Foundations and Trends® in Information Retrieval
Vol. 11, No. 2-3 (2017) 143–296

DOI: 10.1561/1500000030

now
the essence of knowledge

Applications of Topic Models

Jordan Boyd-Graber
Department of Computer Science, UMIACS, Language Science
University of Maryland[1]
jbg@umiacs.umd.edu

Yuening Hu
Google, Inc.[2]
ynhu@google.com

David Mimno
Information Science
Cornell University
mimno@cornell.edu

[1]Work completed while at University of Colorado
[2]Work completed while at Yahoo!

Contents

Abstract

How can a single person understand what's going on in a collection of millions of documents? This is an increasingly common problem: sifting through an organization's e-mails, understanding a decade worth of newspapers, or characterizing a scientific field's research. Topic models are a statistical framework that help users understand large document collections: not just to find individual documents but to understand the general themes present in the collection.

This survey describes the recent academic and industrial applications of topic models with the goal of launching a young researcher capable of building their own applications of topic models. In addition to topic models' effective application to traditional problems like information retrieval, visualization, statistical inference, multilingual modeling, and linguistic understanding, this survey also reviews topic models' ability to unlock large text collections for qualitative analysis. We review their successful use by researchers to help understand fiction, non-fiction, scientific publications, and political texts.

J. Boyd-Graber, Y. Hu and D. Mimno. *Applications of Topic Models*. Foundations and Trends® in Information Retrieval, vol. 11, no. 2-3, pp. 143–296, 2017.

DOI: 10.1561/1500000030.

1

The What and Wherefore of Topic Models

Imagine that you are an intrepid reporter with an amazing scoop: you have twenty-four hours of exclusive access three decades of e-mails sent within a corrupt corporation. You know there's dirt and scandal there, but it has been well-concealed by the corporation's political friends. How are you going to understand this haystack well enough to explain it to your devoted readers under such a tight deadline?

1.1 Tell Me about Your Haystack

Unlike the vignette above, interacting with large text data sets is often posed as a needle in a haystack problem. The poor user—faced with documents that would take a decade to read—is looking for a single needle: a document (or at most a handful of documents) that matches what the user is looking for: a "smoking gun" e-mail, the document that best represents a concept [Salton, 1968] or the answer to a question [Hirschman and Gaizauskas, 2001].

These questions are important. The discipline of information retrieval is built upon systematizing, solving, and evaluating this problem. Google's search service is built on the premise of users typing a few

keywords into a search engine box and seeing quick, consistent search results. However, this is not the only problem that confronts those interacting with large text datasets.

A different, but related problem is *understanding* large document collections, common in science policy [Talley et al., 2011], journalism, and the humanities [Moretti, 2013a]. The haystack has more than one precious needle. At the risk of abusing the metaphor, *sometimes you care about the straw*. Instead of looking for a smoking gun alerting to you some crime that was committed, perhaps you are looking for a sin of omission: did this company never talk about diversity in its workforce? Instead of a single answer to a question, perhaps you are looking for a diversity of responses: what are the different ways that people account for rising income inequality? Instead of looking for one document, perhaps you want to provide population level statistics: what proportion of Twitter users have ever talked about gun violence?

At first, it might seem that answering these questions would require building an extensive ontology or categorization scheme. For every new corpus, you would need to define the buckets that a document could fit into, politely ask some librarians and archivists to put each document into the correct buckets, perhaps automate the process with some supervised machine learning, and then collect summary statistics when you are done.

Obviously, such laborious processes are possible—they have been done for labeling congressional speeches[1] and understanding emotional state [Wilson and Wiebe, 2005]—and remain an important part of social science, information science, library science, and machine learning. But these processes are not always possible, fast, or even the optimal outcome if we had infinite resources. First, they require a significant investment of time and resources. Even creating the *list* of categories is a difficult task and requires careful deliberation and calibration. Even if it were possible, a particular question might not warrant the time or effort: the œuvre of a minor author (only of interest to a few), or the tweets of a day (not relevant tomorrow).

[1] `www.congressionalbills.org/`

Table 1.1: Five topics from a twenty-five topic model fit on Enron e-mails. Example topics concern financial transactions, natural gas, the California utilities, federal regulation, and planning meetings. We provide the five most probable words from each topic (each topic is a distribution over all words).

Topic	Terms
3	trading financial trade product price
6	gas capacity deal pipeline contract
9	state california davis power utilities
14	ferc issue order party case
22	group meeting team process plan

This survey explores the ways that humans and computers make sense of document collections through tools called topic models. Topic models allow us to answer big-picture questions quickly, cheaply, and without human intervention. Once trained, they provide a framework for humans to understand document collections both directly by "reading" models or indirectly by using topics as input variables for further analysis. For readers already comfortable with topic models, feel free to skip this chapter; we will mostly cover the definitions and implementations of topic models.

The intended audience of this book is a reader with some knowledge of document processing (e.g., knows what "tokens" and "documents" are), basic understanding of some probability (e.g., what a distribution is), and interested in many application domains. We discuss the information needs of each application area, and how those specific needs affect models, curation procedures, and interpretations.

By the end of the book (Chapter 9), we hope that readers will be excited enough to attempt to embark on building their own topic models. In this chapter, we go deeper into more of the implementation details. Readers who are already topic model experts will likely not learn much technically, but we hope our coverage of diverse applications will expose a topic modeling expert to models and approaches they had not seen before.

Yesterday, SDG&E filed a motion for adoption of an electric procurement cost recovery mechanism and for an order shortening time for parties to file comments on the mechanism. The attached email from SDG&E contains the motion, an executive summary, and a detailed summary of their proposals and recommendations governing procurement of the net short energy requirements for SDG&E's customers. The utility requests a 15-day comment period, which means comments would have to be filed by September 10 (September 8 is a Saturday). Reply comments would be filed 10 days later.

Topic	Probability
9	0.42
11	0.05
8	0.05

Figure 1.1: Example document from the Enron corpus and its association to topics. Although it does not contain the word "California", it discusses a single California utility's dissatisfaction with how much it is paying for electricity.

1.2 What is a Topic Model

Returning to our motivating example, consider the e-mails from Enron, the prototypical troubled corporation of the turn of the century. A source has provided you with a trove of emails, and your editor is demanding an article by yesterday. You know that wrongdoing happened, but you do not know who did it or how it was planned and carried out. You have suspicions (e.g., around the California energy spot market), but you are curious about other skeletons in the closet and you are highly motivated to find them.

So you run a topic model on the data. True to its name, a topic model gives you "topics", each of which is a ranking of all the distinct words in the e-mails by relevance to a topic. Taking the top five most relevant words in each topic results in collections of words that make sense together (Table 1.1). For example, one topic seems to have words relating to finance and trading. Another seems to involve to gas pipelines, their capacity, and deals or contracts relating to those pipelines. This all makes sense: Enron was an energy trading company. Others seem to involve language used in any business, such as meetings and plans.

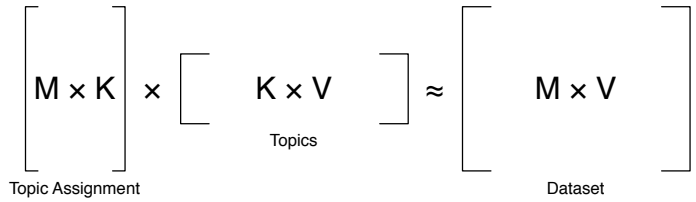

Figure 1.2: A matrix formulation of finding K topics for a dataset with M documents and V unique words. While this view of topic modeling includes approaches such as latent semantic analysis (LSA, where the approximation is based on SVD), we focus on probabilistic techniques in the rest of this survey.

The first half of a topic model connects topics to a jumbled "bag of words". When we say that a topic is about X, we are manually assigning a *post hoc* label (more on this in Chapter 3.1). It remains the responsibility of the human consumer of topic models to go further and make sense of these piles of straw (we discuss labeling the topics more in Chapter 3).

Making sense of one of these word piles by itself can be difficult. The second half of a topic model links topics to individual documents. For example, the document in Figure 1.1 is about a California utility's reaction to the short-term electricity market and exemplifies Topic 9 from Table 1.1. Considering examples of documents that are strongly connected to a topic, along with the words associated with the topic, can give us a more complete representation of the topic. If we get a sense that Topic 9 is of interest, we can explore deeper to find other documents.

1.3 Foundations

You might notice that we are using the general term "topic model". There are many mathematical formulations of topic models and many algorithms that learn the parameters of those models from data. Although we will focus on particular models and algorithms, we choose

our terminology to emphasize that the similarities between formulations, models, and algorithms are often greater than their differences.

Topic modeling began with a linear algebra approach [Deerwester et al., 1990] called latent semantic analysis (LSA): find the best low rank approximation of a document-term matrix (Figure 1.2). While these approaches have seen a resurgence in recent years [Anandkumar et al., 2012, Arora et al., 2013], we focus on probabilistic approaches [Hofmann, 1999a, Papadimitriou et al., 2000, Blei et al., 2003], which are intuitive, work well, and allow for easy extensions (as we see later in many of our later chapters).

The two foundational probabilistic topic models are latent Dirichlet allocation [Blei et al., 2003, LDA] and probabilistic latent semantic analysis [Hofmann, 1999a, pLSA]. We describe the former in significant detail in Chapter 1.4, but we want to take a moment to address some of the historical connection between these two models.

pLSA was historically first and laid the foundation for LDA. pLSA was used extensively in many applications such as information retrieval. However, this survey focuses on LDA because more researchers have not just *used* LDA—they have also *extended* it. LDA is not just widely used, but it is also widely modified. Because of these prolific modifcations, we focus on the mechanics of LDA, which many researchers have used as the foundations of new models. However, as we explain below (Chapter 1.5.4), the similarities between pLSA and LDA outweigh the differences.

In any technical field it is common for general terms to take on specific, concrete meanings, and this can be a source of confusion. In topic modeling the word "topic" takes on the specific meaning of a probability distribution over words, while still alluding the to more general meaning of a theme or subject of discourse. Because other areas of information retrieval have similarly developed specific meanings for the word "topic", we distinguish them here. The most common definition is a specific information need, as in the TREC evaluation corpora developed by NIST [Voorhees and Harman, 2005]. TREC topics are generally much more specific than topic model topics, and may relate to particular aspects or perspectives on a subject. An example from

the 2003 TREC Robust Track is "Identify positive accomplishments of the Hubble telescope since it was launched in 1991" [Voorhees, 2003]. Similarly to information retrieval, the related field of topic detection and tracking also has a specific technical definition of "topic" [Allan, 2002]. In TDT, a "topic" is usually closer to an event or an individual story. In contrast, topic models tend to identify more abstract latent factors. For example, a TDT topic might include an earthquake in Haiti, whereas a topic model might represent the same event as a combination of topics such as Haiti, natural disasters, and international aid.

There has been some work on using topic models to detect emerging events by searching for changes in topic probability [AlSumait et al., 2008]. But these methods tend to identify mainly the fact that an event has occurred, without necessarily identifying the specific features of that event. Other work has found that more lexically specific methods than topic models are best for identifying memes and viral phrases [Leskovec et al., 2009].

1.3.1 Probabilistic Building Blocks

In probabilistic models we want to find values for unobserved model variables that do a good job of explaining the observed data. The first step in inference is to turn this process around, and assert a way to generate data given model variables. Probabilistic models thus begin with a generative story: a recipe listing a sequence of random events that creates the dataset we are trying to explain. Figure 1.3 lists some of the key players in these stories, how they are parameterized and what samples drawn from these distributions look like. We will briefly discuss them, as we will use them to build a wide variety of topic models later.

Gaussian If you know any probability distribution already, it is (probably) the Gaussian. This distribution does not have a role in the most basic topic models that we will discuss here, but it will later (e.g., Chapter 7). We include it because it is a useful point of comparison against the other distributions we *are* using (since it is perhaps the easiest to understand and best known). A Gaussian is a distribution over all real numbers (e.g., $0.0, 0.5, -4.2, \pi, \ldots$). You can ask it to spit

Distribution	Density	Example Parameters	Example Draws
Gaussian	$\frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	$\mu = 2, \sigma^2 = 1.1$	$x = 2.21$
Discrete	$\prod_i \phi_i^{\mathbb{1}[w=i]}$	$\phi = \begin{bmatrix} 0.1 \\ 0.6 \\ 0.3 \end{bmatrix}$	$w = 2$
Dirichlet	$\frac{\prod_{i=1}^K \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^K \alpha_i\right)} \prod_{i=1}^K \theta_i^{\alpha_i - 1}$	$\alpha = \begin{bmatrix} 1.1 \\ 0.1 \\ 0.1 \end{bmatrix}$	$\theta = \begin{bmatrix} 0.8 \\ 0.15 \\ 0.05 \end{bmatrix}$

Figure 1.3: Examples of probability distributions used in the generative stories of topic models. In the case of the discrete draw, $w = 2$ denotes that the second element (the one with probability 0.6) was drawn.

out a number, and it will give you some real number between negative infinity and positive infinity. But not all numbers have equal probability. Gaussian distributions are parameterized by a mean μ and variance σ^2. Most samples from the distribution will be near the mean μ; how close is determined by the variance: higher variances will cause the samples to be more spread out.

Discrete While Gaussian distributions are over a continuous space, documents are combinations of discrete symbols, usually word tokens.[2] Thus, we need a distribution over discrete sets.

A useful metaphor for thinking about discrete distributions is a weighted die. The number of faces on the die is its dimension, and each face is associated with a distinct outcome. Each face has its own probability of how likely that outcome is; these probabilities are the parameters of a discrete distribution (Figure 1.3).

Topic models are described by discrete distributions (sometimes called multinomial distributions) that describe the connection between words and topics (the first half) and topics and documents (the second half). A distribution over words is called a topic distribution; each of

[2] An emerging trend in natural language processing research is to view words as embedded in a continuous space. We discuss these "representation learning" approaches and their connection to topic modeling in Chapter 10, but even then models are still defined over a discrete set of words.

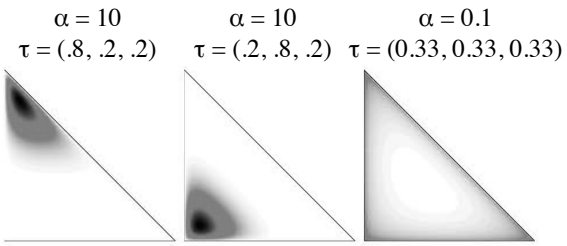

Figure 1.4: Given different Dirichlet parameters, the Dirichlet distribution can either be informative (left, middle) or sparse (right). Sparse distributions encourage distributions to favor a few elements but do not care which ones. This is consistent with our intuitions of how documents are written: they are only about a few things, and topics contain only a handful of words.

the topics gives higher weights to some words more than others (e.g., in Topic 9 from the Enron corpus, "state" and "california" have higher probability than other words). Each document also has an "allocation" for each topic: documents are about a small handful of topics, and most documents have very low weights for most of the possible topics.

Dirichlet Although discrete distributions are the star players in topic models, they are not the end of the story. We often begin with Dirichlet distributions. Just as Gaussians produce real numbers and discrete distributions produce symbols from a finite set, Dirichlet distributions produce probability vectors that can be used as the parameters of discrete distributions. Like the Gaussian distribution, they have parameters analogous to a mean and variance. The mean is called the "base measure" τ and is the expected value of the Dirichlet distribution: the values you would get if you averaged many draws from the Dirichlet. The concentration parameter α_0 controls how far away individual draws

are from the base measure. We often combine these parameters into a single value for each dimension: $\alpha_k = \alpha_0 \tau_k$.

If α_0 is very large, then the draws from a Dirichlet will be very close to τ (Figure 1.4, left). If α_0 is small, however, the discrete distributions become sparse (Figure 1.4, right). A sparse distribution is a distribution where only a few values have high probability and all other values are small.

Because topic models are meant to reflect the properties of real documents, modeling sparsity is important. When a person sits down to write a document, they only write about a handful of the topics that they could potentially use. They do not write about every possible topic, and the sparsity of Dirichlet distributions is the probabilistic tool that encodes this intuition.

There are several important special cases of the Dirichlet distribution. If the base measure τ is the same for every dimension, we call the resulting distribution *symmetric*. This case is appropriate when we do not expect any one element to be, on average, more likely than any other element across all samples from the distribution. In the symmetric case the distribution has only one parameter, the concentration α_0. If the base measure is uniform and the concentration parameter α_0 is equal to the number of dimensions K (or, equivalently, $\alpha_k = 1.0$ for all k), the distribution is uniform, placing equal probability on all K-dimensional probability distributions.

1.4 Latent Dirichlet Allocation

We now have all the tools we need to tell the complete story of the most popular topic model: latent Dirichlet allocation [Blei et al., 2003, LDA]. Latent Dirichlet allocation[3] posits a "generative process" about how the data came to be. We assemble the probabilistic pieces to tell this

[3]The name LDA is a play on LSA, its non-probabilistic forerunner (latent semantic analysis). Latent because we use probabilistic inference to infer missing probabilistic pieces of the generative story. Dirichlet because of the Dirichlet parameters encoding sparsity. Allocation because the Dirichlet distribution encodes the prior for each document's allocation over topics.

story about generating topics and how those topics are used to create diverse documents.

Generating Topics The first part of the story is to create the topics. The user specifies that there are K distinct topics. Each of the K topics is drawn from a Dirichlet distribution with a uniform base distribution and concentration parameter λ: $\phi_k \sim \text{Dir}(\lambda \boldsymbol{u})$. The discrete distribution ϕ_k has a weight for *every* word in the vocabulary.

However, when we summarize topics (as in Figure 1.1), we typically only use the top (most probable) words of a topic. The lower probability words are less relevant to the topic and thus are not shown.

Document Allocations Document allocations are distributions over topics for each document. This encodes what a document is about; the sparsity of the Dirichlet distribution's concentration parameter α_0 ensures that the document will only be about a few topics. Each document has a discrete distribution over topic: $\theta_d \sim \text{Dir}(\alpha \boldsymbol{u})$.

Words in Context Now that we know what each document is about, we create the words that appear in the document. We assume[4] that there are N_d words in document d. For each word n in the document d, we first choose a **topic assignment** $z_{d,n} \sim \text{Discrete}(\theta_d)$. This is one of the K topics that tells us which topic the word token is from, but not what the word is.

To select which word we will see in the document, we draw from a discrete distribution again. Given a word token's topic assignment $z_{d,n}$, we draw from that topic to select the word: $w_{d,n} \sim \phi_{z_{d,n}}$. The topic assignment tells you what the word is about, and then this selects which distribution over words we use to generate the word.

For example, consider the document in Figure 1.1. To generate it, we choose a distribution over all of the topics. This is θ. For this document, the distribution favors Topic 9 about California. The value for this topic

[4] We can model this in the generative story as well, e.g., with a Poisson distribution. However, we often do not care about document *lengths*—only what the document is about—so we can usually ignore this part of the story.

is higher than any other topic. For each word in the document, the generative process chooses a topic assignment z_n. For this document, any topic is theoretically possible, but we expect that most of those will be Topic 9.

Then, for each token in the document, we need to choose which word type will appear. This comes from Topic 9's distribution over words (multiple topics have word distributions shown in Figure 1.1). Each is a discrete draw from the topic's word distribution, which makes words like "California", "state", and "Sacramento" more likely.

It goes without saying that the generative story is a fiction [Box and Draper, 1987]. Nobody is sitting down with dice to decide what to type in on their keyboard. We use this story because it is *useful*. This fanciful story about randomly choosing a topic for each word can help us because if we assume this generative process, we can work backwards to find the topics that explain how a document collection was created: every word, every document, gets associated with these underlying topics.

This simple model helps us order our document collection: by assuming this story, we can discover *topics* (which certainly do not exist) so we can understand the common themes that people use to write documents. As we will see in later chapters, slight tweaks of this generative story allow us to uncover more complicated structures: how authors prefer specific topics, how topics change, or how topics can be used across languages.

1.5 Inference

Given a generative model and some data, the process of uncovering the hidden pieces of the probabilistic generative story is called *inference*. More concretely, it is a recipe for generating algorithms to go from data to *topics that explain a dataset*.

There are many flavors of algorithms for posterior inference: message passing [Zeng et al., 2013], variational inference [Blei et al., 2003], gradient descent [Hoffman et al., 2010], and Gibbs sampling [Griffiths and Steyvers, 2004]. All of these algorithms have their advocates and

reasons you should use them. In this survey, we focus on Gibbs sampling, which is simple, intuitive, and—with some clever tricks specific to topic models—fast [Yao et al., 2009]. (We discuss variational inference in Chapter 9.)

We present the results of Gibbs sampling without derivation, which—along with the history of its origin in statistical physics—are well described elsewhere.[5] We use a variety of Gibbs sampling called *collapsed* Gibbs sampling, which allows inference to side-step some of the pieces of the generative story: instead of explicitly representing the parameters of a discrete distribution, distinct from any observations drawn from that distribution, we represent the distribution solely through those observations. We can then recreate the topic and document distributions through simple formulas.

1.5.1 Random Variables

Topic Assignments Since every individual token is assumed to be generated from a single topic, we can consider the *topic assignment* of a token as a variable. For example, an instance of the word "compilation" might be in a Computer topic in one document and in an Arts topic in another document. Because each token has its own topic assignment, the same word might be assigned to *different* topics in the *same* document. To estimate *global* properties of the topic model we use aggregate statistics derived from token-level topic assignments.

Document Allocation The document allocation is a distribution over the topics for each document; in other words, it says how popular each topic is in a document. If we count up how often a document uses a topic, this gives us its popularity. We define $N_{d,i}$ as the number of times document d uses topic i. This is larger for more popular topics; however, it is not a probability because it is larger than one. We make it a probability by dividing by the number of words in a document

$$\frac{N_{d,i}}{\sum_k N_{d,k}}, \tag{1.1}$$

[5] We recommend Resnik and Hardisty [2009] for additional information on derivation.

but this is problematic because it can sometimes give us zero and ignores the influence of the Dirichlet distribution; a better estimate is[6]

$$\theta_{d,i} \approx \frac{N_{d,i} + \alpha_i}{\sum_k N_{d,k} + \alpha_k}. \tag{1.2}$$

This must never become zero because we do not want it to rule out the possibility that a topic is used in a particular document (hence, each α must be non-zero). This helps the sampler explore more of the possible combinations.

Topics Each topic is a distribution over words. To understand what a topic is about, we look at the profile of all of the tokens that have been assigned to that topic. We estimate the probability of a word in a topic as

$$\phi_{i,v} \approx \frac{V_{i,v} + \beta_v}{\sum_w V_{i,w} + \beta_w}, \tag{1.3}$$

where β is the Dirichlet parameter for the topic distribution.

1.5.2 Algorithm

The collapsed Gibbs sampling algorithm for learning a topic model is only based on the topic assignments, but we will use our estimates for the topics ϕ_k and the documents θ_d discussed above. We begin by setting topic assignments randomly: if we have K topics, each word has equal chance to be associated with any of the topics. These topics will be quite bad, looking like noisy copies of the overall corpus distribution. But we will improve them one word at a time.

The algorithm proceeds by sweeping over all word tokens in turn over and over. At each iteration we change the topic assignments for each word in a way the reflects the underlying probabilistic model of the data. On average, each pass over the data makes the topics slightly better until the model reaches a steady state. There is no easy way to tell when such a steady state has been reached, but eventually the topics will "converge" to reasonable themes and you can consider yourself done.

[6] To be technical, Equation 1.1 is a maximum likelihood estimate and Equation 1.2 is the maximum *a posteriori*, which incorporates the influence of both the prior and the data.

The equation for the probability of assigning a word to a particular topic combines information about words and about documents[7]

$$p(z_{d,n} = i \mid \dots) = \theta_d \phi_j i = \left(\frac{N_{d,i} + \alpha_i}{\sum_k N_{d,k} + \alpha_k} \right) \left(\frac{V_{i,w_{d,n}} + \beta_v}{\sum_w V_{i,w} + \beta_w} \right). \quad (1.4)$$

Computing this value for each topic will result in a probability distribution over the topic assignment for this word token, given all the other topic assignments. The next step is to randomly choose one of those indices with probability proportional to the vector value. You now assign that word to the topic, update $N_{d,\cdot}$ and $V_{\cdot,w_{d,n}}$, and move on to the next word and repeat. The two terms provide two "pressures", for global and local coherence. Sparsity in the topic-word distributions encourages tokens of the same word type to be assigned to a small number of topics, regardless of where they occur. Sparsity in the document-topic distributions encourages tokens in the same document to be assigned to a small number of topics, regardless of what type they are. For example, knowing that a word is "compilation" narrows down the number of potential topics considerably, but leaves ambiguity: is it *program* compilation or a *music* compilation? Knowing that the word occurs in a document with many other words in the Arts topic resolves this ambiguity, leaving the Arts topic as the most probable assignment.

At the very end of the algorithm, we can use the estimates of each topic (Equation 1.3) to summarize the main themes of the corpus and the estimates of each document's topic distribution (Equation 1.2) to start exploring the collection automatically (Chapter 2) or with a human in the loop (Chapter 3).

The algorithm that we have sketched here is the foundation of many of the more advanced models that we will discuss later in the survey. While we will not describe the algorithms in detail, we will occasionally reference this sketch to highlight challenges or difficulties in implementing topic models.

[7] To be theoretically correct, it is important not to include the count associated with the token you are sampling in these counts, which becomes more clear if the probability is written as $p(z_{d,n} = j \mid z_{d,1} \dots z_{d,n-1}, z_{d,n+1} \dots z_{d,N_d}, w_{d,n})$ to show the dependence on the topic assignments of *all other* tokens but not this token.

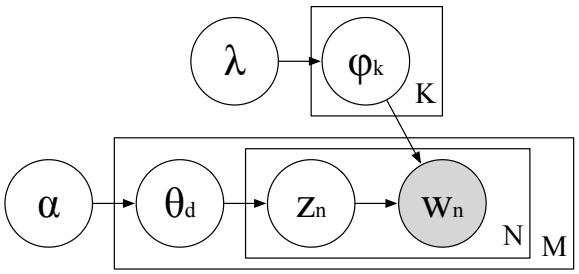

Figure 1.5: Plate diagram for LDA. Nodes show random variables, lines show (possible) probabilistic dependence, rectangles show repetition, and shading shows observation.

1.5.3 Plate Diagrams

Plate diagrams provide a shorthand for quickly explaining which random variables are associated with each other. If you look up many of the references used in this survey, you will likely see plate diagrams (we also use a plate diagram later in Figure 2.1b).

Let's begin with a plate diagram for LDA (Figure 1.5). You can compare these to the generative story in Chapter 1.4. All of the random variables are there, each in its own circle. The lines between random variables tell more of the story. You can see that if a random variable is conditioned on another, there is a line going from the variable that is *conditioned on* **to** the variable that is *conditionally dependent*. For example, a word depends on the token assignment $z_{d,n}$ and a topic ϕ_k, so we draw lines from both.

You can think about the rectangular boxes as repetition. The letter in the bottom right of the box shows how often what is inside the box is replicated. There is a box for each document (there are M in total) and each token (the box of words is inside the box for documents).

When a variable is shaded, this means that it is observed. These are the data we start with. The unshaded variables must either be inferred (e.g., topics ϕ) or are hyperparameters that must be set or inferred (e.g., Dirichlet parameter α).

Plate diagrams allow a reader to quickly see a "family resemblance" between related models, and once someone has become fully immersed in topic models, it is often possible to at a glance understand a model from its plate diagram. However, plate diagrams are imperfect; they lack some of the key information you need to understand the model. For instance, the exact probabilistic relationship between variables is underspecified.

1.5.4 What is so Great about Dirichlet?

Now that we have described what LDA is, we can return to its history. What is the innovation that separates LDA from pLSA, its predecessor? Naïvely, the difference is changing an "s" to a "d" (i.e., changing pL**S**A to L**D**A). The deeper story is about as consequential.

Instead of having a Dirichlet prior over θ, pLSA assumes that θ is a discrete parameter. In practice, this means that documents are not encouraged to focus on a limited number of topics and often "spread out" to have small weights for many different topics. In theory, this means that there is not as sound a generative story for how a document came to be: you cannot run the generative process forward from scratch if you must have θ as a parameter to start with.

These differences are relatively minor. LDA has slightly easier inference—particularly when it comes to tweaking the model—which has caused it to become the more popular of the two models. Thus, we will focus on comparing models to LDA. This is not to diminish from pLSA and its unquestionable place in the literature, but it helps us present a more unified narrative for our reader.

1.5.5 Implementations

Hopefully the previous algorithm sketch has convinced you that implementing topic models is not a Herculean task; most skilled programmers can complete a reasonable implementation of topic models in less than a day. However, we would suggest not trying to implement basic LDA if you just want the output of a topic model, many solid implementations can help users get to useful results more quickly, particularly as topic models often require extensive preprocessing.

Mallet is fast and is a widely used implementation in Java [McCallum, 2002]. This is where you should probably start, in our biased opinion. It runs in Java, uses highly-optimized Gibbs sampling implementations, and can work from a variety of text inputs. It is well documented, mature, and runs well on a multi-core machine, allowing it to process up to millions of documents. Variational inference is the other major option [Blei et al., 2003, Langford et al., 2007], but often requires a little more effort for new users to get a first result.

However, not all users are comfortable with Java; many implementations are available on other platforms and in many programming languages.[8] Many of these implementations are well-built, but check whether they have all of the features of mature implementations like Mallet so that you know what (if anything) you're missing.

However, if your corpus is truly large, consider techniques that can be parallelized over large computer clusters. These techniques can be based on variational inference [Narayanamurthy, 2011, Zhai et al., 2012] or on sampling [Newman et al., 2008].

While these implementations allow you to run *specific* topic models, other frameworks allow you to specify arbitrary generative models. This enables quick prototyping of topic models and integrating topic models with other probabilistic frameworks like regression or collaborative filtering. Examples of these general frameworks include Stan [Stan Development Team, 2014], Theano [Theano Development Team, 2016], and Infer.net [Minka et al., 2014].

If you cannot find the specific model that you want among these existing software packages, the flexibility and simplicity of topic models and inference makes it relatively simple to adapt topic models to model specific phenomena (as we describe in following chapters).

