12

Care and Feeding of Topic Models: Problems, Diagnostics, and Improvementes

Jordan Boyd-Graber

University of Maryland, Institute for Advanced Computer Studies, College Park, MD 20742, USA

David Mimno

Cornell University, Information Science, Ithaca, NY 14850, USA

David Newman

Google Los Angeles, Venice, CA 90291, USA

CONTENTS

Topic models are a versatile tool for understanding corpora, but they are not perfect. In this chapter, we describe the problems users often encounter when using topic models for the first time. We begin with the preprocessing choices users must make when creating a corpus for topic modeling for the first time, followed by options users have for running topic models. After a user has a topic model learned from data, we describe how users know whether they have a good topic model or not and give a summary of the common problems users have, and how those problems can be addressed and solved by recent advances in both models and tools.

12.1 Introduction

Topic models are statistical models for learning the latent structure in document collections, and have gained much attention in the machine learning community over the last decade. Topic models improve the ways users find and discover text content in digital libraries, search interfaces, and across the web, through their ability to automatically learn and apply subject tags to documents in a collection. However, this potential requires practitioners to overcome the problems often associated with topic models: when to use them, how to know when there are problems, how to fix those problems, and how to make topic models more useful.

Topic modeling is an increasingly popular framework for simultaneously soft clustering terms and documents into a fixed number of topics, which take the form of a multinomial distribution over terms in the document collection. Topic models are useful for a variety of research tasks and user-facing applications described below. We start by introducing notation for the original generative topic model, latent Dirichlet allocation (LDA) (Blei et al., 2003).

Latent Dirichlet allocation and its extensions form one popular class of topic models and will be the basis of discussion for this chapter. The LDA topic model is based on the assumption that documents have multiple topics.

In LDA topic modeling, each of D documents in the corpus is modeled as a discrete distribution over T latent topics, and each topic is a discrete distribution over the vocabulary of W words. In the LDA topic model, the number of topics T is fixed and specified by the modeler. For document d, the distribution over topics, $\theta_{t|d}$, is drawn from a Dirichlet distribution Dir[α], where α might either be a symmetric constant vector (say $\alpha_0\mathbf{1}$) or a hyperparameter with variable values (say $(\alpha_1, ..., \alpha_T)$) which can be estimated. Likewise, each distribution over words, $\phi_{w|t}$, is drawn from a Dirichlet distribution Dir[β].

For the ith token in a document, a topic assignment z_{id} is drawn from $\theta_{t|d}$ and the word, x_{id}, is drawn from the corresponding topic, $\phi_{w|z_{id}}$. Hence, the generative process in LDA is given by

$$\theta_{t|d} \sim \text{Dir}[\alpha] \qquad \phi_{w|t} \sim \text{Dir}[\beta] \tag{12.1}$$

$$z_{id} \sim \text{Mult}[\theta_{t|d}] \qquad x_{id} \sim \text{Mult}[\phi_{w|z_{id}}]. \tag{12.2}$$

We can compute the posterior distribution of the topic assignments via Gibbs sampling or variational inference. Given samples from the posterior distribution we can compute point estimates of the document-topic proportions $\theta_{t|d}$ and the word-topic probabilities $\phi_{w|t}$. We will henceforth denote $\boldsymbol{\phi}_t$ as the vector of word probabilities for a given topic t.

The original LDA topic model has been extended in dozens of ways. Most of the extensions are a result of addressing a potential limitation of LDA, or taking advantage of an opportunity made available by additional data. Some notable extensions include: the correlated topic model (Blei and Lafferty, 2005); the nonparametric topic model, or hierarchical Dirichlet process model (Teh et al., 2006); the hierarchical topic model (Blei et al., 2007); and the dynamic topic model (Blei and Lafferty, 2006). To a large extent, these particular extensions have not directly addressed some of the usability issues we focus on in this chapter.

Nevertheless, there has been a thriving cottage industry adding more and more information to topic models to correct some of the shortcomings we are interested in, either by modeling perspective (Paul and Girju, 2010; Lin et al., 2006), syntax (Wallach, 2006; Gruber et al., 2007), or authorship (Rosen-Zvi et al., 2004; Dietz et al., 2007). Similarly, there has been an effort to inject semantic knowledge into topic models (Boyd-Graber et al., 2007).

12.1.1 Using Topic Models

In the academic literature, topic modeling has been demonstrated to be highly effective in a wide range of research-oriented tasks, including multi-document summarization (Haghighi and Vanderwende, 2009), word sense discrimination (Brody and Lapata, 2009), sentiment analysis (Titov and McDonald, 2008), machine translation (Eidelman et al., 2012), information retrieval (Wei and Croft, 2006), discourse analysis (Purver et al., 2006; Nguyen et al., 2012), and image labeling (Fei-Fei and Perona, 2005). In these tasks the topics are used as features in some larger algorithm, and not as first-order outputs of interest.

Beyond these research-type tasks, topic modeling has been demonstrated in several user-facing applications. Here, the topics themselves are of direct interest. Applications range from search and discovery interfaces to other types of collection analysis interfaces. There are several noteworthy examples, including two from the U.S. funding agencies, NIH and NSF. The NIH Map ViewerTopic[1] is both a topic-based search interface and a map visualizing the research funded by NIH (Talley et al., 2011). The STAR METRICS Portfolio Explorer[2] features topics describing NSF-funded research. Another example is the topic model browser for the journal *Science*.[3]

The remainder of this chapter is organized as follows. In this section, we further introduce topic modeling: how one goes from raw data to a topic model. In Section 12.2, we talk about problems and issues with topic modeling. In Section 12.3, we discuss diagnostics that are useful for detecting and measuring these problems. Finally, in Section 12.4 we review new methods aimed at improving the performance and utility of topic models in addition to those aimed at addressing some of their problems.

12.1.2 Preprocessing Text Data

Topic models take documents that contain words as input. This seems simple enough, but often the process of going from a source document to a form that can be understood by topic models drastically changes the final output. Suppose, for example, that we wanted to build a topic model using Wikipedia as our data source. How would we turn that into a sequence of words that could be used as input to a topic model?

Readers experienced with data processing and natural language processing can safely skip to Section 12.1.3, where we assume that we have the necessary input data for topic modeling.

First, let's take a look at what an individual Wikipedia page looks like:[4]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
 <html lang="en" dir="ltr" class="client-nojs"
 xmlns="http://www.w3.org/1999/xhtml"> <head> <title>Princess
 Ida - Wikipedia, the free encyclopedia</title> <meta
 http-equiv="Content-Type" content="text/html; charset=UTF-8"
 /> <meta http-equiv="Content-Style-Type" content="text/css"
 /> <meta name="generator" content="MediaWiki 1.18wmf1" />
```

[1] See https://app.nihmaps.org.

[2] See http://readidata.nitrd.gov/star/.

[3] See http://topics.cs.princeton.edu/Science/.

[4] For this example, we use the HTML representation of a Wikipedia article. This is because it's easy to inspect on the web, isn't restricted by copyrights, and has many of the problems that web corpora have. For real applications, you should **not** use HTML served by Wikipedia's web servers but instead download their XML dumps available at http://dumps.wikimedia.org. This will make your life easier (it lacks many of the problems that we address in this section) and will save both you and Wikipedia bandwidth.

Little in this raw format is what we would call a word, and being able to effectively use this as an input to topic models would require us to do substantial preprocessing. Once we remove extraneous material, we still have to determine what "words" we're going to use and how to extract them from the remaining text. We go through each of these steps to produce a document in a form that is usable for topic modeling.

Many times the files that comprise our corpus have extraneous information that do not add to the *content* of the data. With the *Princess Ida* example, HTML obscures what the underlying words are. We can remove them using a regular expression or a variety of text processing tools (e.g., using the *Natural Language Toolkit* (Bird et al., 2009)).

```
Princess Ida - Wikipedia, the free encyclopedia Princess Ida From
Wikipedia, the free encyclopedia Jump to: navigation , search
Princess Ida; or, Castle Adamant is a comic opera with music by
Arthur Sullivan and libretto by W. S. Gilbert. It was their eighth
operatic collaboration of fourteen.

⋮

Personal tools Log in / create account Namespaces Article Discussion
Variants Views Read Edit View history Actions Search Navigation Main
page Contents Featured content Current events Random article
Donate to Wikipedia Interaction Help About Wikipedia Community por-
tal Recent changes Contact Wikipedia Toolbox What links here Related
changes Upload file Special pages Permanent link Cite this page
Print/export Create a book Download as PDF Printable version
Languages Fran\xc3\xa7ais Italiano This page was last modified on 23
September 2011 at 23:59. Text is available under the Creative
Commons Attribution-ShareAlike License; additional terms may apply.
See Terms of use for details. Wikipedia&reg; is a registered trade-
mark of the Wikimedia Foundation, Inc., a non-profit organization.
Contact us Privacy policy About Wikipedia Disclaimers Mobile view
```

Now that we've removed some of the HTML that obscured the content, we can see content that is ofter referred to as boilerplate: text that is repeated verbatim across many documents. Many forms of boilerplate (Freedman, 2007) text appears on this Wikipedia page. Some of it fulfills a legal function ("Text is available under the Creative Commons"), a navigation function ("Search Navigation"), and some of it provides metadata ("last modified on").

While these data are useful and necessary for an HTML page, they do not tell us about the *content* of the document, which is the goal of topic modeling. Failing to remove this boilerplate material can result in the discovery of topics that include just this boilerplate text. Because such text is on many pages, this is often a suboptimal result.

Typically, boilerplate can be removed by heuristics (e.g., removal of the first or last N bytes), or failing that, methods that can discover boilerplate (Kohlschütter et al., 2010). Such text can take many forms: signatures from prolific posters in a newsgroup, legalese in advertisements, contact information in press releases, or quotes appearing at the start of book chapters.

Removing such boilerplate gives us:

