TOPIC MODELING IN THEORY AND PRACTICE

by
Chandler Camille May

A dissertation submitted to Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland
March 2022

© 2022 Chandler May
All rights reserved

Abstract

Topic models can decompose a large corpus of text into a relatively small set of interpretable themes or topics, potentially enabling a domain expert to explore and analyze a corpus more efficiently. However, in my work, I have found that theories put forth by topic modeling research are not always borne out in practice. In this dissertation, I use case studies to explore four theories of topic modeling. While these theories are not explicitly stated, I show that they are communicated implicitly, some within an individual study and others more diffusely. I show that this implicit knowledge fails to hold in practice in the settings I consider. While my work is confined to topic modeling research and moreover concentrated on the latent Dirichlet allocation topic model, I argue that these kinds of gaps may pervade scientific research and present an obstacle to improving the diversity of the research community.

Primary Reader and Advisor: Benjamin Van Durme

Secondary Reader: Mark Dredze, David Yarowsky

Acknowledgments

At lot has changed since I started my Ph.D., and I am at a loss as to how to acknowledge all of the people who have brought me to where I am now.

To my advisor, Ben Van Durme: Thank you for your incredible advising and support. Thank you for letting me take your ideas and run with them, and thank you for your openness to my own. Thank you for your enthusiasm when things went well and encouragement when they didn't. Thank you for taking what felt like failures and turning them into an inside joke—and then a dissertation topic. I wish everyone could have as good an experience with their advisor as I did with you.

To my readers, David Yarowsky and Mark Dredze: Thank you for your openness to such an unusual dissertation topic and for your always thoughtful and timely feedback. Mark, thank you for your similar openness and feedback in my GBO.

To my advisors during my internship at MSR, Hanna Wallach, Been Kim, and Solon Barocas: Thank you so much for the opportunity you gave me, for such an intriguing problem, and for your excellent mentorship.

To my coauthors and colleagues, including Shikha Bordia, Sam Bowman, Jordan

ACKNOWLEDGMENTS

Boyd-Graber, Annabelle Carrell, Tongfei Chen, Yunmo Chen, Alex Clemmer, Cash Costello, Ryan Cotterell, Kevin Duh, Frank Ferraro, Daniel Garcia-Romero, Shudong Hao, Craig Harman, Nils Holzenberger, Os Keyes, Ashwin Lall, Tom Lippincott, Patrick Martin, Alan McCree, Adam Poliak, Guanghui Qin, Rachel Rudinger, Max Thomas, Ben Van Durme, Siddharth Vashishtha, Alex Wang, Aaron White, Jonathan Wintrode, Travis Wolfe, Patrick Xia, Mahsa Yarmohammadi, and Michelle Yuan: I am so grateful for your collaboration. Thank you for sharing your knowledge and experience, for helping me when I needed it, and for your flexibility and positivity.

To the current and past administrative staff, including Ruth Scally, Kim Franklin, Jennifer Linton, Yamese Diggs, Carl Pupa, Debbie DeFord, Zack Burwell, and Cathy Thornton: Thank you so much for keeping CLSP and Computer Science running smoothly; for having such a wonderful impact on the culture here; and for listening to our problems, solving them, and encouraging us to be our best selves.

To my counselor, Ro, who has helped me grow, navigate the toughest situations, and find a strength I couldn't have imagined eight years ago: Thank you so much for your help. I don't know how I could have gotten this far without you.

To my friends and acquaintances, Pamela and Kira, Sarah and Rogue, Tricia, Madeleine, D. Scott and Spice, Lian, Annabelle, Amy, Rachel R., Rebecca, Keith, Katie, Huda, Naomi, Kate, Sabrina, Ben, Melanie, Tslil, Claire, Zoey, Patrick, Rachel S., Maggie, Lee, Karter Jayme, Craig, Frank, Max, Travis, Jonathan, Darcey, Wes, Juneki, Pushpendre, Courtney, Tim, Dee Ann, Nikita, Abbie, Megan, Mallory, Neal,

ACKNOWLEDGMENTS

Sarah, Max, Sarah, Ryan, Kristine, Doo, Andrew, Yannan, and others: I am so lucky to have you all in my life. Thank you for being here for me.

To all of my family, but especially to Mom, Dad, and Madeleine: Thank you for your love, support, kindness, and openness. I love you so much.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Natural language processing (NLP) is a largely empirical research field, but even empirical research relies on implicit theories. In unsupervised NLP, a wealth of research builds on the latent Dirichlet allocation (LDA) topic model, adding online learning, hierarchical structure, increased computational efficiency, and other improvements, or developing alternative topic modeling approaches. These lines of research generally treat LDA as a strong baseline model, a system capable of decomposing a corpus of text into human-interpretable topics. Hence, we might conclude there is an implicit theory that LDA does satisfactorily decompose text into human-interpretable topics; if there were not, we might expect the research following LDA to (at least initially) focus on getting it to "work" before pursuing other extensions. And while that theory is arguably well-supported—the efficacy of LDA is readily observable—a naive implementation and application of LDA to a text corpus typi-

cally yields a collection of relatively uninterpretable topics represented by words like "the," "of," "a," and "in." In practice, such *stop words* are generally filtered out of the corpus before training (or out of the resultant topics after training), but that process is considered a pre-processing (or post-processing) step, not a part of the LDA training algorithm itself, and it is rarely given much (if any) mention in the literature. Thus, while topic modeling research at large might appear to have an implicit theory that LDA automatically decomposes text, the usage of LDA *in practice* is slightly more complex than the literature suggests: It is only automatic if the necessary manual interventions are performed in concert.

Taking this example as a starting point, I examine several areas of topic modeling research, eliciting a simple implicit theory from each area and testing whether each theory holds in practice. Specifically, I examine the following theories:

- Rejuvenation enables streaming learning of LDA by particle filtering (Ch. 3).
- Under limited supervision, topic model features aid topic identification (Ch. 4).
- The effective usage of LDA is similar across languages (Ch. 5).
- The nested hierarchical Dirichlet process obviates the need for stop-word filtering (Ch. 6).