1.6 The Rest of this Survey

In each of the following chapters, we focus on an application of topic models, gradually increasing the complexity of the underlying models.

[8] So many that change so quickly; thus, we are reluctant endorse specific ones here.

The chapters do occasionally refer to each other, but a reader should be able to read each of the chapters independently.

The next chapter returns to the distinction between high level overviews and finding a needle in a haystack. We show how a high level overview can help users and algorithms find documents of interest. We show how a high level overview can help algorithms (Chapter 2) and users (Chapter 3) find documents of interest.

These tools help enable new applications of topic models: how understanding newspapers (Chapter 4) reveals the march of history, how the corpus of writers of fiction (Chapter 6) illuminates societal norms, how the writings of science reveal innovation (Chapter 5), or how politicians' speeches (Chapter 7) reveal schisms in political organizations.

Finally, the survey closes with thoughts about how interested researchers can start building their own topic models (Chapter 9) and how topic models may change in the future (Chapter 10).

2

Ad-hoc Information Retrieval

Topic models explore and summarize document collections outside the context of any specific information need, when we do not necessarily know what we are looking for. This approach to information retrieval stands in contrast to traditional IR systems, which retrieve relevant documents given users' explicit information needs. Where IR systems might look for the "needle in the haystack", topic models will tell you about the overall proportion of hay and needles, and perhaps inform you about the mice that you did not know were there. But topic models can also be useful in situations when we do have a specific information need, but we do not quite know how to search for it. Despite their differences in purpose, there are strong mathematical and conceptual connections between these two approaches. In this chapter we consider the use of topic modeling in IR to balance specific user queries with more open-ended discovery.

In the most direct sense, topic models can be used as a simple indexing method. Users can find topics that assign high probability to a particular query term, and then find documents with a high probability of these topics. Such topic-based search may additionally provide some level of query disambiguation, since it may be clear from topic-word

distributions that one or another topic is more relevant to the user's information need. More sophisticated approaches blur the boundary between query-driven retrieval and unsupervised topic modeling. Erlin [2017] searches for passages related to epistemology in English and German books by "seeding" topic models with words thought to be relevant to that subject. This approach can be successful, but does not guarantee that relevant topics will be found, or that topics will match the intended subject.

In the more formal *ad-hoc* retrieval setting, users start with an information need expressed in queries. Many IR systems treat both the queries and documents as "bags of words", and retrieve and rank the documents by measuring the word overlap between queries and documents. However, the ability of this direct and simple matching is always limited. Words with similar meaning or in different forms should also be considered as matched instead of being ignored. **Language modeling** has been one of the most popular frameworks to capture such semantic relationships. But humans would also like to use background knowledge to interpret and understand the queries and "add" missing words [Wei, 2007], which provides another approach called **query expansion** to improve retrieval and ranking results.

Both directions can be pursued by learning and discovering the semantic relations between words and, further, the semantic relations between queries and documents. Topic models provide semantic relations between query words and documents [Deerwester et al., 1990, Hofmann, 1999b] by describing each topic using probabilistically-weighted words and modeling each document as a distribution over all topics. This adds a layer of abstraction between a document and the exact words present in that document.

Appealing to the generative "story" of a model, we want to recover the words that *could* have been available to an author based on the words that were chosen. Such semantic relations can be applied to smoothing the language models, or introducing related words in query expansion. This chapter focuses on how to apply topic models in document language modeling [Lu et al., 2011, Wei and Croft, 2006] and query

expansion [Park and Ramamohanarao, 2009, Andrzejewski and Buttler, 2011] to further improve ranking results of information retrieval.

2.1 Document Language Modeling

The language modeling approach [Ponte and Croft, 1998, Song and Croft, 1999, Croft and Lafferty, 2003] is one of the main frameworks for using topic models in IR systems, since it is an effective probabilistic framework for studying information retrieval problems [Ponte and Croft, 1998, Berger and Lafferty, 1999]. A statistical language model estimates the probability of word sequences, denoted as $p(w_1, w_2, \cdots, w_n)$. In practice, the statistical language model is often approximated by n-gram models. A unigram model assumes each word in the sequence is independent,

$$p(w_1, w_2, \cdots, w_n) = p(w_1)p(w_2)\cdots p(w_n) \tag{2.1}$$

A trigram model assumes the probability of the current word only depends on the previous two words, and it is represented as

$$p(w_1, \cdots, w_n) = p(w_1)p(w_2|w_1)p(w_3|w_1, w_2)\cdots p(w_n|w_{n-2}, w_{n-1}). \tag{2.2}$$

Language models are used in information retrieval to estimate the similarity of documents and queries [Zhai and Lafferty, 2001a]. Each document is treated as a sample from a particular language model, which we estimate based on the terms present in the document. Given a language model trained from a document, we can then calculate the probability of any other sequence of words under that model. For a unigram language model, this probability is just the product of the individual term probabilities (or, equivalently, the sum of their log probabilities). We can therefore easily find the score for a given user query under every document's language model, and rank the documents by that score.

Given a sample document d, the simplest way to estimate the associated language model is the maximum likelihood principle. The resulting distribution is the one that places the largest possible probability on

the observed document. The probability of generating a word w is the proportion

$$p_{\texttt{ml}}(w \mid d) = \frac{n_{d,w}}{n_{d,\cdot}} \tag{2.3}$$

where $n_{d,w}$ is the term frequency of word w in document d, and $n_{d,\cdot}$ is the total number of tokens in document d. Then the probability of generating the given query q is

$$p(q \mid d) = \prod_{w \in q} p(w \mid d) = \prod_{w \in q} \frac{n_{d,w}}{n_{d,\cdot}}. \tag{2.4}$$

Then the documents are ranked based on this probability $p(q \mid d)$. Higher probability implies the corresponding document is more relevant to the given query [Song and Croft, 1999]. However, a document often contains limited number of words and maximum likelihood estimation gives zero probability to those unseen words. If a query contains any word not in the document, the probability of generating the whole query given this document is zero, which may throw out perfectly good documents.

This data sparsity problem can be fixed by smoothing, which allocates some non-zero probability to the missing terms. Another solution—which also provides other benefits—is topic models. They provide a unique way to extract the word probabilities given the corpus, which can be used to smooth document language models. We summarize two simple smoothing methods, and then show how topic models fit into this smoothing framework.

There are two major directions for smoothing: **interpolation** [Jelinek and Mercer, 1980, Mackay and Peto, 1995, Ney et al., 1994, Ponte and Croft, 1998, Zhai and Lafferty, 2001a] and **backoff** [Katz, 1987, Song and Croft, 1999]. The interpolation-based method discounts the counts of the seen words and distribute the extra counts to both seen words and unseen words. An alternative backoff smoothing strategy trusts the maximum likelihood estimation for high count words, discounts and redistributes mass only for the less common words [Zhai and Lafferty, 2001a].

Here we review two popular and simple interpolation smoothing methods, which are further extended with topic models to smooth document language models.

Jelinek-Mercer The Jelinek-Mercer method [Jelinek and Mercer, 1980] is a linear interpolation of the maximum likelihood model in a document with the model based on the whole corpus, and a coefficient λ combines the two parts:

$$p(w|d) = (1 - \lambda)p_{\texttt{ml}}(w|d) + \lambda p(w|\mathcal{C}), \tag{2.5}$$

where $\mathcal{C}$ denotes the whole corpus. This simple mixture solves the data sparsity problem. For terms that occur in the document d, the maximum likelihood estimator (Equation 2.3) is not accurate given the limited size of a document, thus it is smoothed with the more reliable corpus level probability. For a missing term w in the document d, the probability of generating word w is not zero any more, but falls back to the corpus level probability $p(w|\mathcal{C})$. This smoothing method has been explored and successfully applied in information retrieval tasks [Ponte and Croft, 1998, Song and Croft, 1999].

Bayesian Smoothing using Dirichlet Priors A language model can be viewed as a discrete distribution, thus it can be smoothed by applying the Dirichlet distribution as the conjugate prior [Mackay and Peto, 1995]. We made a similar observation in the previous chapter comparing Equation 1.1 and Equation 1.2; the same intuition can be extended through multiple layers of discrete distributions with Dirichlet priors to create a smoothing model. Intuitively, this smoothing adds an extra prior count for each word to smooth the probability of unseen words,

$$p(w \,|\, d) = \frac{n_{d,w} + \beta p(w|\mathcal{C})}{\sum_{v \in V} n_{d,v} + \beta}, \tag{2.6}$$

where the Dirichlet prior is decided by concentration parameter β and the corpus-level probabilities $p(v \,|\, \mathcal{C})$,

$$(\beta p(v_1 \,|\, \mathcal{C}), \beta p(v_2 \,|\, \mathcal{C}), \cdots, \beta p(v_n \,|\, \mathcal{C})). \tag{2.7}$$

2.2 Topic-based Document Language Models

Topic models, which model each document as a mixture of topics and each topic as a mixture of words, offer an interesting framework to model documents in information retrieval. Popular topic models such as probabilistic latent semantic analysis (pLSA) and latent Dirichlet allocation (LDA) have been both explored to improve document language models.

Hofmann [1999b] introduces pLSA to to learn the relationship between query words and documents, and the conditional probability of a query word w given a document d is computed as marginalizing all topics k,

$$p_{\texttt{TM}}(w \mid d) = \sum_{k} p(w \mid k) p(k \mid d) \tag{2.8}$$

Following this idea, Wang et al. [2013] further add regularizations—changing the shape of distributions to be more or less spread out—on document topic representations which is useful for retrieval. Instead of using pLSA, Wei and Croft [2006] apply the same idea to learn the topic-smoothed document-word distribution using LDA. Vosecky et al. [2014] also explore LDA for document language models on twitter search.

Because the posterior estimates for topics are smoothed by the Dirichlet priors, topic models learn a better and smoothed semantic relationship between document words and documents. As a result, even though this approach loosens the connection between query words and documents, it is a good approach to complement the original document language models. Thus Wei and Croft [2006] further propose to combine the LDA-based document model with the original document model (Equation 2.5) through a linear interpolation,

$$p(w \mid d) = \lambda' \Big((1-\lambda) p_{\texttt{ml}}(w \mid d) + \lambda p(w \mid \mathcal{C}) \Big) + (1-\lambda') p_{\texttt{TM}}(w|d) \tag{2.9}$$

where λ' is the coefficient which combines the LDA-based document model with the general smoothed language model.

Following Wei and Croft [2006], Lu et al. [2011] further evaluate the performance of applying topic models into the document language model framework. Instead of combining with the language model with

Jelinek-Mercer smoothing (Equation 2.5), Lu et al. [2011] smooth the document language model with Bayesian smoothing (Equation 2.7), and the final linear combination with topic models

$$p(w \mid d) = \lambda \frac{n_{d,w} + \beta p(w \mid \mathcal{C})}{\sum_{v \in V} n_{d,v} + \beta} + (1 - \lambda) p_{\texttt{TM}}(w \mid d) \tag{2.10}$$

While using different smoothing strategies, both approaches apply topic models to connect the query words with documents through hidden topics. As the example shown in Wei and Croft [2006], given a query "buyout leverage", a relevant document talks about "Farley Unit Defaults On Pepperell Buyout Loan" without the exact word "leverage", thus the ranking for this relevant document is very low. However, topic models connect this document with two topics that that have strong connections with the term "leverage": one economic topic contain words like "million", "company" and "bankruptcy", and the other money market topic is connected to "bond". Since a better semantic relationship between the query and the document is learned, this relevant document is ranked much higher and the retrieval performance improves.

2.3 Query Expansion

The document language models in information retrieval [Ponte and Croft, 1998] attempt to model the query generation process based on the document models. However, a big problem is that these models abandon modeling the query-document relevance explicitly [Lavrenko and Croft, 2001], which is important in traditional information retrieval tasks.

In fact, queries, which are normally brief and using informal language from users, diverge significantly from the language in documents [Müller and Gurevych, 2009]. This semantic gap or lexical gap can result in perceived poor query-document relevance, even though the document is quite relevant from the users' view point. For example, after typing in "apple products" to a search engine, a frustrated user might append "computer" to the query after faced with a screen of fruit-based search results.

Query expansion tries to automatically simulate a similar process to prevent this frustration. Query expansion normally analyzes the relationships between the query words and other words and tries to find potential related words so that the original query is better represented; thus improving better query-document relevance. For example, without much context, the query "dtd amc" is hard to understand [Jiang et al., 2016]. Through query expansion, it is possible to build up the relationship between "dtd" and "disneyland downtown", which is more helpful in document retrieval. Next, this section reviews the classic query expansion frameworks in information retrieval, and the related works about using topic models for query expansion are introduced in the next section.

2.3.1 Learning Query-Word Relationships for Query Expansion

There are two main steps for query expansion. The first step is to find the relationships between queries and words and select the top related words to expand the query. The second step is to apply the expanded queries for ranking and compute the final ranking relevance scores. We start with the first step. Two major directions have been explored: query language models [Zhai and Lafferty, 2001b] and relevance models [Lavrenko and Croft, 2001].

Query Language Model To learn the query-word relationship, Zhai and Lafferty [2001b] build up a query language model to estimate the probability $p(w \mid q)$ of a word w given a query q. However, it is not easy to learn a good query language model since the query content is too limited.

Zhai and Lafferty [2001b] propose to use both the query content and the relevant documents $\mathcal{F}$ (sometimes referred as feedback documents or clicked documents) to estimate the query language model. Let $\hat{\theta}_{\mathcal{F}}$ be the estimated query language model based on the relevant documents and $\hat{\theta}_Q$ is the original query language model estimated purely based on queries, the combined query model $\hat{\theta}_{Q'}$ is

$$\hat{\theta}_{Q'} = (1 - \lambda)\hat{\theta}_Q + \lambda\hat{\theta}_{\mathcal{F}} \tag{2.11}$$

Estimating $\hat{\theta}_Q$ is obvious based on query words, and $\hat{\theta}_{\mathcal{F}}$ can also be simply estimated by a unigram language model θ which generates each word in $\mathcal{F}$ independently. However, most documents contain not only the query relevant information, but also the background information. As a result, Zhai and Lafferty [2001b] propose to generate a relevant document by a mixture model, which combines a language model $p(w \mid \theta)$ with a collection language model $p(w \mid \mathcal{C})$, and the log-likelihood of the relevant document is,

$$\log p(\mathcal{F} \mid \theta) = \sum_i \sum_w n_{d_i,w} \log((1-\lambda')p(w|\theta) + \lambda' p(w \mid \mathcal{C})) \qquad (2.12)$$

where $n_{d_i,w}$ is count of word w in document d_i. By combing information from relevant documents, the query language model is more robust, thus query-word relationships are better represented.

Relevance Model Unlike the query language model approach, Lavrenko and Croft [2001] assume both the query and the relevant documents are random samples from an unknown relevance model R. Given the query q, they approximate the probability $p(w \mid R)$ based on the observed query q as

$$p(w \mid R) \approx p(w \mid q) = \frac{p(w,q)}{p(q)}. \qquad (2.13)$$

To estimate the joint probability $p(w,q)$, Lavrenko and Croft [2001] assumes the word w and the query q are sampled independently from the same distribution, e.g., from a unigram distribution, then the joint probability is

$$p(w,q) = \sum_{d \in \mathcal{C}} p(d)p(w,q|d) = \sum_{d \in \mathcal{C}} p(d)p(w \mid d)p(q \mid d). \qquad (2.14)$$

Then the discrete distribution $p(w \mid q)$ for a given query q is

$$p(w \mid q) = \frac{p(w,q)}{p(q)} = \sum_{d \in \mathcal{C}} p(w \mid d)p(d \mid \mid q). \qquad (2.15)$$

Both the query language model and the relevance model capture the relationship between the query and other words, based on which the top related words can be selected for query expansion.

2.3.2 Ranking Relevance with Query Expansion

Given the related words for query expansion, the next question is how to apply the expanded queries for computing the final ranking relevance score. The combination can happen either before or after the relevance score is computed.

Zhai and Lafferty [2001b] combine the expanded query language model $\hat{\theta}_{\mathcal{F}}$ with the original query language model $\hat{\theta}_Q$ as one query language model $\hat{\theta}_{Q'}$ (Equation 2.11). Given a query q generated from the expanded query model $p(q \mid \hat{\theta}_{Q'})$, and a document d generated from a document model $p(d \mid \hat{\theta}_D)$, they compare how similar the topic distributions are between these two "documents" are.

In contrast to Zhai and Lafferty [2001b], Lavrenko and Croft [2001] compute the relevance score using the original query and the expanded query respectively, and then linearly combine the two scores. Thus the final query-document relevance $\hat{s}_d(q)$ is computed as,

$$\hat{s}_d(q) = \lambda s_d(e) + (1 - \lambda) s_d(q) \tag{2.16}$$

where $s_d(q)$ is the relevance between the original query q and documents d, and $s_d(e)$ is relevance between the expanded query terms e and document d.

2.3.3 Applying Topic Models For Query Expansion

Topic models capture the semantic relationships of words through learning the latent topics, which are presented as distributions over different words. Such semantic relationships among words provide a unique way to match or expand words at the semantic level rather than by a direct spelling matching. For example, given a short query "diabetes", topic models can easily find the related words such as "insulin", "glucose", "coronary" and "metformin" etc., as they often occur in the same context [Zeng et al., 2012]. As a result, topic models have been successfully applied into query expansion [Yi and Allan, 2009, Park and Ramamohanarao, 2009, Zeng et al., 2012].

Smoothing Query Language Model The most intuitive way to use topic models for query expansion is to extract the words' relevance from

topics directly as Yi and Allan [2009]. They train a topic model, from which the probability $p_{\texttt{TM}}(k \mid q)$ of a topic k in a query q is learned. Then the query-word relevance $p(w \mid q)$ is computed based on topics:

$$p(w \mid q) = \sum_k p_{\texttt{TM}}(w \mid k) p_{\texttt{TM}}(k \mid q). \tag{2.17}$$

This query-word relevance $p(w \mid q)$ from topic models smooths the original query language model through linear interpolation. However, queries are normally too short to learn meaningful topics, thus the quality of query-word relevance is relatively limited. To improve the quality of extracted topics, Yi and Allan [2009] also train topic models from the relevant documents (e.g., top documents retrieved by a query), and extract the query-word relationships based on the Equation 2.17 for query expansion.

Improving Relevance Model In addition to this direct approach, Yi and Allan [2009] also apply topic models to improve the relevance model in Equation 2.15 for query expansion. In this approach, topic models capture the document-word relationship $p(w \mid d)$ given the query q as

$$p_{\texttt{TM}}(w \mid d, q) = \sum_k p(w \mid k) p(k \mid d, q) \tag{2.18}$$

where,

$$p(k|d, q) = \frac{p(k|d)p(q|k)}{p(q|d)} \approx p(k|d)p(q|k) \tag{2.19}$$

where $p(k \mid d)$ is the topic probability in document d, and $p(q \mid k)$ is the probability of generating a query q given the topic k. Then the topic-based document-word relationship $p_{\texttt{TM}}(w \mid d, q)$ is applied to smooth the document-word relationship $p(w \mid d)$ in the relevance model (Equation 2.15) through a linear interpolation,

$$p(w \mid q) = \sum_{d \in \mathcal{C}} (\lambda p(w \mid d) + (1 - \lambda) p_{\texttt{TM}}(w \mid d, q)) p(d \mid q) \tag{2.20}$$

where λ is a constant weight to combine the original relevance model and the topic-based relevance model. Because the topic models capture the word relationships on a semantic level, this improved relevance model better captures the query-word relationships and improve query expansion.

Learning Pair-wise Word Relationships Park and Ramamohanarao [2009] also apply topic models for query expansion but in a different way. They model the pair-wise relationships between words through topic models and then apply it for query expansion. More specifically, based on the topics extracted from topic models, they compute the probabilistic relationships of each word pair (w_x, w_y),

$$\begin{aligned} p(w_x \mid w_y, \alpha) &= \sum_k p(w_x, k \mid w_y, \alpha) \\ &= \sum_k p(w_x \mid k, \alpha) p(k \mid w_y, \alpha) \end{aligned} \tag{2.21}$$

where α is the concentration parameter of the Dirichlet prior for document-topic distributions and $p(w_x \mid k, \alpha)$ is the probability of word w_x in topic k which can be learned from topic models, and $p(k \mid w_y, \alpha)$ is

$$p(k \mid w_y, \alpha) = \frac{p(w_y|k, \alpha)p(k|\alpha)}{\sum_{k'} p(w_y|k', \alpha)p(k' \mid \alpha)}$$

where $p(w_y \mid k, \alpha)$ is the probability of word w_y in topic k. Park and Ramamohanarao [2009] also show that $p(k|\alpha) = \frac{\alpha_k}{\sum_j \alpha_j}$. As a result, we have,

$$p(k|w_y, \alpha) = \frac{p(w_y \mid k, \alpha)\alpha_k}{\sum_{k'} p(w_y|k', \alpha)\alpha_{k'}}$$

The final probabilistic relationships of each term pair can be represented as

$$p(w_x|w_y, \alpha) = \frac{\sum_k p(w_x|k, \alpha)p(w_y|k, \alpha)\alpha_k}{\sum_{k'} p(w_y|k', \alpha)\alpha_{k'}} \tag{2.22}$$

Once this term relationship is obtained, they choose the top related terms as the expanded terms e for the given query q, and the final document ranking score is computed as Equation 2.16.

Interactive Feedback Relevance feedback involves the users in the retrieval process to improve the ranking result set [Rocchio, 1971]. The basic idea is to ask users to give feedback on the relevance of documents in an initial set of retrieval results, and users' feedback on relevance is

Table 2.1: Given query "euro opposition", seven topics are selected and shown to a user. The user selected Topic 79 as the feedback topic (Example from Andrzejewski and Buttler [2011]).

Topic	Terms
196 (debate)	Tory Euro sceptics, social chapter, Liberal Democrat, mps, Labour, bill, Commons
404 (ratification)	ratification Masstricht treaty, Poul Schluter, Poul Rasmussen, Danish, vote, Denmark, ec
79 (Emu)	economic monetary union, Masstricht treaty, member states, European, Europe, Community, Emu
377 (George)	President George Bush, White House, Mr Clinton, administration
115 (power)	de regulation bill, Sunday trading, Queen Speech, law, legislation, government, act
446 (years)	chairman chief executive, managing director, finance director, Sir, board, group, company
431 (cabinet)	Mr John Major, prime minister, Mr Major, party, tory, government, Conservative

further used to improve the ranking results. This process can go through one or more iterations.

Andrzejewski and Buttler [2011] present a new framework for obtaining and exploiting user feedback at the latent topic level. They learn the latent topics from the whole corpus and construct meaningful topic representations. At query time, they decide which latent topics are potentially relevant and present the topic representations along keyword search results. When a user select a latent topic, the original query is expanded with the top words in this selected topic, and the search results are refined. Andrzejewski and Buttler [2011] use the query "euro opposition" as an example: users want to find documents about opposition to the introduction of the single European currency. 500 topics are learned using the corpus and relevance judgements. Seven of the 500 topics are selected to show to users as shown in Table 2.1 and the user select the Topic 79 as the user feedback. Using the top terms in Topic 79 as the expanded query terms, the ranking of the relevant documents improves.

This direction is related to topic labeling and interactive visualization for topic models, which will be further discussed in Chapter 3.

2.4 Beyond Relevance—Search Personalization

Traditional search systems retrieve documents based on the queries only, regardless of who submitted the queries. As more Web pages become available, queries are normally too short to express users' needs, and users may prefer different results even the input queries are the same [Jansen et al., 2000, Dou et al., 2007]. As the examples shown in Dou et al. [2007], the query "mouse" may mean "rodents" for biologists, while the programmers may use the same query to search for computer peripherals. Even for queries without ambiguity, for example "online shopping", some users may prefer "`www.amazon.com`" while others may prefer "`www.ebay.com`".

Understanding users' preference and context helps meet their information needs. Thus, IR systems must adapt the ranking results [Pitkow et al., 2002, Micarelli et al., 2007]. This is referred as personalized search, or search personalization.

There are multiple ways to do search personalization, and two of major directions are normally referred as contextualization [Melucci, 2012] and individualization [Pitkow et al., 2002]. The former is to consider users' conditions in a search activity, for example, time and location. The latter focuses more on users' individual characteristics and activities, which is also described as the users' profile. Topic models have been investigated to model users' preference on the topic space [Song et al., 2010, Carman et al., 2010].

Modeling Users' Preference via the Output Topics Song et al. [2010] apply topic models to model users' preference from users' search history. The idea is very similar to smoothing query language models $p(w \mid q)$ by topic models, as explained in Equation 2.17. The estimation of $p(w \mid q)$ is split into two parts $p(w \mid k)$ and $p(k \mid q)$.

For each users' query, they concatenate the clicked documents (or the top n ranked documents if no click happened) as a big preference

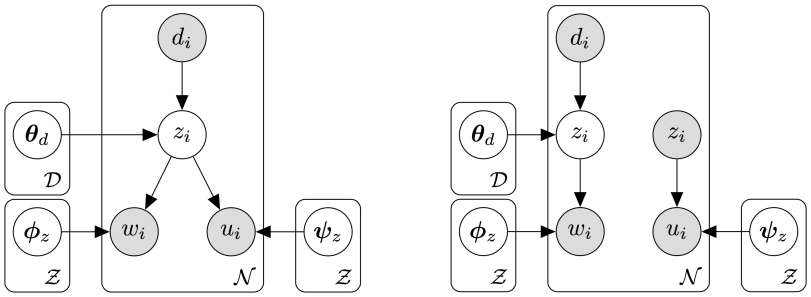

Figure 2.1: The plate diagrams for personalized retrieval in Figure 2.1a and the actual simplified model used for parameter estimation in Figure 2.1b (Both figures from Harvey et al. [2013].)

document. Then the topic model pLSA is applied on the preference collection to extract the latent topics as users' preference ($p(w \mid k)$ in the Equation 2.17). The second part is to estimate the query-topic distribution $p(k \mid q)$ in Equation 2.17. However, the queries are too short to estimate the query topics directly. Instead, they first estimate a language model θ_q from the big preference document of query q and compare the cosine similarity against each topic to estimate the query-topic distribution $p(k \mid q)$,

$$p(k \mid q) = \frac{p(k,q)}{\sum_k p(k,q)} \approx \frac{\text{sim}(\theta_k, \theta_q)}{\sum_k \text{sim}(\theta_k, \theta_q)}. \tag{2.23}$$

This can then direct the user to documents that match the user's interests (e.g., that match those topics well).

Encoding Users into Topic Models Carman et al. [2010] also investigate topic models on large query logs for search personalization and propose a personalization topic model as shown in Figure 2.1a. The idea is given the topic distribution of the document, there will be words chosen at random to generate the query and users who chose to click that document.

As shown in Figure 2.1a (this uses plate diagram formalism, explained in Chapter 1.5.3), this model has three observed variables,

document d_i, query word w_i and user u_i. Given a topic k sampled from a discrete distribution θ_d, the corresponding query words w_i are sampled from a topic-word discrete distribution ϕ_k and the user u_i who submitted the corresponding query is sampled from a topic-user discrete distribution ψ_k.

To estimate the probability of $p(k \mid w_i, d_i, u_i)$, they further assume the conditional independence among the word w_i, the user u_i and document d_i given the topic k, and the model can be simplified as

$$p(k \mid w_i, d_i, u_i) = \frac{p(k, w_i, u_i \mid d_i)}{p(w_i, u_i|d_i)} \propto p(w_i \mid k)p(u_i \mid k)p(k \mid d_i). \quad (2.24)$$

By directly including the user in the topic model, this model assumes the user's topical interests for describing a document that the user clicked is equally important as the words to describe the document. This assumption is too strong [Carman et al., 2010]. As a result, Harvey et al. [2013] further propose to ignore the user during inference and simplify the model (Figure 2.1b). In this model, the topics are used to infer the topic-user distribution $p(u_i|k)$ once the Markov chain is converged. The intuition is to capture the idea that a user clicks on a document given a specific query due to his/her interests expressed over the topic space [Harvey et al., 2013].

Based on the estimates of this personalized topic model, the documents are ranked by the likelihood given the query and the user as follows,

$$\begin{aligned} p(d \mid q, u) &\propto p(d) \prod_{w \in q} p(w, u \mid d) \\ &\propto p(d) \prod_{w \in q} \sum_{k} p(w \mid k)p(u|k)p(k \mid d) \end{aligned} \quad (2.25)$$

By subtly incorporating users' profiles as part of the ranking algorithms, Harvey et al. [2013] significantly improved the personalized ranked document lists than the non personalized baselines.

2.5 Summary

Because topic models analyze documents on a semantic level, they offer an interesting and unique framework for modeling relationships between

words, and between documents and words. As a result, topic models have been successfully applied in smoothing language models, query expansion and search personalization.

There is also some work focusing on the diversification of search results [Dang and Croft, 2013, Santos et al., 2015]. The goal of search result diversification is to identify the different aspects of the ambiguous query, retrieve documents for each aspect and make the search results contain more varied documents [Dang and Croft, 2013]. Personalization and diversification are orthogonal, and can be combined in unified models [Vallet and Castells, 2012, Liang et al., 2014].

These successful applications on information retrieval are based on a good understanding of the outputs of topic models. In the next chapter, we will introduce how to visualize, label, and evaluate topics to help users better understand topics and how they are expressed in collections.

3

Evaluation and Interpretation

While the previous chapter focuses on algorithmic uses of topic models, one of the reasons for using topic models is that they produce human-readable summaries of the themes of large document collections. However, for users to use the results of topic models, they must be able to understand the models' output. This depends on model *visualization*, *interaction*, and *evaluation*.

We begin this chapter with a discussion of how best to show individual topics to users. From these foundations, we move to how we can display entire models—with many topics—to users. Finally, we close with how users can provide feedback through these interfaces to detect errors and improve the underlying model.