```
Princess Ida; or, Castle Adamant is a comic opera with music by Arthur
Sullivan and libretto by W. S. Gilbert. It was their eighth operatic
collaboration of fourteen. Princess Ida opened at the Savoy Theatre on
January 5, 1884, for a run of 246 performances. The piece concerns a
princess who founds a women's university and teaches that women are
superior to men and should rule in their stead. The prince to whom she
had been married in infancy sneaks into the university, together with
two friends, with the aim of collecting his bride. They disguise them-
selves as women students but are discovered, and all soon face a
literal war between the sexes.
```

which is finally getting us the content we want. Now we can begin extracting words from the text. Recall that most topic models treat documents as a bag-of-words, so we can stop caring about the order of the tokens within the text and concentrate on how many times a particular word appears in the text.

With this in mind, below we show the sixty most frequent "words" sorted by frequency if we consider words to be anything delimited by whitespace.

the	and	of	to	in	a	The	[	]
Princess	Ida	Gilbert	that	Sullivan	,	%	was	his
(	by	is	with	.	for	as	Carte	at
D'Oyly	her	p.	on	£	).	King	not	she
Lady	had	Act	I	opera	edit	)	but	Opera
are	from	1884	Hilarion	London	Savoy	has	women's	you
were	Hilarion,	In	first	Company	Gama	W.	he	if

Many of these strings are not what we would consider to be words but are instead punctuation. In most applications of topic modeling, we do not care about the punctuation used, so we likely want to remove them. Many of these words are also not content words; words like "the," "and," "of," etc. are functional words that don't provide any information about what the article is about. Such terms are typically called stopwords.

In addition to including items that are not helping us understand what the document is about, we are also making distinctions between words that under most reasonable interpretations should be viewed as identical. For example, the words "Hilarion" and "Hilarion" are considered to be distinct. Similarly, "opera" and "Opera" are considered to be distinct. This suggests that we need to be more aggressive when separating words.

On the other hand, there are also clues that we need to be less aggressive in separating words. For example, there are multi-word expressions that we might want to treat as pseudowords—e.g., "gilbert and sullivan" might be a reasonable multiword expression to treat as a fixed unit, as would "princess ida" and "king gama."[5]

How do we address these issues? These problems are typically viewed as problems of stopword removal, normalization, tokenization, and collocation discovery. We discuss each of them in turn.

Stopword Removal

The most common way to remove words that do not contribute to the meaning of a document is to use a fixed list. Such lists are available in many languages and typically take care of most stopwords. However, such lists are not complete, and there are often corpus-specific stopwords that such lists

[5]There has been considerable interest in simultaneously discovering multiword expressions either *after* topic modeling (Blei and Lafferty, 2009) or as part of the process for discovering topics (Johnson, 2010; Hardisty et al., 2010) . However, we view it as a preprocessing step (which is much more efficient).

would never discover. For example, in the Wikipedia corpus, "edit" or "citation" might appear so often in the HTML pages of Wikipedia that they do not serve to differentiate documents.[6]

Rather than having a set list of stopwords, other approaches take an adaptive threshold for which words are stopwords. For example, one could compute the tf-idf (Salton, 1968) of each term in a document and only consider terms that are above some reasonably set threshold.

Normalization

Here, we use normalization in a very broad sense. For a particular concept, there may be many different character strings that can represent it in a language. For instance, "Dog," "Dogs," "dog," and "dogs" both refer to the same underlying concept, except that some are plural, and some are capitalized. For the purposes of topic modeling, we may wish to assume that these are actually the same word. Converting to lower case and applying a stemming algorithm (Porter, 1980) can convert all of these to a cannonicalform, "dog."

For languages with a richer morphology (Taghva et al., 2005), this is particularly critical. Failing to do so can lead to an overly large vocabulary (which slows inference) and can lead to poorer topics, as identical words in slightly different syntactic contexts are treated as distinct. However, for English, this is more a matter of taste. When topics are designed for human inspection, many users prefer not to see stemmed words.

Tokenization

Tokenization (or segmentation) is the process for breaking a string of text into its constituent words. For English, whitespace is a good proxy for detecting word boundaries. However, it is not perfect (as we saw above), and there may be other conventions for breaking a string of text into constituent words. For example, Treebank tokenization (Marcus et al., 1993) separates "won't" into "wo" and "n't." Other languages with implicit word boundaries may require more involved preprocessing (Goldwater et al., 2006).

Collocation Discovery

Often, a word's meaning is constrained by its local context (Schemann and Knight, 1995). For example, "house" means one thing, but when it appears together with "white house," it means quite another. Discovering multi-word expressions is a common task in natural language processing (Manning and Schütze, 1999). Often, topic modeling is done while ignoring multiword expressions.

This can lead to suboptimal outcomes for a number of reasons. First, it can lead to topics that join together unrelated concepts. For example, by treating "soviet" and "union" as separate tokens, a topic model might group together documents on the soviet union and the civil war (Chang et al., 2009a) . Even when topic models don't make such errors, it can annoy savvy users who see obvious multi-word expressions separated or displayed in the wrong order (e.g., displaying a topic as "bush," "clinton," "house," and "white" as a topic).

Let us now return to our Wikipedia article on *Princess Ida*, where we identified bigrams scored by point-wise mutual information (PMI), removed stopwords, and tokenized based on all punctuation and whitespace. We did not perform any normalization beyond converting everything to lower case. This gives a much more reasonable list of the most frequent words (seen in the following table).

Note, however, that there are still some problems: "opera" and "operas" are still distinct, "d'oyly carte" was turned into "oyly carte", and "edit" (a wikipedia-specific stopword) are still present. If we believe that these were problematic (or if we saw such issues in the output), we could apply a

[6] In practice, one should use Wikipedia XML dump, which would avoid some of these issues; again, we're using the HTML version to give examples of some of the issues that might arise with web corpora

princess_ida	sullivan	opera	princess	gilbert
chorus	ida	oyly_carte	gilbert_and_sullivan	edit
london	women	1884	first	hilarion
king_hildebrand	may	king_gama	university	act
college	hildebrand	company	lady_blanche	men
one	ainger	melissa	musical	new
piece	productions	florian	gently	lady_psyche
plot	production	recordings	richard	role
rollins_and_witts	savoy	savoy_theatre	tennyson	three
castle_adamant	cyril	early	gama	january_1884
john	man	music	operas	revival
1870	1954	also	although	arthur_sullivan

stemming algorithm that would strip terminal "s" on plurals (at the risk of diminishing interpretability), improve tokenization (at the risk of allowing spurious punctuation to enter words), or add to our stop list (at the risk of removing real content-bearing contributions to documents).

At this point, it's often helpful to look at the most frequent words summed over all documents. This often gives you an idea of where problems might lie. If the results look reasonable, then you can press ahead with inference.

12.1.3 Running Topic Models

There are many different implementations of topic modeling software available;[7] each has (or should!) have its own discussion of how to specifically run the models and prepare input. The goal of this section is not to describe how to run any particular implementation but to talk about what needs to happen to go from raw data to an inferred topic model.

Broadly, implementations fall into two general categories: those that use variational inference (Blei et al., 2003) or Gibbs sampling (Griffiths and Steyvers, 2004)While describing these techniques is outside the scope of this chapter, they both attempt to discover the latent variables that best explain a dataset.

Preparing Data

After completing the steps in Section 12.1.2, the data must be converted into a form that is efficiently readable by software. This takes two steps: selecting the vocabulary and representing the data.

Typically this is done by converting strings into integers (e.g., "opera" is 0, "princess_ida" is 1). Typically you do not want to create an integer for every unique string that appears as a type in your corpus. It increases the amount of memory and time needed to run inference and can also introduce errors from misspellings or tokenization errors. Because natural languages have a power-law distribution, many types only appear in a handful of documents (or one). Including such types is useless for topic models, which attempt to generalize across documents.

Next, the data are reduced to this integer form. There are two ways to do this: representing a document by a single array of integers, with each element in the array corresponding to one appearance of a word, or as two paired arrays a and b, where $a[i]$ represents the identity of a word and $b[i]$ represents the frequency of the word in a document. The former is more common for inference using sampling; the latter is more common for variational inference.

[7] For most uses, we suggest Mallet, http://mallet.cs.umass.edu. For particularly large datasets, we suggest Yahoo LDA (Narayanamurthy, 2011) or Mr. LDA (Zhai et al., 2012).

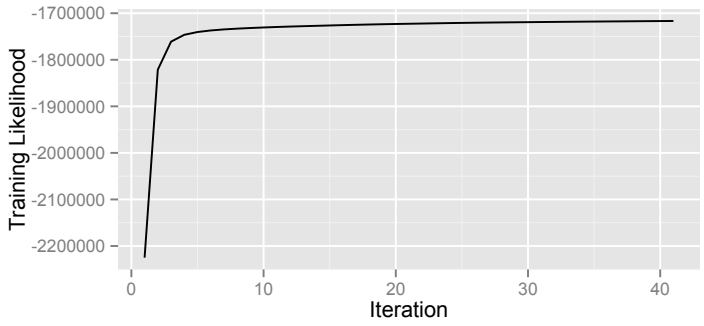

FIGURE 12.1
Training likelihood for variational inference. The shape of the curve shows that inference is increasing likelihood and is nearly converged. Other inference methods may have different convergence profiles, but it should have a similar shape.

Initialization

Both variational inference and Gibbs sampling can be viewed as a search over latent variables. Variational inference searches over variational parameters that induce a distribution over a model's latent variables, and states in the Markov chain for Gibbs sampling are direct assignments of latent variables. Thus, in either case, models must be initialized.

The most important aspect of initialization is to avoid local minima. Some initializations are 'good enough' so that inference will not want to leave the initial state. One common example of this is initializing the variational distributions as uniform distributions; this is a local optimum and will not allow inference to improve upon the initialized state (with boring, identical, uniform topics).

A better approach is to initialize randomly. In practice, this results in either perturbing the initial variational distributions from uniform slightly or, in a Gibbs sampler, setting topic assignments uniformly at random.

Another approach is to initialize the state in a way that might give your algorithm a boost to speed convergence. For example, one could initialize a topic model by initializing each topic with a single document. For other models, other initializations are also possible, but it is important to be aware of the possibility of falling into a local minimum. If inference is working correctly, your model should not be that sensitive to initialization.