Topic models have recently been used in original research in the social sciences [Schwemmer and Jungkunz, 2019, Curry and Fix, 2019] as well as in surveys of past research in other disciplines [Bohr and Dunlap, 2018], suggesting that gaps between the theory

and practice of topic modeling are potentially relevant to current and future research. Although these inconsistencies may result in a loss in productivity and hinder the expansion of knowledge, they may also have more insidious consequences. If practical knowledge needed to make topic models work is communicated primarily through back-channels and not through the more public and accessible communications of the field, the seclusion of that information presents a barrier to entry. Researchers *in the know* and those they associate with, a population that historically skews male, white, and otherwise privileged, will have an advantage in performing successful research. Researchers without those connections, meanwhile, may have to rediscover what some researchers already know before achieving publishable results. Thus, gaps between theory and practice have the potential to exacerbate the under-representation of minority groups in topic modeling research and hinder efforts to improve diversity.

My thesis is similar to the topic of the independently developed NeurIPS 2020 workshop "I Can't Believe It's Not Better! Bridging the gap between theory and empiricism in probabilistic machine learning," which investigates discrepancies between theory and practice across all of probabilistic machine learning.[1] The existence and success of this workshop illustrate a couple of points: first, that other researchers are aware of and interested in studying gaps between theory and practice; second, that those gaps are not limited to topic modeling, but may extend to probabilistic machine learning and beyond.[2]

[1] `https://neurips.cc/Conferences/2020/ScheduleMultitrack?event=16124`

[2] Per the workshop description, the scope is limited to probabilistic machine learning to facilitate

the exchange of ideas and evaluation of hypotheses, not because gaps between theory and practice are necessarily confined to that domain.

Chapter 2

Background

2.1 Topic modeling

I provide a brief introduction to topic modeling with latent Dirichlet allocation; see Blei [2012] for a more thorough presentation.

Topic modeling is the task of finding the themes, or *topics*, that run through a collection of documents. Topic models commonly consist of a set of topics, each of which is modeled as a probability distribution or weighting over a fixed vocabulary of words, and a representation of each document in terms of those topics.

Topic models typically make a *bag-of-words* assumption, meaning that each document is modeled as a bag (multiset) of discrete words, and the order of words in a document is not modeled. Other assumptions abound, however. The number of words in each document is widely assumed to be given and is not modeled; the num-

ber of topics is also typically assumed to be given (but see Chapter 6 for a model that relaxes that assumption). Additionally, topic models generally make a more implicit assumption that *stop words*, common but relatively uninformative words like prepositions, conjunctions, pronouns, determiners, and auxiliary verbs, have been filtered out of the corpus. Although stop word filtering is a necessary step in reproducing many topic modeling results, it is rarely allocated more than a cursory mention in the literature.

2.1.1 Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) is a generative Bayesian topic model: It models a data set as a draw from a probability distribution, and the parameters of that model are themselves modeled as draws from another probability distribution called the *prior*. Specifically, for N words collected into D documents of varying length (number of words), denote the length of the d-th document by N_d, denote the i-th word in the data set by w_i, and denote the document w_i occurs in by d_i. LDA "explains" the occurrence of each word by postulating that a document was generated by repeatedly: (1) Sampling a topic assignment z_i from a document-specific probability distribution over K topics, Categorical($\boldsymbol{\theta}^{(d_i)}$), and (2) sampling a word w_i from a topic-specific probability distribution over a vocabulary of W words, Categorical($\boldsymbol{\phi}^{(z_i)}$). The topic parameters $\boldsymbol{\phi}$ and document-wise topic proportions $\boldsymbol{\theta}$ are in turn modeled as draws

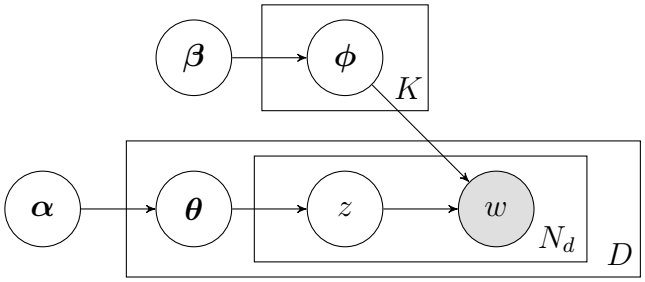


Figure 2.1: Plate diagram for latent Dirichlet allocation topic model.

from Dirichlet priors [Blei et al., 2003b]. Altogether, the model is as follows:

$$w_i|z_i, \boldsymbol{\phi}^{(z_i)} \sim \text{Categorical}\left(\boldsymbol{\phi}^{(z_i)}\right),$$

$$\boldsymbol{\phi}^{(k)} \sim \text{Dirichlet}\left(\boldsymbol{\beta}\right),$$

$$z_i|\boldsymbol{\theta}^{(d_i)} \sim \text{Categorical}\left(\boldsymbol{\theta}^{(d_i)}\right),$$

$$\boldsymbol{\theta}^{(d)} \sim \text{Dirichlet}\left(\boldsymbol{\alpha}\right).$$

Additionally, a plate diagram for the model is provided in Figure 2.1.

2.1.2 Training

Learning the values of $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ from a given data set typically involves estimating the posterior distribution and then taking a point estimate of that posterior to represent the "learned" model. Computing the distribution of $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ exactly is intractable, requiring the enumeration of exponentially many combinations of values of discrete variables (topic assignments z). This intractability motivates approximate inference methods such as expectation propagation [Minka and Lafferty, 2002], vari-

ational Bayesian inference [Blei et al., 2003b, Hoffman et al., 2010], and collapsed Gibbs sampling [Griffiths and Steyvers, 2004]. Once the posterior is approximated, its mean is typically used for point estimates of $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$.

