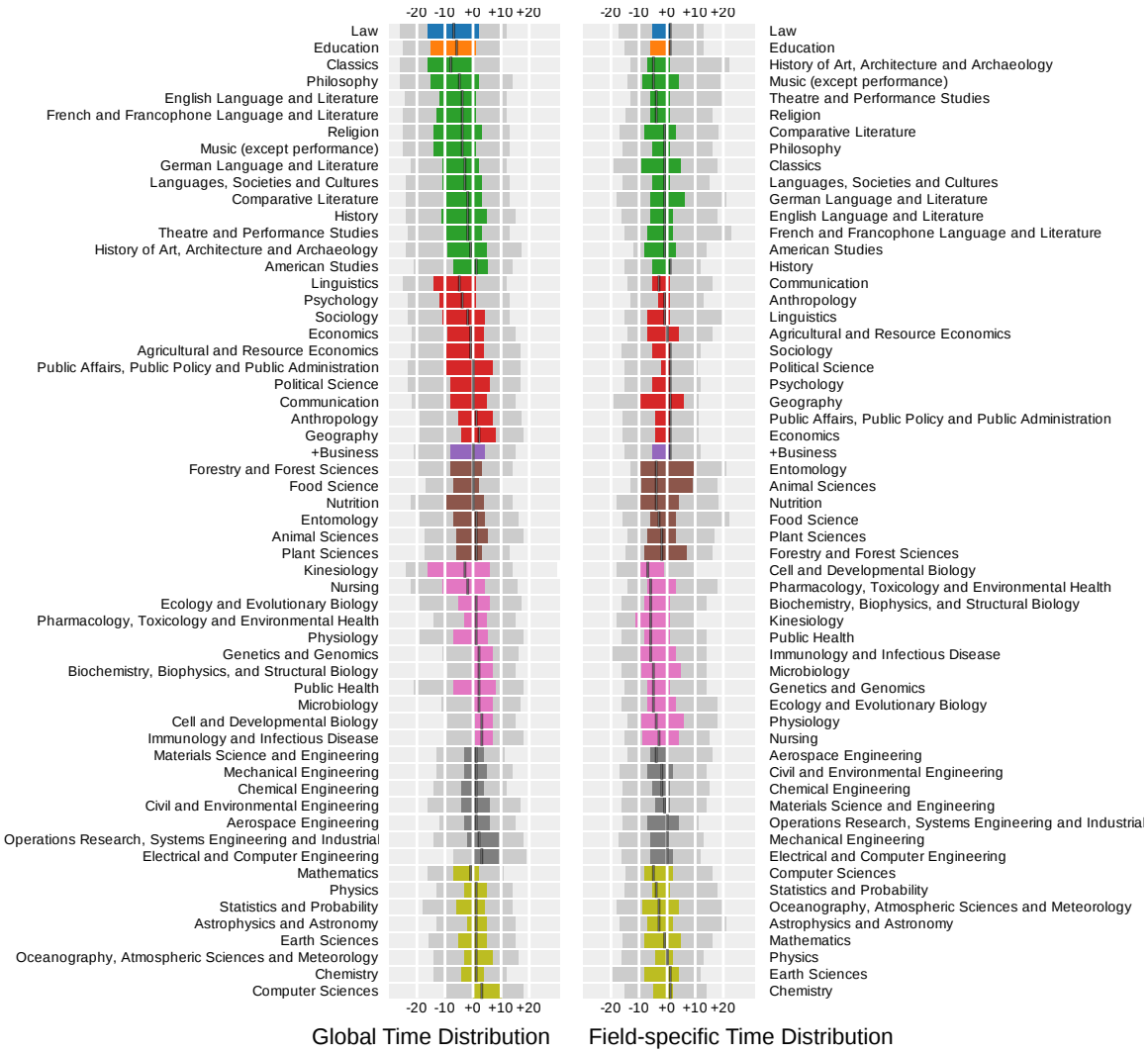

Figure 7.11: Temporal distributions of language usage by field under two models: a global model of language use (left) corresponding to the "% (overall)" features in Figure 7.10, and a field-specific model of language use (right) corresponding to the "% (field)" features.

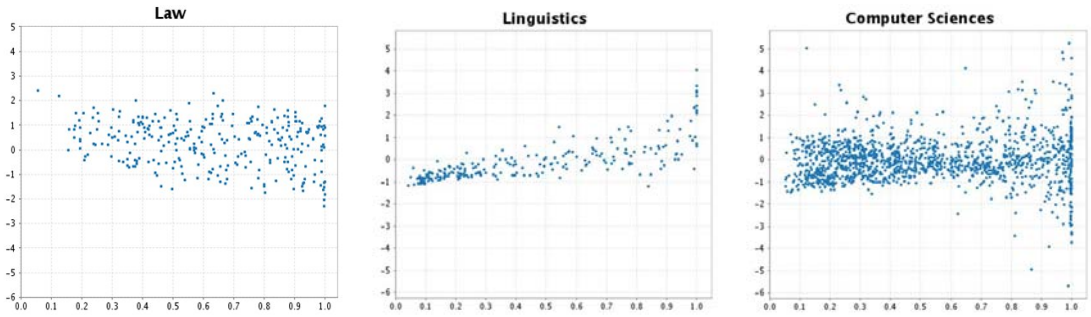

Figure 7.12: Interdisciplinarity versus future score for three fields, Law, Linguistics, and Computer Sciences. Law shows negative returns from interdisciplinarity; Linguistics shows a typical pattern of positive returns; and Computer Sciences shows a more complicated interplay.

in stable, consistent way, looking both like their own past and future as seen in the wide green bar.