3.1 Displaying Topics

Recall from the previous chapters that topics are distributions over words; the words with the highest weight in a topic best explain what the topic is about. While the simplest answer—just show the most probable words—is a common solution, there are refinements that can improve

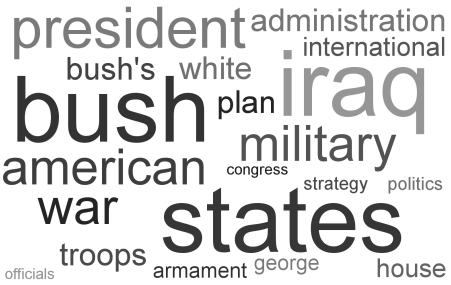

Figure 3.1: Word clouds use a 2D layout to show which words appear in a topic. Word size is related to its probability in the topic, showing which words are more prominent.

a user's understanding of a collection by showing the relationships between words or explicitly showing words' probability.

Word Lists Just showing a list of the most common words (a visualization that we will call "word list") is very simple, and works well. Users can quickly understand how words are arranged, and it is an efficient use of space. Topics have been represented horizontally [Gardner et al., 2010, Smith et al., 2015] or vertically [Eisenstein et al., 2012, Chaney and Blei, 2012], with or without commas separating the individual words, or using set notation [Chaney and Blei, 2012]. Smith et al. [2015] go further by adding bars representing the probabilities of the word.

Word Clouds Word clouds (e.g., Figure 3.1) are another popular approach for displaying topics. Unlike word lists, they also use the size of words to convey additional information. Word clouds typically use the size of words to reflect the probability of the words. This uses more of a given visualization area to be used to display a topic.

However, word clouds have been criticized for providing poor support for visual search [Viégas and Wattenberg, 2008] and lacking contextual information between words [Harris, 2011]; users can sometimes draw false connections between words that are placed next to each other randomly in a word cloud. Another alternative is to use word associations

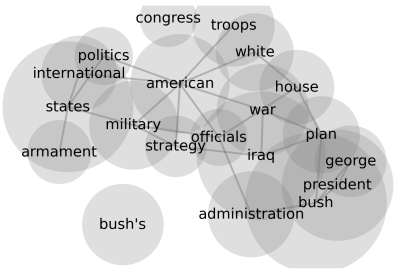

Figure 3.2: A topic-in-a-box visualization for topics—like a word cloud—shows words in a 2D context. However, it uses local co-occurence (whether words appear together in a sentence) to decide which words to place next to each other.

to set the position of words [Smith et al., 2014]; Figure 3.2 places words that appear together next to each other in the visualization.

3.2 Labeling Topics

Throughout this survey, we have been referring to topics with *labels* such as Information Technology or Arts. These are convenient descriptors, but completely removed from the raw distribution over words. Thus, it is often useful to assign labels to topics within an interface.

In contrast to the previous *visualization* approaches, labeling focuses on showing not the original words of a topic but rather a clearer label more akin to what a human summary of the data would provide.

Approaches for automatic labeling can be divided into those that only use internal information from the topic model against those that also use external knowledge resources. While purely internal methods are more robust and consistent with the philosophy of unsupervised topic models, external resources often produce higher quality labels.

Of the techniques that use external resources, we further separate those that use direct supervision for labeling (i.e., knowing what constitutes a good labeling) from those that use general knowledge resources such as Wikipedia or knowledge bases.

Internal Labeling Mei et al. [2007b] propose an internal labeling method that takes prominent phrases from the topic and compares how consistent the phrase's context is with the topic distribution. Phrases whose contexts closely resemble the topic often appear in regions of text that summarize the document, making them good candidates for labels. Mao et al. [2012] extend the technique to hierarchies, using the insight that parents' labels should be consistent with their children's.

Labeling with Supervised Labels Lau et al. [2010] use a supervised approach to rerank the words in a topic to ensure that the user sees the "best" word in the topic. Each candidate word forms a feature vector consisting of features such as the following:

- the conditional probability of a word given the other words in a topic (which implies topic coherence, as discussed in Chapter 3.4);
- whether the word is a hypernym of other words in the topic (e.g., "dog" in a topic that also contains "terrier" and "poodle"); and
- the original probability of the word in the topic.

While these can be used alone as an unsupervised reranking, Lau et al. [2010] use user-selected best topic words to weight which of these features are most important for selecting the best topic word. These weights are learned using support vector regression. Lau et al. [2011] extend their technique by adding candidates from Wikipedia to the set. The weakness of this approach is that Wikipedia may not have coverage of the topics in the collection; if Wikipedia ignores the theme captured by a topic model, then it will fail to find an appropriate label for that topic.

Labeling with Knowledge Bases Mao et al. [2012] align topic models with an external ontology of labels. They argue that labels should match topic words (as labeling with flat topics); a topic's words should be consistent with a labels' children in the hierarchy; and the topic's labels should be unique.

Aletras et al. [2014] instead query the whole Web and then build a graph that includes the words in the titles of the retrieved webpages.

Their goal is to find words that are "central" in the graph: these words should make for a good title. Words have edges between them if they appear close to each other more than one would expect by chance. This property is measures through the normalized pointwise mutual information (NPMI) metric. They find the central words by using the PageRank [Page et al., 1999] algorithm, which finds words that are highly probable in the topic and appear frequently with many other words in the topic. This is the same algorithm that search engines use to find pages that have "high authority" on the Internet.

Just like a search engine should return `simpsonsarchive.com` for a search on "the Simpsons" because everyone links to it, this labeling method will find the word in a topic that all of the other words "vote for". For each word in a topic, find all of the words that are likely to also appear with that word and then take the winner of that election as our label for a topic. For example, given the topic

> cell response immune lymphocyte antigen cytokine t-cell induce receptor immunity

the algorithm selects the topic Immune System, as it appears near many of the other terms in the topic [Aletras et al., 2014].

Using Labeled Documents The task of associating labels with topics becomes much easier if many of your documents are themselves labeled. Labeled LDA [Ramage et al., 2009] associates topics to each of the labels and forces labeled documents to only use the topics associated with the document. This constraint forces the topics to be consistent with the original labels (Figure 3.3). Bakalov et al. [2012] extend this to hierarchical label sets (e.g., NY Times subjects that place Russia under International), while Nguyen et al. [2014] extend it to learning hierarchies of topics from unorganized labels, learning that Ska[1] is a kind of Music without provided links.

[1] A musical genre, familiar to reggae fans and cruciverbalists.

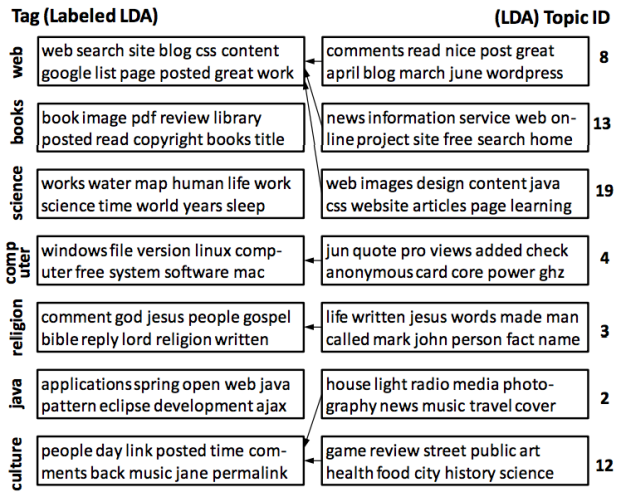

Figure 3.3: Example of topics learned by labeled LDA (Figure from Ramage et al. [2009]). Each topic in labeled LDA is associated with a label, which encourages the topics to be consistent with the ontology of labels. LDA, in contrast, uses the empirical frequency of topics to divide the collection, resulting in three topics (8, 13, 19) associated with the labeled LDA Web topic.

3.3 Displaying Models

However, topics are not the end of the story. Users often want to use topics to find relevant documents within the collection. Going back to our example in a previous chapter, a user may want to find the "smoking gun" in the Enron corpus, not just use topics to understand the main themes in a collection.

Thus, a good topic model visualization must also show the documents associated with a topic. The Topic Model Visualization Engine [Chaney and Blei, 2012, TMVE] shows the top documents associated with a topic (Figure 3.4). Recall that each document has a distribution over topics θ_d, which is a vector with an entry for each topic. We focus on the dimension associated with a particular topic and then sort the documents based on that topic coordinate from largest to smallest.

The topical guide [Gardner et al., 2010] extends this approach by enriching topic views with additional metadata. For instance, if

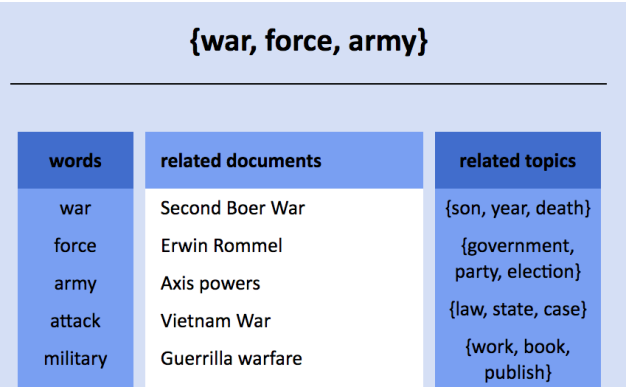

Figure 3.4: The Topic Model Visualization Engine [Chaney and Blei, 2012] shows the most related documents to a topic along with related topics.

the collection has dollar amounts or sentiment [Pang and Lee, 2008] associated with a document, it provides a histogram of the metadata associated with the topic. It also provides *in context* examples of topic words, allowing to see how a word is used within a topic (helping to address the topic model's bag of words assumptions).

Interactive TOpic Model and MEtadata [Eisenstein et al., 2014, Interactive TOpic Model and MEtadata] focuses on a specific type of metadata: time. It allows users to view the evolution of topics over time to understand, for example, how the issue of slavery is reframed from an economic argument to an argument over human rights. It supports filtering to specific topics or to see how words are used over time across topics.

Rather than showing how topics relate to metadata, Chuang et al. [2012] focus on how topics relate to *each other*. Their "Termite" topic visualization (Figure 3.5) shows the term-by-term similarity between topics. By presenting topic-term probabilities on a grid with topics as the columns and terms as the rows, users can see when topics share words or when topics are only about a handful of words.

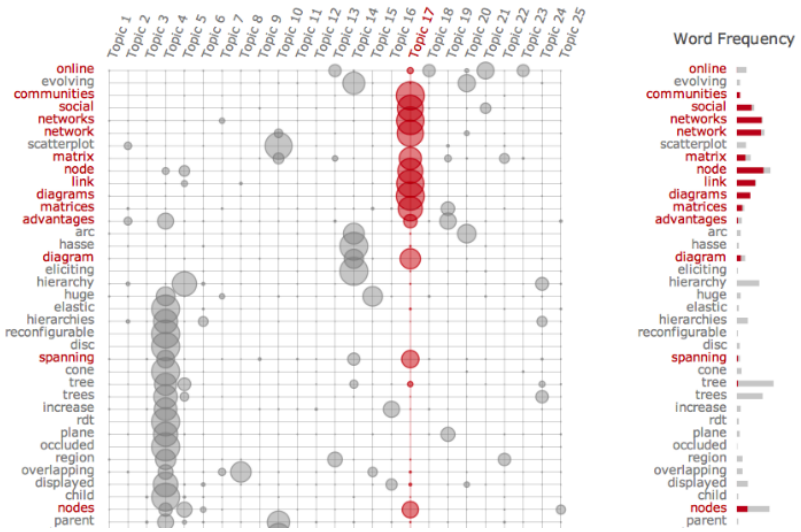

Figure 3.5: The Termite visualization of topics helps reveal which topics use similar words and are thus likely talking about similar things.

3.4 Evaluation, Stability, and Repair

Visualizations can help show users where topic models have issues. Topic models are rarely perfect, and quality can vary within a model. Even in good models we often find several poorly fit or improperly combined topics.

For many years, the primary metric for evaluating the quality of a topic model was the held-out likelihood of a model [Wallach et al., 2009b]. Because a topic model is a generative probabilistic model—like a language model [Chen et al., 1998]—we can ask how well the model can predict unseen text: run the generative process forward for the document and see how well that matches up with the held-out document. If the model does a good job of using topics to predict what words will appear in new documents, then it is a good model, and if it fails to do so, it is a bad model.

In some ways held-out likelihood makes sense, but it is incomplete. We should be able to detect some kinds of failure: topics that are just random noise will have poor held-out likelihood. On the other hand,

hundreds topics so specific that any held-out document is modeled well yields an excellent held-out likelihood. The topics would nevertheless lack generalizability and interpretability in the eyes of a user.

Chang et al. [2009] show that held-out likelihood, a traditional measure of probabilistic model quality, emphasizes *complexity* rather than the ease of interpretability that users are looking for. User ratings of how good topics negatively correlate with held-out likelihood: a more complex model (e.g., a model with more complicated equations or a model with more topics) can better fit a random held-out document. More complex models, however, are more confusing for users.

Automated measurements [Newman et al., 2010, Mimno et al., 2011, Lau et al., 2014] of topic quality may serve as a proxy for human interpretability ratings. However, these approaches may not be able to tell you whether a topic model is suitable for a specific application, which parts of a model are reliable, or why. Showing the relationships between multiple models can also help distinguish stable from spurious topics [Chuang et al., 2015], and adjusting the "hyperparameters" of distributions (the Dirichlet parameters of models discussed in Chapter 1) can have a large effect of what the final models are [Wallach et al., 2009a].

Tang et al. [2014] provide a diagnosis manual for what properties of a dataset can cause the failure of a topic model: a mismatch between the number of topics and documents, topics that are "too close together", or a mismatch between the number of topics in a document and the Dirichlet parameter α.

Interactive topic modeling—in conjunction with visualizations—can help correct the problems of topic models. A user first gets an overview of the collection using a visualization of the topics and documents and can then see and correct instances where the model makes mistakes.

For example, Figure 3.6 shows a topic learned from abstracts of grants funded by the American National Institutes of Health (NIH, discussed more in Chapter 5.1). Most topics were "good": they summarized the data and told a story about a coherent slice of research supported by the NIH. However, this topic is more problematic; it combines words about the central nervous system with words about the urinary system.

Topic Words (before)	Topic Words (after)
bladder, sci, spinal_cord, spinal_cord_injury, spinal, urinary, urinary_tract, urothelial,injury, motor, recovery, reflex, cervical, urothelium, functional_recovery	sci, spinal_cord, spinal_cord_injury, spinal, injury, recovery, motor, reflex, urothelial, injured, functional_recovery, plasticity, locomotor, cervical, locomotion

Figure 3.6: Example topics before and after interactive topic modeling from Hu et al. [2014a]. Initially, the topic conflates two topics (urinary and central nervous system), which is undesirable. Adding a constraint that the words "bladder" and "spinal cord" should not appear together in a topic makes the topic more coherent and discovers concepts that were not present before.

Such a topic [Mimno et al., 2011] does not give a clear understanding of the documents it should represent.

Hu et al. [2014a] address this problem by allowing a user to add probabilistic constraints to the model [Boyd-Graber et al., 2007, Andrzejewski et al., 2009]. For example, the user might say that "bladder" and "spinal cord" do not belong in the same topic together. Figure 3.6 shows how the topic is more focused after the user provides this feedback. In contrast to probabilistic constraints, Choo et al. [2013] and Lund et al. [2017] use matrix factorization constraints to guide changes to topics, which can be much faster.

3.5 Summary

While topic models provide users with overviews of corpora, topic models cannot be much help if the users cannot effectively see or understand the underlying topics and how they relate to specific documents. Evaluations help to identify which portions of a model to trust and which to use with caution. Interactive visualizations allow users to discover and refine insights. In the next chapters we will talk about specific applications of these insights, but these insights are often built on the

initial understanding of a model offered by the visualizations discussed in this chapter.

4

Historical Documents

Topic models play an important role in the analysis of historical documents. Historical records tend to be extensive and difficult to manage without intense and time-consuming organization. Records are complicated: they resist categorization, and may even lack standard spelling and formatting. But there is more to history than the management of documents. The task of a historian is not only to absorb the contents of historical records, but to generalize; to find patterns and regularities that are true to the documents, but also beyond any single piece of evidence. Topic models are useful because they address these issues. They are scalable, robust to variability, and able to generalize while remaining grounded in observation.

Automated methods are an especially valuable counterpoint to traditional scholarly methods. Studying history is about encountering the unexpected, often in contexts that seem familiar. We do not necessarily know how people in the past talked about particular issues, or how they organized their lives. Perhaps more dangerously, we assume that we know these things, and that our ancestors saw the world in the same way we do. Topic models give us a perspective that is interpretable but

at the same time alien, based on patterns in documents and not on our own conceptions of how things should be.

Time is a critical variable in the study of historical documents. Although many modern collections have a significant aspect of time variation (see for example scientometrics), time is a defining element of historical research. Collections of historical documents are necessarily situated in a time other than our own, but also tend to cover long periods—decades or even centuries. As a result, many of the examples cited in this chapter organize documents along a temporal axis. The associated analysis is particularly concerned with how language, as reflected in topic concentrations and topic contents, changes over time.

This chapter is organized around different formats for historical documents. A recurring focus is the desire to plot events and discourses against time. We begin with historical newspapers, which are relatively close to the modern news articles that are a more familiar use case in topic modeling. We then consider other forms of historical records, such as annals and diaries. These show the flexibility of topic modeling, including a corpus not in English and corpus in English with irregular spelling. Finally, we consider studies of historical scholarly literature.

4.1 Newspapers

Newman and Block [2006] present an example of topic modeling on historical newspapers,[1] in a collection of articles from the *Pennsylvania Gazette* from 1728 to 1800.[2] These articles comprise 25 million word tokens in articles and advertisements, and cover several generations of everyday life before, during, and after the founding of the United States of America. The authors contrast their study to manually created keyword-based indexes, which focus on specific terms and can be applied inconsistently across large corpora. Spurious patterns in index term use could complicate historical research. They cite an example of the tag *adv*, which is used extensively in the early and late decades of the

[1] Mei and Zhai [2005] present an earlier example of *contemporary* news analysis (i.e., where the data are already digitized). Their work also uses topic models to show the evolution of themes over time.

[2] http://www.accessible-archives.com/

corpus, but not in the middle. The topic-based approach is attractive because it is consistent across the collection (as long as the terms used in the documents are themselves consistent) and because it is abstract, reducing the chance that modern historians miss key terms.

They compare three methods for finding semantic dimensions, latent semantic analysis [Deerwester et al., 1990], k-means clustering, and a topic model [Hofmann, 1999a]. The difference between these methods can be described via expressivity. LSA embeds word types and documents in a low-dimensional space well, but the individual dimensions of this space are not interpretable as themes. LSA is too expressive: it places no constraints, such as positivity, on the learned dimensions, and therefore produces uninterpretable results that nevertheless fit the document set well. The k-means clustering is more similar to the topic model, and more successful at finding recognizable themes. But it is also prone to repeating similar clusters with small variations. Because of the single-membership assumption (a document can only belong to one cluster), the clustering model cannot represent documents with varying combinations of somewhat independent themes. The k-means model is therefore insufficiently expressive: it forced to "waste" clusters on frequent combinations of simpler themes. The topic model, in contrast, has both modeling flexibility along with constraints to support interpretable results.

The authors find that the learned topics are a good representation of dynamics in the corpus, although not always in a direct manner. There is a large increase in discussions of Politics in the period immediately around the American Revolution (*state government constitution law united power*). There is also evidence of economic factors: a topic relating to descriptions of Cloth (*silk cotton ditto white black linen*) rises in the 1750s, but then declines as Americans turned to domestic "homespun" cloth production in response to British trade policies. Other topics point to more subtle changes in language. A topic that is less immediately interpretable (*say thing might think own did*) corresponds to a series of long "public letters" that contain more academic "argument making". This is consistent with other results [Viermetz et al., 2008] that suggest

topics may be long-term or transient, which is captured directly by Viermetz et al. [2008].

Nelson studies topics in Civil War-era newspapers, including the Confederate paper of record, the Richmond Daily Dispatch.[3] Like Block and Newman, Nelson's goal is to organize the collection into themes and to measure the variation in prevalence of those themes over time. The web interface highlights a temporal view of the collection as a series of topic-specific time series. The mode of analysis is neither fully automated nor manual, but rather combines the two approaches. Nelson manually labels the topics and groups them into larger categories such as "slavery", "nationalism and patriotism", "soldiers", and "economy".

He validates the model by comparing topics to a known and previously annotated category, the "fugitive slave ads". These documents were pre-photographic descriptions of runaway slaves, and have a specific language consisting of aspects of personal appearance and possible locations where enslaved people might have hidden. He finds a near perfect correspondence between the prevalence over time of manually labeled fugitive slave ads and documents that have a high concentration of a specific topic, which places high probability on terms such as *reward, years,* and *color* (manual labels were not used in training the model). Nelson notes that few if any of these documents are assigned completely to this topic: he uses a cutoff of 21.5% as a criterion.

Nelson's larger-scale groupings of topics pick out threads of discourse that may or may not be correlated over time. The model identifies three topics that have similar temporal distribution, peaking at the beginning of the war in 1861 and largely disappearing afterwards. These are related but distinct themes: anti-Northern sentiment expressed in poetic form, anti-Northern sentiment expressed in vitriolic prose, and discussion of secession. All three form aspects of the same process, the rhetorical push for war. Other related topics have slightly different temporal distributions. Nelson groups six topics related to soldiers, and displays them in the order of their maximum concentration over time. They move from "military recruitment" and "orders to report" to later topics related to "deserters", "casualties", and "war prisoners." Again, these

[3]Mining the Dispatch, http://dsl.richmond.edu/dispatch/

are related themes but rather than comprising a single event they trace the development of the increasingly dire military situation of the Confederacy.

Yang et al. [2011] model a collection of historical newspapers from Texas spanning from the end of the Civil War to the present day. The goal is both exploratory, to find out about the interests of Texans through the 19th and 20th centuries, and *semi-exploratory*, to find out more about the history and context of specific, pre-specified themes such as cotton production. In the topic model setting, semi-exploratory analysis starts by identifying one or more topics that seem to correspond to the theme of interest, and then using those topics as an axis of investigation into the corpus. For example, a historian considered documents with topics related to cotton, and the topics that co-occur in those documents. The study also led to more fully exploratory results. A Battle of San Jacinto topic, the final conflict in the Texas Revolution that led to separation from Mexico, appeared earlier than expected. Further investigation suggested that the significance of the pivotal battle of San Jacinto was established much earlier than historians had previously anticipated.

The Texas newspaper study raises several interesting methodological issues relating to pre-processing and iterative modeling. The authors put considerable work into dealing with the quality of digitization. Many factors that affect the quality of digitized historical newspapers, from the quality of the original printing to scanning, article segmentation, and optical character recognition (OCR). For this study extensive work was applied to automated spelling correction. Another notable factor in this study is its prominent use of multiple topic models. There is often a tacit assumption that a single corpus should result in a single model, but in practice modeling is often iterative, and intimately bound to the development of pre-processing systems. AlSumait et al. [2008] iteratively refine their model with each segment of a news collection, while Yang et al. [2011] train different models on different temporal slices of the corpus. Although there is some advantage to maintaining a consistent topic space over time, dividing the corpus into separate sections has certain advantages. In this case, historians were interested in specific historic periods, such as the full run of a newspaper during several pivotal

years, that are smaller than the full corpus but yet too large to be read easily. The authors also describe an iterative workflow that involves comparing topic model output after each of several pre-processing steps. Topic models are often effective at identifying consistent data-preparation errors, such as end-of-line hyphenation and consistent OCR errors.

4.2 Historical Records

Other types of records besides newspapers are of interest, and present their own challenges. This section considers two case studies, in which the simplicity of the bag-of-words document model is an asset because it allows for substantial variability in spelling and language, both in English and in other languages.

Erlin [2017] search for work related to epistemology in a large corpus of English and German books. They "seed" the models for each language with several query words that the authors expect to be related to that subject. This approach is closer to standard information retrieval than many other topic model applications, since the model is used both as a way of organizing the corpus and as a way of focusing attention on specific aspects. Their use of a topic model differs from standard IR in that they are more deliberately open to related terms and concepts: epistemology is expected to be broad, and more likely to be represented by a combination of words than by any one query.

Miller [2013] uses Chinese records to investigate the meaning of the word *zei*, or "bandit" in Qing dynasty China (1644–1912). The word by itself can imply several different forms of anti-social behavior, which are difficult to distinguish from word frequencies alone. A topic model uses contextual information to separate these effects.

The application of topic models in Chinese highlights the importance of tokenization. We usually receive documents in the form of long strings, but we are interested in identifying *tokens* that are short strings with a specific meaning. Breaking a document into distinct tokens is an often-overlooked part of the document analysis process. In European languages we can achieve good results by separating strings of letter

characters from sequences of non-letter characters, although there are many special cases [Boyd-Graber et al., 2014]. Tokens may contain non-letter characters such as apostrophes and hyphens, and may span multiple words (*Queen Victoria*, *black hole*). In many East Asian writing systems we cannot rely on orthographic conventions to identify tokens. Miller argues that in Classical Chinese a single character can be treated as a token without harming modeling, but for Japanese and modern Chinese we must often rely on pre-processing tools that are themselves potentially unreliable.

Cameron Blevins models the diary of Martha Ballard (1735–1812), a revolutionary war-era midwife who recorded entries over 27 years.[4] The model provides a useful way to discover connections between words and repeated discourses. As with other historical corpora, Blevins focuses on the connection between topics and time. Specific events, like a birth, can be highlighted by looking at spikes in a certain topic in the day-to-day time series. But larger trends are also evident. As a calibration experiment, Blevins measures the association of a topic that appears to refer to Cold Weather (*cold, windy, chilly, snowy, air*) to months of the year. As expected, the concentration of this topic is lowest from May to August, rises from September to January, and falls from February to April.

Blevins identifies several other topics that appear to change in their concentration over time. Two topics involving house work focusing roughly on Cleaning and Cooking appear to be correlated in time, rising over the decades. Blevins connects this finding to suggestions that as Ballard grew older and her children moved away, she had less help from family members. A more subtle topic involves descriptions of Fatigue and Illness. This topic also increases over time, and appears to correlate with the housework topics, except in the last year of the diary, where fatigue and illness reach their highest concentration and housework declines.

This analysis exemplifies the exploratory nature of topic modeling: by themselves, these observations are not conclusive, but they are suggestive and point to areas of further analysis. A scholar might take

[4] http://www.cameronblevins.org/posts/topic-modeling-martha-ballards-diary/

the diary entries that score high on an individual topic as a reading list, and determine how well a particular automatically detected discourse maps to themes in Ballard's personal experience. For example, one might check whether Ballard's references to fatigue and illness are referring to herself or to patients. The model does not tell the whole story, but it points to where stories might lie.

Blevins argues that characteristics of the diary form make it well-suited for topic analysis: "Short, content-driven entries that usually touch upon a limited number of topics appear to produce remarkably cohesive and accurate topics." In addition, the topic model's lack of linguistic sophistication is an asset. The diary is written in a terse style with many abbreviations and with irregular, 18$^{\text{th}}$ century spelling: "mrss Pages illness Came on at Evng and Shee was Deliverd at 11h of a Son which waid 12 lb." Models trained on modern text corpora might not even recognize this example as English, but the topic modeling algorithm is still capable of finding semantically meaningful groups of words.

4.3 Scholarly Literature

The historical record of scholarship is a valuable source for intellectual history. Many users make use of the JSTOR "Data for Research" API.[5] The DFR API is an important example, because it provides access to articles that have been scanned by JSTOR and may be under copyright. Access to the underlying documents in their original form as readable sequences of words may be restricted for legal or commercial reasons. DFR provides a simple view into selected articles by only providing the frequency of word unigrams. While the bag-of-words assumption used by topic models is restrictive, in this case it can be an advantage, because the original sequence of words is not used for inference anyway.

Mimno [2012] studies a collection of Classics journals digitized by JSTOR to detect changes in the field over the 20$^{\text{th}}$ century. A distinctive aspect of this study is the use of a *polylingual* topic model [Mimno et al., 2009]. The details of this model and how it contrasts with other

[5] http://dfr.jstor.org/

models are described in more detail in Chapter 8.1; for this discussion we need some topic model that can discover topics that are consistent across languages. An English-language journal is compared to a German-language journal by learning a common set of topics that each have a vocabulary in both languages. In other words, a topic has two "modes", one in which it emits words drawn from a distribution over English terms, and another in which it emits words drawn from a distribution over German terms. The linkage between English and German words is constructed using Wikipedia articles. Wikipedia articles exist in many different languages, and articles in one language often link to comparable articles in another language. The author first selects English Wikipedia articles matching key terms in the English-language journals, and then collects the German Wikipedia articles that are listed as being comparable to the selected English-language articles.

By training the topic model jointly on the combined corpus of the original journal articles and the comparable Wikipedia articles, the model provides insight into the relative concentration of scholarly interests across the two language communities. The German-language journal articles contain more work on Law and Oratory, themes that are present in the English-language articles but less prevalent. The model also shows a large increase in interest in poetry in the German journal in the period following the second world war. In the English journals there is a large increase starting in the 1980s in cultural and economic studies along with critical theory, which does not appear in the German journals.

Riddell [2012] also approaches German scholarly literature from the 20^{th} century. He finds that topics align well with authors such as Goethe and subjects such as folklore. Apparent spikes in the use of these topics appear to align with anniversaries of authors (Göthe, the Grimm brothers). Riddell emphasizes that models are useful in raising issues but not a substitute for scholarship. He comments that "it becomes essential that those using topic models validate the description provided by a topic model by reference to something other than the topic model itself."

Goldstone and Underwood [2014] use a topic model as a tool to structure an exploration of a corpus that spans more than a century. They are interested both in changes at the topic level and at the level of word use within topics. For these authors the appeal of topic modeling is that models are better able to represent contextual meaning than simple lists of keywords. They write that "[t]he meanings of words are shifting and context-dependent. For this reason, it is risky to construct groups of words that we imagine are equivalent to some predetermined concept."