Perhaps the most widely known, and the simplest, training algorithm for LDA is the collapsed Gibbs sampling algorithm introduced by Griffiths and Steyvers [2004]. Gibbs sampling is a Markov chain Monte Carlo (MCMC) method in which, at each step, one variable is sampled from the posterior while conditioning on the current values of all the other variables. The collapsed Gibbs sampling algorithm simplifies this procedure for LDA by leveraging Dirichlet-Multinomial conjugacy to integrate out $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$. The algorithm begins with random topic assignments $\mathbf{z}$, then at each step, one topic assignment z_i is sampled (conditioning on all other topic assignments $\mathbf{z}_{i-1}$) according to

$$\mathrm{P}\left[z_i|\mathbf{z}_{N\setminus i}, \mathbf{w}\right] = \frac{n^{(w_i)}_{z_i,N\setminus i} + \beta_{w_i}}{n^{(\cdot)}_{z_i,N\setminus i} + \sum_{t=1}^{W}\beta_t} \cdot \frac{n^{(d_i)}_{z_i,N\setminus i} + \alpha_{d_i}}{n^{(d_i)}_{\cdot,N\setminus i} + \sum_{k=1}^{K}\alpha_k}$$

where: $n^{(w_i)}_{z_i,N\setminus i}$ is the number of times word w_i has been assigned topic z_i, not including word i; $n^{(\cdot)}_{z_i,N\setminus i}$ is the number of times any word has been assigned topic z_i, not including word i; $n^{(d_i)}_{z_i,N\setminus i}$ is the number of times topic z_i has been assigned to any word in document d_i, not including word i; and $n^{(d_i)}_{\cdot,N\setminus i}$ is the number of words observed in document d_i, not including word i [Griffiths and Steyvers, 2004].

Sampling is run until some stopping criteria, such as a lower threshold on the

change in log likelihood between sweeps of the data, is met. At that point, the most recent step in the Markov chain is typically selected as a point estimate of the trained model. After (or during) sampling, the topics and topic proportions can be estimated as

$$\hat{\phi}_w^{(k)} = \frac{n_{k,N}^{(w)} + \beta_w}{n_{k,N}^{(\cdot)} + \sum_{t=1}^{W} \beta_t},$$
$$\hat{\theta}_k^{(d)} = \frac{n_{k,N}^{(d)} + \alpha_d}{n_{\cdot,N}^{(d)} + \sum_{t=1}^{K} \alpha_t},$$

respectively [Griffiths and Steyvers, 2004].

2.1.3 Evaluation and Application

Topic models are evaluated in the literature through both *quantitative* (numerical, statistical) and *qualitative* (descriptive, conceptual) methods. Topics are often represented to human users and other researchers by their *keys*, or top-m words (for some small number m).[1] Accordingly, many evaluation methods assess the quality of these topic keys.

One of the most common quantitative measures is the *log-likelihood*, which conveys

[1] Typically, $m = 5$ or $m = 10$.

how likely a data set is under the model. In notation, the log-likelihood is

$$\begin{aligned}\log \mathcal{L}\left(\mathbf{\Phi}, \mathbf{\Theta} | \mathbf{w}\right) &= \log \mathrm{P}_{\mathbf{\Phi},\mathbf{\Theta}}\left[\mathbf{w}\right] \\ &= \sum_{i=1}^{N} \log \mathrm{P}_{\mathbf{\Phi},\boldsymbol{\theta}^{(d_i)}}\left[w_i\right]\end{aligned}$$

where $\mathrm{P}_{\mathbf{\Phi},\boldsymbol{\theta}}\left[w; d\right]$ is the probability of w under the model with global parameters $\mathbf{\Phi}$ and local parameters $\boldsymbol{\theta}$, marginalizing over the latent variable z. While this naive formulation computes how likely a corpus is given all global parameters as well as each document's local parameters $\boldsymbol{\theta}^{(d)}$, *predictive log-likelihood* marginalizes over document-local parameters as well:

$$\log \mathcal{L}_{\text{pred}}\left(\mathbf{\Phi} | \mathbf{w}\right) = \sum_{i=1}^{N} \log \mathrm{P}_{\mathbf{\Phi}}\left[w_i\right].$$

Best practices when evaluating with likelihood include computing predictive log-likelihood on a held-out test set (using, for example, the left-to-right or Chib-style estimator of Wallach et al. [2009b] or the first-order approximation of Scott and Baldridge [2013]) in order to convey how well the model generalizes.

Another popular and more recent method for quantitative evaluation is the *topic coherence* metric introduced by Mimno et al. [2011]. Topic coherence measures the degree to which a topic's keys, or most probable words, co-occur in the corpus (occur in the same documents). If $D(w, w')$ is the number of documents in which words w and w' co-occur, $D(w)$ is the number of documents in which w occurs (its document

frequency), and $v_i^{(k)}$ is the i-th topic key (the i-th most probable word) of topic k, then the topic coherence $TC(k)$ of topic k is

$$TC(k) = \sum_{i=2}^{m} \sum_{j=1}^{i-1} \log \frac{D\left(v_i^{(k)}, v_j^{(k)}\right) + 1}{D\left(v_j^{(k)}\right)}$$

And the overall topic coherence for a model is typically computed by taking the average over the model's topics.

As Roberts et al. [2014] point out, this metric quantifies the *cohesiveness* of each topic, the degree to which the topic keys tend to co-occur (occur in the same documents), but there is another property of topic decompositions that we often care about: *exclusivity*, the degree to which a topic's keys tend not to occur in the keys of another topic. Roberts et al. [2014] note that they are not the first to make this observation; rather, it has been made before in several forms.

Both log-likelihood and topic coherence evaluate the distributions of words and topics and do not require additional labeled data. Sometimes, however, a data set will be labeled with *gold standard* or *ground truth* labels, and the inferred topics can be evaluated for their alignment with those labels. For example, the classic "20 newsgroups" data set contains a collection of newsgroup posts from twenty different newsgroups,[2] and posts from multiple newsgroups are typically combined into a single data set for topic modeling, treating the original newsgroup for each post as its gold-standard topic labeling. One popular approach for quantifying the alignment

[2] `http://qwone.com/~jason/20Newsgroups/`

between inferred and gold-standard topics is *normalized mutual information (NMI)*, an information theoretic measure of similarity between two clusterings. There is no universal definition of NMI due to different choices in normalization, but a common definition is as follows.

Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two clusterings (partitions) of a set of points $\mathcal{X}$ into K clusters (partition blocks). Let C_1 be a random variable indicating the cluster index k under clustering $\mathcal{C}_1$ of a point sampled uniformly with replacement from $\mathcal{X}$, and define C_2 analogously. So, if there are $n = |\mathcal{X}|$ points in total and $n_k^{(1)}$ of them are assigned to cluster k under $\mathcal{C}_1$, then

$$\mathrm{P}\left[C_1 = k\right] = \frac{n_k^{(1)}}{n}.$$

Moreover, if $n_{k'}^{(2)}$ points are assigned to cluster k' under $\mathcal{C}_2$ and $n_{k,k'}^{(1,2)}$ points are assigned to both cluster k under $\mathcal{C}_1$ and cluster k' under $\mathcal{C}_2$, then

$$\mathrm{P}\left[C_2 = k'\right] = \frac{n_{k'}^{(2)}}{n}$$

and

$$\mathrm{P}\left[C_1 = k, C_2 = k'\right] = \frac{n_{k,k'}^{(1,2)}}{n}.$$

If $H(C_1)$ is the entropy of C_1, $H(C_2)$ is the entropy of C_2, $H(C_1, C_2)$ is their joint

entropy, and $I(C_1, C_2)$ is their mutual information, then the NMI of C_1 and C_2 is

$$\text{NMI}(C_1, C_2) = \frac{I(C_1, C_2)}{H(C_1) + H(C_2)}$$

or equivalently

$$\text{NMI}(C_1, C_2) = \frac{H(C_1) + H(C_2) - H(C_1, C_2)}{H(C_1) + H(C_2)}.$$

NMI is often applied to topic models by treating a set of documents as the data $\mathcal{X}$ and using the topic with highest topic proportion in each document as that document's inferred cluster.

Becker [2011, Appendix A] shows that this definition of NMI is equivalent to another clustering metric, the *V-measure*. The V-measure, in turn, is a harmonic mean of two components: *homogeneity,* the degree to which each cluster from C_1 only contains data from a single cluster in C_2, and *completeness,* the degree to which all points from a cluster in C_2 are assigned to the same cluster in C_1 [Rosenberg and Hirschberg, 2007]. Put succinctly, the first component measures how homogeneous a cluster in C_1 is while the second component measures how complete it is. These components closely resemble the cohesiveness and exclusivity criteria (respectively) described by Roberts et al. [2014]. However, while V-measure (hence NMI) is often applied to the topics assigned to documents, cohesiveness and exclusivity are described in terms of the topics assigned to words. And whereas V-measure involves comparison of inferred

topics to a gold-standard clustering, the cohesiveness and exclusivity criteria pertain solely to the distributions of words and inferred topics.

A closely related metric, the *variation of information (VOI)*, measures how much information is lost when moving between two clusterings [Meilă, 2007]. The VOI between clusterings C_1 and C_2 is defined as

$$VI(C_1, C_2) = H(C_1) + H(C_2) - 2I(C_1, C_2)$$

or equivalently

$$VI(C_1, C_2) = H(C_1|C_2) + H(C_2|C_1).$$

where $H(C_1|C_2)$ is the conditional entropy of C_1 conditioned on C_2 and $H(C_2|C_1)$ is defined analogously [Meilă, 2007].

Qualitative evaluation of topic models is more nascent, but no less important. Topic models are often motivated as potential tools for human analysts working with text, so qualitative evaluation of topic models reflects the circumstances of their use. Typically, to distill a topic model into something that can be assessed qualitatively, each topic is represented by its keys, or top m words for some small number m. Many authors analyze the topic keys for each topic informally, discussing how well they fit the authors' intuitions and reproducing the lists directly in their reports so that readers may make their own judgments. The *word intrusion* and *topic intrusion*

tasks of Chang et al. [2009] use human judgments to approximately quantify two model qualities one might find intuitive: the distinguishability of words in a topic from words out of the topic and the distinguishability of topics used in a document from topics not used in the document, respectively. These tasks combine simple qualitative tests performed by human judges with quantitative aggregation methods.

Although the top m words by probability are by far the most common choice of topic keys, some researchers have proposed other selection criteria. For example, Bischof and Airoldi [2012] treat frequency and exclusivity of words in a topic as competing desiderata for topic keys and use the harmonic mean of the two properties to score how informative each word is in each topic. This frequency-exclusivity score, abbreviated *FREX*, can be used to select topic keys in a manner that is more robust to the presence of stop words [Bischof and Airoldi, 2012].

Topic modeling research is motivated by potential applications in the social sciences [Roberts et al., 2014, Boyd-Graber et al., 2017, Schwemmer and Jungkunz, 2019, Curry and Fix, 2019], the humanities [Buurma, 2015, Boyd-Graber et al., 2017], education [Reich et al., 2014], natural language processing [Boyd-Graber et al., 2014], and surveys of past research in other disciplines [Boyd-Graber et al., 2017, Bohr and Dunlap, 2018].[3] Topic models are also used in the development of search interfaces, visualizations, and other tools in government and industry [Boyd-Graber et al., 2014].

[3]Indeed, Schwemmer and Jungkunz [2019], Curry and Fix [2019], and Bohr and Dunlap [2018] all use a probabilistic graphical topic model called the structural topic model [Roberts et al., 2013], suggesting that research on probabilistic graphical topic models like LDA is still relevant in the era of neural topic models and deep learning in natural language processing. More information on usage of the structural topic model is available at `https://www.structuraltopicmodel.com/`.