7.6 Returns from interdisciplinary research

By combining the measure of interdisciplinarity developed in Section 7.4.4 with our validated notion of future-leaning dissertations in 7.5, we can examine the effective returns from interdisciplinary research as determined by the impact of increased interdisciplinarity on future leaning score. Overall, we find a large return to interdisciplinary work: on average a dissertation that is one standard deviation more interdisciplinary than other dissertations in its field and year will be about one half of a standard deviation more future leaning. Figure 7.12 shows the plot of future score (normalized for its area and year) versus interdisciplinarity (normalized for its area and year) for three representative areas: Law, which shows a negative return to interdisciplinarity; Linguistics, which shows a more typical positive return; and Computer Sciences, which appears to have no simple correlation between interdisciplinarity and future score.

The relationship between interdisciplinarity and future score for the Computer Sciences becomes more interesting when we examine the dynamics of the relationship. Figure 7.13 plots the normalized interdisciplinarity score versus the normalized

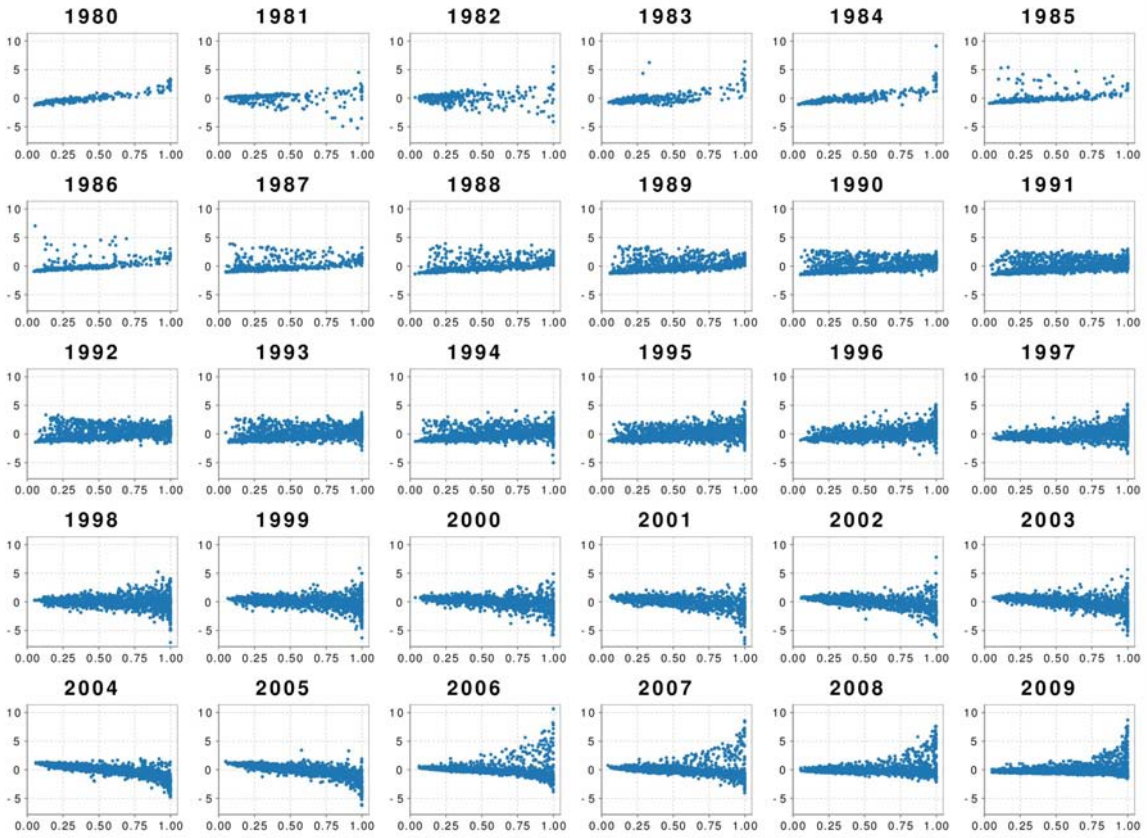

Figure 7.13: Diminishing returns from interdisciplinary work over time for computer science dissertations (all universities). The % of words borrowed from other fields is displayed on the x-axis versus the normalized future score (for computer science dissertations in each year) is shown on the y-axis.