Regardless of how you initialize your model and regardless of what inference technique you use, it's important to have many multiple starting points to inference. This guards against problems of local optima and allows you to make better estimates about the stability of your inferred latent variables.

Inference

Running inference itself is the most important step in the process; it produces a learned model from raw data. If you've implemented inference yourself, it is also likely that this aspect has taken the most time.

Typically, implementations work based on a series of iterations. Each iteration updates slightly the state of the algorithm, working slowly toward finding a local optimum. With each iteration, the model should estimate the data likelihood, i.e., given the current guess of the latent variables, what is the probability of recreating the data?

You should watch this quantity closely. If the quantity is consistently going down, it probably means you have a bug. If the quantity is improving steadily, it is a good sign that inference is making

progress (although there could be other problems lurking underneath). It is difficult to say how many iterations are needed for inference; it depends on initialization, the data size, the complexity of the model, and what form of inference you're using. However, once the likelihood converges to a value, it is usually a sign that your inference has converged (although this is not always a sure-fire indicator, particularly for MCMC (Neal, 1993)).

Ready-made implementations should provide this information to you; even if you trust the code, you should still pay attention to verify that inference is progressing as it should.

Hyperparameter Updating

Hyperparameters in topic models are those that are not latent variables in the model but instead are the 'most basic' parameters of the topic model. Typically, these are the Dirichlet parameters that are assumed to have generated the per-document topic distributions and the per-topic distributions over words. More generally, these are any unknown parameters that govern latent variables (and are a part of any statistical model, not just topic models).

Particularly if you've derived inference for the model yourself, it's very tempting to set hyperparameters and forget them. After all, you're getting good results, the models are learning interesting things, and you've proved your point. At the risk of editorializing, we would encourage authors to explore sampling hyperparameters:

- It is not that hard, both from the programmer's perspective and from the amount of time it takes the computer;
- If you're using any kind of perplexity or likelihood-based evaluation, you will almost certainly lose to anything that does hyperparameter optimization (Wallach et al., 2009a) ; and
- It will improve the (qualitative) quality of the results.

12.1.4 Evaluation of Topic Models

One of the most important features of topic modeling is that it does not require 'supervision' in the form of annotations. In addition to text documents, many text mining and NLP tasks require additional information such as document-level labels for classification, word-level labels for part-of-speech tagging, phrase-structure trees for parsing, and relevance judgments for information retrieval. With the exception of classification and translation, document creators do not naturally produce such labels, and hiring experts to add annotations can be expensive and time-consuming. In contrast, topic models require only a segmentation of documents into word tokens. They can therefore be applied quickly to large volumes of data.

The benefit of supervised models, however, is that if we take the human-generated labels as a gold standard, measuring and comparing the performance of different methods is simple: we hold out a section of the labeled data as a testing set, train a model on the remaining data, and ask that model to predict labels for the testing set. If the predicted labels match the 'true' labels, the model is effectively learning the association between input data and output labels. In topic modeling, where the model is not trained to predict specific topics, there is no supervised gold standard.

Finding patterns in data is the central goal of topic modeling, but in order to make scientific statements, we must also be able to make predictions about future observations. As an alternative to predicting annotations given previously unseen documents, we can attempt to predict the unseen documents themselves. Simply generating documents and comparing them to a held-out set, however, is not feasible. In classification, there are a finite number of possible document labels. For a given testing document, even random guessing has a reasonably good probability of selecting the correct label. In contrast, the number of possible sequences of words from a vocabulary is exponential in the length of the document. Therefore, rather than measuring accuracy or some rank-based

metric, we calculate the marginal probability of the held-out documents under the model. This metric measures the degree to which the model concentrates its probability mass on a relatively small set of 'sensible' documents rather than the vastly larger set of completely random documents.

If, given some held-out document set $\mathbf{w}$, some model A assigns greater marginal probability $p(\mathbf{w}|A)$ than some model B, we assume that model A has more effectively learned the language of the document set than model B. Model A is, in some sense, less 'surprised' by the real documents than model B. Borrowing a term from statistical language modeling, we refer to the negative log probability of the held-out set divided by the number of tokens $-\log p(\mathbf{w}|A)/|\mathbf{w}|$ as the *perplexity* of model A.

Unfortunately, even measuring the marginal probability of a document under a topic model is not computationally tractable due to the exponentially large number of possible topic assignments for words. Good approximations, however, can be evaluated tractably (Wallach et al., 2009b; Buntine, 2009).

Although measurements of held-out probability are important, they are not, by themselves, sufficient. There are several common problems:

- People use topic models to summarize the semantic components of a large document collection, but good predictive power does not necessarily mean that a model provides a meaningful representation of concepts.
- Users frequently distinguish between the quality of different topics: some are seen as coherent or pure, while others are seen as random or illogical. Marginal probability, however, depends on all topics, and therefore cannot be easily decomposed as a function of individual topics.
- Calculations of marginal probability can be sensitive to hyperparameter settings.

12.2 Problems

The topic model is based on the simple assumption that documents contain multiple topics. But is this assumption valid? An article on salary caps in the NFL may be about *sports* and *remuneration*, but do those two topics account for every word written in that article? And is the bag-of-words assumption (that word order is irrelevant) valid? In topic models, every word in a document is probabilistically assigned a topic label, and therefore topics need to explain or account for all words that appear. Is this a reasonable assumption?

Topic models are based on a generative model that clearly does not match the way humans write. However, topic models are often able to learn meaningful and sensible models. Of course, models are learned from the data—a collection of documents—so the quality of the model depends on the quality of the training data.

Most evaluation of topic models has focused on statistical measures of perplexity or likelihood of test data. But this type of evaluation has limitations. The perplexity measure does not reflect the semantic coherence of individual topics learned by a topic model, nor does perplexity necessarily indicate how well a topic model will perform in some end-user task. Recent research has shown potential issues with perplexity as a measure— Chang et al. (2009b) suggests that human judgments can be contrary to perplexity measures.

With this in mind, we pose the following overarching questions relating to evaluating topic models:

Q1 Are individual topics meaningful, interpretable, coherent, and useful?
Q2 Are assignments of topics to documents meaningful, appropriate, and useful?

Q3 Do topics facilitate better or more efficient document search, navigation, understanding, browsing?

While the final question is ultimately the most important for assessing the end-user utility of topic models, it is appropriate to address these questions in order. It doesn't make sense to talk about the quality of assignments of topics to documents if one can't agree on what a topic is about. Although topics themselves are not the end goal (the end goal is to use topics to improve some end-user task), the evaluation framework is built on the usability and usefulness of individual topics, and our focus in this chapter is primarily on the first of the three questions.

12.2.1 Categories of Poor Quality Topics

Before considering bad topics, it is helpful to consider what we are looking for in a topic. The following topic has several good, though not essential, properties:

trout fish fly fishing water angler stream rod flies salmon...

It is specific. There is a clear focus on words related to the sport of trout fishing. It is coherent. All of the words are likely to appear near one another in a document. Some words (*water, fly*) are ambiguous and may occur in other contexts, but they are appropriate for this context. It is concrete. We can picture the angler with his rod catching trout in the stream. It is informative. Someone unfamiliar with the topic can work from general words (*fishing*) to learn about more unfamiliar words (*angler*). Relationships between entities can be inferred (trout and salmon both live in streams and can be caught in similar ways).

There are a variety of ways topics can be "bad," and we list some of them here. This value judgement is contextual: "good" or "bad" depends on a variety of factors that may involve the task, user, experience, etc. Here we take "bad" as some general idea of lack of usability, usefulness, utility, etc.

General and Specific Words

In any natural language, the most frequent words have less specific meaning, while rare words have very precise meanings. Stopwords such as *the, and, of* are the most extreme examples, but this gradient in specificity remains even after removing such words. For example, in a collection of publications from an artificial intelligence conference, words in the 99th percentile by token frequency might include *algorithm, model, estimation*. At the opposite end, there are large numbers of words that occur only once or twice, such as *dopaminergic* and *phytoplankton*.

notion sense choice situation idea natural explicitly explicit definition refer...

level significantly_higher significantly_lower lower higher_lever measured significantly_ different investigate differ tended positive_correlation significantly_increased...

might doesn't fact anyone does isn't mean anyway point quite...

quite rather couple wasn't far seems less three however point...

Topic models often contain one or more topics consisting of frequent, non-specific words. Users perceive these topics as overly general and therefore not useful in understanding the divisions within a corpus. Such topics often consist of the most frequent words that were not removed as part of the stoplist.

Low-frequency words can also be problematic. Topics that contain many specific words are often perceived as unhelpful because they do not provide a general overview of the corpus. Such topics are also more vulnerable to random chance than topics containing more frequent words because they rely on words with small sample sizes.