They analyze the proceedings of the Modern Language Association[6] to find shifts in focus in English literature. A model trained with 150 topics on 21,000 articles identifies a topic associated with descriptions of Violence: *power, violence, fear, blood, death, murder, act, guilt.* Using a temporal plot they argue that the concentration of this topic is greater in the second half of the 20$^{\text{th}}$ century than during the first half. They contextualize this finding by comparing the frequency of these words in a more general corpus from Google n-grams; there is no comparable change. This approach holds the topic fixed and searches for associated words. They then pivot and hold the word "power" fixed and search for associated topics. In this case the Violence topic appears to be relatively stable in its association with the target word. The largest increase is in a different topic characterized by the words *own power text form*, in which context it appears almost exclusively after 1980. Like many topics, the content of this topic is difficult to assess from top words alone. Further exploration through exploration of individual documents would be necessary (e.g., through the tools discussed in Chapter 3).

4.4 Summary

This chapter focuses on finding themes in document that reflect temporal trends. When we consider newspapers, historical records, and historical scholarly journals we are looking not just for the topical foci of each time period, but how those topics shift in concentration as they are influenced by historical events. Modeling large collections of documents

[6]The MLA is a professional organization for literary scholars in the United States.

allows us to reveal how events are reflected in writing and how ideas and emotions emerge in response to changing events.

In the next chapter, we extend our discussion of scholarly journals to focus more directly on how new ideas emerge. Unlike newspapers and diaries which reflect the reality of the world, the writing in scientific manuscripts can actually change the world by introducing innovative technologies. The next chapter asks whether we can detect and describe these innovations.

5

Understanding Scientific Publications

In Chapter 4, we discuss how scholars use topic models to understand non-fiction documents. This chapter focuses on a particular sub-genre of non-fiction: scientific documents. Scientific documents deserve their own chapter because these documents use very specialized vocabulary, they are the vehicles for innovation, and shape important policy decisions. We discuss each of these aspects in turn.

Specialized Vocabularies Define Fields of Study First, scientific documents are unique because unlike general documents, their vocabulary is precise and carefully measured. "Resistance", "splice", "utilization", and "demand" are common words with radically different meanings when used in specialized, technical contexts. Their use is a marker for membership in a specific discipline (Figure 5.1). Thus, the ability of topic models to capture patterns of word usage also captures community and affiliation; this goes well beyond the thematic organization of topic models described in previous chapters.

Scientific Documents Innovate Not every scientific publication is innovative; in fact, most are not. However, some scientific publications are

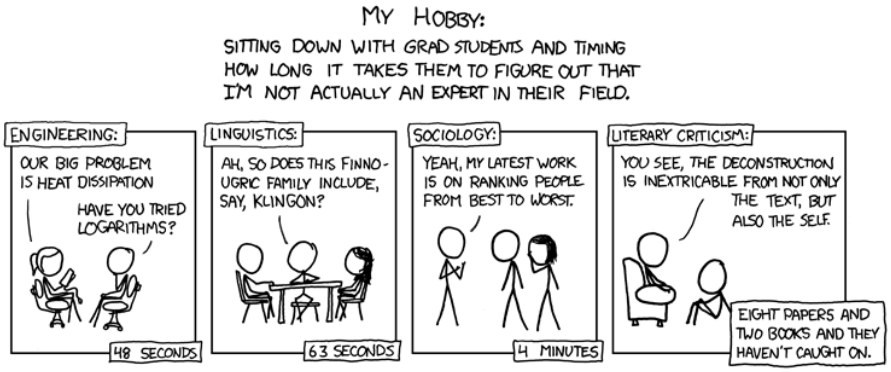

Figure 5.1: Using the appropriate language is a prerequisite for being part of a field (but not sufficient). Topic models use this to automatically discover fields of study. Source: `xkcd.com`

Earth-shaking. Such developments might be theoretical, methodological, or empirical. Physics was revolutionized by relativity. Genetics was revolutionized by the discovery of polymerase chain reaction methods. Geology was revolutionized by the discovery of evidence for plate tectonics in the form of magnetic traces in the ocean floor. Unlike the other domains we have discussed, scientific documents are not just *reports* of news or events; they *are the news*.

What makes the analysis of scientific document collections both challenging and interesting is that innovation is hard to detect and hard to attribute. Einstein's groundbreaking 1905 papers were not fully recognized until many years later; important ideas are often proposed by an obscure researcher but only accepted once popularized and supported by other research. Which document (or researcher) in this case was the true source of the innovation? As we will see in this chapter, topic models can help answer this question.

Science and Policy Understanding scientific publications is important for funding agencies, lawmakers, and the public. Government funding of science can create jobs, improve culture, and is an important form of international "soft power". However, knowing which research to fund is difficult, as the nature of science means that fields constantly change,

which precludes rigid classifications [Szostak, 2004]. One challenge of modeling scientific documents is modeling how fields change; the static models we have discussed thus far are not always appropriate.

5.1 Understanding Fields of Study

One of the first uses of topic models was to understand the "fields of science". Griffiths and Steyvers [2004] found that they were able to reconstruct the official Proceedings of the National Academy of Sciences (PNAS) topic codes automatically using topic models (Figure 5.2). This is a useful sanity check: yes, topic models correlate with what we often think of as scientific disciplines. They use distinct language for methods, subjects of study, and have different key players.

This work sought to find divisions between the fields of science, but from a retrospective point of view. In contrast, Talley et al. [2011] sought to map the funding priorities of the American National Institutes of Health (NIH) from within the organization.

The National Institutes of Health are America's premiere funding agency for biological and health research. The NIH consists of several institutes that focus on particular diseases, research techniques, or body systems; each of these institutes manages its own independent funding portfolio, sometimes making it difficult to understand the "big picture" of funding.

Talley et al. [2011] use topic models to help create this big picture, in contrast to more labor-intensive techniques (e.g., keywords from a meticulously organized ontology). Their analysis discovered unexpected overlaps in research priorities across institutes. For example, many institutes study angiogenesis, the formation of new blood vessels; as a treatment for cancer, in heart imaging, the molecular basis of angiogenesis in the eye, and how angiogenesis might signal complications in diabetes.

In practice, applying topic modeling to the NIH grant abstracts collection was not straightforward. Creating models that were acceptable to users accustomed to manually applied keywords required extensive curation of the vocabulary. Topic models try to find topics to explain

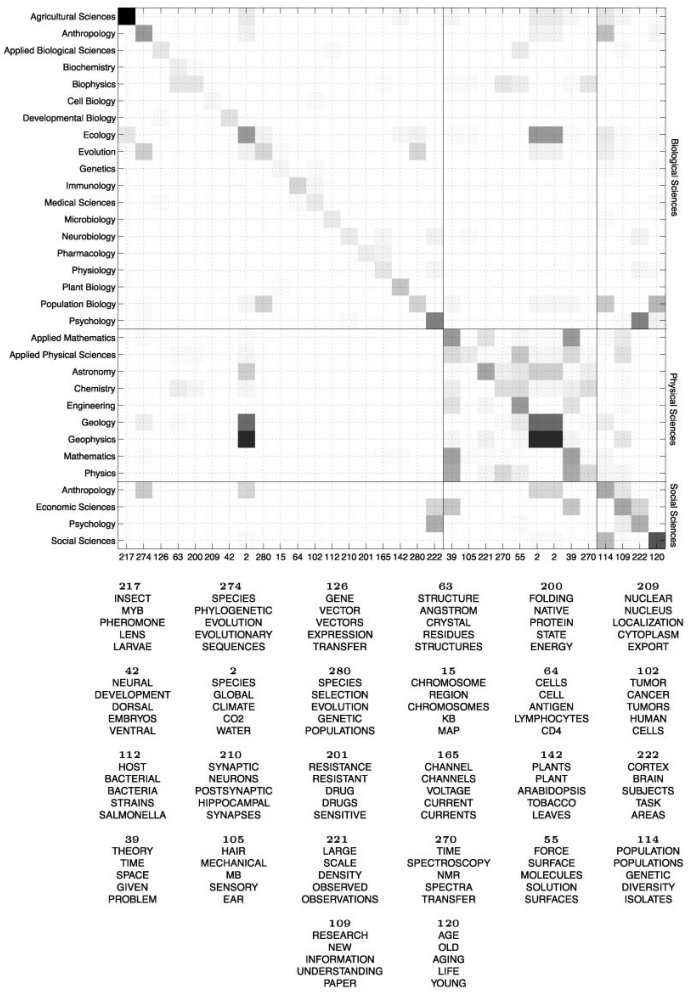

Figure 5.2: After running a topic model on PNAS, Griffiths and Steyvers [2004] found topics (x-axis) that could recreate the manually defined fields of study covered by PNAS (y-axis).

all aspects of a training corpus. A collection of research proposals will have significant discourse related to non-research themes, represented by words such as "propose," "support," and "funding". Large numbers of corpus-specific stopwords were identified for removal, mainly associated with the non-research topics. In addition, Talley et al. [2011] found that preprocessing documents to combine an extensive list of multi-word terms into single tokens made a substantial difference. Scientific and other technical vocabulary often forms non-compositional compound terms because of the need for specificity. As an example, "amino acid" contains the word "acid", but amino acids have no functional similarity to "fatty acid" or "hydrochloric acid". Combining such terms into single-token compounds resulted in substantially improved specificity and comprehensibility in topics.

5.2 How Fields Change

One way that science is distinct from the fields discussed in the previous chapters in that scientists see themselves as building a single coherent structure of knowledge. Each paper in its own way stands on the shoulders of giants. Topic models for science thus need to be aware of the connections between documents across time. Another way that science is different is that the documents themselves introduce new ideas (we discuss detecting these innovative ideas in the next section).

One of the first techniques to *model* topic change viewed topics as subtly changing each year with a dynamic topic model [Blei and Lafferty, 2006, DTM]. Each topic has a distinct distribution over words for each time period. For example, the probability that the physics topic emits the word "string" in 1910 might be low, but after 1970 much higher. Of course, we do not want the topics to be completely different every year—we want topics to change, but not *too much*.

The DTM views topics as changing through *Brownian motion*: the topic at year t is drawn from a Gaussian distribution with mean at the topic for year $t-1$ (a separate variance parameter controls how much topics can vary each year). At this point, you may object given our discussion of distributions from Chapter 1.3.1: Gaussians produce

continuous observations that might be negative or greater than 1.0, while topics are multinomial distributions over discrete outcomes.

To move from Gaussian draws from $\vec{x} \in \mathbb{R}^d$ to a discrete distributions over d outcomes, Blei and Lafferty [2006] use the logistic normal form to create a multinomial distribution

$$p(z = k \mid \vec{x}) = \frac{\exp x_k}{\sum_i \exp x_i}, \tag{5.1}$$

rather than drawing the discrete distribution from a Dirichlet distribution (c.f. Equation 1.3). This greatly complicates inference, but allows the topics to change gradually from year to year.

With this model, the DTM discovers how fields change over time. At the start of the twentieth century, the language of physics focused on understanding how the "æther" propagates waves and the fundamental forces; by midcentury, understanding "quantum" effects took precedence; by the end of the century, experimental physics with large particle accelerators lead the search for ever more exotic members of the subatomic menagerie. While the final topic is nearly unrecognizable given the first, they all are clearly physics; the modeling assumptions of the DTM capture these nearly imperceptible changes in each year.

The flipping of a calendar page does not rule science, however; changes can happen at any time. Wang et al. [2008] captures changes in topics in continuous time; each document gets its "own" view of a topic that can change slightly from the previous version of a topic. This can help capture sudden changes in scientific topics, e.g., from an innovative contribution.

An alternative view of innovation is one directed by authors. Steyvers et al. [2004], Rosen-Zvi et al. [2004] build a generative model that includes the identity of authors. An author has a collection of topics that they write about and each document is a combination of the topics that its set of authors care about. Zhou et al. [2006] extend this argument through modeling the use of a topic based on a Markov transition from other topics. They then use this model to discover the authors that drive those transitions.

5.3 Innovation

The changes to fields happen because of *innovation*. Scientists develop new techniques, new terminologies, and new understanding of the world. These concepts require new words which are reflected in their scientific publications. Unlike other fields, where documents merely report the changing world, scientific documents are themselves the force that can change the world: from Darwin's *Origin of Species* to Einstein's papers on relativity.

Can we find where this change is happening? From a modeling perspective, we thus need models that can also change. Unlike the approach described in Chapter 4.3, which focused on *static* topics, here we focus on *dynamic* topics that can change and *who* is responsible for the change.

From a historical perspective, we might want to know who introduced groundbreaking research first. Measuring the context of innovations may also be useful for policy makers [Largent and Lane, 2012]. Rather than awaiting "magic" or serendipitous findings, we might want to measure how conditions, teams, and forms of research collaboration lead to breakthroughs. Better predictive models could then help direct new initiatives to recreate the settings that lead to important findings.

From a topic perspective, assessing impact amounts to detecting *who* was responsible for changing topics. Mann et al. [2006] find highly cited papers within the context of individual topics. This approach helps to contextualize impact relative to sub-domains: a massively influential paper in mathematics may have the same number of citations as a moderately successful paper in molecular biology simply because the latter field is much larger. They also search for papers that have topically diverse impact by measuring the topic entropy of papers that cite a given paper. These broadly impactful papers tend to be methods and tools. From an institutional perspective, Ramage et al. [2010b] take a *post hoc* perspective: after fitting a standard LDA topic model, find the distribution over research topics in the entire research community at time t and find the places in the past whose research most resembles the present. They hypothesize that these institutions "lead" other institutions to adopt their ideas.

Citations are a useful but incomplete guide to solving this problem. Dietz et al. [2007] develop a "copycat" model that uses the citation network to model the use of language across cited documents: if you cite a paper, you are likely to reuse some of the language in the original paper. He et al. [2009] extend this idea to more complex graphs. However, this assumes a perfect citation graph, which isn't always available for some fields.

Capturing more nuanced effects at either the individual or lab level requires refining the model. Gerrish and Blei [2010] adapt the random walk model of Wang et al. [2008] (Chapter 5.2) for scientific change. Instead of topic randomly wandering into new concepts, Gerrish and Blei [2010] propose that innovative articles "nudge" topics to look adopt the word usage of those innovative documents. This model is called the "dynamic influence model" (DIM). The assumption is that concepts and ideas are represented by language. If we can identify changes in word usage, this change implies that there were underlying changes in concepts.

For example, the Penn Treebank [Marcus et al., 1993] revolutionized natural language process and helped enable the statistical revolution in computational linguistics. Among its many effects is that people started using the word "treebank" much more than they had in the past. DIM captures this by explicitly modeling the influence $l_{k,d}$ of a document d in topic k.

Documents that do not make a splash have no measurable influence, while influential documents are absorbed by other scientists, who adopt the influential ideas and, critically, their language. Most documents will not move topics at all so it is reasonable to assume that $l_{d,k}$ is zero for most documents. However, influential documents will change a topic.

A topic is changed by an influential document by making the topic's distribution over words look more like the words in the influential document. For example, the article introducing the Penn Treebank uses "treebank" much more than "potato", so the topic will have a higher probability for "treebank" after incorporating a document's influence.

This incorporation happens similar to the drift of the dynamic topic model (DTM, Chapter 5.2). Instead of drifting randomly, the *direction*

of topic change is based on the words used in influential documents and the *magnitude* of the drift is how influential a document is.

Each of these terms are random variables; inference in the model discovers the settings of the random variables that best explain the data. The DIM's estimates of influence correlate well with the number of citations an article gets (the traditional measure of influence). Unlike citations, however, the DIM can be used in more informal settings to detect influential documents: for example, when a blog or a letter introduces influential ideas.

5.4 Summary

Understanding science communication allows us to see how our understanding of nature, technology, and engineering have advanced over the years. Ostensibly, topic models can capture how these fields have changed and have gained additional knowledge with each new discovery. As the scientific enterprise becomes more distributed and faster moving, these tools are important for scientists hoping to understand trends and development and for policy makers who seek to guide innovation.

In contrast to scientific trends, the next chapter looks at less literal word usage. Unlike science, fiction and literature use words and phrasing to reveal emotion and mood. However, just like with science, researchers can use topic models to reveal patterns of how words are used that reflect artistic and literary trends.

6

Fiction and Literature

This chapter considers documents that are valued not just for their information content but for their artistic expression. There are many ways to read fiction, poetry, and rhetoric. How we choose to read affects the conclusions we are able to make. Scholars have traditionally focused on a "close reading" approach, in which the goal is to identify the specific features of a passage that convey a more general meaning, or emotion, or atmosphere. These features might include nuances of word selection, echoes of sound through rhyme or alliteration, or prosodic features like rhythm or cadence.

6.1 Topic Models in the Humanities

While close reading is a foundational tool in the study of literature, it is necessarily limited by its scale. We value literature because it is one of the best ways to capture the spirit of an age, and the experiences of those who lived through it. But standard close reading methods require narrow focus and thorough interpretation. Topic models complement close reading in two ways, as a survey method and as a means for tracing and comparing large-scale patterns.

The survey method is relatively simple, linking passages that a reader may not have known about. Close reading is the best way to analyze a short passage of text, but which short passages of text do we want to analyze? Because no one can read—much less close read—all the available material from a culture or time period, scholars are often left trying to make large-scale arguments about the history of literature from small-scale evidence. And this small-scale exploration is not randomly selected: the same small canon is studied in detail while the vast proportion remains the "great unread" [Moretti, 2000]: works that are *never* studied. Identifying broad themes and then mapping those themes to their realization in different contexts may reveal works or sections of works that are "hiding in plain sight", unknown to modern scholarship through obscurity.

An alternative, and less traditional, mode of analysis is often called "distant reading" [Moretti, 2013a]. This approach uses computer-assisted methodologies. Topic modeling has emerged as a central tool in distant reading, as a way to organize our reading of large scale patterns [Blei, 2012].

Topic analysis, viewed as a way of identifying repetitions of language or discourse through multiple works, resonates with many more familiar approaches to the study of literature. At the broadest scale, to define a genre or a literary period is to separate a corpus into sections based on some observable criterion. We posit a "gothic" literature characterized by atmospheric descriptions of castles, or a "cyberpunk" literature characterized by conflicted relationships with information technology. At a smaller scale, themes or tropes reappear in different contexts. At the most detailed level, scholars identify repeated phrases, such as the descriptive epithets used in Homeric oral poetry.

Statistical topic analysis has a similar goal, but pursues it through different means. Rather than rigid boundaries specified by date of publication or nationality, algorithms identify genre through the repeated words that form the traces of those themes. Topics do not represent themes themselves, but rather identify the implicit statistical regularities in word use brought about by the presence of genres, themes, and discourses.

Applying topic models to fiction, however, brings new challenges. Jockers [2013] trains a 500-topic model on a corpus of 4000 English-language novels. Several issues emerge from this corpus. These are present in other contexts, but they are much more readily apparent in fiction.

6.2 What is a Document?

In most literature about topic models, the term "document" is used on the implicit assumption that users have things called documents. In the canonical LDA journal article [Blei et al., 2003], this word is used 143 times, but never defined. The meaning of a "document" is often fairly clear: a news article, or a scientific abstract. What was not clear in this earlier work was that this definition can be problematic, especially for documents longer than a few pages of text.

Treating novels as a single bag of words, for example, does not work. Topics resulting from this corpus treatment are overly vague and lack thematic coherence. We should not be surprised by this finding. The assumption of a topic model is that the concentration of topics over a document is fixed and unchanging from the beginning of a document to the end. Natural writing rarely fits the topic model assumption, and a novel that had no thematic variation over its entire length is unlikely to have been published.

We need to find a good segmentation into shorter contexts (in contrast to social media, which often needs to be combined into longer documents, c.f. Chapter 7.5.1). We assume that themes are expressed in different sections of a long document like a novel. If a segmentation does a good job of identifying the boundaries between these sections, each resulting segment should have relatively few themes. If a segmentation does not do a good job of identifying boundaries, we should see segments that contain more themes on average, because our segments combine fragments of multiple thematic segments.

Jockers [2013] chooses to avoid relying on structural markers such as chapter divisions and divides novels into 1000-word chunks. This treat-

ment results in coherent, tightly focused topics that can be reasonably used as proxies for recognizable themes.

Although fixed-length segmentation is effective, it is not necessarily ideal. Algee-Hewitt et al. [2015] compare varying fixed-length segmentations to segmentation based on paragraphs. They evaluate the difference between treatments by measuring the concentration of topics in each segment of text after modeling. The Herfindahl index is a measure of concentration in discrete probability distributions, calculated as the sum of the squared probabilities of each possible value:

$$\text{Herfindahl}(P) = \sum_x P(x)^2. \tag{6.1}$$

For example, consider two distributions P and Q over a set of symbols $\{a, b, c, d, e\}$. If P has non-zero probability only on a single symbol $P(a) = 1.0$ and zero probability for all other symbols, the Herfindahl index of P will be 1.0. If Q has uniform probability on all five symbols $Q(a) = Q(b) = ... = Q(e) = 0.2$, the Herfindahl index will be $5 \cdot \frac{1}{5} \cdot \frac{1}{5} = 0.2$.

When a corpus of 19^{th}-century novels is divided by paragraphs, the Herfindahl index over concentration of topics within each segment is consistently larger than the same index calculated when the same corpus is divided into evenly sized 200-word slices, implying that the distribution over topics for each segment is more focused on a smaller number of topics. Setting the slice size to the average length of paragraphs in the corpus, eighty-two words, increases the Herfindahl concentration metric, but the resulting value is still smaller than the value based on paragraphs. This result is reassuring, in that it suggests that paragraphs do indeed have some consistent meaning, at least in this collection of 19^{th}-century novels.

6.3 People and Places

Because most works of fiction are set in imaginary worlds that do not exist outside the work itself, they have words such as character names that are extremely frequent locally but never occur elsewhere. This word co-occurrence pattern is problematic for topic models because

they can be thought of as machines for finding groups of words that occur frequently together and not in other contexts. Character names are—by that criterion—a perfect topic. Modeling these documents can result in topics that are essentially lists of character names.

As an example, consider a model with fifty topics of fourteen novels by Charles Dickens and its top words from a selection of topics (Table 6.1). Upper-case letters are not reduced to lower-case to emphasize the presence of proper names. Several topics are dominated by capitalized names, with individual novels clearly identifiable: Topic 4 is *Oliver Twist*, Topic 5 is *Nicholas Nickleby*, Topic 6 is *The Pickwick Papers* and Topic 7 is *A Tale of Two Cities*. In fact, exactly half of the distinct words in the top 20 words for all topics are capitalized, and almost all of these are proper names.

Focusing on characters is not always uninformative, and can in some cases highlight structure within works. Topics 1–3 all refer primarily to *Bleak House* (with the exception of *Scrooge*), but focus on different interlocking subplots. The first focuses on Lady Dedlock, the second on Mr. Jarndyce and his two wards, Richard and Ada, and the third on the investigations of the detective Mr. Bucket. The plot centers around the revelation of the connections between these apparently unrelated groups.

Jockers [2013] approaches this problem by constructing a stopword list that removes all character names before modeling. There are many ways to construct such lists. Lists of common names are a good start, but may not be aligned with a specific corpus. Some languages mark proper names with orthographic conventions like capitalization, but these tend to be noisy. A useful heuristic in English is to identify terms that appear capitalized in more than 90% of instances. Even then, names that are also common words, such as *daisy* and the aforementioned Mr. Bucket, or words that appear capitalized for other reasons, such as *god*, may lead to unintended results. Furthermore, some languages do not differentiate letter cases (Hebrew, Korean) and others use it for other purposes (all nouns in German). Named-entity recognition tools scan text for patterns of language that indicate personal names, and may result in greater precision than simpler methods. Nevertheless, there is

Table 6.1: Sample topics from Charles Dickens novels, without removal of character names (ordered manually).

Topic	Terms
1	Lady Leicester Scrooge Dedlock Rouncewell ladyship Wold Chesney Ghost Volumnia Christmas Tulkinghorn family Spirit Baronet nephew Rosa Scrooge's housekeeper Lady's
2	Richard Jarndyce guardian Ada Charley Caddy dear Skimpole Miss Summerson Esther Jellyby miss Vholes Kenge Woodcourt quite myself Guppy Chancery
3	says George Bucket Snagsby Guppy returns Smallweed Bagnet comes Tulkinghorn looks takes trooper does makes friend goes asks cries Chadband
4	Oliver replied Bumble Sikes Jew Fagin boy girl Rose Brownlow dear gentleman Monks Noah doctor Giles Dodger lady Nancy Bill
5	Nicholas Nickleby Ralph Kate Newman replied Tim Mulberry Mantalini Creevy brother N oggs Madame Gride Linkinwater Smike Arthur rejoined Wititterly Ned
6	Pickwick Winkle replied Tupman Wardle gentleman Snodgrass Pickwick's Perker fat boy Bardell dear Jingle inquired Fogg Dodson friends friend lady
7	Lorry Defarge Doctor Manette Pross Carton Darnay Madame Lucie Monseigneur Cruncher Jerry Stryver prisoner Charles Monsieur Tellson's Marquis father Paris
8	coach uncle gentleman lady box coachman gentlemen landlord get London guard inside horses waiter boys mail passengers large better hat
9	street door streets windows houses room window few iron walls wall rooms dark within shop doors corner small stood large
10	money letter paper business read pounds papers five hundred office thousand clerk paid years pen next law desk letters week

no known way to avoid careful consideration of the meaning of words in context.

Novels describe people and places, but they are also created by people (authors) who are influenced by their cultural setting. Jockers and Mimno [2013] perform a post-hoc analysis on Jockers' earlier 500-topic model to determine whether there is a connection between the use of specific topics and metadata variables such as author gender, author nationality, and year of publication. They find that the concentration of many topics is strongly correlated with author gender, and that these correlations are statistically significant. Such significance testing can be carried out by randomization and bootstrap tests. Both methods create "fake" corpora that are similar to the real corpus but different in specific ways. Randomization or permutation tests randomly shuffle the assignment of labels (such as author gender). If an observed correlation between a topic and an external variable is within the range of the correlations generated by randomly assigning documents to labels, there is little statistical evidence that that observed correlation is meaningful. Bootstrap tests preserve the relationship between documents and metadata variables, but resample documents with replacement. This test indicates whether a result depends on the presence or absence of a specific document. If there is wide variation between randomly generated corpora, the observed correlation may be the result of unusual outliers rather than a consistent pattern.

While the use of statistical hypothesis testing methods is valuable in the context of large-scale distant reading, a literary analysis is not—and should not be—like a clinical trial, there are differences between their use in a scholarly context and their use in more typical scientific studies. First, the presence of unusual outliers or singular examples can in fact be a positive result. The suggestion that a particularly work may be radically different from supposedly similar examples could be the beginning of a new perspective. At the very least, it can identify editing and curation issues. Second, a critical variable in an analysis of statistical significance is sample size. Unlike a designed experiment, this sample size is usually not within our control: we have the literature that we have. Finally, it is vitally important to avoid the impulse to

treat a significance score as a binary valid/invalid result. If numeric scores should be used at all, they should be presented as a "level of support" given the documents that are available. Humanists may also be fundamentally more comfortable with dubious hypotheses: an observed association with a 10% chance of being purely random could still be a very strong result.

As an example, Jockers and Mimno [2013] evaluate an intriguing hypothesis, that a topic about religious foundations (Convents and Abbeys) is used more by unknown authors[1] than by either (known) male or female authors. The conjecture is authors were choosing to remain anonymous to write about politically and religiously touchy subjects. This correlation, however, showed large variability under a bootstrap test, and indeed one of the supposedly anonymous works turned out to be an abridgment of an Anne Radcliffe novel. The pattern is still present without the effect of these works, but there is not a clear and undeniable association between anonymity and the questioning of religious authority.

Fiction is sometimes set in the context of real places. Tangherlini and Leonard [2013] look at nested models of sub-corpora within Danish literature in a way that highlights connections between real-world events and cultural movements and fictional echoes. Their method, which they describe as a topical "trawl line," uses a user-specified sub-corpus as a query and then searches the remainder of the corpus for works that match to that query. As examples, they find works influenced by the translation of Charles Darwin into Danish, works influenced by the "Modern Breakthrough", and works influenced by folklore and regional literature.

6.4 Beyond the Literal

One of the hallmarks of fiction and literature is the use of figurative language. It is not obvious that unintelligent machines with no cultural understanding would have any ability to process such metaphors. How-

[1] Authors unknown to modern scholarship, not authors publishing under known pseudonyms.

ever, Rhody [2012] demonstrates on a corpus of poetry that although topics do not represent symbolic meanings, they are a good way of detecting the concrete language associated with repeated metaphors.

Specifically, Rhody explores a corpus of 4600 poems, 276 of which describe works of art (*ekphrastic* poems). She trains a sixty-topic model, and highlights several particularly interesting topics. One of these topics places high probability on *night, light, moon, stars, day, dark, sun, sleep, sky, wind, time, eyes, star, darkness, bright.* The apparent meaning of the topic is clear, and well summarized by the single top word: night. But Rhody finds that when she explores the *context* of this topic, the poems are all using a consistent metaphor relating night and sleep to death. The concept of death does not appear in the top words—poets are not addressing the issue directly. Nevertheless, the model has identified an example of non-literal, figurative language even though, because it is grounded in the actual words, it has no ability to represent poets' deeper meaning. This is because the poets use a consistent "surface" language to represent a consistent metaphor. The metaphor is not detectable directly, but a poet's use of a metaphor has a signature that is observable.

Rhody highlights a second topic that provides an example of a different type of non-literal meaning. This topic places high probability on *death, life, heart, dead, long, world, blood, earth, man, soul, men, face, day, pain, die.* Unlike the previous topic, the topic directly references death and life, but it also lacks what Rhody calls the "unambiguous comprehensibility" of the *night* topic. But examining the context of poems that contain the topic reveals a different pattern. These poems have a consistent *form* that Rhody describes as elegiac. She writes that "Paul Laurence Dunbar's 'We Wear the Mask' never once mentions the word 'death', the discourse Dunbar draws from to describe the erasure of identity and the shackles of racial injustice are identified by the model as drawing heavily from language associated with death, loss, and internal turmoil—language which 'The Starry Night' indisputably also draws from".

6.5 Comparison to Stylometric Analysis

In addition to discussing what researchers have done in literary analysis with topic models, it is useful to consider how other technologies have been used in the same setting. One of the most established applications of computation in the study of literature is stylometry, or more specifically the question of authorship attribution [Juola, 2006]. It is illustrative to contrast the goals and methods of stylometry with those of topic modeling.