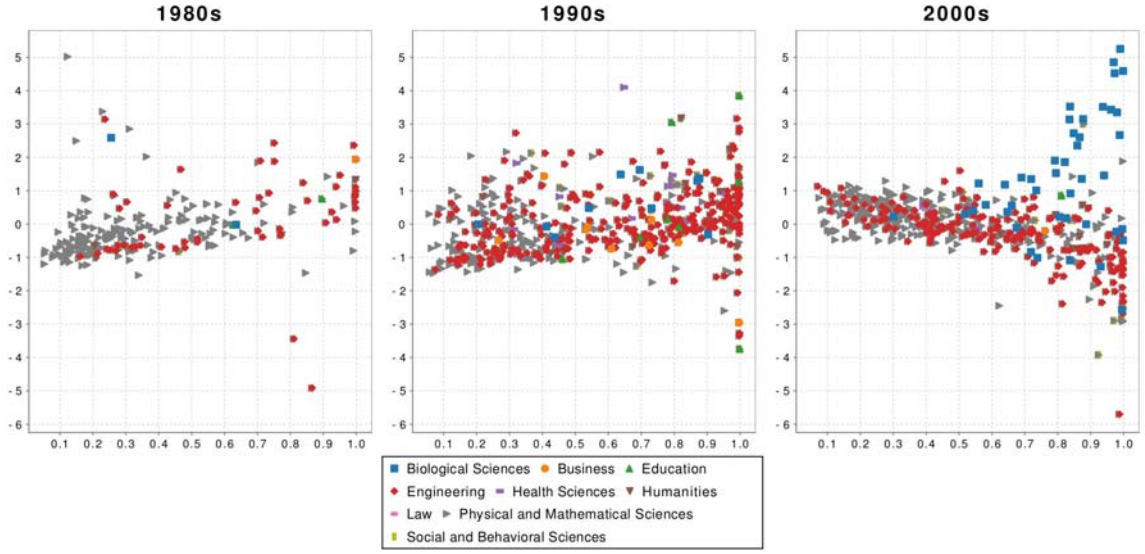

Figure 7.14: Dissertations from Stanford University computer science are plotted and colored by external area designations. Most red engineering dissertations are electrical engineering co-listed. The returns from collaborating with engineering fall over time, but in the 2000s computational biology's growth begins to make interdisciplinary dissertations again look future leaning.

future score for all dissertations filed as Computer Science in every year in the dataset, from 1980 to 2010 in its own cloud. Note that these terms are indeed correlated, but that the direction of correlation actually changes over the course of the dataset. In particular, we find that in the 1980s, interdisciplinary work tends not to be particularly future-leaning, but in later years there is a greater return to crossing disciplinary boundaries for computer scientists. In a sense, we see that as dissertations become more interdisciplinary (the mass of tends to move toward the right) the return to being interdisciplinary falls.

The reason why is illustrated in Figure 7.14, where we examine Computer Sciences dissertations at Stanford by decade, with each dissertation colored by any other brother areas in which it participates. Here we find that in the early years, most interdisciplinary work is interdisciplinary within the Engineering disciplines (red)—and is usually Systems dissertations that borrow from Electrical Engineering, to be precise—but that over time the returns from incorporating language from other parts

of Engineering diminishes greatly, eventually having a negative correlation in the 2000s. However, not all forms of interdisciplinary language incorporation are created equal: by the 2000s, language incorporated from the Biological Sciences (blue) is substantially more future-leaning than other dissertations in the field.

7.7 Conclusion

Understanding the role of language incorporation and interdisciplinarity in academia promises to shed light on both our academic understanding of the interplay of ideas and institutions as well as having practical impact in data-driven techniques for assessing impact. This chapter illustrates the broad scope of analysis made possible by leveraging the human interpretable metadata—implicit domain expertise in the case of subject codes—to perform larger scale analyses of language use in academia than are possible with previous techniques. We use these techniques to study ideas—we document fundamental changes in the language of academia that have occurred over the past three decades, including broad scale splits in the biological sciences and the growing dominance of gender and ethnic studies in the social sciences and humanities. We also use these techniques to study the organization—the academic structure of departments and universities—in which these ideas are made. Together with the study of *people* in Chapter 6, this chapter complete our illustration of the ways in which the statistical text models developed in the first half of this dissertation can apply to studies of *people, organizations,* and *ideas*: three primary areas of inquiry in the computational social sciences.

Chapter 8

Conclusion

Computer science has dramatically changed society with technologies like the web, online social networks, mobile computing, large scale databases, and so on. As a discipline, we have developed and studied the content generation and storage tools that enable these changes. Furthermore, from web search to genetics, techniques from our field are now indispensable to the analysis of that content. As textual datasets grow in size and scope, computer science has the potential to occupy an even more central role in society by speaking directly to social questions in the world at large.