Mixed and Chained Topics

Many topics are perceived as low quality by users because they are "mixed" or "chained."

zinc migraine veterans zn headache magnesium military war zn2 csd affairs episodic deficiency...

A mixed topic can be defined as a set of words $\mathcal{T} = \{w_1, w_2, w_3, ..., w_N\}$ that do not make sense together, but that contains subsets $\mathcal{S}_1, \mathcal{S}_2, ...$, each of which individually form a sensible combination of words. For example, the words

dog, cat, bird, honda, chevrolet, bmw

do not make sense together, but *dog, cat, bird* describe animals and *honda, chevrolet, bmw* describe makes of cars.

A chained topic is like a mixed topic: a set of words that is low quality overall but contains high quality subsets. The difference is that in a chained topic every high quality subset shares at least one word with another subset. For example, the set

reagan, roosevelt, clinton, lincoln, honda, chevrolet, bmw

combines the names of U.S. presidents with makes of cars, but *lincoln* can be both categories. Chained topics can be caused by ambiguous words such as *lincoln*, but can also result from hierarchical relationships. A broader concept like *tax* may include several narrower concepts (*sales tax, property tax*). These more specific individual words (*sales, property*) may by themselves form non-sensical combinations.

Identical Topics

One common problem with the topic models learned on corpora is that the topics all look the same (or nearly so). Since topic models are meant to explain a corpus, having identical topics is clearly a suboptimal outcome. We discuss some of the possible causes of this outcome and how you can fix them.

company customer market product business revenue companies software...

market product company sale patent companies commercial cost...

One reason that topics might appear to be identical is that the *prior* topic distribution is being observed. Normally, the prior distribution is combined with data to produce a posterior conditioned on that data. However, the prior is still a model of text even without data, and most implementations will happily provide the prior distribution as the "result," even if it has not been supplied with data.

This result might be of particular concern if the inference took a suspiciously short amount of time or if inference chose not to use some of the topics available to it. Both problems are relatively easy to fix—perhaps preprocessing created empty documents or too many topics were chosen.

Incomplete Stopword List

In contrast, one of the symptoms of an incomplete stopword list is topics filled with highly frequent words (but the topics are not identical). Often, the topics discovered are perfectly reasonable, but buried underneath the convention of displaying the n most probable words in a topic.

vii viii xiv xiii xii xvi xix xviii xvii xxix xxx xxi xxii xxiv xiii...

david nick elizabeth brad kelsey ted drew theresa ricky russell...

This is often resolved by adding the most frequent words in the topics to the stopword list and then rerunning inference. Alternatively, one could adopt models that have asymmetric priors (Wallach et al., 2009a) or explicitly model syntax (Griffiths et al., 2005; Boyd-Graber and Blei, 2008).

Nonsensical Topics

Another possible problem is that the topics learned will be distinct, but otherwise inscrutable. This is often the result of preprocessing errors or providing the model with too much information.

tree plum ink blossom **chp** branch bird paper...

Remember that topic models discover words that often appear together in documents. If your "documents" evince a structure that has similar correlation patterns between "words," it will gladly create a topic (we use scare quotes to highlight that the determination of what a document and word is is often subjective and is often impacted by preprocessing steps).

For example, if some documents are created by optical character recognition (OCR), frequent OCR errors will likely occur together; this can create a topic of such errors. Similarly, if metadata are included in the specification of a document, this also might create topics to model this boilerplate material (e.g., as we did in Section 12.1.2).

12.3 Diagnostics

Now we have topics, but how do we know how good the topics are? Traditionally in the literature, measurements have focused on measures based on held-out likelihood (Blei et al., 2003; Blei and Lafferty, 2005) or an external task that is independent of the topic space such as sentiment detection (Titov and McDonald, 2008) or information retrieval (Wei and Croft, 2006). This is true even for models engineered to have semantically coherent topics (Boyd-Graber et al., 2007).

For models that use held-out likelihood, Wallach et al. (2009b) provides a summary of evaluation techniques. These metrics borrow tools from the language modeling community to measure how well the information learned from a corpus applies to unseen documents. These metrics generalize easily and allow for likelihood-based comparisons of different models or selection of model parameters such as the number of topics. However, this adaptability comes at a cost: these methods only measure the probability of observations; the internal representation of the models is ignored.

However, not measuring the internal representation of topic models is at odds with their presentation and development. Most topic modeling papers display qualitative assessments of the inferred topics or simply assert that topics are semantically meaningful, and practitioners use topics for model-checking during the development process. Hall et al. (2008), for example, used latent topics

deemed historically relevant to explore themes in the scientific literature. Even in production environments, topics are presented as themes: Rexa,[8] a scholarly publication search engine, displays the topics associated with documents.

In this section, we focus on metrics that *do* pay attention to the underlying topics either by asking individuals directly or by measuring the properties of the discovered topics.

12.3.1 Human Evaluation of Topics

Chang et al. (2009b) presented the following task to evaluate the latent space of topic models. In the word intrusion task, the subject is presented with six randomly ordered words. The task of the user is to find the word which is out of place or does not belong with the others, i.e., the *intruder*.

When the set of words minus the intruder makes sense together, then the subject should easily identify the intruder. For example, most people readily identify *apple* as the intruding word in the set: dog, cat, horse, apple, pig, cow because the remaining words: dog, cat, horse, pig, cow make sense together—they are all animals. For the set: car, teacher, platypus, agile, blue, Zaire, which lacks such coherence, identifying the intruder is difficult. People will typically choose an intruder at random, implying a topic with poor coherence.

In order to construct a set to present to the subject, they select a topic from the model. They then select the five most probable words from that topic. In addition to these words, an intruder word is selected at random from a pool of words with low probability in the current topic (to reduce the possibility that the intruder comes from the same semantic group) but high probability in some other topic (to ensure that the intruder is not rejected outright due solely to rarity). All six words are then shuffled and presented to the subject.

What Topics Make Sense?

The word intrusion task was applied to two corpora: The *New York Times* (Sandhaus, 2008) and Wikipedia,[9] two real-world corpora that are viewed by millions of people each day. Figure 12.2 shows the spectrum from incoherent to coherent topics.

An additional finding was that there was not a clear association between traditional measures of topic models, such as held-out log-likelihood and more intuitive measures such as the word intrusion task.

12.3.2 Topic Diagnostic Metrics

While the techniques described in the previous section are useful, they are time consuming and relatively expensive. Are there ways to measure topic quality without relying on human judgments? Fortunately, there are several useful topic diagnostic metrics that depend only on statistics of individual words in a topic without considering relationships between words or external knowledge sources. None of these metrics is conclusive by itself, but taken together they can provide a useful automated summary of topic quality. As a running example, we consider a model trained with 100 topics on a corpus of political blogs from the 2008 U.S. presidential election.

Topic Size

Most topic model inference methods work by assigning the word tokens in a corpus to one of K topics. We can add up the number of tokens or fractions of a token assigned to a given topic to get a measure of topic size, where the unit is the number of word tokens. There is a strong relation between this measure of topic size and perceived topic quality: very small topics are frequently

[8] See http://rexa.info.
[9] See www.wikipedia.org.

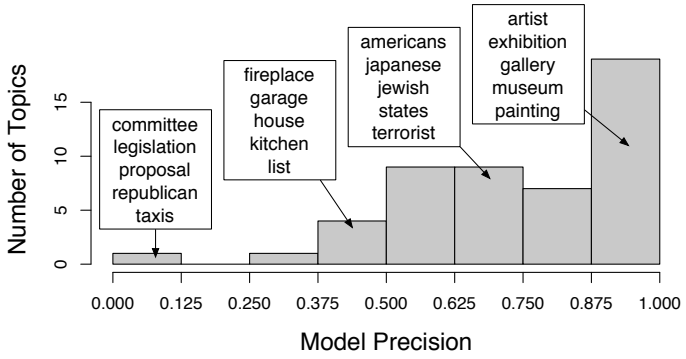

FIGURE 12.2
A histogram of the model precisions (the proportion of times users found the "intruder") on the *New York Times* corpus evaluated on a 50-topic LDA model. Example topics are shown for several bins; the topics in bins with higher model precision evince a more coherent theme.

bad (Talley et al., 2011). As an example, in the 2008 political blog model, the smallest topic by token count is *http player video window flag script false scriptalreadyrequested www*, with around 6,500 words (most topics lie between 15,000 and 20,000). This topic appears to represent URLs for embedded videos. Although it is arguably interpretable, it is not the sort of conceptual topic that many users may be looking for.

There are several possible explanations for this relationship. The most common topics in a corpus are usually well-represented in many documents. For the less-frequent topics, the model must estimate their word distribution from a smaller sample size. Smaller topics are also more vulnerable to become mixed with other topics because they do not 'own' their distribution as well.

Word Length

This metric measures the average length of the top N words in a topic. The usefulness of this metric varies by corpus, but in many cases it can be useful in picking up anomalous topics. The intuition is that words with more specific meaning tend to be longer, and vice versa. Examples include topics consisting of stopwords from a language other than the primary language of the corpus, and topics with many short acronyms, which are frequently ambiguous. In the political blog corpus, the topics with the smallest average word length are *legislator usmc aye nc ny fl pa oh ca tx va* (2.7 characters) and *re ll exit don doesn ve isn didn maverick guy* (4.15 characters). The *legislator* topic appears to represent abbreviations for U.S. states, perhaps related to legislative roll call voting. As with the previous metric, word length in this case does not necessarily indicate that a topic is uninterpretable, but it flags the fact that this is a different sort of cluster of co-occurring words. The *re* topic is more problematic, and indicates that there may be problems with tokenization of contractions such as *you're* or *don't*, possibly due to differences in character encodings for the apostrophe.

Distance from a Corpus Distribution

A topic is a probability distribution over the vocabulary of a corpus. We can define a "global" topic by counting the number of times each word is used in all documents and normalizing those counts. Topic distributions that are similar to this corpus-level distribution according to some measure of similarity between distributions, such as Jensen-Shannon distance or Hellinger distance, consist