The critical insight of modern stylometry is that it is easy for authors to shift the focus of their work, but much more difficult to alter the semi-conscious style of their language [Mosteller and Wallace, 1964]. The implication is that content-bearing words, such as nouns and adjectives, are a relatively poor indicator of authorship or at least authorial style, while functional words, such as determiners, conjunctions, and prepositions, carry more information about authorship. Therefore, measures such as Burrows' delta [Burrows, 2002] restrict attention to the most frequent words in a corpus.

The contrast to topic modeling is clear: stylometric analysis focuses on frequent, low-information words and ignores content-bearing words, while topic modeling generally does the exact opposite. We generally remove high frequency words using a stop list, and in fiction go even further in removing words that are overly distinctive of a particular work. An assumption of topic modeling is therefore that the goal is to find thematic components that are *not* specific to one author, but rather repeat, with more or less variation, across multiple works. Where stylometry seeks to see past what authors are saying and focus on how they are saying it, a use of topic modeling is to find instances where different authors are writing about the same thing.

6.6 Operationalizing "Theme"

The use of topic modeling in the study of literature has been beneficial both for humanities scholars and for machine learning researchers. For scholars, these models offer the possibility of a more precise approach to concepts that have traditionally been vague and impressionistic, such as

theme, genre, and motif. At the same time, and somewhat paradoxically, literary documents present such a radically different mode of language than news articles or scientific publications that they lead us to question the apparent precision of statistical approaches.

Topic models provide a way of operationalizing the concept of distant reading. Moretti [2013b] defines this term as "Taking a concept, and transforming it into a series of operations". He attributes this definition to Bridgman [1927], who introduces the term in the context of measurement in physics: "To find the length of an object, we have to perform certain physical operations. The concept of length is therefore fixed when the operations by which length is measured are fixed: that is, the concept of length involves as much as and nothing more than the set of operations by which length is determined". While topic models are an imperfect tool for measuring theme in literature, they do provide a much more powerful approximation of theme than anything that we have had previously.

But applying statistical models to literature also brings forward a series of challenges that highlight the amount of human interpretive work that must go into successful topic modeling. Literary documents are of varied lengths, describe self-contained imaginary worlds, and are suffused with symbolic language. We can address these issues through corpus curation and through interpretive reading of models, but in doing so we must necessarily confront the fact that we are not applying Bridgman's fixed set of operations.

6.7 Summary

Topic models cannot by themselves study literature, but they are useful *tools* for scholars studying literature. Models provide a distinct perspective that can call our attention to connections across different parts of a corpus that might not be obvious from close reading. Literary concepts are complicated, but they often have surprisingly strong statistical signatures. Models can still be useful in identifying areas of potential interest, even if they don't "understand" what they are finding. At the same time, fiction presents a challenge to modeling practices because

each fictional work deliberately creates its own closed world—a world whose characters and settings are so vivid that they can overshadow more subtle connections across works. Addressing these challenges can serve as an invitation to think deeply about words and their contextual meanings.

Just as topic models provide a methodology for analyzing the creative, diverse œuvre of authors and the emotions and thoughts of fictional characters, topic models can also help us build insights of real people. Thanks to social media, we have a wealth of information about people sharing their thoughts and views online. Topic models can help us use these data to better capture emotion, beliefs, and relationships. The next chapter discusses how topic models can understand these messy, interesting properties of text.

7

Computational Social Science

While the previous chapters were mostly retrospective analyses, computational social science is mostly in the "here and now". The role of text analysis is to provide evidence for how people relate to each other and to their environment in particular contexts, for example social, political, or economic interactions. The specific expression of any particular document is usually of less importance. As a result, social science focuses on data being generated in the most recent hours, days, or weeks to inform intelligence analysts, brand monitors, journalists, or social scientists. The underlying problem is the same, however: these stakeholders are interested in what people have to say but cannot read all of the data at their disposal.

Historically, social science asks questions about opinions. What candidate is preferred in a particular part of the country? Do people like a new restaurant or product? These questions are often answered by polling: social scientists would head out into the world, gather a statistically significant survey sample, and extrapolate to the broader population.

These techniques remain foundational, but they take time. A company needs to know if it has an issue with a product immediately,

particularly if its good name is being dragged through the mud on social media [Bowen, 2016]. However, the reason for the acute time pressure can also be the solution: if a company is able to quickly see that it has a social media problem, it can more quickly intervene and correct the issue.

Traditional social science methods are labor intensive, take a long time, or are impossible for sensitive subjects. For instance, surveys of influenza take too long to be useful compared to the life cycle of influenza's progression [Broniatowski et al., 2015], and approval ratings may be too slow in the run-up to an election [O'Connor et al., 2010]. Using Twitter and Google searches results in more accurate information faster.

Directly communicating with some populations may be difficult. Using social media presents an alternative [Wang et al., 2015], as individuals share information more freely than official news agencies (which, for example, may suffer from official censorship in the case of opinions about pollution in China) or in school-administered surveys (which can suffer from self-censorship in the case of sexuality or drug use). Topic models and other large-data approaches that can look at vast quantities of text help overcome some of the obstacles to fast-response social science.

The observational nature of topic models is both a weakness and a strength. Analyzing documents through social media collection can induce threats to validity. Researchers cannot necessarily control the populations producing documents, the forum in which documents are written, or the subject of documents. At the same time, observational models have the advantage of increased potential for discovery. In a survey, researchers have to specify every question that will be asked. That control is good for ensuring validity, but risks missing whole categories of opinion that might not be obvious to researchers. Topic models can complement this type of carefully designed survey by unearthing issues or factors important to participants whether or not those issues were anticipated. Thus unsupervised models can help in identifying questions that researchers "forgot to ask".

Prediction and Interpretation A common theme in using topic models is whether models should prioritize *prediction* or *interpretation*. Different topic models privilege each of these approaches. The distinction between model applications mirrors, to some extent, the distinction between quantitative and qualitative social science. Predictive applications are closer to quantitative methodologies that focus on regression-based methods that have clear input and output variables. Interpretative applications are closer to qualitative methodologies that use human intuition to explain complex processes. At the same time, the use of topic models blurs the boundary between these two methodologies. Even when used to support qualitative work, topic models apply computation, and insulate researchers to an extent from pre-conceived biases (although they bring their own modeling assumptions). Similarly, even when used to support quantitative work, topic models enable researchers to establish quantitative relationships between document metadata variables and variables derived from messy, unstructured text that might otherwise not have supported quantitative analysis.

The previous chapters have focused on interpretation: can a user understand the output of a model? But for supervised models, there is a question of how well the model can predict some parameter of interest, such as sentiment or user engagement.

To some extent, these are not always in conflict. Ramage et al. [2010a] show that topic model features can improve tweet categorization, as do Blei and McAuliffe [2007] for supervised LDA. However, changing the objective function can further improve predictions [Zhu et al., 2009].

However, sometimes improved interpretability (Chapter 3.4) hampers the ability of the model to predict content. This is true of both words within a document and document labels. Chang et al. [2009] showed that complicated topic models do a better job of predicting held-out documents but make less sense to a user. Nguyen et al. [2015a] show that supervised models offer better predictions with additional topics but the topics are less interpretable.

7.1 Topic Models for Qualitative Analysis

A common task in qualitative social science is to develop high-level theories that explain social processes based on low-level observations, such as field reports or ethnographic notes. One way to operationalize this process is the grounded theory method [Glaser and Strauss, 1967]. Grounded theory describes a process for iteratively developing theories through repeated reading of source material. Baumer et al. [2017] compare a manual grounded theory analysis and a topic model-based analysis of survey response text describing people's experiences attempting to voluntarily leave Facebook.

They find that there are both theoretical and empirical connections between these two approaches. At the theoretical level, both grounded theory and probabilistic topic model algorithms are iterative, beginning with rough, low-quality models/theories and refining them through repeated passes through the documents. Both methods also seek to maintain a close connection between the abstract representation and the original data set: in Gibbs sampling, topics are "grounded" in specific word tokens. At the empirical level, Baumer et al. [2017] find close connections between the themes discovered by researchers manually applying grounded theory procedures and topics discovered by an LDA model. But this result should not imply that human analysis is of no additional value. They report that the thematic meaning of LDA topics was not immediately apparent from simple lists of high-frequency words. Rather, the model was most useful as a way of suggesting a topic-specific "reading list." The "meaning" of topics was only clear at the theory level by browsing the documents that had unusually high representation of that topic. As such the topic model can best be thought of as a tool for applying grounded theory more efficiently and with greater insulation from human biases.

7.2 Sentiment Analysis

A useful way to think of the application of topic models in quantitative social science is as a means of deriving a numeric variable from text. Specific topics in a document can stand in for themes that may not be

otherwise easily measurable. These inferred topic variables can then be added to standard statistical methods to find connections between topics and non-textual variables. As a motivating example, we first consider sentiment analysis [Pang and Lee, 2008]. Here, the goal is to determine the "sentiment"—e.g., positive or negative opinions—associated with a piece of text. For example, "Chipotle is great!" would be associated with positive sentiment, while "Chipotle made me sick" would be associated with negative sentiment.

While commercial applications of sentiment analysis are mostly for identifying whether people like a product or company, there are wider social science applications of examining large corpora to determine authors' *internal state*. For example, political scientists may want to classify social media users as liberal or conservative based on their online commentary.

Topic models can help these tasks by dividing a problem into topics. The contextual disambiguation provided by topics can be useful in narrowing the range of applicable subjects. For example, "Apple" can appear in tech news as well as a food ingredient; someone monitoring the seller of iPods and iPhones would not want to be confused by social media commentary complaining about the low quality of a Red Delicious. Contextual disambiguation can also be useful because the implied sentiment of words can vary considerably between domains. The term "surprising" could be positive in a book review, but strongly negative in an automobile review.

Topic-based sentiment analysis can, however, be problematic. The objective of a topic model is to identify and separate the latent factors that best explain the way combinations of words appear together in documents. Sentiment can be subtle, and may not present the strongest signal to an algorithm. For example, even though restaurant reviews are specifically intended to be sentiment-bearing, topic models trained on restaurant reviews mostly align with *types* of restaurants, producing topics representing cuisines, such as Chinese, Mexican, and Thai. Topic models also lose their value if you want to *contrast* sentiment within a topic. While a topic model can find people discussing Chipotle burritos online, it cannot separate the lovers from haters. Thus, *distinguishing*

topics based on their sentiment can help a user better understand how topics and sentiment interact in a dataset. This requires modifying the topic model to make it aware of the underlying sentiment.

7.3 Upstream and Downstream Models

To distinguish topics based on their sentiment, the model must be aware of a non-textual variable that represents sentiment. In the language of probabilistic models, sentiment and topic are modeled *jointly*. That is, there is for each document a probability distribution over both the sentiment variable y and the topic assignments z.

There are two general kinds of joint models that incorporate metadata such as sentiment: upstream and downstream models. The distinction is based on the generative story of topic models (Chapter 1): is sentiment before (upstream) or after (downstream) topics in the generative story?

Upstream models assume that external variables such as sentiment come first in the generative story. That is, there will be different topics given the underlying sentiment. This can come in the form a hard-coded distribution [Mei et al., 2007a], a prior learned from observed sentiment [Mimno and McCallum, 2008], or from a latent variable that can serve as a proxy for sentiment [Lin and He, 2009]. Upstream models are often easier to implement and are more flexible [Roberts et al., 2014] because they do not need to specify a generative distribution that matches the form of the variable.

In contrast, downstream models explicitly predict variables such as sentiment *given* text. If the goal is to later predict sentiment given raw text with the help of topic models, downstream models can work better than upstream models. These models are often called "supervised" topic models after supervised LDA [Blei and McAuliffe, 2007, SLDA], which use a document's topics to predict the downstream sentiment using regression: a document's sentiment y_d is assumed to come from a Gaussian distribution with mean $\eta^\top \bar{z}$, where $\bar{z}$ is a normalized vector of all of the topics that a document uses and η is a regression parameter that describes the sentiment of each topic.

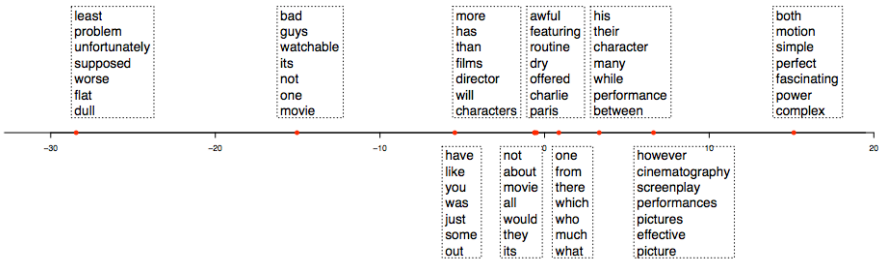

Figure 7.1: Example topics learned by supervised LDA from Blei and McAuliffe [2007]. Each topic is not just a collection of words but also has a regression score η that explains whether it is associated with positive sentiment (right) or negative sentiment (left).

During inference, the words and sentiments work together to find combinations of topic and sentiment that make sense. While "vanilla" topic models seek to find clusters of words that make sense together, if a topic is associated with documents that have many different sentiment values, it will have to learn a less focused distribution over sentiment scores, resulting in lower probability.

Consider Figure 7.1. If a topic has an inconsistent sentiment value (for example, a negative sentiment document in a positive sentiment topic), inference will try to move the negative sentiment documents to topics with consistent sentiment η **and** consistent words.

These models form the foundation for the models and problems we discuss in the rest of this section.

7.4 Understanding Stance and Polarization

Another form of internal state is *stance*: which side does a person take on an issue. This can take many forms: are you for or against a proposal, are you a Democrat or a Republican, or are you a fan of the original Star Trek or the new version?

Upstream models can discover these sides by incorporating stance into the generative model. For example, several authors—Zhai et al. [2004], Lu and Zhai [2008], and Paul and Girju [2010]—develop topic

models that allow readers to compare aspects of a topic. They posit that each comparative "side" has a distribution over words that it uses generally *and* that each side had its own take on how it discusses a topic. Within a document, each word is chosen either from a side's background distribution, a side's version of the topic, or from the topic's "neutral" words. For instance, Israelis and Palestinians both use "attacks", "civilians", and "military" in discussing unrest in Israeli-occupied Palestine, but the Israeli side uses "terrorist" and "incitement", while the Palestinian side focuses on "resistance" and "occupation".

The interaction between sentiment and aspect is unclear. Some aspects are independent of sentiment, and other aspects are particularly charged. Jo and Oh [2011] develop a model that first draws sentiment distributions that constrain the aspects discussed in a document.

Downstream models can also capture these divisions as well. Nguyen et al. [2013] predict whether a speaker is Republican or Democrat[1] based on the versions of topics they discuss, extending the non-predictive model of Grimmer [2010]. For example, Republicans are more likely to discuss taxes than Democrats, but Democrats focus on the good that comes out of taxes (Figure 7.2).

However, there are not always two sides to an issue. A probabilistic solution to this model is the nested Dirichlet process [Blei et al., 2010]. These hierarchies induce a non parametric hierarchy over an unbounded number of topics. This corresponds to agenda setting from political science [Nguyen et al., 2015b].

7.5 Social Networks and Media

We have talked about metadata that are independent for each user. Sometimes, however, we are interested in metadata that describe the relationships *between* documents: which users follow each other on

[1]In downstream models, which variables to use in the prediction is often up for debate. Using lexical terms [Titov and McDonald, 2008, Zhao et al., 2010] in a log-linear model typically works better: it is able to capture word-specific nuances of sentiment and model situational sentiment (e.g., "unpredictable" is good for a book but bad for a car's steering). However, it leads to a more complicated, less interpretable model.

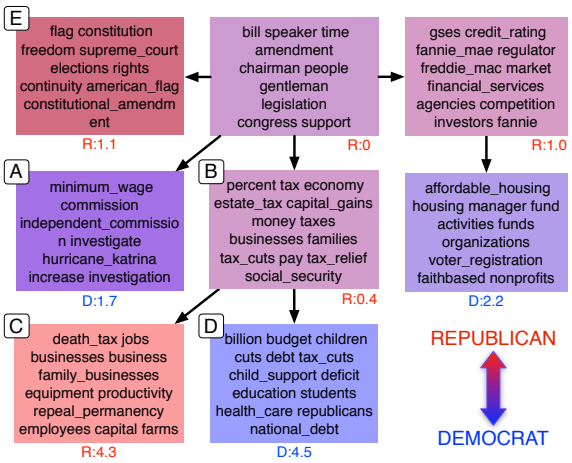

Figure 7.2: Topics discovered from Congressional floor debates using a downstream model to capture speaker's ideology. Many first-level topics are bipartisan (purple), while lower level topics are associated with specific ideologies (Democrats blue, Republicans red). For example, the "tax" topic (B) is bipartisan, but its Democratic-leaning child (D) focuses on social goals supported by taxes ("children", "education", "health care"), while its Republican-leaning child (C) focuses on business implications ("death tax", "jobs", "businesses"). The number below each topic denotes the magnitude of a learned regression parameter associated with that topic. Colors and the numbers beneath each topic show the regression parameter η associated with the topic. From Nguyen et al. [2013].

Twitter, which scientific papers cite each other, or which webpages link to each other. This makes modeling more difficult, but we still see the same division between upstream and downstream models: upstream models assume that the communities form before we see words, while downstream models use the words to explain which links we see.

The stochastic block model [Holland et al., 1983] and its mixed-membership descendant [Airoldi et al., 2008] are prototypes for upstream models. They posits that there are intrinsic groups of documents and links are more likely inside the group than outside the group. These groups are analogous to the topics in topic models, except that the links are "shared" between documents.

However, the first probabilistic models of network structure ignored the words in documents. Because the network structure is tied to author identity, it is natural to combine author identity with an upstream model McCallum et al. [2007], Liu et al. [2009], conditioning topics on authors and the communities the authors belong to.

Link LDA is the exemplar for downstream models [Nallapati and Cohen, 2008] and include the text in the documents. It uses a regression on the topic allocations (θ) rather than topic assignments (z), in contrast to supervised LDA above. Similarly, Cha and Cho [2012] use followed users to model downstream documents.

Conditioning on the topic assignments can improve the algorithm's ability to predict links on held-out documents, however [Chang and Blei, 2009]. This is because a regression based on the allocations alone can use topics to explain links that aren't in the document. For example, if the model thinks there's a link between documents because they both use Topic 14 but no words in the document are assigned ($z_n = 14$), then the model is unable to recreate this prediction in a held-out document.

Not all topic models applied to social network attempt to predict links. Weng et al. [2010] use the network structure of Twitter to find who is influential within a topic, and several models use links to constrain documents that are linked together to be similar [Mei et al., 2008, Sun et al., 2009, Daumé III, 2009].

In addition to explicit links in social networks, social media is also shaped by implicit links between people in similar contexts—events, cultural, or regional patterns that affect how people talk and what they talk about. Mei et al. [2006] capture how topics vary across region and time (e.g., when a hurricane strikes a region, those closest to the eye of the storm will talk about it with greater volume and more specifically). Later work builds models location and topic jointly [Yin et al., 2011].

In constrast, Eisenstein [2017] focuses on lexical variation, capturing how social media neologisms like "af" (a post-position intensifier, particularly of adjectives: e.g., "the description was evasive af") spread from twin epicenters in southern California and Atlanta to the whole of the US.

7.5.1 Peculiarities of Social Media

Hong and Davison [2010] discuss how the short documents of social media platforms like Twitter can confuse topic modeling algorithms, and Zhao et al. [2011] expand on the analysis, showing topical differences (e.g., Twitter often briefly follows fleeting topics passionately). Because a document is limited to 140 character, the admixture assumptions of topic models are limited. To capture trends over time or across users, algorithms must also know connections between users or group messages together over time [Mehrotra et al., 2013]. Other researchers develop models with specific sparsity properties [Lin et al., 2014] to accomodate the peculiarities of Twitter.

7.6 Summary

Computational social science can unlock the emotion and hidden factions often present in online discussions. This is useful for companies trying to understand their customers, for politicians trying to target voters, for first responders reacting to a disaster [Kireyev et al., 2009], and for academics trying to understand how online communication morphs social norms.

However, as social networks increasingly span the entire globe, assuming that a topic model is only in a single language is often a poor assumption. Indeed, even within a single country, topic models can discover regional variation [Eisenstein et al., 2010]. In the next chapter, we discuss how to cope with multilingual datasets and still discover coherent, language-independent topics.

8

Multilingual Data and Machine Translation

So far, we have been focusing on monolingual topic models and their applications. But many collections contain documents in more than one language. In practice, we often discover this phenomenon unexpectedly after running an initial monolingual topic model: topic models turn out to be very good at language identification. This behavior makes sense because the model is looking for groups of words that appear frequently together but not in other contexts, and separate languages have this property. In some cases we may choose to filter out small numbers of documents in other languages, but we would like to take advantage of connections across many languages.

Multilingual topic models have been developed to analyze and understand a corpus in multiple languages. Vulić et al. [2015] provide a good overview. The applications in multi-language corpora can be divided into two categories. The first, and simpler category is those that align languages at the topical level, but not at the level of individual word types. These models are useful for organizing corpora, but make no attempt to support analysis for users unfamiliar with any particular language. The second category is those that explicitly model word-level

alignments across languages. These models support applications in statistical machine translation (SMT).

As one of the most frequent applications for multilingual topic models, SMT tries to find a sequence of words in one language which match the meaning of a text input in another language. While the training data of SMT requires explicit aligned sentences in different languages, multilingual topic models relax this data restriction and are flexible to explore only loosely aligned data. In this chapter, we first discuss how topic models are adapted to use multiple languages, and then show how these multilingual topic models can help SMT.

Before discussing specific methods, it is useful to define terms related to data sources. The most salient feature for multilingual corpora is their degree of alignment. Parallel corpora are the most closely aligned. These collections comprise subsets of documents such that each set contains documents in different languages that have the same semantic content (up to the limits of translation). Common examples include translations of literary works or translated government documents, where a transcript of a speech in French is accompanied by a transcript of the same speech in German, with as little semantic difference as possible. Comparable corpora are less closely aligned. These collections also contain subsets of documents, but each set is only constrained to be *topically* similar, and not necessarily a direct translation. A common example is articles in Wikipedia. The articles for the French city of Lille in English and French Wikipedia are referring to the same place and contain much of the same information, but the French version is considerably longer. Mixed corpora are the least aligned. These collections simply contain documents in more than one language, but there is not necessarily any connection between any one document in one language and a document in another language. An example might be a journal that publishes in English, French, German, and Italian. No article is a translation of any other article. There are likely to be topical overlaps between articles, but there are not necessarily any structural indications of such relationships. A last category of useful data, not necessarily in the form of documents, is a bilingual lexicon that maps words in one language to words in another. Lexicons of this form can be considered to be

examples of parallel corpora with single-token documents, but it is often useful to treat them specially.

8.1 Document-level Alignment from Multilingual Corpora

In a case where a user is browsing a multilingual collection with only monolingual knowledge to find relevant documents, multilingual topic models can help. Such collections contain multiple languages, but does not necessarily have the exact matching or translations on words and sentences. Only a coarse document alignment is necessary, as long as the documents discuss the same topics, e.g., Wikipedia articles in different languages. Such connection between languages is also helpful to infer more robust topics, since different languages can complement each other to reduce ambiguity.

This approach pre-dates probabilistic topic models. Landauer and Littman [1990] connect aligned documents in different languages by projecting both documents to a shared latent semantic indexing space.

Similarly, bilingual topic models [Zhao and Xing, 2006, De Smet and Moens, 2009] and—more specifically—polylingual Latent Dirichlet Allocation [Mimno et al., 2009, pLDA] assume that the aligned documents in different languages share the same topic distribution and each language has a unique topic distribution over its word types. Thus the generative process of polylingual topic model is as follows: given a document pair (d_{l_1}, d_{l_2}), we first sample a document-topic distribution θ_d; for a document d_{l_i} in language l_i, we then sample a topic z_{dn} from θ_d, and generate a word from topic ϕ_{z_{dn},l_i} in language l_i.

Topic models trained from document-level alignments have applications in exploratory data analysis and in information retrieval [Vulić et al., 2013]. Mimno et al. [2009] use a model trained on multiple languages in Wikipedia to compare relative interest in different topics across linguistic domains. For example, the Persian-language Wikipedia has a larger than average number of articles about science, while the Finnish-language Wikipedia has a larger than average number of articles about skiing. These methods require parallel or comparable corpora, but for mixed corpora the training data can be augmented with a sup-

plemental corpus of comparable documents as long as the comparable documents cover similar enough topics [Mimno, 2012].

It is not necessary to take "language" in its strict meaning. Loosely aligned models have been applied in information retrieval for query expansion [Gao et al., 2011, 2012]. While the topic models' application in information retrieval (Chapter 2) focus on a single language, Gao et al. [2011] assume the queries and Web documents are in different "languages". The query language from users are normally informal oral language, which are less formatted and may include abbreviation as well. However, the document language are more formal and well organized written language. For example, given a query "dtd amc", the relevant Web document may contain "downtown disney amc" [Jiang et al., 2016].

This semantic gap [Müller and Gurevych, 2009] between queries and documents provides the possibility to treat queries and documents as different languages, and the relevance between queries and documents make them loosely aligned. Based on this assumption, they further assume queries and documents share the same document-topic distributions θ^Q, but have different topic-word distributions ϕ_z^Q and ϕ_z^D respectively. In this way, documents and queries are connected through the hidden topics, even though their vocabularies (topic-word distributions) are different. By summing over all possible topics, the relationship between document term e and query q is

$$p(e \mid q) = \sum_k p(e \mid \phi_k^D) p(k \mid \theta_q). \tag{8.1}$$

Some forms of query expansion across multiple languages do not require explicit modeling of connections between topics [Vulić et al., 2011a, 2015]. As noted in Chapter 2, Erlin [2017] use two independent seeded models on English and German books to search for works about epistemology. After manually identifying one Epistemology topic from each language's model, these two topics are used as a form of query expansion to identify documents related to the target subject.

Although it is relatively easy to get comparable topics from comparable corpora, identifying specific words that are translations of each other in different languages is more difficult. Given a word w_e that has high probability in topic k in language e, it is likely that a good translation

of w_e in language f has high probability in topic k as well. Vulić et al. [2011b] evaluate several methods for identifying such translation pairs given a bilingual or polylingual topic model. They find two methods that work well, both of which consider the frequency of a given target word across many topics. The intuition is that words that have high probability in a given topic because they are specific to that topic are more likely to be good translations than words that have high probability in a topic because they are frequent in the corpus overall, and are thus represented in many topics. The authors were then able to derive an algorithm for finding high-quality aligned translation pairs [Vulić and Moens, 2012]. This method is capable of using word patterns as hints for etymologically related words when available, but is also effective even for unrelated languages. However, there are limits to our ability to find direct translation pairs with only document-level alignment, both because the data is not sufficient, and because languages may align more at the level of *concepts* rather than specific lexical items [Vulić and Moens, 2014].

8.2 Word-level Alignment from Lexical Data

Aligned documents are useful when a collection is designed to be lightly multilingual: e.g., when the creators are building a native-language version of Wikipedia. Documents links are cheap and easy. However, they require active support of those creating the collection, which is not always available. Many collections are written in isolation.

However, one of the most ubiquitous multilingual tools is the dictionary. This section discusses how we can use *lexical information* like multilingual dictionaries [Zhang et al., 2010] and orthographic relations between words [Boyd-Graber and Blei, 2009] to help users who want to understand a collection.

For instance, tree-based topic models such as tree-based latent Dirichlet allocation [Boyd-Graber et al., 2007, Andrzejewski et al., 2009, TLDA] incorporate positive correlations between words in the same or different languages by encouraging words that appear together in a

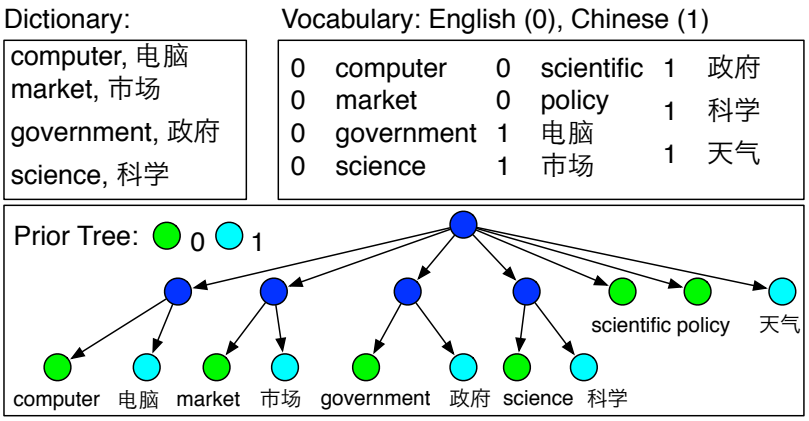

Figure 8.1: An example of constructing a prior tree from a bilingual dictionary: word pairs with the same meaning but in different languages are concepts; a common parent node is created to group words in a concept, and then is connected to the root; uncorrelated words are connected to the root directly.

concept to have similar probabilities given a topic.[1] These concepts can come from WordNet [Boyd-Graber and Resnik, 2010], domain experts [Andrzejewski et al., 2009], or user constraints [Hu et al., 2014a]. If these concepts are in the same language, the backend model is the same as monolingual interactive topic modeling introduced in Chapter 3. However, when we gather concepts from bilingual resources, these concepts can connect different languages. For example, if a bilingual dictionary defines "电脑" as "computer", we combine these words in a concept.

These concepts (positive correlations) are organized into a **prior tree** structure. Words in the same concept share a common parent node (Figure 8.1). That concept then becomes one of many children of the root node. Words that are not in any concept—**uncorrelated words**—are directly connected to the root node. Thus a topic becomes a distribution over all paths in this prior tree and each path is associated with a word.

The probability of a path in a topic depends on the transition probabilities in a topic. Each concept i in topic k has a distribution over its child nodes that is governed by a Dirichlet prior: $\pi_{k,i} \sim \text{Dir}(\beta_i)$.

[1] Zhang et al. [2010] use topic-level soft constraints to achieve a similar effect.