One of the limitations and challenges of applying topic modeling in practice is that many topic models, including LDA, are not *identifiable*. This means that even in theory, a given data set can be equivalently represented by many different parametrizations of LDA. Concretely, the ordering of topics in LDA has no theoretical impact on the observation model, so a topic model with two topics, so there are $K!$ equivalent parametrizations of an LDA model with K topics. Moreover, when training a topic model, the initial model provided to the training algorithm is often randomly generated and that initialization step is often symmetric with respect to the topics, so each an LDA model is trained on a given data set the topics may appear in a different order. And because of estimation error, multiple runs of the training algorithm may even produce different sets of topics altogether. These complications are a nuisance in the evaluation of topic models in the literature, but they can be much more of a problem in real-world applications where random effects are less acceptable. For example, the use of LDA in a judicial trial would be problematic because different random seeds can potentially produce qualitatively different results.

See Boyd-Graber et al. [2017] for further discussion of applications of topic models.

2.1.4 Practice

Training, inference, and analysis of LDA topic models has been implemented numerous times in open-source software. The most popular implementation is likely MALLET [McCallum, 2002], a Java library that implements collapsed Gibbs sampling

for LDA as well as algorithms for other NLP models. Another popular implementation is Gensim [Řehůřek and Sojka, 2010], a Python library that implements stochastic variational inference for LDA as well as algorithms for other topic models and vector representations of text.

Some of the practical concerns of topic modeling that I investigate in the following chapters have already been observed by researchers and practitioners before me. For example, Manning et al. [2008, Ch. 2] discuss varying preprocessing requirements of different languages and stop word filtering in the context of information retrieval. Topic modeling, in fact, descends directly from information retrieval: Latent Dirichlet allocation is based on probabilistic latent semantic analysis [Hofmann, 1999], which is in turn based on latent semantic analysis [Deerwester et al., 1990], a classic method in information retrieval. Accordingly, early topic modeling researchers may have assumed that their audience was familiar with practical concerns of information retrieval like those outlined in Manning et al. [2008, ch. 2], and that assumption might have been reasonable at the time. However, modern publications on topic modeling make little or no mention of information retrieval, and information retrieval practice is not widely known by researchers outside of information retrieval. The practical concerns of information retrieval are therefore poised to be *unknown unknowns* to modern topic modeling researchers, concepts that researchers not only don't know, but don't know they don't know.

Topic modeling publications sometimes enumerate practical concerns of topic

modeling themselves; see, for example, Boyd-Graber et al. [2014]. However, my experience as a student researcher indicates that such sporadic expositions are not sufficient, and the premise of the NeurIPS 2020 "I Can't Believe It's Not Better!" workshop suggests I am not be alone.

2.2 Theory

In general, topic modeling research spans multiple spaces, disciplines, communities, and cultures. The research I study similarly spans these contexts, although it is concentrated in computer science and largely does not extend to the social sciences and humanities. Although out of scope for my work, the variability of topic modeling theory and practice across these contexts would be an interesting object of study for future work.

Topic modeling, in the sense I consider, is a largely empirical field of research; progress is determined largely by the outcomes of computational experiments. Additionally, there is little in the way of explicit theory developed in topic modeling research. For example, the "distributional hypothesis" from linguistics may be considered to be a theoretical foundation for topic modeling. While the distributional hypothesis is often attributed to Harris [1954], it is more concisely stated by Rubenstein and Goodenough [1965], and this abbreviated form is more often used in modern research:

Theory. *"Words which are similar in meaning occur in similar contexts" [Rubenstein and Goodenough, 1965].*

In contrast, topic modeling research rarely makes such formal claims; when it does, the claims are often limited to the mathematical properties of learning algorithms, as in Arora et al. [2013]. However, building off empirical work requires abstracting from the reported observations, which are inevitably laden with contextual information that may have little relevance to the objects of study. Thus, topic modeling research must produce theory in some more implicit form.

A small, self-contained example of implicit theories in NLP appears in the history of the scikit-learn stop word list recounted by Nothman et al. [2018]: Scikit-learn, a Python machine learning library, provided an English stop word list as of July 2010, but it was disabled by default as the person who contributor it claimed it did not improve text classification results. In November 2010, a different contributor enabled it, arguing that filtering out stop words was a reasonable default. Then, in March 2012, it was disabled again [Nothman et al., 2018]. Although there is no explicit theory of stop words in NLP, the developers involved in these changes clearly had mental models of how stop word filtering affected text classification (and perhaps other tasks) in general. In fact, these models—theories—appear to contradict each other. Because such theories are not stated, tested, and communicated explicitly, it is easy for competing theories to coexist in any given community, a phenomenon I will revisit in Chapter 5.

To make the notion of *implicit* theory more concrete, I draw on key ideas in the philosophy of science and technology; for a more detailed philosophical exposition, see Franssen et al. [2018]. In this section so far, I have assumed that topic modeling is a scientific area of research and proceeds largely by developing and testing hypotheses. Surely, some proportion of topic modeling research *is* science, but many (if not most) topic modeling papers have the following structure: First, present a task or problem to be solved, and second, present a solution. This problem-solving structure suggests that to some extent, topic modeling research is not science but technology research. Specifically, Bunge [1966] argues that theories in technology take two forms: *operative theories*, which are theories of action, and *substantive theories*, which are theories of the object of that action. In the context of topic modeling research, substantive theories may be considered to include theories of language, like the distributional hypothesis, and theories of machines and data, like the central limit theorem. While these substantive theories are inherited from the sciences (and mathematics), operative theories arise in the topic modeling research itself and pertain to the *action* of topic modeling. While operative theories do not contain the substance of science, they do use scientific methods, including making use of theoretical concepts and abstractions and testing against empirical data. These operative theories, theories of how to do topic modeling, are the implicit theories I aim to study.