of the most common words in a corpus. These topics are often perceived as useless or overly general (AlSumait et al., 2009). The most common non-stopwords need to be assigned somewhere, so having a small number of these overly general topics may help to improve the quality of other topics, but it may not be necessary to display them to users.

Distance from the corpus distribution is most useful for documents that contain formulaic or administrative language, such as grant proposals. In corpora focused on a particular issue, this metric may be less useful. The most frequent words in the 2008 political blog corpus are *iraq war country states military security*, indicating that the corpus is dominated by discussion of the Iraq war. The closest topic to this overall distribution is *iraq troops war surge iraqi withdrawal security petraeus military forces*, which is a useful, coherent topic.

Difference between Token and Document Frequencies

We typically rank words within a topic by the number of word tokens (or fractional tokens) of a particular type that have been observed in the topic. We can also rank words within a topic by the number of documents that contain at least one token of a particular type in that topic. The difference between the token-based distribution over words and this document-based distribution is useful in identifying words that are prominent in a topic due to the burstiness of words. When a corpus contains many long documents, it is common for a word that is specific to a single document to occur often enough in that one document that it appears in the list of N top words for a topic. The highest ranking topic according to this metric is the *re* topic mentioned previously, where the tokens *re* and *ll* are the most bursty, possibly reflecting occasional use of second-person pronouns. The metric can also detect outlier words in otherwise more usable topics. The second most bursty topic is *financial crisis bailout fannie mortgage loans wall banks*, where the term *fannie* (referring to the U.S. financial entity known as Fannie Mae) is the most bursty.

Prominence within Documents

Topics often represent the major themes of a document, but they can also be clusters of "methodological" words, like words describing measurement (*larger, smaller, fast*) or days of the week. A good method for distinguishing between important topics and these more functional topics is to examine the proportion of documents assigned to a topic. The names of months may occur many times in a corpus, and more consistently with each other than any other words, but no documents are dominated by month names in the way that a document might be about molecular biology or a political debate. This property can be defined mathematically in several ways. One method is to count the number of documents such that the estimated probability of topic k $\widetilde{\theta}_k$ is above some threshold, such as 0.2. Another is to count the number of documents for which topic k is the single largest topic. For example, the topic *meeting official officials conference visit senior reported event friday* is relatively large, with over 42,000 tokens, but it never appears as the single largest topic in any document. Meetings and conferences occur frequently, but are not by themselves worth discussing in great depth. In contrast, the topic *franken coleman ballots minnesota votes recount al board counted* has only a quarter of the total tokens of the *meeting* topic, but is the largest topic in 12.5% of the documents it appears in. This topic, about a contested senate election, refers to a specific event that is discussed in depth when it is discussed at all.

Burstiness

Many of the problems people observe in topic models are caused by the phenomenon of *burstiness* in natural language documents. This property states that within a context, for example a short document, there will be a small set of words that are globally rare but locally common.

Burstiness is related to, but distinct from, well-known power law properties of natural language. If we construct a list of all the distinct words in a corpus of documents and record, for each word,

the number of documents that contain at least one instance of that word, the vast majority of those words will be rare, that is, they will occur in very few documents. The most common words, on the other hand, will make up roughly one half of the tokens in any given document. This relationship is known as Zipf's law.

Zipfian dynamics suggest that many of the words in a document will be rare, but burstiness describes an additional level of non-uniformity. It is not only likely that many of the tokens in a document will be rare, but it is also likely that many of them will be the *same rare word*. For example, assume you know the overall word-frequency statistics of a corpus. You can estimate the probability of every distinct word by dividing the number of occurrences of that word by the total number of tokens in the corpus $\mathcal{C}$. If you know nothing about a certain document, these corpus-level frequencies provide a reasonable estimator of the probability that a randomly selected word from that document will be, for example, *elephant*. For most words this probability will be a small number $p(w|\mathcal{C}) = \epsilon$. Once you have observed a particular word, however, the probability that the next word sampled at random from the same document will be of the same type is much larger than ϵ.

This phenomenon of burstiness violates the assumptions of a topic model, which assert that if we know the topic for a token position in a document, the probability that the word at that position is a particular type is independent of the document. When a topic is well represented in a corpus and most documents are short, the violations of this assumption may be averaged out. If there are long documents, however, the bursty words in those documents may have high prevalence in a topic despite not being representative of the central concept of a topic. Similarly, if a topic appears in only a few documents, each of which has its own bursty subset of the words that are associated with the topic, the topic may appear idiosyncratic or nonsensical.

When confronted with a bursty corpus, it may be useful to filter your documents so that documents are of similar length, perhaps by removing abnormally long documents or by breaking very long documents into smaller documents. It may also be worthwhile to consider particularly bursty words as stopwords to prevent them from dominating topics.

12.3.3 Topic Coherence Metrics

Our goal of answering whether individual topics are interpretable and coherent is partly addressed by the human evaluation of topics in Section 12.3.1. But how can we automatically measure topic coherence? And can we do this without disturbing the topic by adding intruder words? Earlier work presented an unsupervised approach to ranking topic significance and identifying what they call "junk" or "insignificant" topics (AlSumait et al., 2009). However, it was unclear to what extent their unsupervised approach and objective function agreed with human judgments, as they presented no user evaluations.

Subsequent work demonstrated that it is possible to automatically measure topic coherence with near-human accuracy (Newman et al., 2010a;b) using a topic coherence score based on pointwise mutual information of pairs of terms taken from topics. In both Newman et al. (2010a) and Newman et al. (2010b), 6000 human evaluations are used to show that their coherence score broadly agrees with human-judged topic coherence. Similar approaches further confirmed that humans agree with word-pair based topic coherence metrics (Mimno et al., 2011).

Topic coherence metrics are motivated by measuring word association between pairs of words in the list of the top-10 most likely topic words (here, top-10 is chosen arbitrarily as the typical number of terms displayed to a user; other settings such as top-20 could work equally as well). The intuition is that a topic will likely be judged as coherent if pairs of words from that topic are associated. Devising word association measures is a long-studied problem in computational linguistics. We opt for co-occurrence-based metrics that use corpus aggregates of the number of times two words are seen in a document. There are two flavors of counting term co-occurrences: either using a sliding window of fixed size (e.g., Do two terms appear in a window of 20 consecutive words?), or binarized

at the document level (e.g., Does this document contain both these terms?). The former makes the metric more biased toward short-range dependencies.

For a final twist, we could either use the training corpus to count term co-occurrences, or we could opt for an *external* corpus to obtain these counts. The former is certainly easier, but one may be concerned that the training corpus is not representative—or may be polluted by unusual termwise statistics—as may happen in a text collection of blogs or tweets. In this case, the external corpus could come from a variety of sources, for example the entire collection of English Wikipedia articles.

Our topic coherence metrics take the form

$$\text{TC-f}(\mathbf{w}) = \sum_{i<j} f(w_i, w_j),\ i, j \in \{1 \dots 10\}, \tag{12.3}$$

where $\mathbf{w} = \{w_1, w_2, \dots, w_{10}\}$ are the top-10 most likely terms in a topic, and f is some function measuring the association between words w_i and w_j.

Let $N(w_i, w_j)$ be the number of times word w_i and w_j co-appear in a sliding window of fixed width (say 20 terms), applied to every document in the corpus used to obtain co-occurrence counts. Furthermore, $N(w_i)$ is the total count of times w_i appeared in that sliding window. Let $M(w_i, w_j)$ be the number of distinct documents where words w_i and w_j co-appear, and $M(w_i)$ is the total number of distinct documents that include term w_i. We create different metrics by using N or M to convert counts to probabilities, using the appropriate normalization. Two obvious quantities are pointwise mutual information (PMI) and log conditional probability (LCP). Note that PMI is symmetric, whereas LCP is one-sided.

$$\text{PMI}(w_i, w_j) = \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}, \tag{12.4}$$

$$\text{LCP}(w_i, w_j) = \log \frac{P(w_i, w_j)}{P(w_j)}. \tag{12.5}$$

Using these, we define the following three topic coherence metrics:

$$\text{TC-PMI}(\mathbf{w}) = \sum_{i<j} \text{PMI}(w_i, w_j), \tag{12.6}$$

$$\text{TC-LCP}(\mathbf{w}) = \sum_{i<j} \text{LCP}(w_i, w_j), \tag{12.7}$$

$$\text{TC-NZ}(\mathbf{w}) = \sum_{i<j} \mathbf{1}[N(w_i, w_j) = 0], \tag{12.8}$$

where all sums are over $i, j \in \{1 \dots 10\}$. Note, we have added a third metric that simply counts the number of word pairs that are *never* observed in the reference corpus. These topic coherence metrics can be computed four different ways: using sliding window (N) or binarized (M) counts, obtained from training data or external data. For LCP, we can also do a symmetric metric instead of a one-sided metric by switching $i < j$ for $i \neq j$. When $N(w_i, w_j) = 0$, smoothing is required to compute a finite LCP, and for PMI we simply assume independence, PMI $= 0$.

Using TC-PMI computed with a 20-word sliding window on the entire 3M articles in English Wikipedia, Newman et al. (2010a;b) compared computed topic coherence to 6000 human-judged coherence scores, and obtained a Spearman rank correlation of $\rho = 0.8$, approximately the same as the inter-rater correlation computed on a leave-one-out basis. This topic coherence metric was used by Lau et al. (2010) for their best topic word task, and it performed well at detecting Chang et al.'s intruder word (Chang et al., 2009b).

We conclude this section by showing how these three different topic coherence metrics differ. Here, we focus on the metrics' ability to identify poor quality topics. We list sample topics learned from a collection of *New York Times* news articles, showing the lowest-scoring topics using the three metrics:

TC-PMI