Each path ends in a word (i.e., a leaf node) and the probability of a path is the product of all of the transitions between topics it traverses. Topics have correlations over words because the Dirichlet parameters can encode positive or negative correlations [Andrzejewski et al., 2009].

As a result, to sample a word w_{dn} given a topic z_{dn}, a path y_{dn} from the topic tree of topic z_{dn} is sampled: we start from the root n_0 and first sample a child node n_1 of the root; if node n_1 is a concept node, we continue to sample a word node n_2 and generate the word associated with n_2; if node n_1 is a word node already, we generate the word directly.

When this tree serves as a prior for topic models, words in the same concept are positively correlated in topics. For example, if "电脑" has high probability in a topic, so will "computer", since they share the same parent node. With the tree priors, each topic is no longer a distribution over word types; instead, it is a distribution over paths, and each path is associated with a word type. The same word could appear in multiple paths, and each path represents a unique sense of this word.

8.3 Alignment from Parallel Corpora and Lexical Information

Bilingual dictionaries and other sources of word-level information are valuable in training multilingual models, because they can easily specify simple lexical relationships that might be difficult to extract from parallel corpora. But such manually generated data may be brittle, low-quality, or missing contextual differences in actual usage. These two approaches are not mutually exclusive, however; they reveal different connections across languages. Hu et al. [2014b] bring existing tree-based Latent Dirichlet Allocation (tLDA) and polylingual Latent Dirichlet Allocation (pLDA) together and create the polylingual tree-based Latent Dirichlet allocation (ptLDA) that incorporates both word-level correlations and document-level alignment information.

To build up the prior tree structure, Hu et al. [2014b] consider two resources that correlate words across languages. The first is multilingual dictionaries, which match words with the same meaning in different

languages together. The other is the word alignments extracted from aligned sentences in a parallel corpus. These relations between words are used as the concepts [Bhattacharya, 2006] in the prior tree (Figure 8.1).

Given the prior tree structure, the generation of documents is a combination of tLDA and pLDA. For each aligned document pair (d_{l_1}, d_{l_2}), we first sample a distribution over topics θ_d from a Dirichlet prior $\text{Dir}(\alpha)$. For each token in the aligned document d_{l_i}, we first sample a topic z_{dn} from the multinomial distribution θ_d, and then sample a path y_{dn} along the tree of topic z_{dn}. Because every path y_{dn} leads to a word w_{dn} in language l_{dn}, we append the sampled word w_{dn} to document $d_{l_{dn}}$ in language l_{dn}.

If we use a flat symmetric Dirichlet prior in place of the tree prior, the model is equivalent to pLDA. Similarly, if all documents are monolingual (i.e., with distinct distributions over topics θ), the model is equivalent to tLDA. ptLDA connects different languages on both the word level (using the word correlations) and the document level (using the document alignments), thus it learns better topics by considering more information from both languages.

8.4 Topic Models and Machine Translation

The most frequent application of multilingual topic models is in machine translation. Given a text input in one language (source language), statistical machine translation tries to find a similar sequence of words in another language (target language). Modern machine translation systems [Koehn, 2009] use millions of training examples to learn the translation rules and apply these rules on the test data. Topic models are useful in this application when they can help to inform word meaning and word choice in specific contexts. While the translation rules are learned in local context, these systems work best when the training corpus has a consistent *domain*, such as a genre (e.g., sports, business) or style (e.g., newswire, blog-posts).

Translations within one domain are better than translations across domains since they vary dramatically in their word choices and style. A correct translation in one domain may be inappropriate in another

domain. For example, "潜水" in the Sports domain usually means "underwater diving", but in the Social Media domain, it means a non-contributing "lurker". To avoid such translation errors caused by domain shift, train translation must be robust to such systematic variation in the training set. This is called *domain adaptation*.

To train such SMT systems, early efforts focused on building separate models given the hand-labeled domains [Foster and Kuhn, 2007, Matsoukas et al., 2009, Chiang et al., 2011]. However, this setup is at best expensive and at worst infeasible for large data. Topic models provide a promising solution by automatically discovering domains: Each extracted topic is treated as a soft domain. Standard monolingual topic models, trained only on the source documents, have been applied in this way to extract domain knowledge for machine translation [Eidelman et al., 2012].

For the rest of this chapter we will use the term "topic" and "domain" interchangeably. We will use "topic" to refer to a word distribution in topic models and "domain" to refer to SMT corpora, but these will refer to the same entity.

However, the source language the and target language can complement each other to build up more accurate topic models. For example, if we only know the Chinese phrase "潜水", it is hard to decide whether it is a Sports domain or it is a Social Media domain. However, with the help of the aligned English translation "lurker", it is easy to identify the "social media" domain. Thus multilingual topic models [Ni et al., 2009, De Smet and Moens, 2009] have been applied to extract domain knowledge for machine translation [Hu et al., 2014b].

8.5 The Components of Statistical Machine Translation

Statistical machine translation represents translation as a combination of probabilistic processes, a phrase-level translation model and a sentence-level language model [Koehn et al., 2003, Koehn, 2009]. Topic models have been applied to both aspects of this process.

The core of translation is to find the best translation target sentence $\mathbf{e}$ given the source sentence $\mathbf{f}$. This process is also known as *decoding*.

Given a source sentence $\mathbf{f}$, the best translation sentence in the target language $\mathbf{e}_{\texttt{best}}$ is

$$\mathbf{e}_{\texttt{best}} = \mathbf{argmax}_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) = \mathbf{argmax}_{\mathbf{e}} p(\mathbf{f}|\mathbf{e}) p(\mathbf{e}), \tag{8.2}$$

which is split to a *translation model* $p(\mathbf{f}|\mathbf{e})$ and a *language model* $p(\mathbf{e})$. Intuitively, a good translation should be both a good match for the source sentence (scoring high in the translation model) and a good sentence in its own right (scoring high in the language model).

In the *decoding* phase, the source sentence $\mathbf{f}$ is segmented into multiple source phrases $\bar{f}_n$, which are translated to a set of target phrases $\bar{e}_n$. Thus the translation probability $p(\mathbf{f}|\mathbf{e})$ can be further decomposed to the phrase translation probability $p(\bar{f}_n|\bar{e}_n)$. In the *reordering* phase target phrases may then need to be repositioned to get the best translation result. Reordering is captured by a relative distortion probability distribution $d(a_n - b_{n-1})$, where a_i denotes the start position of the source phrase that was translated to the n^{th} target phrase, and b_{n-1} denotes the end position of the source phrase translated into the $(n-1)^{\text{th}}$ target phrase. As a result, the translation model decomposes into

$$p(\mathbf{f} \,|\, \mathbf{e}) = \prod_n p(\bar{f}_n \,|\, \bar{e}_n) d(a_n - b_{n-1}) \tag{8.3}$$

In phrase-based SMT, the phrase probability $p(\bar{f}_n \,|\, \bar{e}_n)$ can be further estimated by combining lexical translation probabilities of words contained in that phrase [Koehn et al., 2003], which is normally referred as *lexical weighting*. Lexical conditional probabilities $p_w(f \,|\, e)$ are maximum likelihood estimates from relative lexical frequencies,

$$p_w(f \,|\, e) = c(f, e) \Big/ \textstyle\sum_f c(f, e) \tag{8.4}$$

where $c(f, e)$ is the count of observing lexical pair (f, e) in the training dataset. Given a word alignment a, the lexical weight for this phrase pair $p_w(\bar{f} \,|\, \bar{e}; a)$ is the normalized product of lexical probabilities of the aligned word pairs within that phrase pair:

$$p_w(\bar{f} \,|\, \bar{e}; a) = \prod_i \frac{1}{\{|j \,|\, (i,j) \in a\}|} \sum_{\forall (i,j) \in a} p_w(f_i \,|\, e_j) \tag{8.5}$$

where i and j are the word positions in target phrase $\bar{e}$ and source phrase $\bar{f}$.

Next we introduce how to apply topic models to improve translation models, language models, and reordering models respectively.

8.6 Topic Models for Phrase-level Translation

Translation models map words and phrases from one language to another. Both monolingual topic models and bilingual topic models are useful for improving translation models. As we have mentioned in Section 8.4, the most prominent application of topic models is in *domain adaptation*.

Early work for extracting domain knowledge focus on the hand-labeled domains [Foster and Kuhn, 2007, Matsoukas et al., 2009, Chiang et al., 2011]. These labels are not only expensive and time consuming to obtain, but also unsmoothed and sensitive to labeling errors and inconsistency. Besides, such hard domain labels are difficult to apply and can decrease the robustness of translations: domains are fundamentally uncertain, and if you get the domain wrong, you may cut off useful information.

Topic models provide a way of **automatically** discovering **soft** domain assignments. If we equate the K topic distributions over the vocabulary in a topic model with K SMT domains, each document's topic distribution can be viewed as a soft domain assignment for that document. If there are two topics Sports and Social Media and a test example is most likely about Sports, it may have a soft domain distribution as 85% for Sports domain and 15% for Social Media domain. These automatically obtained soft domain labels are well smoothed, and they are not only cheap to obtain but also much more robust to topic errors. We next describe applications of monolingual and multilingual topic models to improve translation models [Eidelman et al., 2012, Hu et al., 2014b].

Translations from Monolingual Topic Models We can train a translation model by counting the frequency of pairs from word-level alignment data. Eidelman et al. [2012] builds topic-specific translation models by reweighting the frequency of word pairs based on soft topic/domain assignments for documents. Since a translated document is assumed to

have the same topics in both languages, we only require a monolingual topic model trained on one or the other language. The document-topic distribution $p(k \mid d)$ is used to smooth the expected count $\hat{c}_k(f, e)$ of a word translation pair under topic k,

$$\hat{c}_k(f, e) = \sum_d p(k \mid d) c_d(f, e), \tag{8.6}$$

where $c_d(\bullet)$ is the number of occurrences of the word pair in document d. The lexical probability conditioned on topic k is the unsmoothed probability estimate of those expected counts

$$p_w(f|e; k) = \hat{c}_k(f, e) / \sum_f \hat{c}_k(f, e), \tag{8.7}$$

from which we can compute the lexical weight of this phrase pair $p_w(\bar{f} \mid \bar{e}; a, k)$ given a word alignment a [Koehn et al., 2003]:

$$p_w(\bar{f} \mid \bar{e}; a, k) = \prod_{i=1}^{n} \frac{1}{\{|j \mid (i, j) \in a\}|} \sum_{\forall (i,j) \in a} p_w(f_i \mid e_j; k) \tag{8.8}$$

where i and j are the word positions in target phrase $\bar{e}$ and source phrase $\bar{f}$ respectively. Equations 8.7 and 8.8 are equivalent to Equations 8.4–8.5, but with the addition of soft topic/domain assignments. Eidelman et al. [2012] combine the standard $f(\bar{f}|\bar{e})$ and $f(\bar{e}|\bar{f})$ with two directions of topic-adapted probabilities $p_w(\bar{f}|\bar{e}; a, k)$ and $p_w(\bar{e}|\bar{f}; a, k)$, equivalent to introducing $2K$ new word translation tables. Feature weights are optimized through using the Margin Infused Relaxed Algorithm [Crammer et al., 2006, MIRA].

For a test document d, the document topic distribution $p(k \mid d)$ is inferred based on the topics learned from training data. The lexical weight feature of a phrase pair $(\bar{f}, \bar{e})$ is

$$f_k(\bar{f} \mid \bar{e}) = -\log \left\{ p_w(\bar{f} \mid \bar{e}; k) \cdot p(k \mid d) \right\}, \tag{8.9}$$

a combination of the topic dependent lexical weight and the topic distribution of the document, from which we extract the phrase.

These adapted features allow us to bias the translations according to the topics. For example, if topic k is dominant in a test document, the feature $f_k(\bar{f} \mid \bar{e})$ will be large, which may bias the decoder to a translation that has small value of the standard feature $f(\bar{f} \mid \bar{e})$. In

addition, combining the adapted features with the standard features makes this model more flexible. For a test document with less clear topics, the topic distribution will tend toward being fairly uniform. In this case, the topic features will contribute less to the translation results and the standard features will dominate the translation results.

Hasler et al. [2012] also use monolingual topic models for domain adaptation like Eidelman et al. [2012], except they replace LDA with *hidden topic Markov models* [Gruber et al., 2007, HTMM]. While words are conditionally independent in an LDA document, HTMM models the words' topics via a Markov chain that encourages nearby words to share topics. Thus, topics within phrases in aligned sentences are consistent and can be directly used as features.

Su et al. [2012] use HTMM to incorporate topic information into the phrase probability directly, rather than through the word translation probability. Given the bilingual translation training data without any specific domain information (referred as out-of-domain bilingual data), they incorporate topic information from the source language into translation probability estimation, and decompose the phrase probability $p(\bar{e} \mid \bar{f})$ as

$$p(\bar{e} \mid \bar{f}) = \sum_{k_{out}} p(\bar{e}, k_{out} \mid \bar{f}) = \sum_{k_{out}} p(\bar{e} \mid \bar{f}, k_{out}) \cdot p(k_{out} \mid \bar{f}) \tag{8.10}$$

where $p(\bar{e} \mid \bar{f}, k_{out})$ is the translation probability given the source side topic k_{out}, and $p(k_{out}|\bar{f})$ denotes the phrase probability in topic k_{out}.

In addition, Su et al. [2012] assume a monolingual corpus in the same domain as the test sentence (called "in-domain monolingual data"). Thus they also apply HTMM to estimate the in-domain topic k_{in} and $p(k_{in} \mid \bar{f})$. However, the in-domain topics k_{in} and the out-of-domain topics k_{out} may not be in the same space, so Su et al. [2012] introduce the topic mapping probability $p(k_{out} \mid k_{in})$ to map the in-domain topic to the out-of-domain topic:

$$p(k_{out} \mid \bar{f}) = \sum_{k_{in}} p(k_{out} \mid k_{in}) \cdot p(k_{in} \mid \bar{f}) \tag{8.11}$$

As a result, the final phrase probability becomes

$$p(\bar{e} \mid \bar{f}) = \sum_{k_{out}} \sum_{k_{in}} p(\bar{e} \mid \bar{f}, k_{out}) \cdot p(k_{out} \mid k_{in}) \cdot p(k_{in} \mid \bar{f}). \tag{8.12}$$

The topic and topic mapping relationship between the training data and test data can be built offline, so the whole process adds no additional burden to the translation system.

Monolingual topic models can add contextual information about word choice to translation models, but do not by themselves take advantage of multilingual information. We next turn to topic models that explicitly learn multilingual connections between words.

Multilingual Information for Domain Adaptation Using bilingual data adds modeling complexity, but can also improve topic model quality. One can think of topic models as tools for disambiguating the meaning of words based on their context. Aligning across multiple languages is a common way of resolving such ambiguities. For example, "木马" in a Chinese document can be either "hobbyhorse" in a Children's topic, or "Trojan virus" in a Technology topic. A monolingual topic model might not be able to tell the difference in a short Chinese document, but these terms are unambiguous in English, more accurately indicating the relevant topic.

While many of the approaches described in this chapter try to model the source and target languages simultaneously to extract topics, some of the benefit of multilingual models can be achieved by aligning monolingual models. Xiao et al. [2012] apply topic models on the source documents and target documents separately to learn the document-topic distributions $p(k_f \mid d_f)$ and $p(k_e \mid d_e)$, and then estimate the phrase-topic probabilities $p(\bar{e}, k_f \mid \bar{f})$ and $p(\bar{e}, k_e \mid \bar{f})$ from each model. They further compute the topic similarity scores between the phrase topic distribution and document topic distribution as features for decoding to improve SMT results.

To translate a new document d_f, they first estimate the document-topic distribution $p(k_f \mid d_f)$. Then for a given phrase $\bar{f}$ in the source document they search for the target phrase $\bar{e}$ that maximizes the similarity between the source document's topic distribution $p(k_f \mid d_f)$ and the phrase-topic distribution $p(\bar{e}, k_f \mid \bar{f})$ according to squared Hellinger distance $H^2(p, q) = \sum_k \left(\sqrt{p_k} - \sqrt{q_k}\right)^2$. Second, they calculate a projection between the two monolingual topic models $p(k_f \mid k_e)$ by normalizing

the co-occurrence count in the aligned training sentences, and use this relationship to calculate the conditional distribution of target phrases and target topics $p(\bar{e}, k_e \mid \bar{f})$.

This topic projection resembles the topic mapping by Su et al. [2012], but it is applied between the source language and the target language. Compared to the lexical features in Eidelman et al. [2012] and Hu et al. [2014b], Xiao et al. [2012] introduce a new framework to apply topic information directly to measure the relationship between phrases and present two topic similarity features for decoding. These two approaches can be combined to further improve SMT.

8.7 Topic Models for Sentence-level Language Modeling

A critical component of machine translation systems is the language model, which provide local constraints and preferences to make translations more coherent. A language model describes the probability of a word w occurring given the previous context words, which is also mentioned as the history h (Chapter 2.1 discusses language models for information retrieval). They also help choose the correct or more appropriate word during statistical machine translation. For example, the English words "house" and "home" are often synonymous, but the translation "I am going home" is better than "I am going house".

Domain adaptation for language models [Bellegarda, 2004, Wood and Teh, 2009] use extra knowledge to adjust this probability $p(w \mid h)$ to reflect a change in context, which is an important avenue for improving machine translation. As Bellegarda [2004] points out, "an adaptive language model seeks to maintain an adequate representation of the current task domain under changing conditions involving potential variations in vocabulary, syntax, content, and style".

Topics from topic models can be one of the resources to provide such knowledge for language model adaptation. For example, the Chinese phrase "很多粉丝" is "a lot of vermicelli" in a Food domain, but means "a lot of fans" in an Entertainment domain. Such ambiguity can be reduced by using topic/domain knowledge. If the Entertainment topic is known based on the previous context, this Chinese phrase will be

translated to "a lot of fans" without any ambiguity. We next describe how to capture thesis topical knowledge for language model adaptation.

Language Model Adaptation from Monolingual Topic Models Early work [Clarkson and Robinson, 1997, Seymore and Rosenfeld, 1997, Kneser and Peters, 1997, Iyer and Ostendorf, 1999] focuses on partitioning the training data to multiple topic-specific subsets and building up language models for each subset. Then the topic-specific language models $p_k(w \mid h)$ are linearly combined with a general language model $p_g(w \mid h)$ built from all training data as Equation 8.13. The weights λ_k can be tuned based on the topics of the test documents.

$$p_{\text{adapted}}(w \mid h) = \sum_k \lambda_k p_k(w \mid h) + \lambda_g p_g(w \mid h) \tag{8.13}$$

Seymore et al. [1998] further identify the most appropriate topic for each word in the vocabulary and choose either a topic-specific language model or the general language model. The intuition is that the general language model best estimates general words, but the topic language better models more specific words. As a result, they split the vocabulary words into three groups: the general subset, on-topic subset and off-topic subsets. They use the general language model for the general subset and the off-topic subset and the topic-specific language model for the on-topic subset.

All of these methods use a traditional n-gram model, which conditions on a finite, bounded history. These models also assume each document or history belongs to exactly one topic cluster. To fix these problems, models with topic mixtures, such as *Latent Semantic Analysis* [Deerwester et al., 1990, LSA] and its probabilistic interpretation probabilistic latent semantic indexing (PLSI) [Hofmann, 1999a, PLSI], learn large-span language models [Bellegarda, 1997, Coccaro and Jurafsky, 1998, Gildea and Hofmann, 1999]. Gildea and Hofmann [1999] decomposes the language model for a unigram w as a summation over topics,

$$p(w \mid h) = \sum_k p(w \mid k) p(k \mid h) \tag{8.14}$$

where the topics are learned by setting these probabilities to optimize the sum of the logs of the marginal probabilities of all words in the training corpus,

$$l(\theta; N) = \sum_{w} \sum_{d} n(w, d) \log \sum_{k} p(w \mid k) p(k \mid d) \tag{8.15}$$

where d is the training documents, and $n(w, d)$ is the word frequency of w in document d. $p(w \mid k)$ and $p(k \mid d)$ are learned through the EM algorithm. For test documents, they fix $p(w \mid k)$ to estimate $p(k \mid h)$ and then compute $p(w \mid h)$ using Equation 8.14.

This way of applying topic models to language models is similar to document language modeling for information retrieval as in Chapter 2.1. However, unlike information retrieval, two different languages are involved in the process of SMT, and they can complement each other to learn more accurate topics. Next, we discuss multilingual topic models for language model adaptation.

Language Model Adaptation from Multilingual Topic Models As we explain in Section 8.6, the information from different languages can complement each other to extract better topics. Latent semantic models such as LSA have been used in multilingual information retrieval for many years [Carbonell et al., 1997]. We now describe approaches to add *multilingual* information to probabilistic topic models for language model adaptation.

Tam et al. [2007] introduce bilingual latent semantic analysis (BLSA) to learn the topics for both source language and target language and apply the learned topics to language model adaptation for SMT. Similar to polylingual topic models [Mimno et al., 2009], BLSA transfers the inferred topics from the source language to the parallel target language.

More specifically, Tam et al. [2007] assume the aligned source document and the target document share the same document-topic distribution. They first learn an LSA model on the source language and then use the document-topic vector from the source document as the document-topic vector for the aligned target document, and then infer the topic-word vector on the target side. The topics for the target

language are not learned iteratively, thus the topics in a parallel corpus can be learned very efficiently.

To apply the topics for language adaptation, the word marginal distribution $p_{lsa}(w)$ for document d is computed,

$$p_{lsa}(w) = \sum_{k=1}^{K} p(w \mid k)p(k \mid d) \tag{8.16}$$

Then this word marginal distribution is integrated into the target background language model by minimizing the KL divergence between the adapted language model and the background language model [Kneser et al., 1997]:

$$p_a(w \mid h) \propto \left(\frac{p_{lsa}(w)}{p_{bg}(w)}\right)^{\beta} \cdot p_{bg}(w \mid h) \tag{8.17}$$

Ruiz and Federico [2011] apply a similar idea for language model adaptation. Instead of using BLSA, Ruiz and Federico [2011] merge the aligned source and target document as one document, and train PLSI. Both ideas are based on the assumption that the aligned source document and target document share the same document-topic distribution. The final adapted language model combines the topic-based language model with the general background language model, thus it is more robust in improving the results of SMT.

Yu et al. [2013] present a hidden topic Markov model (HTMM) to improve the language model in SMT. They build up a topic model on the source side and target side respectively, and learn a topic-specific language model based on the target side by estimating the maximum-likelihood. To smooth the sharply distributed probabilities, they back off to other distributions:

$$p(w_i \mid w_{i-n+1}^{i-1}, k_e) = \lambda_{w_{i-n+1}^{i-1}} p_{MLE}(w_i \mid w_{i-n+1}^{i-1}, k_e) \tag{8.18}$$

$$+ (1 - \lambda_{w_{i-n+1}^{i-1}}) p_{MLE}(w_i \mid w_{i-n+2}^{i-1}, k_e) \tag{8.19}$$

where λ is the normalization parameter

$$\lambda_{w_{i-n+1}^{i-1}, k_e} = \frac{N_{1+}(w_{i-n-1}^{i-1}, k_e)}{N_{1+}(w_{i-n-1}^{i-1}, k_e) + \sum_{w_i} c(w_{i-n+1}^{i}, k_e)} \tag{8.20}$$

where $N_{1+}(w_{i-n-1}^{i-1}, k_e)$ is the number of words following w_{i-n-1}^{i-1} in topic k_e, and $c(w_{i-n+1}^{i}, k_e)$ is the count of n-gram w_{i-n+1}^{i} in k_e.

During decoding, since no target sentence is available, they extract the topics on the source side and project the source topic to the target side. The target probability is:

$$p(e) = \sum_{k_e} p(e \mid k_e) p(k_e) = \sum_{k_e} p(e \mid k_e) \cdot \sum_{k_f} p(k_e \mid k_f) p(k_f) \qquad (8.21)$$

where $p(k_e \mid k_f)$ is the topic projection probability, estimated by the co-occurrence of the source-side and the target-side topic assignment.

8.8 Reordering with Topic Models

In addition to translation models and languages models, a third important component of a phrase-based SMT system is reordering models, which learn how the order of words in the source sentences influences the order of words in the target sentences and how to make the translations in the right order. The usefulness of topic models in reordering is less clear than their usefulness for domain adaptation of translation models and language models, but it is nevertheless significant. The primary advantage is that word order in different domains of the same language may be different: Chen et al. [2013] find that training corpora in different domains vary significantly in their reordering characteristics for particular phrase pairs. As the example shown in Table 8.1 [Wang et al., 2014], in an Economy topic, the Chinese word 比 is on the left of 五; but in a Sports topic, 比 is on the right of 五. As a result, it is necessary to introduce domain knowledge to model such order variance, and topic models provide a good data-driven way to do so.

Xiong et al. [2006] treat the reordering problem as a classification with two labels: straight and inverted between two consecutive blocks, and build up a maximum entropy classification model as the reordering model. Chen et al. [2013] manually divide the training data into multiple domains, instead of using automatic techniques such as topic models. Wang et al. [2014] integrate two more types of topic-based features into the reordering model, in addition to the boundary word features [Xiong et al., 2006]. First, they choose the topic with maximum probability in a

Topic	Type	Example
Economy	Source	··· **比五** 月份下降3.8% ···
	Target	··· down 3.8% from May ···
Sports	Source	··· **五比**一3.8% ···
	Target	··· five to one ···

Table 8.1: Topics influence the word orders: the Chinese words in bold are in different orders in different topics. (Example from Wang et al. [2014])

document to be the *document topic feature* for that document. Besides, they also use the topics of the content words that locate at the left and rightmost positions on the source phrases as the *word topic features* to capture topic-sensitive reordering patterns.

During the decoding process, Xiong et al. [2006] infer the topic distributions of the test documents first and then apply this proposed topic-based reordering model as one sub-model to the log-linear maximum entropy model to obtain the best translation:

$$e_{\texttt{best}} = \texttt{argmax}_e \left\{ \sum_{m=1}^{M} \lambda_m h_m(e, f) \right\} \tag{8.22}$$

where $h_m(e, f)$ are the sub-models or features of the whole log-linear model, λ_m are their weights accordingly, which are tuned on the development set.

This framework is very flexible and can encode any topic-based features. Any multilingual topic models we have discussed so far can be applied to extract better topics.

8.9 Beyond Domain Adaptation

In addition to translation models, language models and reordering models, there are also other modules of SMT, such as word alignment, where topic models have also been applied. The Bilingual topical admixture model Zhao and Xing [2006, BiTAM] assumes each document pair is an admixture of topics, and the topics for each sentence pair within that document pair are sampled from the same document-topic distribution.

Each topic also has a topic-specific translation table. Therefore, the sentence-level word alignment and translations are coupled by the hidden topics. BiTAM captures the latent topical structure and generalizes word alignments and translations via topics shared across sentence pairs, thus the quality of the alignments is improved.

In addition, coherence, which ties sentences of text into a meaningfully connected structure [Xiong and Zhang, 2013], is another important piece to SMT. Xiong and Zhang [2013] introduce a topic-based coherence model to improve the document translation quality. They learn the sentence topic for source documents, based on which they predict the target topic chain; they then incorporate the predicted target coherence chain into the document translation decoding process.

8.10 Summary

Topic models are not limited to a single language and different languages can be connected on either document level or word level. Multilingual topic models obtain topics with high quality, since different languages can complement each other to reduce topic ambiguity. Many different approaches apply multilingual topic models to improve different pieces of the statistical machine translation pipeline. With such topic knowledge, the variations of different languages can be better captured to make the translations more natural and coherent.

9

Building a Topic Model

Previous chapters have focused on *existing* models. We have thus far described models that researchers have created to capture particular nuances of documents or document-creating processes that exist in the world. This chapter focuses on how a researcher can create, implement, and validate a new model.

Before going into the details of constructing new models, we encourage users to consider whether questions can be answered through post-hoc analysis of a simple topic model with additional information. For example, a simple dynamic topic model can be constructed from a standard LDA model by slicing the corpus into sections and estimating the probability distribution over words for each topic in each section. Building and validating custom topic models is a powerful tool, but requires significant investment in coding and debugging, and may not be able to take advantage of computational optimizations available for simpler models. Posterior predictive checks [Mimno and Blei, 2011] provide a good means of determining whether LDA topics or words within topics already show patterns that are present but not explicitly modeled.

It is impossible to cover all of the details of research in topic models or machine learning generally, but this chapter introduces some of the common techniques for creating new models. We will focus on a running example, creating a new model for predicting the ideology of a political speaker.

9.1 Designing a Model

The first step in creating a new model is to define what is important. For example, previous chapters have focused on measuring innovation, cross-language connections, and sentiment. These are high-level concepts that we want to discover from text. We believe these properties exist in the world, but we want the variables of our models to represent these concepts.

In a topic model, incorporating a new concept into the model usually involves adding a new random variable to the model. This is where intuitions and domain knowledge come to the forefront. Because the generative process attempts to model the real world, a new model must balance several components that are often in tension: fidelity, performance, tractability, and interpretability.

Fidelity A good model should reflect the world. One of the ways we can model the relationship between words and non-word information is to define conditioning patterns. If political scientists believe that a politician's ideology is a property that changes how they speak, then the model should condition topic choices on a speaker's ideology. Mimno and McCallum [2008] describe this formulation as an "upstream" model (for more details see Chapter 7). If they believe that electoral success is a result of political speech, then the model should condition success on topics (a "downstream" model).

Modeling reality is a good idea. But just as building a scale model of a building requires compromising materials and level of detail, building a statistical model sometimes requires unreal assumptions. If a generative model exactly matches the process that produces data, we can prove that it will converge to the correct answer [Neal, 1993]. But humans

and text are not Dirichlet and discrete distributions; all models will be an approximation.

In addition to determining *where* to model a feature in a generative model, you must decide *how* to model it. Is it a continuous value, a binary value, a member of a discrete set, or something else? Often, we have multiple pre-defined notions of how to represent a quantity of interest. Sentiment is sometimes represented as a continuous positive or negative value, while review corpora include discrete, positive star ratings. Political ideology could be represented by membership in one of a fixed number of parties, but political scientists more often assign a continuous value to a politician's ideology.