Finally, note that topic modeling research—or indeed any area of research—is not only performed through the publication of peer-reviewed papers. Broadly, topic

modeling research takes place over peer-reviewed papers as well as book chapters, preprints that have not gone through peer review, presentations at conferences, discussions in the hallways and other meeting places of those conferences, discussions in research labs and coffee shops and myriad other settings, distribution of research software and documentation, bug reports and feature requests on that software, classroom lectures and activities, blogs and blog comments, private email threads, mailing lists, and social media in general. Indeed, the theories that I claim are *implicit* in published research papers are likely made explicit in these other channels. However, an analysis of the broader research discourse is out of scope in my work, and because published, peer-reviewed papers are generally considered the gold standard of research, I will restrict my attention to them.

Chapter 3

Streaming Learning

3.1 Preface

This chapter describes work originally appearing in May et al. [2014], of which I was the primary author. Our goal was to use reservoir sampling to implement the rejuvenation step of a particle filtering training algorithm for LDA introduced in prior work, thereby eliminating that algorithm's linear memory requirement and allowing LDA models to be learned on unbounded data streams in principle.

Many data sets are too large to fit into main memory or traverse more than once. Using a batch learning method like Gibbs sampling, we can train an LDA topic model by iterating over the documents in a corpus. However, a good fit requires traversing the corpus multiple times, and we must maintain document-level state for all documents in the corpus throughout training. In response, a particle filtering

algorithm was proposed,[1] allowing LDA to be trained by sampling in one pass and with memory constant in the size of the corpus [Canini et al., 2009], that is, in a *streaming* data setting. In order to update the model with new data without losing or "forgetting" what was learned before, the particle filtering algorithm uses a technique called *rejuvenation* in which the sampler is periodically re-run on a random sample of documents from the data stream. Prior work found this algorithm to yield high performance on some data sets, measuring by distributional similarity to gold-standard assignments. I seek to implement the rejuvenation step using a reservoir sample, a technique that produces a random sample at every point in a data stream in constant memory and linear runtime requirements. I implement and study the particle filtering algorithm empirically, making the following contributions to science:

- I perform a parameter study of a particle filtering training algorithm for LDA introduced in prior work [Canini et al., 2009], providing a more comprehensive analysis of the algorithm.
- I show that tuning an LDA model by perplexity is just as good as tuning it by the evaluation metric (using gold-standard forum labels) on several qualitatively different subsets of the common 20 newsgroups data set, suggesting that there may be little room for improvement in topic modeling on that data set from an information theoretic perspective.
- I have publicly released my LDA particle filter training and experiment code

[1] See, for example, Godsill [2019] for a recent introduction to particle filtering.

for the benefit of future research.[2]

In studying the prior work, I develop the following implicit theory:

Theory. *Rejuvenation enables streaming learning of LDA by particle filtering.*

However, I find that the topic model learned by this particle filtering algorithm is sensitive to initialization, and rejuvenation has a negligible effect at practical scales. I was only able to approximately reproduce the results of prior work with an initialization sample one-fifth the size of the entire training set, and the particle filter itself yielded a relatively small improvement on the initial model [May et al., 2014]. Thus, while prior work suggests that topic models can be effectively learned by sampling in the streaming data setting, I find that this approach is dominated by initialization, limiting its utility in practice.

3.2 Introduction

In this chapter, I study the training of a latent Dirichlet allocation topic model in cases where the data is effectively unbounded. To formalize this setting, I define a *streaming* algorithm as an algorithm that requires at most

- one pass over the data,
- constant storage, and

[2] `https://github.com/ccmaymay/pflda`

- linear runtime.[3]

Specifically, then, in this chapter, I study LDA training in the streaming setting.

Canini et al. [2009] presented a method for LDA inference based on particle filtering in which a weighted sample of particles (in this case, LDA models) is updated online via importance sampling for each new token observed from a stream. In general, the particle sample becomes degenerate (almost all importance weight goes to just one particle) over time, and must be periodically *resampled* and *rejuvenated* [Gilks and Berzuini, 2001]. In the resampling step, particles are resampled according to their importance weights so that the total weight is more evenly distributed over the particles; in the rejuvenation step, Markov Chain Monte Carlo (MCMC) steps are taken over the history so that the particles are less redundant. This process is illustrated in Figure 3.1. The particle filter of Canini et al. [2009] rejuvenates over independent draws from the history by storing all past observations and states. This algorithm thus has linear storage complexity and does not satisfy the *streaming* criteria.

I propose approximating the rejuvenation step with a reservoir sampler to reduce the storage complexity of the particle filter to constant, thereby yielding a streaming learning algorithm. This implementation is scientifically interesting in that it recovers some of the cognitive plausibility that motivated the use of a similar particle filter by Börschinger and Johnson [2012]. My proposal hinges on a belief that the particle

[3]Other authors have proposed slightly different definitions of a streaming algorithm, such as allowing multiple passes or $O(\log n)$ storage. However, the relatively simple, restrictive definition proposed here will suffice for my purposes.