```
why bad thing maybe doesn something does let isn really...
self sense often history yet power seems become itself perhaps...
came went told took later didn room began asked away...
need better problem must enough does likely less whether...
```

TC-LCP

```
space canadian station canada nasa mission air shuttle crew hughes...
fight lewis jones tyson vegas las boxing ring murphy elvis...
ball body wright arms watson puerto club rico hands swing...
blood thompson wilson cell test gladwin disease nixon gas sickle...
```

TC-NZ

```
eminem connor shea hanson mile daniels abbott seymour black trupia...
porter amin burke olsen omar horse horses martinez ruettgers botai...
hart hunter troy mack willis oxygen scooter terry chayes farrell...
greene weber sims fashion fairchild malley fletcher crosby sawyer
mccann...
```

The above examples show how PMI, LCP, and NZ-based topic coherence metrics identify different types of poor quality topics. TC-PMI tends to show poor quality topics that include terms that are more general and more frequent. TC-LCP shows topics that appear to relate to a nameable subject, but nevertheless are relatively incoherent. Finally, TC-NZ appears to do a good job at identifying the classic topic-of-names that is often learned by topic models.

12.4 Improving Topic Models

Now that we know what problems can appear in topic models and how to detect them, what can we do about them? At a high level, the problems can be interpreted as topics containing words that should not be together but are (e.g., "mixed" or "chained" topics) or distinct topics that should be together but aren't.

In this section, we discuss techniques to adapt the statistical formulating of topic models to incorporate these intuitive descriptions of problematic topics to create analysis of datasets that are more useful and more understandable. We also include a discussion on automatic topic labeling, another technique to improve the utility of topic models.

12.4.1 Interactive Topic Models

First, let's begin with a common-use case: a frustrated consumer of topic models staring at a collection of topics that do not make sense. In this section, we discuss interactive topic modeling (ITM), an in situ method for incorporating human knowledge into topic models.[10]

[10]For full details, see Hu et al. (To Appear).

Recall that LDA views topics as distributions over words, and each document expresses an admixture of these topics. For "vanilla" LDA, these are symmetric Dirichlet distributions. A document is composed of observed words, which we call tokens, to distinguish specific observations from the word (type) associated with each token. Because LDA assumes a document's tokens are interchangeable, it treats the document as a bag-of-words, ignoring potential relations between words.

Constraints Change the Topics Discovered

This problem with vanilla LDA can be solved by encoding constraints, which will 'guide' different words into the same topic. Constraints change the underlying distribution by forcing words to either be positively or negatively correlated with each other. If a user sees two words that should appear in the same topic but do not, they can impose a positive correlation between the words. If the user sees two words that appear in a topic together but should not, they can impose a negative correlation between the words.

These correlations work by changing the underlying probabilistic model; while vanilla topic models assume that each topic is a distribution over words, we use tree-structured topics (Boyd-Graber et al., 2007; Andrzejewski et al., 2009). These models instead assume that topics first have a distribution over *concepts* and these concepts in turn have a distribution over words. By encoding word distributions as a tree, we can preserve conjugacy and relatively simple inference while encouraging correlations between words that are grouped together in concepts.

While these models can encourage words to be negatively or positively correlated, these constraints on the model must be added interactively as the user sees problems that must be corrected.

Interactively Adding Constraints

Interactively changing constraints can be accommodated in ITM, smoothly transitioning from unconstrained LDA to constrained LDA with one constraint, to constrained LDA with two constraints, etc.

A central tool that we use to transition between models is the strategic unassignment of states, which we call *ablation* (distinct from feature ablation in supervised learning). Gibbs sampling inference stores the topic assignment of each token. In the implementation of a Gibbs sampler, unassignment is done by setting a token's topic assignment to an invalid topic and decrementing any counts associated with that word.

The constraints created by users implicitly signal that words in constraints don't belong in a given topic. In other models, this input is sometimes used to 'fix,' i.e., deterministically hold constant topic assignments (Ramage et al., 2009). Instead, we change the underlying model, using the current topic assignments as a starting position for a new Markov chain with some states strategically unassigned; this is equivalent to performing online inference (Yao et al., 2009).

An alternative would be to not pursue this interactive strategy but instead restart inference from a new initialization. This, however, is counter to the goals of pursuing topic modeling interactively: restarting inference increases the latency users have to wait to see an updated model, restarting the model destroys any mental mapping of the model, and restarting the model could create additional problems into the model.

Merging Topics

To examine the viability of ITM, we begin with a qualitative demonstration that shows the potential usefulness of ITM. For this task, we used a corpus of about 2000 *New York Times* editorials from the years 1987 to 1996. We started by finding 20 initial topics with no constraints, as shown in Table 12.1 (left).

Notice that Topics 1 and 20 both deal with Russia. Topic 20 seems to be about the Soviet Union, with Topic 1 about the post-Soviet years. We wanted to combine the two into a single topic, so we

created a constraint with all of the clearly Russian or Soviet words (*boris, communist, gorbachev, mikhail, russia, russian, soviet, union, yeltsin*). Running inference forward 100 iterations with the **Doc** ablation strategy yields the topics in Table 12.1 (right). The two Russia topics were combined into Topic 20. This combination also pulled in other relevant words that were not near the top of either topic before: "moscow" and "relations." Topic 1 is now more about elections in countries other than Russia. The other 18 topics changed little.

While we combined the Russian topics, other researchers analyzing large corpora might preserve the Soviet vs. post-Soviet distinction but combine topics about American government. ITM allows tuning for specific tasks.

Topic	Words
1	election, yeltsin, russian, political, party, democratic, russia, president, democracy, boris, country, south, years, month, government, vote, since, leader, presidential, military
2	new, york, city, state, mayor, budget, giuliani, council, cuomo, gov, plan, year, rudolph, dinkins, lead, need, governor, legislature, pataki, david
3	nuclear, arms, weapon, defense, treaty, missile, world, unite, yet, soviet, lead, secretary, would, control, korea, intelligence, test, nation, country, testing
4	president, bush, administration, clinton, american, force, reagan, war, unite, lead, economic, iraq, congress, america, iraqi, policy, aid, international, military, see
⋮	⋮
20	soviet, lead, gorbachev, union, west, mikhail, reform, change, europe, leaders, poland, communist, know, old, right, human, washington, western, bring, party

Topic	Words
1	election, democratic, south, country, president, party, africa, lead, even, democracy, leader, presidential, week, politics, minister, percent, voter, last, month, years
2	new, york, city, state, mayor, budget, council, giuliani, gov, cuomo, year, rudolph, dinkins, legislature, plan, david, governor, pataki, need, cut
3	nuclear, arms, weapon, treaty, defense, war, missile, may, come, test, american, world, would, need, lead, get, join, yet, clinton, nation
4	president, administration, bush, clinton, war, unite, force, reagan, american, america, make, nation, military, iraq, iraqi, troops, international, country, yesterday, plan
⋮	⋮
20	soviet, union, economic, reform, yeltsin, russian, lead, russia, gorbachev, leaders, west, president, boris, moscow, europe, poland, mikhail, communist, power, relations

TABLE 12.1
Five topics from a 20-topic topic model on the editorials from the *New York Times* before adding a constraint (left) and after (right). After the constraint was added, which encouraged Russian and Soviet terms to be in the same topic, non-Russian terms gained increased prominence in Topic 1, and "Moscow" (which was not part of the constraint) appeared in Topic 20.

However, user constraints are not absolute. For example, in experiments some users attempted to merge topics about Apple computers and IBM-compatible personal computers discovered from the 20 Newsgroups corpus.[11] However, the model preferred to explain the data using two separate topics.

Separating Topics

Another possible imperfection in a topic model is that a single topic conflates two concepts that should be in distinct topics. This can be corrected by adding a constraint that two words cannot appear in the same topic. For example, in a collection of biomedical publications, a topic might be discovered that contains both words related to spinal cord and the urinary tract. Upon showing this to a domain expert—an NIH program manager—it was found that this was incorrect clustering. Introducing a constraint that these two words should not appear together results in the new topics in Table 12.2.

12.4.2 Generalized Pólya Urn Models

A topic model claims that, given topic assignments, the observed words are selected i.i.d. from a single set of topic distributions. If this assumption is true, then the expected number of documents that contain any pair of words w_i, w_j assigned to topic k should be a function of $p(w_i|k)$ and $p(w_j|k)$. Under this model, if those two probabilities are both large, it is unlikely that there will be

[11] See http://people.csail.mit.edu/jrennie/20Newsgroups/.

Before	After	
bladder	spinal_cord	bladder
spinal_cord	spinal_cord_injury	women
sci	spinal	oc
spinal_cord_injury	injury	pelvic_floor
spinal	recovery	incontinence
urinary	motor	urinary_incontinence
urothelial	reflex	pelvic
cervical	urothelial	ui
injury	injured	prolapse
recovery	functional_recovery	ul
urinary_tract	plasticity	contraceptive
locomotor	locomotor	treatment
lumbar	cervical	stress
reflex	pathways	disorders

TABLE 12.2
Example of a topic being split using interactive topic modeling under the constraint that "bladder" and "spinal_cord_injury" should not be in the same topic. This results in "bladder" now being associated with incontinence.

no documents containing both words. Several of the topic quality metrics described in this chapter measure mismatch between the theoretical co-occurrence implied by a model and actual word co-occurrence observed in documents.

The power that these simple metrics hold raises the question of why such topics should arise in the first place: if they are so easy to detect, why do they appear at all? The answer is that under standard specifications, topic models such as LDA cannot directly represent co-occurrence information. Mimno et al. (2011) presents an alternative model based on generalized Pólya urns (Mahmoud, 2008) that addresses this problem by encoding word co-occurrence information into the prior.

The generative process of a topic model is usually described in terms of discrete variables drawn from multinomial distributions that are themselves drawn from Dirichlet distributions. In this representation, the "meaning" of a topic is defined once and for all when the multinomial parameters for the topic-word distribution are sampled, and does not change no matter how many words are observed. An alternative generative model for LDA, which does not involve these intermediate multinomial parameters, is a standard Pólya urn process. Under this representation, the "meaning" of a topic evolves as words are sampled.