To rise to this challenge, computational social scientists will need tools that can discover and characterize patterns of language use at a scale beyond what individuals can achieve by reading. The discovered patterns must ultimately support or inspire hypotheses about the world from the texts people write. But in order to be trusted, the tools must first confirm the broad trends and facts already known to be true—or they must provide compelling arguments against the conventional wisdom. Without such support in finding the known, text mining models have limited utility in discovering the unknown, i.e., in quantifying known trends or discovering unexpected ones. Specifically, we need tools that are *trustworthy* in that they consistently find patterns that match what we know to be true; they must be *interpretable* in that their output speaks to hypotheses about the world, not just model mechanics; and they must be *flexible* enough to incorporate input from domain experts while still letting the data speak (Chapter 1).

The approach I take in this dissertation is to incorporate labels or tags—representing implicit domain expertise within the data itself—in order to build text mining tools that are trustworthy, interpretable, and flexible. I start by simply using labels to improve upon the trustworthiness of latent topic models in the MM-LDA model of Chapter 3. However, MM-LDA suffers from the same interpretability difficulties faced by all latent topic models, suggesting a better alternative use for label data: as an anchor for organizing the learned topics. The first such model using labels in this way, Labeled LDA in Chapter 4, restricts each topic to explicitly align with a single label, addressing the c*redit attribution problem* of determining which label is most responsible for the presence of any given word in a document. I extend Labeled LDA in Chapter 5 with the PLDA and PLDP models, which re-introduce the flexibility of latent topics to Labeled LDA while retaining the former's interpretability advantages. As a whole, the four models present a portrait of how labels can be used to build trustworthy, interpretable, and flexible text mining tools.

This dissertation has demonstrated the viability of using datasets rich with human-interpretable metadata to discover—and place in context—patterns about *people*, *organizations*, and *ideas*. These three categories of interest identified in Chapter 1 cover a wide range of computational social science questions. Yet despite the power of these models, they are not a one-size-fits-all approach to every question in computational social science. They must be supported by domain-specific validation.

Indeed, this dissertation has devoted a great deal of effort towards validating the learned models in embedded tasks and case studies throughout Chapters 3, 4, and 5. The longer case studies of language on microblogs (in Chapter 6) and academia (Chapter 7) illustrate the central importance of domain-specific validation even more clearly: we take pains to validate our understanding of the latent topic space on Twitter through interviews and surveys, and we check our intuitions of academic language incorporation with known precedents. Only then—once the model has confirmed some of the patterns that we expected to find—can we begin to trust its ability to tell us things we didn't already know or couldn't have known otherwise. This form of domain-specific validation is always needed to establish the trust of the practitioner with his or her audience. Improvements in modeling can extend the scope of

answerable questions, but can never replace the need for well designed validations.

Despite the contributions of this dissertation, what the field continues to lack is a satisfying, general theory that connects the space of problems, the properties of datasets and models, and methodologies for establishing trust. I have not formulated this general theory, although I have illustrated some examples of how it might play out in Chapters 6 and 7. In particular, I believe that when a dataset has labels of interest, explaining the words in terms of those labels as much as possible leads to clearer mechanisms for validation and more interpretable results. As a corollary, when latent structure is unnecessary, it should be avoided.

The reason for both lessons is straightforward: as text mining practitioners, we tend to over-fit our own intuitions. We examine each model we train for the errors and inconsistencies we know to expect, training the models to match our own insights. It is only by showing the output of the model to a third party—preferably a domain expert without knowledge of model internals—that we can get a sense of whether the model matches *other people's* intuitions, too. If the model supports clear, interactive visualization, all the better: visualization can lead to simplified exploration of the individual examples that really validate findings or expose flaws. From practical experience, I've found that models trained to align with a label space—like Labeled LDA and PLDA—are better at matching my own intuitions and those of others than purely unsupervised models like LDA.

Other recent approaches by topic modeling researchers are promising from a modeling perspective, as well: work in evaluating the quality of LDA models with respect to human intuition has shown that some models generate output that is more semantically coherent than others [29]. Indeed, [96] explicitly optimizes the semantic coherence as part of the topic model, while [3] demonstrates ways that domain knowledge can be encoded into LDA models in terms of explicit word-pairing preferences on the learned β distributions. While this dissertation does not explicitly compare to these works, I find the recent interest in incorporating domain knowledge and semantic coherence a heartening development in the field, and indicative of the broad potential of other modeling approaches to core text mining challenges.

In this dissertation, I have posited that we can build models that are simultaneously trustworthy, interpretable, and flexible by using the machinery of latent topic models to learn word distributions constrained to align with the human-interpretable metadata present in so many modern text collections. I use these models to show that tags can improve clustering on the web in Chapter 3; that people's language on microblogs can be categorized into *substance, status, social,* and *style* aspects in Chapter 6; that large scale structural change in academia can be documented (Section 7.4); that interdisciplinary work can be identified (Section 7.4.4); and that dissertations crossing disciplinary boundaries tend to look more like academia's future (Section 7.6). While these results stand on their own, I hope that the approach taken in this dissertation acts as a stepping stone toward even better methodologies and analyses for uncovering the richness of our world through text.