Performance Greater fidelity to our perceptions of how the world works does not always imply that models will be more useful. We sometimes have to trade off fidelity with performance. Recent work seems to agree that *downstream* models work better even though they are less realistic [Nguyen et al., 2013].[1]

It is nearly impossible to know *a priori* whether a model will work well for a particular task given just the model. Knowing what will work best is often a trial-and-error process. However, it is often possible to draw parallels from similar models. Upstream models allow metadata variables to better predict which topics will occur in a context, but they do not necessarily encourage the model to learn different topics than a model without metadata. Downstream models must align topics to best predict a metadata variable, and therefore tend to find different (but not necessarily better) topics. Supervised topic models have therefore worked well for predicting sentiment as a downstream variable, so it might be reasonable to assume that a downstream model would work well for ideology as well.

Tractability Now that we know what variable we want to model, some approaches to model that variable could be easier than others. Political

[1] One might argue that it is more realistic for a *listener* who must interpret speech to assign an ideology after it has been heard, but this is no longer consistent with how the text was *generated*.

ideology is often thought of as a *spectrum* rather than a label: some politicians are more centrist than others even though they might have the same party label.

However, discrete labels are appealing from a modeling perspective. Dirichlet-discrete distributions (Chapter 1) are easy to to combine. Given that basic topic models are built from Dirichlets and discrete distributions, it is easier to add an additional Dirichlet-discrete than to add a continuous random variable to the model. For example, the Topic-Aspect model [Paul and Girju, 2010] is far simpler to implement than Supervised Topic Models [Blei and McAuliffe, 2007].

Dirichlets and discrete distributions play well together because they are conjugate, while Gaussian distributions add additional difficulty. However, Gaussian distributions are more convenient than other distributions. For example, spherical distributions [Batmanghelich et al., 2016] are thought to better model continuous embeddings of words than Gaussian distributions but come at the cost of a less tractable model.

Even worse are combinatorial probability distributions. For example, in Chapter 8 we discussed how to learn mappings across languages. Boyd-Graber and Blei [2009] use a combinatorial distribution to learn the mapping from one language to another. It is very complicated but does not model languages as well as far simpler approaches [Mimno et al., 2009].

Our listing of ever more complicated distributions should not be taken as an admonition against using them: sometimes complicated models are necessary. However, one should not choose a complicated model just because it is complicated (an alluring temptation to young graduate students who want to show off their machine learning chops). It is best to try the simplest possible model that could work; even if it fails, this model can serve as a useful baseline.

Unlike many of the other dimensions when building a model, complexity is often sought and fetishized without improving the other dimensions (perverse incentives from publications often play a role in this). Thus, guard against unnecessarily complicating a model!

Interpretability As we describe in Chapter 3.4, interpretability is a measure of how easy it is for a human to understand the results of a model. Often, the interpretability of a model is an afterthought. That is, if it is thought of at all; many papers neglect to inspect the learned parameters of a model, focusing instead on quantifications of performance.

While always a problem, with the increased prominence of deep learning (discussed more in Chapter 10), this is particularly worrying. Deep learning has a reputation for inscrutable parameters but state-of-the-art performance. One of the strengths of probabilistic models is their interpretability and grounded generative processes. Thus, researchers who choose probabilistic models such as topic models should not ignore the interpretability of their models.

While Chapter 3.4 discusses rigorous evaluations of model interpretability, even a simple once-over of model parameters by the researcher can reveal whether a learned model "makes sense" or not. Inspecting the model can help buttress whether the design of the model ("fidelity", above) was able to capture the intuitions of the modeler.

However, there are often tradeoffs with interpretability (as with the choice of a deep learning model). Chang and Blei [2009] found that the Correlated Topic Model [Blei and Lafferty, 2007] had markedly higher held-out likelihood at the cost of interpretability.

9.2 Implementing the Model

So you have a new model in hand. This is usually described as a generative process (Chapter 1.4): a sequence of probabilistic steps that tells a story of how your data came to be.

Implementing a model requires writing down this model in a form that a computer can understand and then using *probabilistic inference* work backward from data to discover the configuration of the latent variables (topics, other properties of the data you've added as part of the model-building process) that best describe your data.

9.2.1 Automatic Approaches

Automatic approaches for inference are attractive: write down your model and call it a day. However, automatic approaches do have several drawbacks. They are often restricted to specific platforms, are slower than inference "by hand", and restrict the kinds of models you can explore.

Using a tool (no matter how wonderful) developed by someone else means that you must accept their assumptions. It may force you to use a programming language that you are unfamiliar with, it may force you to format your data in odd ways, or it may restrict you to operating systems you do not use (or cannot afford).

Stan Stan [Stan Development Team, 2014] is an inference framework that works best with the R programming language.

Infer.Net Infer.net [Minka et al., 2014] is a Microsoft-created, closed-source framework designed for conjugate models but can only be used on the Windows operating system.

Automatic Differentiation The advent of deep learning has created a variety of auto-differentiation platforms. These can often be used for arbitrary objective functions including variational inference. If you are already familiar with Torch [Collobert et al., 2011] or Theano [Theano Development Team, 2016], it is relatively easy to use these tools to define variational objectives for arbitrary probabilistic models.

9.2.2 Variational Inference

While we focused on Gibbs sampling in Chapter 1.4, the other major class of inference algorithms is variational inference. Compared to Gibbs sampling, variational inference is often considered to be slightly more difficult to both derive and implement.

Variational inference resembles expectation maximization algorithms [Liang and Klein, 2007]. Expectation maximization algorithms find a setting of local latent variables z (which represent specific obser-

vations) and global parameters θ (which represent model properties) that maximize the data likelihood $p(x \mid z, \theta)$. First, we start with some guess of what the latent variables might be z_0. Then we update the parameters to be

$$\arg\max_{\theta} p(x \mid z, \theta), \tag{9.1}$$

the parameters that maximize the likelihood. Then, given those parameters, compute the expected value of the latent variables z to compute the next iteration of latent variables z.

Expectation maximization is a useful tool for inference when the model distribution p is simple enough to solve Equation 9.1 directly. However, for many models (including topic models), this is not feasible. Variational inference solves this intractability by searching for optimal *distributions* $q(z)$ and $q(\theta)$ rather than optimal values of z and θ. It is this search for optimal functions (i.e., probability distribution functions) rather than optimal values that gives this method its name: the "calculus of variations" is the branch of calculus concerning optimal functions.

A variational distribution is a function that assigns a probability score to each setting of the model's latent variables. These are the same variables that would appear in the original model's posterior distribution over latent variables given observed variables and user-defined hyperparameters. Although it is a distribution over the same variables, it is typically simpler. In the original posterior distribution, there are many dependencies between variables: the topics for one document constraints the topics for other documents. In common approaches for topic models [Blei et al., 2003], the distribution is fully factorized to break these constraints. For example, the true distribution over latent variables is

$$p(z, \theta) = \prod_k p(\phi_k \mid \beta) \prod_d p(\theta_d \mid \alpha \boldsymbol{u}) \prod_n p(z_{d,n} \mid \theta_d) p(w_{d,n} \mid \beta_{z_{d,n}}) \tag{9.2}$$

where $\boldsymbol{u}$ is a K-dimensional uniform distribution, while the variational distribution is

$$q(z, \theta, \beta) = \prod_k q(\beta_k \mid \lambda_k) \prod_d q(\theta_d \mid \gamma_d) q(z_{d,n} \mid \phi_{d,n}), \tag{9.3}$$

where λ and γ are Dirichlet parameters that describe the model's latent variables. Although functionally the same as the true distribution p,

because the parameters are no longer tied, they are called variational parameters.

Instead of maximizing the data likelihood, variational inference minimizes the distance between the variational distribution q (a product of independent distributions over each of the latent variables) and the original model posterior distribution p. Here, distance is the Kullback-Leibler *divergence* between the distributions p and q. Minimizing this divergence is equivalent to optimizing a lower bound of the data likelihood,

$$\ell \equiv \mathbb{E}_q \left[\log \left(p(w \mid z, \theta) p(z, \theta | \alpha, \beta)\right)\right] - \mathbb{E}_q \left[\log q(z, \theta)\right] \tag{9.4}$$

There are several options for this optimization: direct optimization, stochastic gradient [Hoffman et al., 2010], or coordinate ascent [Blei et al., 2003].

Deriving the Objective Function Once you have chosen a variational distribution, you need to derive the full form of the objective and compute the updates for each variational parameter. This involves taking the derivative of Equation 9.4 with respect to that variational parameter, setting it equal to zero, and then solving for the variational parameter.

Choosing a Variational Distribution On one hand, you want to choose a variational distribution that is close to the true distribution over the latent variables. In fact, if you choose q so that it is *equal* to p, variational inference reduces to expectation maximization.

However, choosing a more accurate variational distribution often comes at a cost: more complicated computations or a more difficult (or even impossible) derivation. You might not be able to solve the equations to update individual variational parameters or the calculations might be more difficult. The more dependencies in the variational distribution, the more terms you must consider. When there are latent variables for every token in many documents, the number of dependencies can explode.

Often a fully-factored variational distribution is a good choice. After implementing a fully-factorized variational distribution, a researcher

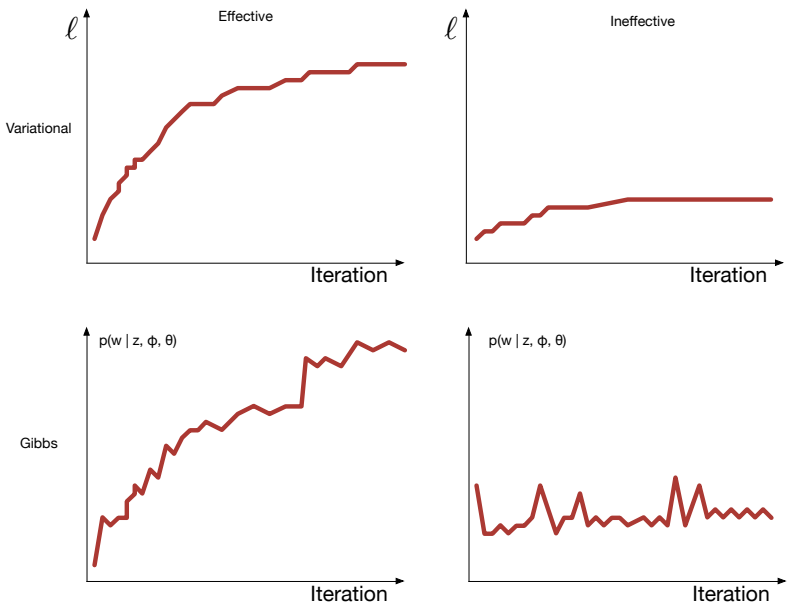

Figure 9.1: Monitoring the objective function is an important component of diagnosing whether inference is working correctly. Correct variational inference should increase monotonically while Gibbs sampling can decrease slightly. However, in both cases, the objective function should increase dramatically over time. If the objective is flat, it may mean that you may need to reconsider design choices in your inference algorithm. Scales for variational inference and Gibbs sampling are not comparable.

should carefully monitor the objective function (Equation 9.4). Ideally it will quickly increase and reach a stable (local) optimum (Figure 9.1, left). However, if it does not (Figure 9.1, right), then it could be that there are coupled variables that are not well served by the variational distribution.

Coupled variables need to change together, and a fully-factorized distribution forces them to change in sequence. For example, let's say that $x = 0$ and $y = 0$ and that they are highly correlated, but a higher probability setting is for $x = 1$ and $y = 1$. Coordinate ascent optimization with a fully-factorized variational distribuion will want to change x or y individually, when in fact they must move together to maintain high performance.

To fully model the interaction between the two variables, we can model x and y jointly in the variational distribution. Instead of independent distributions $q(x)q(y)$, the variational distribution becomes $q(x, y)$ (as appropriate those variables' formulation).

9.2.3 Gibbs Sampling

Gibbs sampling (as discussed at a high level in Chapter 1) finds latent variables by randomly sampling an assignment of each random variable conditioned on all of the other random variables.

Thus, after creating your model, you need to compute the conditional distribution of each random variable conditioned on all of the others. For example, the conditional distribution of word n's topic assignment variable $z_{d,n}$ (highlighted in bold) is

$$p(\boldsymbol{z_{d,n}} \mid z_{d,1} \dots z_{d,n-1}, z_{d,n+1}, \dots z_{d,N_d}, \theta, \phi) = \frac{p(z_{d,1} \dots z_{d,n-1}, \boldsymbol{z_{d,n}}, z_{d,n+1}, \dots z_{d,N_d}, \theta, \phi)}{p(z_{d,1} \dots z_{d,n-1}, z_{d,n+1}, \dots z_{d,N_d}, \theta, \phi)}, \quad (9.5)$$

which simplifies into Equation 1.4.

Deriving conditional distributions is often simpler than expanding variational expectations. Hardisty et al. [2010] provide a step-by-step tutorial on deriving Gibbs sampling for probabilistic models.

Marginalization and Joint Variables Just like there is some art in deciding which variables to merge into a joint distribution for variational inference, in Gibbs sampling, you need to decide which variables to sample jointly and which variables to marginalize.

Jointly sampling variables solves the same problem as merging variables into the same variational distribution above. Instead of sampling x conditioned on all other variables and then sampling y, x and y are sampled *at the same time* from a distribution that conditions x and y on all of the other variables.

9.3 Debugging and Validation

Now that you have implemented inference for your topic model, how do you know if it's working as intended?

9.3.1 Synthetic Data

Because most topic models have a generative story, we can *generate* data. This means running the probabilistic story of the model forward to generate data. For example, for LDA, sample a topic distribution for each document and a type distribution for each topic from Dirichlet distribution. Such a dataset is often called a *synthetic* dataset.

If you create a dataset this way, the latent variables are no longer latent. You know *exactly* what they are. If you run inference on these data you should be able to recreate the unmasked latent variables you generated. If your inference algorithm does not come close, there is likely a problem with your inference procedure.

However, topic models can be tricky. Topic 7 in your synthetic data may correspond to Topic 4 that you discovered in inference. That is not a problem as the topic identifiers are arbitrary; thus, you will have to match topics greedily or measure some other statistic to compare your inferred variables with the true synthetic data.

Synthetic data is most valuable when it is close to the distribution of real data. Toy problems can be a good sanity check, but provide at best a loose upper bound on our ability to model data that we care about. Using *semi-synthetic* data can therefore be a good compromise. In this setting, you train a model from real data, and then generate new documents from that model. The resulting semi-synthetic corpus has many of the correct properties of natural documents, such as vocabulary size and sparsity, but is guaranteed to actually fit the proposed model.

9.3.2 Updates

For both variational inference and Gibbs sampling, the most important step is updating the distribution for individual random variables. Thus, you will want to take as many steps to ensure their accuracy as possible.

For variational inference, each update of the variational parameters increases the objective function (Equation 9.4). Thus, you can check after every update of a variable's variational distribution whether that objective has increased or not. While not every update will exactly increase the objective (due to numerical precision errors or initialization), no update should dramatically *decrease* the objective. If one does, you likely have a bug.

Gibbs sampling is stochastic, so it is more difficult to debug. But for both variational inference and Gibbs sampling, you should write unit tests (worked out with pen and paper) to verify that your updates reach the right answer given the same inputs. Gibbs sampling's randomness can be handled in a unit testing environment using stubs that replace the random number generator (these stubs can also be useful when you later want to generate results with different random seeds, as discussed below).

9.3.3 Baselines and Metrics

Often you will want to compare your model to another model that either has a similar structure or application. This comparison can help you during development.

For example, you may want to extend supervised LDA in some way (Chapter 7). Fortunately, SLDA is a well-defined model and has established performance on widely available datasets.

Thus, a reasonable development strategy would be to create a series of models that take you from SLDA to your final model (e.g., adding one latent variable at a time), at each time comparing how well your model does against SLDA. These comparisons do not just aid development; they will also help document how important each of the changes to the model are.

Comparisons between topic models can be difficult because implementations and algorithms can vary so much. It is often unclear, for example, whether an observed difference is due to one model being better than another or to comparing Gibbs sampling to variational inference. An excellent way to test a new algorithm for a complicated model is to find ways to implement simpler models in the same code. For

example, you can emulate an LDA model using an Author-Topic model [Rosen-Zvi et al., 2004] by assigning each document its own "author".

A standard metric to report for topic models is their perplexity or held-out likelihood. This value is the probability of held-out data given the settings of the model. Wallach et al. [2009b] detail a method for estimating perplexity in topic models given a Gibbs sampling algorithm (which is beyond the scope of this survey).

However, perplexity is often not what you care about for your application of topic models. For example, if you developing a variant of SLDA, you likely care about prediction accuracy. Thus, as you are slowly building out your SLDA variant, you should report prediction performance as well.

9.4 Communicating Your Model

After building a model, implementing inference, and applying it to data, the next exciting step is to tell the world. While one must obviously provide motivation for a new model and describe the technical details, one component of communicating a model that is often overlooked is the interplay between these two facets of describing a model.

The application should not just stand alongside the modeling goal; it should be tightly integrated in the probabilistic story. Rather than listing the steps of the generative process, the generative process should be described with evocative variable names. For example, if your model attempts to capture political polarization, then you may call a discrete variable "polarization π" to make it clear where in the model this aspect will be modeled.

However, naming a variable does not make it so. You will also need to provide qualitative evidence that will convince a skeptical reader that your model is doing what you promised. This is possible by providing overviews of your data at either the micro or macro level.

At the macro level, you can show that your model is doing reasonable things by showing the distribution over words given topics (or whatever the analogous component is). However, you should avoid the temptation to cherry-pick topics. It is better to select topics randomly or—if you

do cherry-pick—to also select "bad" topics that show failure modes of your model.

While macro level cues can show the model finds good summaries, you should also show individual documents and how they interact with data. For example, if your model attempts to show political polarization, you can show polarized (or not) documents and show how the latent variables in your models correctly capture those aspects of the documents (or not; as with topics, show failure modes as well).

In addition, you will also have quantitative metrics for your task: accuracy of prediction, precision at rank K, or translation quality (Chapter 8). When reporting quantitative results, remember the probabilistic foundations of topic models: you are not learning one answer. Regardless of the inference, you are learning a *distribution* over latent variables given a dataset.

Thus, convey the inherent uncertainty in inference. You should run inference multiple times with different random seeds (Gibbs sampling) or random initializations (variational inference). Quantitative results with error bars enables credible comparisons to other models: not only do you have a higher score but you show that your higher score is not just the result of chance.

9.5 Summary

This chapter discusses the process for creating a new topic model from scratch and how to communicate this process to the world. While superficially different than our application-focused chapters, most of the papers were created through this process of model-building, inference, and evaluation. Thus, this chapter helps understand the process for building the models discussed through these chapters.

In the next chapter, we contemplate the future of topic models and how topic models might fit into the broader computer and information science research agenda.

10

Conclusion

While we have attempted to cover a variety of the applications of topic models to help individuals navigate large text datasets, no finite survey could enumerate all of the applications of topic models in text, which have been applied to part of speech tagging [Toutanova and Johnson, 2008], word sense induction [Brody and Lapata, 2009], and entity disambiguation [Kataria et al., 2011]. It goes without saying that we have also omitted many other applications outside text, such as biology [Pritchard et al., 2000], understanding source code [Maskeri et al., 2008], music analysis [Hu and Saul, 2009], and many more.

10.1 Coping with Information Overload

A challenge in topic modeling is how to make inference efficient enough to both scale to large datasets and to provide low-latency interactive experiences to help provide support to a user in the loop. There are three broad strategies for processing documents more quickly.

The first is through decreasing the average number of times a computer needs to look at a document to learn a topic model; i.e., to improve *throughput*. Online algorithms [Hoffman et al., 2010] only

look at a document once, update the topics, and then move on to the next document. This is often much faster than batch approaches which require many passes over the same set of documents. Another option is to *distribute* computation across many machines [Zhai et al., 2012].

A complementary approach is to reduce the time a computer spends on any particular document: improving *efficiency*. This is possible by improving how long it takes to sample document assignments [Yao et al., 2009, Li et al., 2014] or compute variational parameters [Mimno et al., 2012].

The final approach to improve the efficiency of probabilistic algorithms for topic models is to rethink the inference process entirely. Novel approaches view topic model inference as a factorization of a co-occurence matrix [Arora et al., 2013] or as a spectral decomposition [Anandkumar et al., 2012]. These approaches often are much faster than traditional approaches as they use word types—rather than documents—as the central unit of computation.

10.2 Deeper Representations

Part of the benefit of topic models is that the topic distribution of a document (θ) serves as a low-dimensional representation of what the document means. This numerical vector is useful for finding similar documents (Chapter 2), displaying documents to a user (Chapter 3), or connecting documents across languages (Chapter 8).

Increasingly, vector-based, distributed representations have been useful "all the way down". Vector-based representations of words and phrases can improve next word prediction [Bengio et al., 2003], sentiment analysis [Socher et al., 2012], and translation [Devlin et al., 2014]. And this is not just for text—representation learning has taken hold of speech, vision, and machine learning generally.

The effect of representation learning on topic modeling remains unclear as we go to press in 2017. We see several ways that representation learning and topic modeling could benefit each other in the future.

Evaluation Evaluation methods from topic models (Chapter 3.4) have made their way into representation learning [Schnabel et al., 2015, Iyyer et al., 2016], which suggests that some of the lessons learned in making topic models interpretable could also be applied in representation learning. This mollifies some critics of representation learning who argue that the results are often uninterpretable or deceptive [Szegedy et al., 2014].

Synthesis Topic modeling is also blending with more expressive latent representation models [Ranganath et al., 2015]. Topic models could help representation learning solve some of its difficulty summarizing larger segments of text. Paragraphs and sentences are difficult to model as a single vector, and techniques more sophisticated than simple averaging do not seem worth the hassle [Iyyer et al., 2015].

Parallel Evolution Another possible path is less intertwined: topic models and deep-learning representation learning solve different problems and are not directly competitive. Topic models offer advantages of speed and interpretability, while representation learning can do better for prediction-based tasks. Topic models have never been ideal, for example, in inducing features for text classification: it is almost always better to use word-count features. If interpretability and recognizability is not a fundamental goal of your analysis, you are probably better off using something else. However, in cases where interpretability and recognition are the main and final goals of analysis, deep learning methods offer little because their very advantage —greater representational complexity—is also their weakness. Both approaches should be tools that many text miners have in their toolkit, with specific circumstances for using either.

10.3 Automatic Text Analysis for the People

However, in our view, the primary research challenge of topic models is not to make these models and their inference more complicated but

rather to make them more accessible. As we have described, topic models can help scholars and ordinary people navigate large text collections.

However, using topic models still requires extensive data and computer skills. Our job as information scientists is not complete until these tools (or suitable alternatives) are available to everyone who needs them.

This goal requires making the tools more usable. The corpus preprocessing and vocabulary curation required of topic models is not straightforward: should we remove non-English documents, what should we consider a document, how should we use metadata? Nor are the modeling choices needed to make sense of the data trivial: how many topics should we use, which of the many possible models should we use, and what inference technique gives us the best tradeoff between speed and accuracy? Existing topic models do a poor job of communicating what options are available to a user and what consequences these choices have.

However, even if the process of creating a topic model becomes intuitive, the output must also be interpretable. Distributions over words are the language that these models use to create representations of document collections, but it is not how users think about topics: they would much rather have phrases [Mei et al., 2007b], sentences [Smith et al., 2016], or pictures [Lau et al., 2014]. However, providing these representations is non-trivial and requires a deeper understanding of a corpus than today's topic models can manage.

Finally, topic models need a more systematic investigation of how they can assist users' workflow for typical information seeking, organization, and management tasks. While the applications covered in this survey show examples of how people can use topic models from applications from history to political science, how topic models can augment or replace existing workflows lacks the same attention given to—for example—search engines.

10.4 Coda

We hope that you have enjoyed our survey of topic models' applications. For further information, we would encourage the reader to investigate the topic modeling bibliography,[1] join the topic modeling mailing list,[2] or the book's associated webpage.

[1] `https://mimno.infosci.cornell.edu/topics.html`
[2] `https://lists.cs.princeton.edu/mailman/listinfo/topic-models`

References

Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.

Nikolaos Aletras, Timothy Baldwin, Jey Han Lau, and Mark Stevenson. Representing topics labels for exploring digital libraries. In *Proceedings of the IEEE/ACM Joint Conference on Digital Libraries*, pages 239–248, 2014.

Mark Algee-Hewitt, Ryan Heuser, and Franco Moretti. On paragraphs. scale, themes, and narrative form. *Stanford Literary Lab Pamphlets*, 1(10), October 2015.

James Allan, editor. *Topic Detection and Tracking: Event-based Information Organization*. Springer, 2002.

Loulwah AlSumait, Daniel Barbará, and Carlotta Domeniconi. On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *International Conference on Data Mining*, 2008.

Anima Anandkumar, Dean P Foster, Daniel J Hsu, Sham M Kakade, and Yi kai Liu. A spectral algorithm for latent Dirichlet allocation. In *Proceedings of Advances in Neural Information Processing Systems*, 2012.

David Andrzejewski and David Buttler. Latent topic feedback for information retrieval. In *Knowledge Discovery and Data Mining*, 2011.

David Andrzejewski, Xiaojin Zhu, and Mark Craven. Incorporating domain knowledge into topic modeling via Dirichlet forest priors. In *Proceedings of the International Conference of Machine Learning*, 2009.

Sanjeev Arora, Rong Ge, Yoni Halpern, David M. Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. A practical algorithm for topic modeling with provable guarantees. In *Proceedings of the International Conference of Machine Learning*, 2013.

Anton Bakalov, Andrew Kachites McCallum, Hanna Wallach, and David Mimno. Topic models for taxonomies. In *Joint Conference on Digital Libraries*, 2012.

Kayhan Batmanghelich, Ardavan Saeedi, Karthik Narasimhan, and Samuel Gershman. Nonparametric spherical topic modeling with word embeddings. In *Proceedings of the Association for Computational Linguistics*, 2016.

Eric P. S. Baumer, David Mimno, Shion Guha, Emily Quan, and Geri K. Gay. Comparing grounded theory and topic modeling: Extreme divergence or unlikely convergence? *Journal of the Association for Information Science and Technology*, 2017.

Jerome R. Bellegarda. A latent semantic analysis framework for large-span language modeling. In *European Conference on Speech Communication and Technology*, 1997.

Jerome R. Bellegarda. Statistical language model adaptation: review and perspectives. volume 42, 2004.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March 2003.

Adam Berger and John Lafferty. Information retrieval as statistical translation. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999.

Indrajit Bhattacharya. Collective entity resolution in relational data. *PhD Dissertation, University of Maryland, College Park*, 2006.

David M. Blei. Topic modeling and digital humanities. *Journal of Digital Humanities*, 2(1), 2012.

David M. Blei and John Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, pages 17–35, 2007.

David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the International Conference of Machine Learning*, 2006.

David M. Blei and Jon D. McAuliffe. Supervised topic models. In *Proceedings of Advances in Neural Information Processing Systems*, 2007.

David M. Blei, Andrew Ng, and Michael Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.

David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*, 57(2):7:1–7:30, February 2010.

Shannon Bowen. Pseudo-events pay dividends from Cleopatra to Chipotle. *Public Relations Week*, 2016.

George E.P. Box and Norman R. Draper. *Empirical model-building and response surfaces*. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley, 1987.

Jordan Boyd-Graber and David M. Blei. Multilingual topic models for unaligned text. In *Proceedings of Uncertainty in Artificial Intelligence*, 2009.

Jordan Boyd-Graber and Philip Resnik. Holistic sentiment analysis across languages: Multilingual supervised latent Dirichlet allocation. In *Proceedings of Empirical Methods in Natural Language Processing*, 2010.

Jordan Boyd-Graber, David M. Blei, and Xiaojin Zhu. A topic model for word sense disambiguation. In *Proceedings of Empirical Methods in Natural Language Processing*, 2007.

Jordan Boyd-Graber, David Mimno, and David Newman. *Care and Feeding of Topic Models: Problems, Diagnostics, and Improvements*. CRC Handbooks of Modern Statistical Methods. CRC Press, Boca Raton, Florida, 2014.

Percy Williams Bridgman. *The logic of modern physics*. Macmillan, New York, 1927.

Samuel Brody and Mirella Lapata. Bayesian word sense induction. In *Proceedings of the European Chapter of the Association for Computational Linguistics*, 2009.

Andre David Broniatowski, Mark Dredze, J. Michael Paul, and Andrea Dugas. Using social media to perform local influenza surveillance in an inner-city hospital: A retrospective observational study. *JMIR Public Health and Surveillance*, 1(1):e5, May 2015.

John Burrows. Delta: a measure of stylistic difference and a guide to likely authorship. *Lit Linguist Computing*, 17(3):267–287, 2002.

Jaime G Carbonell, Yiming Yang, Robert E Frederking, Ralf D Brown, Yibing Geng, and Danny Lee. Translingual information retrieval: A comparative evaluation. In *International Joint Conference on Artificial Intelligence*, 1997.

Mark James Carman, Fabio Crestani, Morgan Harvey, and Mark Baillie. Towards query log based personalization using topic models. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, 2010.

Youngchul Cha and Junghoo Cho. Social-network analysis using topic models. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2012.

Allison Chaney and David M. Blei. Visualizing topic models. In *International AAAI Conference on Weblogs and Social Media*, 2012.

Jonathan Chang and David M. Blei. Relational topic models for document networks. In *Proceedings of Artificial Intelligence and Statistics*, 2009.

Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *Proceedings of Advances in Neural Information Processing Systems*, 2009.

Boxing Chen, George Foster, and Roland Kuhn. Adaptation of reordering models for statistical machine translation. In *Proceedings of the Association for Computational Linguistics*, 2013.

Stanley Chen, Douglas Beeferman, and Ronald Rosenfeld. Evaluation metrics for language models. Technical report, Carnegie Mellon University School of Computer Science, 1998.

David Chiang, Steve DeNeefe, and Michael Pust. Two easy improvements to lexical weighting. In *Proceedings of the Human Language Technology Conference*, 2011.

Jaegul Choo, Changhyun Lee, Chandan K. Reddy, and Haesun Park. UTOPIAN: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1992–2001, 2013.