Consider an urn containing N balls, each with a single word written on it, such that N_w balls have word w written on them. If we draw and replace balls repeatedly, recording the word on each sampled ball, the frequency of each word in the resulting set of words is a distributed i.i.d. multinomial with $p(w) \propto N_w$.

If instead of replacing just the sampled ball we also add a new ball with the same word, the resulting set of words is distributed as a Dirichlet-compound multinomial. The DCM distribution is equivalent to a Dirichlet-multinomial hierarchical model with the parameters of the multinomial distribution integrated out. This model, the standard Pólya urn, is not i.i.d.: if we draw a ball with word w at time t, the probability that word w will appear on the next ball at $t + 1$ increases and the probability of all other words decreases. The model is, however, exchangeable, as the probability of a sequence of words is invariant to permutation of their order.

The Pólya urn process provides burstiness (a word, once seen, becomes more probable), but it cannot represent covariance since an increase in one word decreases the probability of all other words. The generalized Pólya urn extends the standard urn model by specifying a separate rule for

adding new balls after sampling a ball of each type. For example, we might say that after sampling a ball with word w_2, we should replace it along with two new balls with word w_2, and one each of w_5, w_8, and w_{15}. In this way, w_2 would increase the probability of seeing w_2 again, but also increase the probability of the three other word types.

All three urn models can be represented by specifying a *schema* matrix $\boldsymbol{A}$, which defines the number of balls of each type to add after drawing a ball of each type. To define the simple sampling-with-replacement model we use a matrix of all zeros, indicating that no new balls will be added. For the standard Pólya urn, we use an identity matrix, which specifies that after seeing a ball of type w we add a single new ball of type w and nothing else. The generalized Pólya urn permits arbitrary values in the matrix (negative values are possible, corresponding to permanently removing balls, but can lead to instability). Mimno et al. (2011) defines a matrix with entries proportional to the co-document matrix used in the previously discussed evaluation metrics.

$$\begin{aligned} \boldsymbol{A}_{vv} &\propto \lambda_v D(v), \\ \boldsymbol{A}_{vw} &\propto \lambda_v D(w, v). \end{aligned} \tag{12.9}$$

As with the standard Pólya urn, the flexibility of the generalized Pólya urn comes at the cost of additional complexity. Specifically, the resulting distribution is no longer exchangeable, as the probability of a sequence depends on the order that words are observed. Nevertheless, the model can be effectively trained using a Gibbs sampler as if the distribution were exchangeable.

12.4.3 Regularized Topic Models

Topic models have the potential to improve search and discovery by extracting useful semantic themes from text documents. When learned topics are coherent and interpretable, they can be valuable for faceted browse, results set diversity, and document retrieval. However, when collections are made up of short documents or noisy text (e.g., web search result snippets or blog posts), learned topics can be less coherent, less interpretable, and less useful.

Predicated on recent evidence that a PMI-based topic coherence score is highly correlated with human-judged topic coherence (Newman et al., 2010a), Newman et al. (2011) proposed two Bayesian regularization formulations to improve topic coherence. Both methods use additional word co-occurrence data to improve the coherence and interpretability of learned topics, while still learning a faithful representation of the collection of interest, as measured by likelihood of test data. These *regularized topic models* are an alternative to the generalized Pólya urn models described in the previous section, and have similar objectives and goals.

To learn more coherent topic models for small or noisy collections, they introduced structured priors on $\boldsymbol{\phi}_t$ based upon external data, which have a regularization effect on the standard LDA model. More specifically, the priors on $\boldsymbol{\phi}_t$ depend on the structural relations of the words in the vocabulary as given by external data, which are characterized by the $W \times W$ "covariance" matrix $\mathbf{C}$. Intuitively, $\mathbf{C}$ is a matrix that captures the short-range dependencies between (i.e., co-occurrences of) words in the external data. One is only interested in relatively frequent terms from the vocabulary, so $\mathbf{C}$ is a sparse matrix and computations are still feasible.

Quadratic Regularizer. A standard quadratic form is used with a trade-off factor. Given a matrix of word dependencies $\mathbf{C}$, use the prior:

$$p(\boldsymbol{\phi}_t|\mathbf{C}) \propto \left(\boldsymbol{\phi}_t^T \mathbf{C} \boldsymbol{\phi}_t\right)^{\nu} \tag{12.10}$$

for some power ν. The normalization factor is unknown but for MAP estimation we do not need it. Optimizing the log posterior with respect to $\phi_{w|t}$ subject to the usual constraints, one obtains the following fixed point update:

$$\phi_{w|t} \leftarrow \frac{1}{N_t + 2\nu}\left(N_{wt} + 2\nu\frac{\phi_{w|t}\sum_{i=1}^{W}\mathrm{C}_{iw}\phi_{i|t}}{\boldsymbol{\phi}_t^T\mathbf{C}\boldsymbol{\phi}_t}\right). \tag{12.11}$$

Unlike other topic models in which a covariance or correlation structure is used in the context of correlated priors for $\theta_{t|d}$, (as in the correlated topic model (Blei and Lafferty, 2005)), this method does not require the inversion of $\mathbf{C}$, which would be impractical for even modest vocabulary sizes. (Interactive topic modeling, discussed in Section 12.4.1, also adds correlations without requiring this inversion because it preserves conjugacy.)

By using the update in Equation (12.11) we obtain the values for $\phi_{w|t}$. This means we no longer have conjugate priors for $\phi_{w|t}$ and thus the standard Gibbs-sample update

$$p(z_{id} = t|x_{id} = w, \mathbf{z}^{\neg i}) \propto \frac{N_{wt}^{\neg i} + \beta}{N_t^{\neg i} + W\beta}(N_{td}^{\neg i} + \alpha) \tag{12.12}$$

does not hold. Instead, at the end of each major Gibbs cycle, $\phi_{w|t}$ is re-estimated and the corresponding Gibbs update becomes:

$$p(z_{id} = t|x_{id} = w, \mathbf{z}^{\neg i}, \phi_{w|t}) \propto \phi_{w|t}(N_{td}^{\neg i} + \alpha)\,. \tag{12.13}$$

Convolved Dirichlet Regularizer. Another approach to leveraging information on word dependencies from external data is to consider that each $\boldsymbol{\phi}_t$ is a mixture of word probabilities $\boldsymbol{\psi}_t$, where the coefficients are constrained by the word-pair dependency matrix $\mathbf{C}$:

$$\boldsymbol{\phi}_t \propto \mathbf{C}\boldsymbol{\psi}_t \qquad \text{where} \qquad \boldsymbol{\psi}_t \sim \text{Dirichlet}(\gamma\mathbf{1}). \tag{12.14}$$

Each topic has a different $\boldsymbol{\psi}_t$ drawn from a Dirichlet, thus the model is a convolved Dirichlet. This means that we convolve the supplied topic to include a spread of related words. Optimizing the posterior and solving for $\psi_{w|t}$ one obtains:

$$\psi_{w|t} \propto \sum_{i=1}^{W}\frac{N_{it}\mathrm{C}_{iw}}{\sum_{j=1}^{W}\mathrm{C}_{ij}\psi_{j|t}}\psi_{w|t} + \gamma. \tag{12.15}$$

One follows the same semi-collapsed inference procedure used for the quadratic regularizer, with the updates in Equations (12.15) and (12.14) producing the values for $\phi_{w|t}$ to be used in the semi-collapsed sampler (12.13).

Using thirteen datasets from blog posts, news articles, and web searches, Newman et al. (2011) shows that both regularizers improve topic coherence and interpretability while learning a faithful representation of the collection of interest. Additionally, in an experiment involving 3,650 crowd-sourced topic comparisons, they show that humans judge the regularized topic models as being more coherent than LDA.

12.4.4 Automatic Topic Labeling

In user-facing applications that use topic models, topics are displayed to humans, typically using the top-10 or so terms in the topic. However, it can sometimes be difficult for end-users to interpret the rich statistical information encoded in the topics, or quickly getting the gist of a topic. One way of making topics more readily human interpretable is by annotating the topic with a short label. While this task is best done by a subject matter expert, recent work has shown that one can partially automate the generation of candidate labels for topics.

Short labels for topics are typically best expressed with multiword terms (for example STOCK

MARKET TRADING), or terms that might not be in the top-10 topic terms (for example, COLORS would be a good label for a topic of the form *red green blue cyan* ...). Lau et al. (2011) proposed a novel method for automatic topic labeling that first generates topic label candidates using English Wikipedia, and then ranks the candidates to select the best topic labels. Given the size and diversity of English Wikipedia, they posit that the vast majority of (coherent) topics or concepts are probably well represented by a Wikipedia article title.

Their method of predicting suitable candidate labels has two parts. They first have a system to generate a relatively long list of candidates. Then, they use lexical features of and association measures between candidate labels and topic terms in a support vector regression framework for ranking the labels.

Generating the list of candidates starts with querying Wikipedia using the top-10 topic terms. The top-ranked search results (article titles) returned constitute the initial set of *primary* candidates for each topic. Next we chunk parse the primary candidates using the OpenNLP chunker and extract out all noun chunks. For each noun chunk, we generate all component n-grams, out of which we remove all n-grams which are not in themselves article titles in English Wikipedia. For example, if the Wikipedia document title were the single noun chunk *United States Constitution*, we would generate the bigrams *United States* and *States Constitution*, and prune the latter; we would also generate the unigrams *United*, *States*, and *Constitution*, all of which exist as Wikipedia articles and are preserved.

Ranking candidate labels is premised on the idea that a good label should be strongly associated with the topic terms. To learn the association of a label candidate with the topic terms, we use several lexical association measures: pointwise mutual information (PMI), Student's t-test, Dice's coefficient, Pearson's χ^2 test, and the log-likelihood ratio. We also include conditional probability and reverse conditional probability measures based on the work of Lau et al. (2010). To calculate the association measures, we parse the full collection of English Wikipedia articles using a sliding window of width 20, and obtain term frequencies for the label candidates and topic terms. To measure the association between a label candidate and a list of topic terms, we average the scores of the top-10 topic terms.

These lexical features and association measures were used in a supervised model by training over topics where we have gold standard labeling of the label candidates using a support vector regression (SVR) model over all of the features. Table 12.3 shows examples of the top-ranked label candidate for four topics learned on four different corpora from diverse genres. We see that the top-ranked label candidate does a relatively good job of capturing the gist of each of the four topics.

china chinese olympics gold olympic team win beijing medal sport ... Label: 2008 SUMMER OLYMPICS
church arch wall building window gothic nave side vault tower ... Label: GOTHIC ARCHITECTURE