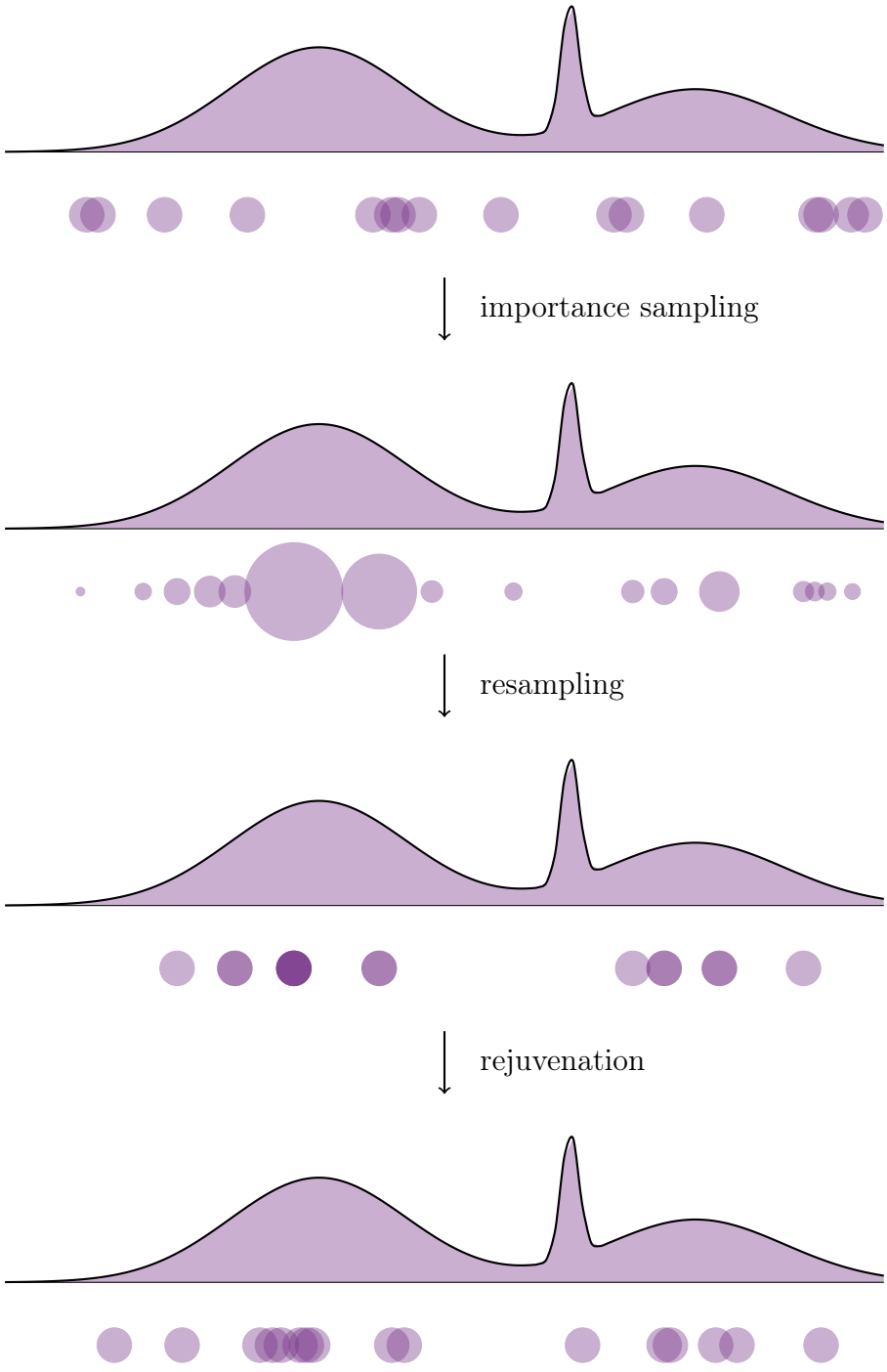


Figure 3.1: Illustration of the resampling and rejuvenation steps for a hypothetical particle filter on a real-valued variable. The curve represents the true posterior; the circles below the curve represent the particles, with location corresponding to value and size corresponding to importance weight.

filter proposed by Canini et al. [2009] works in part due to rejuvenation, that is, that rejuvenation has an impact on the resultant model. I argue this relatively simple *theory* is justified by prior work: "Markov chain Monte Carlo (MCMC) is used after particle resampling to restore diversity to the particle set . . . . The length of [the rejuvenation sequence] can be chosen to trade off runtime against performance" [Canini et al., 2009], implying rejuvenation impacts runtime and performance. Moreover, I assume that everything written in Canini et al. [2009] is *relevant* to their argument;[4] if rejuvenation did not impact their experimental results, Canini et al. [2009] would likely not have expended the effort to write about it and implement it in software.

However, I find rejuvenation makes no noticeable impact on the resultant model in the setting studied by Canini et al. [2009]. Instead, I show that the quality of the resultant model is highly dependent on the quality of the initial model provided to the particle filter, and the particle filter only overcomes this sensitivity when the rejuvenation process is performed at such a large scale that the algorithm no longer satisfies the streaming criteria in practice. This finding re-opens the question of whether LDA models can be effectively trained by particle filtering in the streaming setting.

[4]This assumption follows directly from Grice's cooperative principle, which describes the properties of conversational communication.

3.3 Background

A common way to fit an LDA model to a corpus is by Gibbs sampling, in which a chain of samples from the model is constructed by iteratively sampling one variable while conditioning on all others. While the resultant chain yields an empirical distribution over (parametrized) models, a single model representing the data can be produced by taking the last sample in the chain.

One major benefit of Gibbs sampling is its simplicity: A training algorithm can be implemented in relatively few lines of code and easily explained. Additionally, Gibbs sampling and other sampling techniques estimate the exact distribution of model parameters given the data, accounting for burn-in and autocorrelation in the chain. Improvements to sampling methods, like the extension to the streaming data setting studied here, have the potential to significantly benefit future research and practice.

One limitation of these techniques is they require multiple passes over the data to obtain good samples of $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$—to achieve burn-in. This requirement makes them impractical when the corpus is too large to fit directly into memory and in particular when the corpus grows without bound. This motivates online learning techniques [Banerjee and Basu, 2007, Canini et al., 2009]. However, where these approaches assume the ability to draw independent samples from the full data set, I consider the case when it is infeasible to access elements arbitrarily far back in the history. While a streaming variational Bayes framework has been proposed for training under this constraint [Broderick et al., 2013], I focus on sampling-based meth-

initialize: $\omega_0^{(p)} \leftarrow 1/\mathcal{P}$ for each particle $p = 1, \ldots, \mathcal{P}$
for $i = 1, \ldots, N$ **do**
  for $p = 1, \ldots, \mathcal{P}$ **do**
    $\omega_i^{(p)} \leftarrow \omega_{i-1}^{(p)} \mathrm{P}\left[w_i \middle| \mathbf{z}_{i-1}^{(p)}, \mathbf{w}_{i-1}\right]$
    sample $z_i^{(p)}$ with probability $\mathrm{P}\left[z_i^{(p)} \middle| \mathbf{z}_{i-1}^{(p)}, \mathbf{w}_i\right]$.
  if $\|\boldsymbol{\omega}\|_2^{-2} \leq n_{\text{eff}}$ **then**
    for $j \in \mathcal{R}(i)$ **do**
      for $p = 1, \ldots, \mathcal{P}$ **do**
        sample $z_j^{(p)}$ with probability $\mathrm{P}\left[z_j^{(p)} \middle| \mathbf{z}_{i\setminus j}^{(p)}, \mathbf{w}_i\right]$
    $\omega_i^{(p)} \leftarrow 1/\mathcal{P}$ for each particle $p = 1, \ldots, \mathcal{P}$

Algorithm 1: Particle filtering for LDA.

ods, of which I know no streaming algorithm other than the particle filter proposed by [Canini et al., 2009] when implemented with a reservoir sampler for rejuvenation.