Bibliography

[1] Open directory project. http://dmoz.org/.

[2] Andrew Abbott. *Chaos of Disciplines*. University of Chicago Press, Chicago, 2001.

[3] D. Andrzejewski, X. Zhu, and M. Craven. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 25–32. ACM, 2009.

[4] A. Asuncion, M. Welling, P. Smyth, and Y.W. Teh. On smoothing and inference for topic models. In *UAI*, 2009.

[5] Arthur Asuncion, Padhraic Smyth, and Max Welling. Asynchronous distributed learning of topic models. In *NIPS*, pages 81–88, 2008.

[6] M. Aurnhammer, P. Hanappe, and L. Steels. Integrating collaborative tagging and emergent semantics for image retrieval. In *Collaborative Web Tagging Workshop (WWW'06)*, 2006.

[7] Stéphane Baldi. Departmental quality ratings and visibility: The advantages of size and age. *The American Sociologist*, 28:89–101, 1997. 10.1007/s12108-997-1028-x.

[8] Shenghua Bao, Guirong Xue, Xiaoyuan Wu, Yong Yu, Ben Fei, and Zhong Su. Optimizing web search using social annotations. In *WWW*, pages 501–510, 2007.

[9] S. Basu, M. Bilenko, A. Banerjee, and R. J. Mooney. Probabilistic semi-supervised clustering with constraints. In O. Chapelle, B. Schoelkopf, and A. Zien, editors, *Semi-Supervised Learning*, pages 73–102. MIT Press, 2006.

[10] G. Begelman, P. Keller, and F. Smadja. Automated tag clustering: Improving search and exploration in the tag space. In *Collaborative Web Tagging Workshop (WWW'06)*, 2006.

[11] B. Berendt and C. Hanser. Tags are not metadata, but "just more content"–to some people. In *ICWSM*, 2007.

[12] Eli M. Blatt. *The semantics of success: Cultural evolution and science citation.* PhD thesis, Stanford University, 2008.

[13] D. Blei, T.L. Griffiths, M.I. Jordan, and J.B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *NIPS*, volume 16, page 106, 2003.

[14] D. Blei and J McAuliffe. Supervised topic models. In *NIPS*, volume 21, 2007.

[15] D. M. Blei and J. Lafferty. Correlated topic models. In *NIPS*, volume 18, page 147. MIT, 2006.

[16] David M. Blei, Andrew Ng, and Michael Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[17] D.M. Blei and M.I. Jordan. Modeling annotated data. In *SIGIR*, pages 127–134. ACM Press New York, NY, USA, 2003.

[18] K. W. Boyack. Using detailed maps of science to identify potential collaborations. *Scientometrics*, 79(1):27–44, 2009.

[19] K. W. Boyack, K. Börner, and R. Klavans. Mapping the structure and evolution of chemistry research. *Scientometrics*, 79(1):45–60, 2009.

[20] K.W. Boyack, R. Klavans, and K. Börner. Mapping the backbone of science. *Scientometrics*, 64(3):351–374, 2005.

[21] D. boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: conversational aspects of retweeting on twitter. In *HICSS*, pages 1–10, 2010.

[22] SRK Branavan, H. Chen, J. Eisenstein, and R. Barzilay. Learning document-level semantic properties from free-text annotations. *Journal of Artificial Intelligence Research*, 34(1):569–603, 2009.

[23] C.H. Brooks and N. Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *WWW*, 2006.

[24] W. Burggren, K. Chapman, B. Keller, M. Monticino, and J. Torday. Biological sciences. In R Frodeman, JT Klein, and C Mitcham, editors, *Oxford Handbook of Interdisciplinarity*, chapter 8, pages 119–132. Oxford University Press, New York, 2010.

[25] J.B. Callicott. The environment. In R Frodeman, JT Klein, and C Mitcham, editors, *Oxford Handbook of Interdisciplinarity*, chapter 33. Oxford University Press, New York, 2010.

[26] C. Cattuto, A. Barrat, A. Baldassarri, Schehr G., and V. Loreto. Collective dynamics of social annotation. In *PNAS*, pages 10511–10515, 2009.

[27] Soumen Chakrabarti, Martin van den Berg, and Byron Dom. Focused crawling: a new approach to topic-specific web resource discovery. In *WWW '99: Proceedings of the eighth international conference on World Wide Web*, pages 1623–1640, New York, NY, USA, 1999. Elsevier North-Holland, Inc.

[28] J. Chang, J. Boyd-Graber, and D.M. Blei. Connections between the lines: augmenting social networks with text. In *KDD*, pages 169–178. ACM, 2009.

[29] J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, and D. Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*. Citeseer, 2009.

[30] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-supervised learning*. MIT Press, 2006.

[31] W.W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. In *SIGIR*, 1999.

[32] D.R. Cutting, D.R. Karger, J.O. Pedersen, and J.W. Tukey. Scatter/gather: a cluster-based approach to browsing large document collections. In *SIGIR*, pages 318–329. ACM Press New York, NY, USA, 1992.