israel peace barak israeli minister palestinian agreement prime leader ... Label: ISRAELI-PALESTINIAN CONFLICT
cell response immune lymphocyte antigen cytokine t-cell induce receptor ... Label: IMMUNE SYSTEM

TABLE 12.3
A sample of topics and automatically generated topic labels.

12.5 Conclusion

While topic models are a popular technique for understanding large datasets, how to actually go from raw data to an effective topic analysis is often difficult for new users. This chapter discusses the iterative process for building topic models from preprocessing data to improving and understanding the results users can obtain from models. In time, this process can benefit from continued development by both tool builders and researchers.

However, tool builders will continue to make this process more straightforward by building unified interfaces that can seamlessly adjust tokenization, vocabulary, and topic models within a single interface. Improved visualization tools that can help users identify and correct topic modeling errors would also make the process of curating a topic model more straightwforward.

Researchers can improve the process by building models that are less sensitive to the seemingly arbitrary choices made by users. Models should be less sensitive to the vocabulary, should be able to segment overly long documents, and should detect when the data fail to meet the assumptions of topic models, such as when a corpus is in multiple languages or dialects. Finally, researchers can improve inference throughput and latency so that users can try more models more quickly.

Together, these advances will allow users to move from data to a final, quality model quickly and with minimal hassle.

References

AlSumait, L., Barbará, D., Gentle, J., and Domeniconi, C. (2009). Topic significance ranking of LDA generative models. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part I (ECML PKDD '09)*. Berlin, Heidelberg: Springer-Verlag, 67–82.

Andrzejewski, D., Zhu, X., and Craven, M. (2009). Incorporating domain knowledge into topic modeling via Dirichlet Forest priors. In *Proceedings of the 26th International Conference of Machine Learning (ICML '09)*. New York, NY, USA: ACM, 25–32.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

Blei, D. M., Griffiths, T. L., and Jordan, M. I. (2007). The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM* 57 : 7.1–7.30.

Blei, D. M. and Lafferty, J. D. (2005). Correlated Topic Models. In Weiss, Y., Schölkopf, B., and Platt, J. (eds), *Advances in Neural Information Processing Systems 18*. Cambridge, MA: The MIT Press, 147–154.

Blei, D. M. and Lafferty, J. D. (2006). Dynamic topic models. In *Proceedings of the 23rd International Conference of Machine Learning (ICML '06)*. New York, NY, USA: ACM, 113–120.

Blei, D. M. and Lafferty, J. D. (2009). Visualizing topics with multi-word expressions. http://arxiv.org/abs/0907.1013 [stat.ML].

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research* 3: 993–1022.

Boyd-Graber, J. and Blei, D. M. (2008). Syntactic Topic Models. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds), *Advances in Neural Information Processing Systems 21*. Red Hook, NY: Curran Associates, Inc., 185–192.

Boyd-Graber, J., Blei, D. M., and Zhu, X. (2007). A topic model for word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Stroudsburg, PA, USA: Association of Computational Linguistics, 1024–1033.

Brody, S. and Lapata, M. (2009). Bayesian word sense induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 103–111.

Buntine, W. (2009). Estimating likelihoods for topic models. In *Proceedings of the 1st Asian Conference on Machine Learning: Advances in Machine Learning*. Berlin, Heidelberg: Springer-Verlag, 51–64.

Chang, J., Boyd-Graber, J., and Blei, D. M. (2009a). Connections between the lines: Augmenting social networks with text. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*. New York, NY, USA: ACM, 169–178.

Chang, J., Boyd-Graber, J., Wang, C., Gerrish, S., and Blei, D. M. (2009b). Reading tea leaves: How humans interpret topic models. In Bengio, Y., Schuurmans, D., Lafferty, J. D., Williams, C. K. I., and Culotta, A. (eds), *Advances in Neural Information Processing Systems 22*. Red Hook, NY: Curran Associates, Inc., 288–296.

Dietz, L., Bickel, S., and Scheffer, T. (2007). Unsupervised prediction of citation influences. In *Proceedings of the 24th Annual International Conference on Machine Learning (ICML '07)*. New York, NY, USA: ACM, 233-240.

Eidelman, V., Boyd-Graber, J., and Resnik, P. (2012). Topic models for dynamic translation model adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Vol. 2*. Stroudsburg, PA, USA: Association for Computational Linguistics, 115-119.

Fei-Fei, L. and Perona, P. (2005). A Bayesian hierarchical model for learning natural scene categories. In *Proceedings of the 10th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*. Los Alamitos, CA,USA: IEEE Computer Society, 524–531.

Freedman, A. (2007). *The Party of the First Part: The Curious World of Legalese*. New York, NY: Henry Holt and Company.

Goldwater, S., Griffiths, T. L., and Johnson, M. (2006). Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Comptuational Linguistics, 673–680.

Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences* 101: 5228–5235.

Griffiths, T. L., Steyvers, M., Blei, D. M., and Tenenbaum, J. B. (2005). Integrating topics and syntax. In Saul, L. K., Weiss, Y., and Bottou, L. (eds), *Advances in Neural Information Processing Systems 17*. Cambridge, MA: The MIT Press, 537–544.

Gruber, A., Rosen-Zvi, M., and Weiss, Y. (2007). Hidden topic Markov models. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS 2007). Journal of Machine Learning Research – Proceedings Track* 2: 163–170.

Haghighi, A. and Vanderwende, L. (2009). Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Boulder, Colorado: Association for Computational Linguistics, 362–370.

Hall, D., Jurafsky, D., and Manning, C. D. (2008). Studying the history of ideas using topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 363–371.

Hardisty, E. A., Boyd-Graber, J., and Resnik, P. (2010). Modeling perspective using adaptor grammars. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP '10)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 284–292.

Hu, Y., Boyd-Graber, J., Satinoff, B., and Smith, A. (To Appear). Interactive topic modeling. *Machine Learning Journal*.

Johnson, M. (2010). PCFGs, topic models, adaptor grammars and learning topical collocations and the structure of proper names. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 1148–1157.

Kohlschütter, C., Fankhauser, P., and Nejdl, W. (2010). Boilerplate detection using shallow text features. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM '10)*. New York, NY, USA: ACM, 441–450.

Lau, J. H., Grieser, K., Newman, D., and Baldwin, T. (2011). Automatic Labelling of Topic Models. In *Proceedings of the Association for Computational Linguistics*, 1536–1545.

Lau, J. H., Newman, D., Karimi, S., and Baldwin, T. (2010). Best topic word selection for topic labelling. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Poster presented. Beijing, China, 605–613.

Lin, W. -H., Wilson, T., Wiebe, J., and Hauptmann, A. (2006). Which side are you on? Identifying perspectives at the document and sentence levels. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X '06)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 109–116.

Mahmoud, H. (2008). *Pólya Urn Models*. Chapman & Hall/CRC, 1st edition.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA: The MIT Press.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics* 19: 313–330.

Mimno, D., Wallach, H. M., Talley, E. M., Leenders, M., and McCallum, A. (2011). Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 262–272.

Narayanamurthy, S. (2011). Yahoo! LDA project.

Neal, R. M. (1993). Probabilistic Inference Using Markov Chain Monte Carlo Methods. Tech. report CRG-TR-93-1, University of Toronto.

Newman, D., Baldwin, T., Cavedon, L., Huang, E., Karimi, S., Martinez, D., Scholer, F., and Zobel, J. (2010a). Visualizing search results and document collections using topic maps. *Web Semantics* 8: 169–175.

Newman, D., Bonilla, E., and Buntine, W. (2011). Improving topic coherence with regularized topic models. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P., Pereira, F. C. N., and Weinberger, K. Q. (eds), *Advances in Neural Information Processing Systems 24*. Red Hook, NY: Curran Associates, Inc., 496–504.

Newman, D., Lau, J. H., Grieser, K., and Baldwin, T. (2010b). Automatic evaluation of topic coherence. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 100–108.

Nguyen, V. -A., Boyd-Graber, J., and Resnik, P. (2012). SITS: A hierarchical nonparametric model using speaker identity for topic segmentation in multiparty conversations. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers –Vol. 1*. Stroudsburg, PA, USA: Association for Computational Linguistics, 78–87.

Paul, M. and Girju, R. (2010). A two-dimensional topic-aspect model for discovering multi-faceted topics. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI 2010)*. Palo Alto, CA, USA: AAAI, 545–550.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program: Electronic Library and Information Systems* 14: 130–137.

Purver, M., Körding, K., Griffiths, T. L., and Tenenbaum, J. B. (2006). Unsupervised topic modelling for multi-party spoken discourse. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 17–24.

Ramage, D., Hall, D., Nallapati, R. M., and Manning, C. D. (2009). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Vol. 1*. Stroudsburg, PA, USA: Association for Computational Linguistics, 248–256.

Rosen-Zvi, M., Griffiths, T. L., Steyvers, M., and Smyth, P. (2004). The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*. Arlington, VA, USA: AUAI Press, 487–494.

Salton, G. (1968). *Automatic Information Organization and Retrieval*. McGraw Hill Text.

Sandhaus, E. (2008). The *New York Times* Annotated Corpus. Philadelphia, PA: Linguistic Data Consortium.

Schemann, H. and Knight, P. (1995). *German-English Dictionary of Idioms*. Oxford, UK: Routledge.

Taghva, K., Elkhoury, R., and Coombs, J. (2005). Arabic stemming without a root dictionary. In *Proceedings of the Internaional Conference on Information Technology: Coding and Computing – Vol. 1 (ITCC 2005)*. Los Alamitos, CA, USA: IEEE Computer Society, 152–157.

Talley, E. M., Newman, D., Mimno, D., Herr, B. W., Wallach, H. M., Burns, G. A. P. C., Leenders, M., and McCallum, A. (2011). Database of NIH grants using machine-learned categories and graphical clustering. *Nature Methods* 8: 443–444.

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

Titov, I. and McDonald, R. (2008). A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Stroudsburg, PA, USA: Association for Computational Linguistics, 308–316.

Wallach, H. M., Mimno, D., and McCallum, A. (2009a). Rethinking LDA: Why priors matter. In Bengio, Y., Schuurmans, D., Lafferty, J. D., Williams, C. K. I., and Culotta, A. (eds), *Advances in Neural Information Processing Systems 22*. Red Hook, NY: Curran Associates, Inc., 1973–1981.

Wallach, H. M. (2006). Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference of Machine Learning (ICML '06)*. New York, NY, USA: ACM, 977–984.

Wallach, H. M., Murray, I., Salakhutdinov, R., and Mimno, D. (2009b). Evaluation methods for topic models. In Bottou, L. and Littman, M. (eds), *Proceedings of the 26th Annual International Conference of Machine Learning (ICML '09)*. New York, NY, USA: ACM, 1105–1112.

Wei, X. and Croft, B. (2006). LDA-based document models for ad-hoc retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '06)*. New York, NY, USA: ACM, 178–185.

Yao, L., Mimno, D., and McCallum, A. (2009). Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 937–946.

Zhai, K., Boyd-Graber, J., Asadi, N., and Alkhouja, M. (2012). Mr. LDA: A flexible large scale topic modeling package using variational inference in MapReduce. In *Proceedings of the 21st International Conference on World Wide Web*. New York, NY, USA: ACM, 879–888.