Jason Chuang, Christopher D. Manning, and Jeffrey Heer. Termite: Visualization techniques for assessing textual topic models. In *Advanced Visual Interfaces*, 2012.

Jason Chuang, Margaret E. Roberts, Brandon M. Stewart, Rebecca Weiss, Dustin Tingley, Justin Grimmer, and Jeffrey Heer. TopicCheck: Interactive alignment for assessing topic model stability. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2015.

Philip R. Clarkson and Anthony J. Robinson. Language model adaptation using mixtures and an exponentially decaying cache. In *International Conference on Acoustics, Speech, and Signal Processing*, 1997.

Noah Coccaro and Daniel Jurafsky. Towards better integration of semantic predictors in statistical language modeling. In *International Conference on Acoustics, Speech, and Signal Processing*, 1998.

Ronan Collobert, Koray Kavukcuoglu, and Clement Farabet. Torch7: A Matlab-like environment for machine learning. In *NIPS Workshop on Big Learning (Biglearn)*, 2011.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.

W. Bruce Croft and John Lafferty. Language modeling for information retrieval. In *Kluwer International Series on Information Retrieval*, 2003.

Van Dang and W. Bruce Croft. Term level search result diversification. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2013.

Hal Daumé III. Markov random topic fields. In *Proceedings of Artificial Intelligence and Statistics*, 2009.

Wim De Smet and Marie-Francine Moens. Cross-language linking of news stories on the web using interlingual topic modelling. In *Workshop on Social Web Search and Mining*, 2009.

Scott Deerwester, Susan Dumais, Thomas Landauer, George Furnas, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the Association for Computational Linguistics*, 2014.

Laura Dietz, Steffen Bickel, and Tobias Scheffer. Unsupervised prediction of citation influences. In *Proceedings of the International Conference of Machine Learning*, 2007.

Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of World Wide Web Conference*, 2007.

Vladimir Eidelman, Jordan Boyd-Graber, and Philip Resnik. Topic models for dynamic translation model adaptation. In *Proceedings of the Association for Computational Linguistics*, 2012.

Jacob Eisenstein. *Written dialect variation in online social media*. Wiley, 2017.

Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric P. Xing. A latent variable model for geographic lexical variation. In *Proceedings of Empirical Methods in Natural Language Processing*, 2010.

Jacob Eisenstein, Duen Horng Chau, Aniket Kittur, and Eric Xing. TopicViz: interactive topic exploration in document collections. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems*, 2012.

Jacob Eisenstein, Iris Sun, and Lauren F. Klein. Exploratory text analysis for large document archives. In *Digital Humanities*, 2014.

Matt Erlin. Topic modeling, epistemology, and the English and German novel. *Cultural Analytics*, May 2017.

George Foster and Roland Kuhn. Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, 2007.

Jianfeng Gao, Kristina Toutanova, and Wen tau Yih. Clickthrough-based latent semantic models for web search. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2011.

Jianfeng Gao, Shasha Xie, Xiaodong He, and Alnur Ali. Learning lexicon models from search logs for query expansion. In *Proceedings of Empirical Methods in Natural Language Processing*, 2012.

Matthew Gardner, Joshua Lutes, Jeff Lund, Josh Hansen, Dan Walker, Eric Ringger, and Kevin Seppi. The topic browser: An interactive tool for browsing topic models. In *NIPS Workshop on Challenges of Data Visualization*, 2010.

Sean Gerrish and David M. Blei. A language-based approach to measuring scholarly impact. In *Proceedings of the International Conference of Machine Learning*, 2010.

Daniel Gildea and Thomas Hofmann. Topic-based language models using EM. In *European Conference on Speech Communication and Technology*, 1999.

Barney G. Glaser and Anslem Strauss. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine, 1967.

Andrew Goldstone and Ted Underwood. The quiet transformations of literary studies: What thirteen thousand scholars could tell us. *New Literary History*, 45(3), Summer 2014.

Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl 1):5228–5235, 2004.

Justin Grimmer. A Bayesian Hierarchical Topic Model for Political Texts: Measuring Expressed Agendas in Senate Press Releases. *Political Analysis*, 18(1):1–35, 2010.

Amit Gruber, Michael Rosen-Zvi, and Yair Weiss. Hidden topic Markov models. In *Artificial Intelligence and Statistics*, 2007.

Eric Hardisty, Jordan Boyd-Graber, and Philip Resnik. Modeling perspective using adaptor grammars. In *Proceedings of Empirical Methods in Natural Language Processing*, 2010.

Jacob Harris. Word clouds considered harmful. `http://www.niemanlab.org/2011/10/word-clouds-considered-harmful/`, 2011.

Morgan Harvey, Fabio Crestani, and Mark James Carman. Building user profiles from topic models for personalised search. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, 2013.

Eva Hasler, Barry Haddow, and Philipp Koehn. Sparse lexicalised features and topic adaptation for SMT. In *Proceedings of International Workshop on Spoken Language Translation*, 2012.

Qi He, Bi Chen, Jian Pei, Baojun Qiu, Prasenjit Mitra, and Lee Giles. Detecting topic evolution in scientific literature: How can citations help? In *Proceedings of the ACM International Conference on Information and Knowledge Management*, 2009.

Lynette Hirschman and Rob Gaizauskas. Natural language question answering: The view from here. *Natural Language Engineering*, 7(4):275–300, December 2001.

Matthew Hoffman, David M. Blei, and Francis Bach. Online learning for latent Dirichlet allocation. In *Proceedings of Advances in Neural Information Processing Systems*, 2010.

Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence*, 1999a.

Thomas Hofmann. Probabilistic latent semantic indexing. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999b.

Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

Liangjie Hong and Brian D. Davison. Empirical study of topic modeling in Twitter. In *Proceedings of the First Workshop on Social Media Analytics*, 2010.

Diane Hu and Lawrence K. Saul. A probabilistic model of unsupervised learning for musical-key profiles. In *International Society for Music Information Retrieval Conference*, 2009.

Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith. Interactive topic modeling. *Machine Learning Journal*, 95(3):423–469, June 2014a.

Yuening Hu, Ke Zhai, Vladimir Edelman, and Jordan Boyd-Graber. Polylingual tree-based topic models for translation domain adaptation. In *Association for Computational Linguistics*, 2014b.

Rukmini Iyer and Mari Ostendorf. Modeling long distance dependencies in language: topic mixtures versus dynamic cache models. *IEEE Transactions on Speech Audio Process*, 7:236–239, 1999.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Association for Computational Linguistics*, 2015.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *North American Association for Computational Linguistics*, 2016.

Bernard J. Jansen, Amanda Spink, and Tefko Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing and Management*, 36(2):207–227, 2000.

Fred Jelinek and Robert L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, 1980.

Shan Jiang, Yuening Hu, Changsung Kang, Tim Daly, Dawei Yin, and Yi Chang. Learning query and document relevance from a web-scale click graph. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2016.

Yohan Jo and Alice H. Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of ACM International Conference on Web Search and Data Mining*, 2011.

Matthew L. Jockers. *Macroanalysis: Digital Methods and Literary History*. Topics in the Digital Humanities. University of Illinois Press, 2013.

Matthew L. Jockers and David Mimno. Significant themes in 19th century literature. *Poetics*, 41(6):750–769, December 2013.

Patrick Juola. Authorship attribution. *Foundations and Trends in information Retrieval*, 1(3):233–334, 2006.

Saurabh S. Kataria, Krishnan S. Kumar, Rajeev R. Rastogi, Prithviraj Sen, and Srinivasan H. Sengamedu. Entity disambiguation with hierarchical topic models. In *Knowledge Discovery and Data Mining*, 2011.

S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. In *IEEE Transaction on Acoustics, Speech and Signal Processing*, 1987.

Kirill Kireyev, Leysia Palen, and Kenneth Anderson. Applications of Topics Models to Analysis of Disaster-Related Twitter Data. December 2009.

Reinhard Kneser and Jochen Peters. Semantic clustering for adaptive language modeling. In *International Conference on Acoustics, Speech, and Signal Processing*, 1997.

Reinhard Kneser, Jochen Peters, and Dietrich Klakow. Language model adaptation using dynamic marginals. In *European Conference on Speech Communication and Technology*, 1997.

Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2009.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2003.

Thomas K Landauer and Michael L Littman. Fully automatic cross-language document retrieval using latent semantic indexing. In *Proceedings of the UW Centre for the New Oxford English Dictionary*, 1990.

John Langford, Lihong Li, and Alex Strehl. Vowpal Wabbit, 2007.

Mark A. Largent and Julia I. Lane. STAR METRICS and the science of science policy. *Review of Policy Research*, 29(3):431–438, 2012.

Jey Han Lau, David Newman, Sarvnaz Karimi, and Timothy Baldwin. Best topic word selection for topic labelling. In *Proceedings of International Conference on Computational Linguistics*, 2010.

Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic labelling of topic models. In *Proceedings of the Association for Computational Linguistics*, 2011.

Jey Han Lau, David Newman, and Timothy Baldwin. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the European Chapter of the Association for Computational Linguistics*, 2014.

Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001.

Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Knowledge Discovery and Data Mining*, 2009.

Aaron Q Li, Amr Ahmed, Sujith Ravi, and Alexander J Smola. Reducing the sampling complexity of topic models. In *Knowledge Discovery and Data Mining*, 2014.

Percy Liang and Dan Klein. Structured Bayesian nonparametric models with variational inference (tutorial). In *Proceedings of the Association for Computational Linguistics*, 2007.

Shangsong Liang, Zhaochun Ren, and Maarten de Rijke. Personalized search result diversification via structured learning. In *Knowledge Discovery and Data Mining*, 2014.

Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, 2009.

Tianyi Lin, Wentao Tian, Qiaozhu Mei, and Hong Cheng. The dual-sparse topic model: Mining focused topics and focused terms in short text. In *Proceedings of World Wide Web Conference*, 2014.

Yan Liu, Alexandru Niculescu-Mizil, and Wojciech Gryc. Topic-link LDA: Joint models of topic and author community. In *Proceedings of the International Conference of Machine Learning*, 2009.

Yue Lu and Chengxiang Zhai. Opinion integration through semi-supervised topic modeling. In *Proceedings of World Wide Web Conference*, 2008.

Yue Lu, Qiaozhu Mei, and ChengXiang Zhai. Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA. *Information Retrieval*, 14(2):178–203, 2011.

Jeff Lund, Connor Cook, Kevin Seppi, and Jordan Boyd-Graber. Tandem anchoring: A multiword anchor approach for interactive topic modeling. In *Association for Computational Linguistics*, 2017.

David J. C. Mackay and Linda C. Bauman Peto. A hierarchical Dirichlet language model. *Natural Language Engineering*, 1:1–19, 1995.

Gideon Mann, David Mimno, and Andrew Kachites McCallum. Bibliometric impact measures leveraging topic analysis. In *Joint Conference on Digital Libraries*, 2006.

Xian-Ling Mao, Zhao-Yan Ming, Zheng-Jun Zha, Tat-Seng Chua, Hongfei Yan, and Xiaoming Li. Automatic labeling hierarchical topics. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, 2012.

Mitchell P. Marcus, Beatrice Santorini, and Mary A. Marcinkiewicz. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

Girish Maskeri, Santonu Sarkar, and Kenneth Heafield. Mining business topics in source code using latent dirichlet allocation. In *India Software Engineering Conference*, 2008.

Spyros Matsoukas, Antti-Veikko I. Rosti, and Bing Zhang. Discriminative corpus weight estimation for machine translation. In *Proceedings of Empirical Methods in Natural Language Processing*, 2009.

Andrew Kachites McCallum. Mallet: A machine learning for language toolkit, 2002. http://www.cs.umass.edu/ mccallum/mallet.

Andrew Kachites McCallum, Xuerui Wang, and Andrés Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research*, 30(1):249–272, October 2007.

Rishabh Mehrotra, Scott Sanner, Wray Buntine, and Lexing Xie. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2013.

Qiaozhu Mei and ChengXiang Zhai. Discovering evolutionary theme patterns from text: An exploration of temporal text mining. In *Knowledge Discovery and Data Mining*, 2005.

Qiaozhu Mei, Chao Liu, Hang Su, and ChengXiang Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proceedings of World Wide Web Conference*, 2006.

Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of World Wide Web Conference*, 2007a.

Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. Automatic labeling of multinomial topic models. In *Knowledge Discovery and Data Mining*, 2007b.

Qiaozhu Mei, Deng Cai, Duo Zhang, and ChengXiang Zhai. Topic modeling with network regularization. In *Proceedings of World Wide Web Conference*, 2008.

Massimo Melucci. Contextual search: A computational framework. *Foundations and Trends in Information Retrieval*, 6:257–405, 2012.

Alessandro Micarelli, Fabio Gasparetti, Filippo Sciarrone, and Susan Gauch. Personalized search on the world wide web. In *The Adaptive Web*, volume 4321, 2007.

Ian Matthew Miller. Rebellion, crime and violence in qing china, 17221911: A topic modeling approach. *Poetics*, 41(6):626–649, December 2013.

David Mimno. Computational historiography: Data mining in a century of classics journals. *Journal on Computing and Cultural Heritage*, 5(1):3:1–3:19, April 2012.

David Mimno and David M. Blei. Bayesian checking for topic models. In *Proceedings of Empirical Methods in Natural Language Processing*, 2011.

David Mimno and Andrew Kachites McCallum. Topic Models Conditioned on Arbitrary Features with Dirichlet-multinomial Regression. In *Proceedings of the 2008 Conference on Uncertainty in Artificial Intelligence (UAI)*, 2008.

David Mimno, Hanna Wallach, Jason Naradowsky, David Smith, and Andrew Kachites McCallum. Polylingual topic models. In *Proceedings of Empirical Methods in Natural Language Processing*, 2009.

David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew Kachites McCallum. Optimizing semantic coherence in topic models. In *Proceedings of Empirical Methods in Natural Language Processing*, 2011.

David Mimno, Matthew Hoffman, and David M. Blei. Sparse stochastic inference for latent Dirichlet allocation. In *Proceedings of the International Conference of Machine Learning*, 2012.

Tom Minka, John Winn, John Guiver, and David Knowles. Infer.NET 2.6, 2014. Microsoft Research Cambridge. http://research.microsoft.com/infernet.

Franco Moretti. The slaughterhouse of literature. *Modern Language Quarterly*, 61(1):207–227, 2000.

Franco Moretti. *Distant Reading*. Verso, 2013a. URL `https://books.google.com/books?id=YKMCy9I3PG4C`.

Franco Moretti. Operationalizing, or the function of measurement in literary theory. *New Left Review*, 84, Nov/Dec 2013b.

Frederick Mosteller and David L. Wallace. *Inference and Disputed Authorship: The Federalist*. Addison-Wesley, Reading, Mass., 1964.

Christof Müller and Iryna Gurevych. A study on the semantic relatedness of query and document terms in information retrieval. In *Proceedings of Empirical Methods in Natural Language Processing*, 2009.

Ramesh Nallapati and William Cohen. Link-PLSA-LDA: A new unsupervised model for topics and influence of blogs. In *International Conference on Weblogs and Social Media*, 2008.

Shravan Narayanamurthy. Yahoo! LDA, 2011. URL `https://github.com/shravanmn/Yahoo_LDA/wiki`.

Radford M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993.

David Newman and Sharon Block. Probabilistic topic decomposition of an eighteenth-century american newspaper. *Journal of the American Society for Information Science and Technology*, 18(1):753–767, 2006.

David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. Distributed Inference for Latent Dirichlet Allocation. In *Proceedings of Advances in Neural Information Processing Systems*, 2008.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2010.

Hermann Ney, Ute Essen, and Reinhard Kneser. On structuring probabilistic dependencies in stochastic language modelling. *Computer Speech and Language*, 8:1–38, 1994.

Thang Nguyen, Jordan Boyd-Graber, Jeff Lund, Kevin Seppi, and Eric Ringger. Is your anchor going up or down? Fast and accurate supervised topic models. In *North American Association for Computational Linguistics*, 2015a.

Viet-An Nguyen, Jordan Boyd-Graber, and Philip Resnik. Lexical and hierarchical topic regression. In *Proceedings of Advances in Neural Information Processing Systems*, 2013.

Viet-An Nguyen, Jordan Boyd-Graber, Philip Resnik, and Jonathan Chang. Learning a concept hierarchy from multi-labeled documents. In *Neural Information Processing Systems*, 2014.

Viet-An Nguyen, Jordan Boyd-Graber, Philip Resnik, and Kristina Miler. Tea party in the house: A hierarchical ideal point topic model and its application to Republican legislators in the 112th Congress. In *Association for Computational Linguistics*, 2015b.

Xiaochuan Ni, Jian-Tao Sun, Jian Hu, and Zheng Chen. Mining multilingual topics from Wikipedia. In *Proceedings of World Wide Web Conference*, 2009.

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

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the Web. Stanford Digital Library Working Paper SIDL-WP-1999-0120, Stanford University, 1999.

Bo Pang and Lillian Lee. *Opinion Mining and Sentiment Analysis*. Now Publishers Inc, 2008.

Christos H. Papadimitriou, Prabhakar Raghavan, Hisao Tamaki, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. *Journal of Computer and System Sciences*, 61(2):217 – 235, 2000.

Laurence A. Park and Kotagiri Ramamohanarao. The sensitivity of latent dirichlet allocation for information retrieval. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, 2009.

Michael Paul and Roxana Girju. A two-dimensional topic-aspect model for discovering multi-faceted topics. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, 2010.

James Pitkow, Hinrich Schütze, Todd Cass, Rob Cooley, Don Turnbull, Andy Edmonds, Eytan Adar, and Thomas Breuel. Personalized search. *Communications of the ACM*, 45(9):50–55, September 2002.

Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998.

Jonathan K. Pritchard, Matthew Stephens, and Peter Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155:945–959, 2000.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of Empirical Methods in Natural Language Processing*, 2009.

Daniel Ramage, Susan T. Dumais, and Daniel J. Liebling. Characterizing microblogs with topic models. In *International Conference on Weblogs and Social Media*, 2010a.

Daniel Ramage, Christopher D. Manning, and Daniel A. Mcfarland. Which universities lead and lag? toward university rankings based on scholarly output. In *NIPS Workshop on Computational Social Science and the Wisdom of the Crowds*, 2010b.

Rajesh Ranganath, Linpeng Tang, Laurent Charlin, and David M. Blei. Deep exponential families. In *Proceedings of Artificial Intelligence and Statistics*, 2015.

Philip Resnik and Eric Hardisty. Gibbs sampling for the uninitiated. Technical report, University of Maryland, 2009. URL `http://www.umiacs.umd.edu/~resnik/pubs/gibbs.pdf`.

Lia M. Rhody. Topic modeling and figurative language. *Journal of Digital Humanities*, 2(1), 2012.

Allen Beye Riddell. *How to Read 22,198 Journal Articles: Studying the History of German Studies with Topic Models*, pages 91–113. Camden House, 2012.

Margaret E. Roberts, Brandon M. Stewart, and Dustin Tingley. STM: R package for structural topic models, 2014. URL `http://www.structuraltopicmodel.com`. R package version 1.0.8.

Joseph John Rocchio. *Relevance feedback in information retrieval*, pages 313–323. Englewood Cliffs, NJ: Prentice-Hall, 1971.

Michal Rosen-Zvi, Thomas L. Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of Uncertainty in Artificial Intelligence*, 2004.

Nick Ruiz and Marcello Federico. Topic adaptation for lecture translation through bilingual latent semantic models. In *WMT Workshop on Statistical Machine Translation*, 2011.

Gerard. Salton. *Automatic Information Organization and Retrieval.* McGraw Hill Text, 1968.

Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Search result diversification. *Foundations and Trends in Information Retrieval*, 9(1):1–90, March 2015.

Tobias Schnabel, Igor Labutov, David M. Mimno, and Thorsten Joachims. Evaluation methods for unsupervised word embeddings. In *Proceedings of Empirical Methods in Natural Language Processing*, 2015.

Kristie Seymore and Ronald Rosenfeld. Using story topics for language model adaptation. In *European Conference on Speech Communication and Technology*, 1997.

Kristie Seymore, Stanley F. Chen, and Ronald Rosenfeld. Nonlinear interpolation of topic models for language model adaptation. In *International Conference on Spoken Language Processing*, 1998.

Alison Smith, Jason Chuang, Yuening Hu, Jordan Boyd-Graber, and Leah Findlater. Concurrent visualization of relationships between words and topics in topic models. In *ACL Workshop on Workshop on Interactive Language Learning, Visualization, and Interfaces*, 2014.

Alison Smith, Sana Malik, and Ben Shneiderman. *Visual analysis of topical evolution in unstructured text: Design and evaluation of topicflow*, pages 159–175. Springer, 2015.

Alison Smith, Tak Yeon Lee, Forough Poursabzi-Sangdeh, Leah Findlater, Jordan Boyd-Graber, and Niklas Elmqvist. Evaluating visual representations for topic understanding and their effects on manually generated labels. *Transactions of the Association for Computational Linguistics*, 2016.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of Empirical Methods in Natural Language Processing*, 2012.

Fei Song and W. Bruce Croft. A general language model for information retrieval. In *International Conference on Information and Knowledge Management*, 1999.

Wei Song, Yu Zhang, Ting Liu, and Sheng Li. Bridging topic modeling and personalized search. In *Proceedings of International Conference on Computational Linguistics*, 2010.

Stan Development Team. Stan: A C++ library for probability and sampling, version 2.5.0, 2014. URL `http://mc-stan.org/`.

Mark Steyvers, Padhraic Smyth, Michal Rosen-Zvi, and Thomas Griffiths. Probabilistic author-topic models for information discovery. In *Knowledge Discovery and Data Mining*, 2004.

Jinsong Su, Hua Wu, Haifeng Wang, Yidong Chen, Xiaodong Shi, Huailin Dong, and Qun Liu. Translation model adaptation for statistical machine translation with monolingual topic information. In *Proceedings of the Association for Computational Linguistics*, 2012.

Yizhou Sun, Jiawei Han, Jing Gao, and Yintao Yu. iTopicModel: Information network-integrated topic modeling. In *International Conference on Data Mining*, 2009.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

Rick Szostak. Classifying science. *Classifying Science: Phenomena, Data, Theory, Method, Practice*, pages 1–22, 2004.

Edmund M. Talley, David Newman, David Mimno, Bruce W. Herr, Hanna M. Wallach, Gully A. P. C. Burns, A. G. Miriam Leenders, and Andrew Kachites McCallum. Database of NIH grants using machine-learned categories and graphical clustering. *Nature Methods*, 8(6):443–444, May 2011.

Yik-Cheung Tam, Ian Lane, and Tanja Schultz. Bilingual-lsa based lm adaptation for spoken language translation. In *Proceedings of the Association for Computational Linguistics*, 2007.

Jian Tang, Zhaoshi Meng, XuanLong Nguyen, Qiaozhu Mei, and Ming Zhang. Understanding the limiting factors of topic modeling via posterior contraction analysis. In *Proceedings of the International Conference of Machine Learning*, 2014.

Timothy R. Tangherlini and Peter Leonard. Trawling in the sea of the great unread: Sub-corpus topic modeling and humanities research. *Poetics*, 41(6): 725–749, December 2013.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

Ivan Titov and Ryan McDonald. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of the Association for Computational Linguistics*, 2008.

Kristina Toutanova and Mark Johnson. A Bayesian LDA-based model for semi-supervised part-of-speech tagging. In *Proceedings of Advances in Neural Information Processing Systems*, 2008.

David Vallet and Pablo Castells. Personalized diversification of search results. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2012.

Fernanda B. Viégas and Martin Wattenberg. Tag clouds and the case for vernacular visualization. *Interactions*, 15(4):49–52, 2008.

Maximilian Viermetz, Michal Skubacz, Cai-Nicolas Ziegler, and Dietmar Seipel. Tracking topic evolution in news environments. In *IEEE International Conference on E-Commerce Technology*, 2008.

Ellen M. Voorhees. Overview of TREC 2003. In *Proceedings of the Text REtrieval Conference*, pages 1–13, 2003.

Ellen M. Voorhees and Donna K. Harman. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, 2005.

Jan Vosecky, Kenneth Wai-Ting Leung, and Wilfred Ng. Collaborative personalized Twitter search with topic-language models. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2014.

Ivan Vulić and Marie-Francine Moens. Detecting highly confident word translations from comparable corpora without any prior knowledge. In *Proceedings of the European Chapter of the Association for Computational Linguistics*, 2012.

Ivan Vulić and Marie-Francine Moens. Probabilistic models of cross-lingual semantic similarity in context based on latent cross-lingual concepts induced from comparable data. In *Proceedings of Empirical Methods in Natural Language Processing*, 2014.

Ivan Vulić, Wim De Smet, and Marie-Francine Moens. Cross-language information retrieval with latent topic models trained on a comparable corpus. In *Asia Information Retrieval Societies*, 2011a.

Ivan Vulić, Wim De Smet, and Marie-Francine Moens. Identifying word translations from comparable corpora using latent topic models. In *Proceedings of the Association for Computational Linguistics*, 2011b.

Ivan Vulić, Wim De Smet, and Marie-Francine Moens. Cross-language information retrieval models based on latent topic models trained with document-aligned comparable corpora. *Information Retrieval*, 16(3):331–368, 2013.

Ivan Vulić, Wim De Smet, Jie Tang, and Marie-Francine Moens. Probabilistic topic modeling in multilingal settings: An overview of its methodology and applications. *Information Processing and Management*, 51(1), 2015.

Hanna Wallach, David Mimno, and Andrew Kachites McCallum. Rethinking LDA: Why priors matter. In *Proceedings of Advances in Neural Information Processing Systems*, 2009a.

Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the International Conference of Machine Learning*, 2009b.

Chong Wang, David M. Blei, and David Heckerman. Continuous time dynamic topic models. In *Proceedings of Uncertainty in Artificial Intelligence*, 2008.

Quan Wang, Jun Xu, Hang Li, and Nick Craswell. Regularized latent semantic indexing: A new approach to large-scale topic modeling. *ACM Transactions on Information Systems*, 31(1):5:1–5:44, January 2013.

Shiliang Wang, J. Michael Paul, and Mark Dredze. Social media as a sensor of air quality and public response in China. *Journal of Medical Internet Research*, 17(3):e22, Mar 2015.

Xing Wang, Deyi Xiong, Min Zhang, Yu Hong, and Jianmin Yao. A topic-based reordering model for statistical machine translation. *Natural Language Processing and Chinese Computing*, 496:414–421, 2014.

Xing Wei. *Topic Models in Information Retrieval*. Ph.D. dissertation, University of Massachusetts Amherst, 2007.

Xing Wei and W. Bruce Croft. LDA-based document models for ad-hoc retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006.

Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. TwitterRank: Finding topic-sensitive influential Twitterers. In *Proceedings of ACM International Conference on Web Search and Data Mining*, 2010.

Theresa Wilson and Janyce Wiebe. Annotating attributions and private states. In *CorpusAnno '05: Proceedings of the Workshop on Frontiers in Corpus Annotations II*, 2005.

Frank Wood and Yee Whye Teh. A hierarchical nonparametric Bayesian approach to statistical language model domain adaptation. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 12, 2009.

Xinyan Xiao, Deyi Xiong, Min Zhang, Qun Liu, and Shouxun Lin. A topic similarity model for hierarchical phrase-based translation. In *Proceedings of the Association for Computational Linguistics*, 2012.

Deyi Xiong and Min Zhang. A topic-based coherence model for statistical machine translation. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, 2013.

Deyi Xiong, Qun Liu, and Shouxun Lin. Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of the Association for Computational Linguistics*, 2006.

Tze-I Yang, Andrew J. Torget, and Rada Mihalcea. Topic modeling on historical newspapers. In *ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, 2011.

Limin Yao, David Mimno, and Andrew Kachites McCallum. Efficient methods for topic model inference on streaming document collections. In *Knowledge Discovery and Data Mining*, 2009.

Xing Yi and James Allan. A comparative study of utilizing topic models for information retrieval. In *Proceedings of the European Conference on Information Retrieval*, volume 5478, 2009.

Zhijun Yin, Liangliang Cao, Jiawei Han, Chengxiang Zhai, and Thomas Huang. Geographical topic discovery and comparison. In *Proceedings of World Wide Web Conference*, 2011.

Heng Yu, Jinsong Su, Yajuan Lv, and Qun Liu. A topic-triggered language model for statistical machine translation. In *International Joint Conference on Natural Language Processing*, 2013.

Jia Zeng, W. K. Cheung, and Jiming Liu. Learning topic models by belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5):1121–1134, 2013.

Qing T. Zeng, Doug Redd, Thomas C. Rindflesch, and Jonathan R. Nebeker. Synonym, topic model and predicate-based query expansion for retrieving clinical documents. In *American Medical Informatics Association Annual Symposium*, 2012.

ChengXiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001a.

ChengXiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, 2001b.

ChengXiang Zhai, Atulya Velivelli, and Bei Yu. A cross-collection mixture model for comparative text mining. In *Knowledge Discovery and Data Mining*, 2004.

Ke Zhai, Jordan Boyd-Graber, Nima Asadi, and Mohamad Alkhouja. Mr. LDA: A flexible large scale topic modeling package using variational inference in mapreduce. In *Proceedings of World Wide Web Conference*, 2012.

Duo Zhang, Qiaozhu Mei, and ChengXiang Zhai. Cross-lingual latent topic extraction. In *Proceedings of the Association for Computational Linguistics*, 2010.

Bing Zhao and Eric P. Xing. BiTAM: Bilingual topic admixture models for word alignment. In *Proceedings of the Association for Computational Linguistics*, 2006.

Wayne Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In *Proceedings of Empirical Methods in Natural Language Processing*, 2010.

Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing Twitter and traditional media using topic models. In *Proceedings of the European Conference on Information Retrieval*, 2011.

Ding Zhou, Xiang Ji, Hongyuan Zha, and C. Lee Giles. Topic evolution and social interactions: How authors effect research. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, 2006.

Jun Zhu, Amr Ahmed, and Eric P. Xing. MedLDA: maximum margin supervised topic models for regression and classification. In *Proceedings of the International Conference of Machine Learning*, 2009.

Index