[33] I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, 2006.

[34] M. Dalrymple. *Lexical functional grammar*. Academic Press, New York, 2001.

[35] H. Daumé III. Markov random topic fields. In *ACL*, 2009.

[36] D. De Solla Price. *Little Science, Big Science...and Beyond*. Columbia University Press, 1986.

[37] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[38] O. Dekel, P.M. Long, and Y. Singer. Online learning of multiple tasks with a shared loss. *JMLR*, 8:2233–2264, 2007.

[39] D. Easley and J. Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.

[40] Nadav Eiron and Kevin S. McCurley. Analysis of anchor text for web search. In *SIGIR*, pages 459–460, New York, NY, USA, 2003. ACM.

[41] E. Fisher and D. Beltran del Rio. Mathematics and root interdisciplinarity. In R. Frodeman, J.T. Klein, and C. Mitcham, editors, *Oxford Handbook of Interdisciplinarity*, pages 88–90. Oxford University Press, New York, 2010.

[42] Johannes Fürnkranz. Exploiting structural information for text classification on the WWW. In *IDA '99: Proceedings of the Third International Symposium on Advances in Intelligent Data Analysis*, pages 487–498, London, UK, 1999. Springer-Verlag.

[43] P. Galison. *Image and Logic: a Material Culture of Microphysics*. University of Chicago Press, Chicago, 1997.

[44] Scott A. Golder and Michael W. Macy. Diurnal and seasonal mood vary with work, sleep, and daylength across diverse cultures. *Science*, 333(6051):1878–1881, September 2011.

[45] G. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis*, 2(2):205–224, 1965.

[46] J.T. Goodman. A bit of progress in language modeling. *Computer Speech & Language*, 15(4):403–434, 2001.

[47] P. Graham. A plan for spam, August 2002. http://www.paulgraham.com/spam.html.

[48] T. L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 101:5228–5235, 2004.

[49] J. Grimmer. A bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases. *Political Analysis*, 18(1):1–35, 2010.

[50] D.L.W. Hall, D. Jurafsky, and C.D. Manning. Studying the history of ideas using topic models. In *EMNLP*, 2008.

[51] Paul J. Hanges and Julie S. Lyon. Relationship between u.s. news and world report's and the national research council's ratings/rankings of psychology departments. *American Psychologist*, 60(9):1035–1037, Dec 2005.

[52] Z.S. Harris. Distributional structure. *Word*, 10:146–162, 1954.

[53] T. Haveliwala, A. Gionis, D. Klein, and P. Indyk. Evaluating strategies for similarity search on the web. In *WWW*, 2002.

[54] Taher H. Haveliwala. Topic-sensitive pagerank. In *WWW*, pages 517–526, New York, NY, USA, 2002. ACM.

[55] C. Hayes and P. Avesani. Using tags and clustering to identify topic-relevant blogs. In *ICWSM*, 2007.

[56] Marti A. Hearst and Jan O. Pedersen. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *SIGIR*, pages 76–84, New York, NY, USA, 1996. ACM.

[57] G. Heinrich. Parameter estimation for text analysis. Technical report, 2004. *http://www.arbylon.net/publications/text-est.pdf*.

[58] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Can social bookmarking improve web search. In *WSDM*, 2008.

[59] M. Hoffman, D. Blei, and F. Bach. Online learning for latent dirichlet allocation. In *Neural Information Processing Systems*, 2010.

[60] Thomas Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, New York, NY, USA, 1999. ACM.

[61] C. Honeycutt and S.C. Herring. Beyond microblogging: Conversation and collaboration via twitter. In *HICSS*, 2009.

[62] A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. *The Semantic Web: Research and Applications*, 4011:411–426, 2006.

[63] T. Iwata, T. Yamada, and N. Ueda. Modeling Social Annotation Data with Content Relevance using a Topic Model. In *NIPS*, 2009.

[64] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: an analysis of a microblogging community. In *Advances in Web Mining and Web Usage Analysis*, pages 118–138, 2009.

[65] S. Ji, L. Tang, S. Yu, and J. Ye. Extracting shared subspace for multi-label classification. In *KDD*, pages 381–389, New York, NY, USA, 2008. ACM.

[66] H. Kazawa, H. Taira T. Izumitani, and E. Maeda. Maximal margin labeling for multi-topic text categorization. In *NIPS*, 2004.

[67] M. Kilduff and W. Tsai. *Social networks and organizations*. Sage Publications Ltd, 2003.

[68] J.T. Klein. *Interdisciplinarity: history, theory, and practice*. Wayne State University Press, 1990.

[69] K. Knorr-Cetina. *Epistemic Cultures: How the Sciences Make Knowledge*. Harvard University Press.

[70] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics, 2007.

[71] B. Krishnamurthy, P. P. Gill, and M. Arlitt. A few chirps about twitter. In *WOSP*, 2008.

[72] S. Kübler, R. McDonald, and J. Nivre. *Dependency parsing*, volume 2 of *Synthesis lectures on human language technologies*. Morgan & Claypool, US, 2009.

[73] T.S. Kuhn. *The Structure of Scientific Revolutions*. University of Chicago Press, Chicago, 1962.

[74] S. Lacoste-Julien, F. Sha, and M. I. Jordan. DiscLDA: Discriminative learning for dimensionality reduction and classification. In *NIPS*, volume 22, 2008.

[75] B. Latour. *Science in Action*. Harvard University Press, 1987.

[76] D.D. Lee and H.S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, volume 13, 2001.

[77] D. D. Lewis, Y. Yang, T. G. Rose, G. Dietterich, F. Li, and F. Li. RCV1: A new benchmark collection for text categorization research. *JMLR*, 5:361–397, 2004.

[78] Wei Li and Andrew McCallum. Pachinko allocation: DAG-structured mixture models of topic correlations. In *International Conference on Machine Learning*, pages 577–584, 2006.

[79] C. Lin and Y. He. Joint sentiment/topic model for sentiment analysis. In *CIKM*, pages 375–384. ACM, 2009.

[80] D. Lin and P. Pantel. DIRT-discovery of inference rules from text. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 323–328. Citeseer, 2001.

[81] T. Liu, S. Liu, Z. Chen, and W.Y. Ma. An evaluation on feature selection for text clustering. In *ICML '03*.

[82] X. Liu and W.B. Croft. Cluster-based retrieval using language models. In *SIGIR'04*.

[83] Mary J. Golladay Lori Thurgood and Susan T. Hill. U.S. doctorates in the 20th century. Technical Report NSF 06-319, National Science Foundation, Division of Science Resources Statistics, Arlington, VA 2006, 2006.

[84] C. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[85] C.D. Manning and H. Schütze. *Foundations of statistical natural language processing*. The MIT Press, 1999.

[86] M.P. Marcus, M.A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330, 1993.

[87] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 7, 1998.

[88] A.K. McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI Workshop on Text Learning*, 1999.

[89] A.C. McCormick and C.M. Zhao. Rethinking and reframing the Carnegie classification. *Change: The Magazine of Higher Learning*, 37(5):51–57, 2005.

[90] K.R. McKeown, R. Barzilay, D. Evans, V. Hatzivassiloglou, J.L. Klavans, A. Nenkova, C. Sable, B. Schiffman, and S. Sigelman. Tracking and summarizing news on a daily basis with Columbia's Newsblaster. In *Proceedings of the second international conference on Human Language Technology Research*, pages 280–285. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 2002.

[91] Q. Mei, X. Shen, and C Zhai. Automatic labeling of multinomial topic models. In *KDD*, 2007.

[92] M. Meilă. Comparing clusterings by the variation of information. *Learning theory and kernel machines*, pages 173–187, 2003.

[93] Louis Menand. *The Marketplace of Ideas: Reform and Resistance in the American University*. W.W. Norton & Company, New York, 2010.

[94] V. Metsis, I. Androutsopoulos, and G. Paliouras. Spam filtering with naive Bayes—which naive Bayes. In *Third conference on email and anti-spam (CEAS)*, volume 17, pages 28–69. Citeseer, 2006.

[95] D. Mimno and A. McCallum. Expertise modeling for matching papers with reviewers. In *KDD*, pages 500–509. ACM, 2007.

[96] D. Mimno, H.M. Wallach, E.T.M. Leenders, and A. McCallum. Optimizing semantic coherence in topic models. In *EMNLP*, 2011.

[97] M. Naaman, J. Boase, and C.H. Lai. Is it really about me?: Message content in social awareness streams. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, pages 189–192, 2010.

[98] R. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, June 2000.

[99] M. E. J. Newman. The structure of scientific collaboration networks. *PNAS*, 92(2):404–409, 2001.

[100] US News and World Report. Best graduate school rankings, 2010. http://www.usnews.com/rankings.

[101] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2010.

[102] S. Osinski and D. Weiss. A concept-driven algorithm for clustering search results. *IEEE Intelligent Systems*, 20(3):48–54, 2005.

[103] J. P. Ostriker, C. V. Kuh, and J. A. Voytuk. A data-based assessment of research-doctorate programs in the United States. In *Committee to Assess Research-Doctorate Programs*. National Research Council, 2010.

[104] Chung C.K. Ireland M. Gonzales A. & Booth R.J. Pennebaker, J.W. *The development and psychometric properties of LIWC2007.* [Software manual], Austin, TX, 2007.

[105] J.W. Pennebaker, M.E. Francis, and R.J. Booth. *Linguistic Inquiry and Word Count: LIWC2001.* Erlbaum Publishers, Mahwah, NJ, 2001.

[106] C.J. Pollard and I.A. Sag. *Head-driven phrase structure grammar*. University of Chicago Press, 1994.

[107] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 569–577. ACM, 2008.

[108] M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[109] ProQuest. Proquest dissertation publishing, 2011. http://www.proquest.com/en-US/products/dissertations/.

[110] L. Rabiner and B.H. Juang. *Fundamentals of speech recognition*. Prentice hall, 1993.

[111] D. Ramage, S. Dumais, and D. Liebling. Characterizing microblogs with topic models. In *ICWSM*, 2010.

[112] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled LDA: A supervised topic model for credit attribution in multi-label corpora. In *EMNLP*, pages 248–256, 2009.

[113] D. Ramage, P. Heymann, C. D. Manning, and H. Garcia-Molina. Clustering the tagged web. In *WSDM*, 2009.

[114] Daniel Ramage, Christopher D. Manning, and Susan T. Dumais. Partially labeled topic models for interpretable text mining. In *KDD*, pages 457–465, 2011.

[115] W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, pages 846–850, 1971.

[116] T. Rattenbury, N. Good, and M. Naaman. Towards automatic extraction of event and place semantics from Flickr tags. In *SIGIR '07*.

[117] A. Ritter, C. Cherry, and B. Dolan. Unsupervised modeling of twitter conversations. In *HLT NAACL*, pages 172–180. Association for Computational Linguistics, 2010.

[118] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *UAI*. AUAI Press, 2004.

[119] T.N. Rubin, UC Irvine, A. Holloway, P. Smyth, and M. Steyvers. Modeling Tag Dependencies in Tagged Documents. 2009.

[120] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[121] L. Shih, Y.H. Chang, J. Rennie, and D. Karger. Not too hot, not too cold: The bundled-SVM is just right! In *ICML Workshop on Text Learning*. Citeseer, 2002.

[122] T.D. Snyder et al. *Digest of Education Statistics, 1994*. US Government Printing Office, 1994.

[123] Daniel Stokols, Kara L. Hall, Brandie K. Taylor, and Richard P. Moser. The science of team science: Overview of the field and introduction to the supplement. *American Journal of Preventive Medicine*, 35(2, Supplement):S77–S89, 2008.

[124] A. Strehl. *Relationship-based clustering and cluster ensembles for high-dimensional data mining*. PhD thesis, The University of Texas at Austin, 2002.

[125] A. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. In *AAAI Workshop on AI for Web Search (AAAI 2000)*, pages 58–64, 2000.

[126] David L. Tan. The assessment of quality in higher education: A critical review of the literature and research. *Research in Higher Education*, 24:223–265, 1986. 10.1007/BF00992074.

[127] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[128] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *WWW*, pages 111–120. ACM, 2008.

[129] L.N. Trefethen and D. Bau. *Numerical linear algebra*. Number 50. Society for Industrial Mathematics, 1997.

[130] N. Ueda and K. Saito. Parametric mixture models for multi-labeled text includes models that can be seen to fit within a dimensionality reduction framework. In *NIPS*, 2003.

[131] C. J. Van Rijsbergen. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow, 1979.

[132] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *SIGCHI*, CHI '04, pages 319–326, New York, NY, USA, 2004. ACM.

[133] E.M. Voorhees. The cluster hypothesis revisited. In *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 188–196. ACM, 1985.

[134] C. Wang, B. Thiesson, C. Meek, and D.M. Blei. Markov topic models. In *AISTATS*, pages 583–590, 2009.

[135] David Weakliem, Gordon Gauchat, and Bradley Wright. Sociological stratification: Change and continuity in the distribution of departmental prestige, 1965-2007. *The American Sociologist*, Online First:1–18, 2011. 10.1007/s12108-011-9133-2.

[136] X. Wei and W.B. Croft. LDA-based document models for ad-hoc retrieval. In *SIGIR '06*.

[137] P. Weingart. A short history of knowledge formations. In R Frodeman, JT Klein, and C Mitcham, editors, *Oxford Handbook of Interdisciplinarity*, pages 3–14. Oxford University Press, New York, 2010.

[138] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In *NIPS*, pages 521–528, 2003.

[139] Y. Yanbe, A. Jatowt, S. Nakamura, and K. Tanaka. Can social bookmarking enhance search in the web? In *JCDL '07*.

[140] Y. Yang and J.O. Pedersen. A comparative study on feature selection in text categorization. In *ICML '97*.

[141] Oren Zamir and Oren Etzioni. Web document clustering: a feasibility demonstration. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 46–54, New York, NY, USA, 1998. ACM.

[142] H.J. Zeng, Q.C. He, Z. Chen, W.Y. Ma, and J. Ma. Learning to cluster web search results. In *SIGIR '04*.

[143] Ding Zhou, Jiang Bian, Shuyi Zheng, Hongyuan Zha, and C. Lee Giles. Exploring social annotations for information retrieval. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 715–724, New York, NY, USA, 2008. ACM.

[144] J. Zhu, A. Ahmed, and E.P. Xing. MedLDA: Maximum margin supervised topic models for regression and classification. In *ICML*, pages 1257–1264. ACM, 2009.

[145] X. Zhu. Semi-Supervised Learning Literature Survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.