3.4 Online LDA Using Particle Filters

Particle filters are a family of sequential Monte Carlo (SMC) sampling algorithms designed to estimate the posterior distribution of a system whose state evolves over time [Doucet et al., 2001]. A particle filter approximates the posterior with a weighted sample of points, called particles, from the state space. The particle cloud is updated recursively for each new observation using importance sampling (an approach referred to as *sequential importance sampling*).

Canini et al. [2009] apply this approach to LDA, analytically integrating out $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ to obtain Rao-Blackwellized particle filter [Doucet et al., 2000] that estimates the

collapsed posterior P $[\mathbf{z}|\mathbf{w}]$. In this setting, the $\mathcal{P}$ particles are samples of the topic assignment vector $\mathbf{z}^{(p)}$, and they are propagated forward in state space one token at a time.As Canini et al. [2009] note, this usage of particle filtering is nonstandard because the state space $\mathbf{z}$ grows over time (with each step of the particle filtering algorithm).[5]

In general, the larger the number of particles $\mathcal{P}$ is, the more accurately we approximate the posterior; for small $\mathcal{P}$, the approximation of the tails of the posterior will be particularly poor [Pitt and Shephard, 1999]. However, a larger value of $\mathcal{P}$ increases the runtime and storage requirements of the algorithm.

I now describe the Rao-Blackwellized particle filter for LDA in detail; pseudocode is given in Algorithm 1. At the moment token i is observed, the particles form a discrete approximation of the posterior up to the $(i-1)$-th word:

$$\mathrm{P}\left[\mathbf{z}_{i-1}|\mathbf{w}_{i-1}\right] \approx \sum_{p} \omega_{i-1}^{(p)} \left[\mathbf{z}_{i-1} = \mathbf{z}_{i-1}^{(p)}\right]$$

where $[x = x']$ is the Iverson bracket which evaluates to 1 if $x = x'$ and 0 otherwise. Now each particle p is propagated forward by drawing a topic $z_i^{(p)}$ from the conditional posterior distribution $\mathrm{P}\left[z_i^{(p)} \middle| \mathbf{z}_{i-1}^{(p)}, \mathbf{w}_i\right]$ and scaling the particle weight by $\mathrm{P}\left[w_i \middle| \mathbf{z}_{i-1}^{(p)}, \mathbf{w}_{i-1}\right]$. The particle cloud now approximates the posterior up to the i-th

[5]Particle filters are often used in applications like object tracking in video, where the target distribution changes over time but its dimensionality is constant.

word:

$$\mathrm{P}\left[\mathbf{z}_i|\mathbf{w}_i\right] \approx \sum_p \omega_i^{(p)} \left[\mathbf{z}_i = \mathbf{z}_i^{(p)}\right].$$

Dropping the superscript (p) for notational convenience, the conditional posterior used in the propagation step is given by

$$\begin{aligned}\mathrm{P}\left[z_i|\mathbf{z}_{i-1}, \mathbf{w}_i\right] &\propto \mathrm{P}\left[z_i, w_i|\mathbf{z}_{i-1}, \mathbf{w}_{i-1}\right] \\ &= \frac{n_{z_i,i\backslash i}^{(w_i)} + \beta_{w_i}}{n_{z_i,i\backslash i}^{(\cdot)} + \sum_{t=1}^{W} \beta_t} \cdot \frac{n_{z_i,i\backslash i}^{(d_i)} + \alpha_{d_i}}{n_{\cdot,i\backslash i}^{(d_i)} + \sum_{k=1}^{K} \alpha_k}\end{aligned}$$

where $n_{z_i,i\backslash i}^{(w_i)}$ is the number of times word w_i has been assigned topic z_i so far, $n_{z_i,i\backslash i}^{(\cdot)}$ is the number of times any word has been assigned topic z_i, $n_{z_i,i\backslash i}^{(d_i)}$ is the number of times topic z_i has been assigned to any word in document d_i, and $n_{\cdot,i\backslash i}^{(d_i)}$ is the number of words observed in document d_i. The particle weights are scaled as

$$\begin{aligned}\frac{\omega_i^{(p)}}{\omega_{i-1}^{(p)}} &\propto \frac{\mathrm{P}\left[w_i \middle| \mathbf{z}_i^{(p)}, \mathbf{w}_i\right] \mathrm{P}\left[z_i^{(p)} \middle| \mathbf{z}_{i-1}^{(p)}\right]}{Q\left(z_i^{(p)} \middle| \mathbf{z}_{i-1}^{(p)}, \mathbf{w}_i\right)} \\ &= \mathrm{P}\left[w_i \middle| \mathbf{z}_{i-1}^{(p)}, \mathbf{w}_{i-1}\right]\end{aligned}$$

where Q is the proposal distribution for the particle state transition; in my case,

$$Q\left(z_i^{(p)} \middle| \mathbf{z}_{i-1}^{(p)}, \mathbf{w}_i\right) = \mathrm{P}\left[z_i^{(p)} \middle| \mathbf{z}_{i-1}^{(p)}, \mathbf{w}_i\right],$$

minimizing the variance of the importance weights conditioned on $\mathbf{w}_i$ and $\mathbf{z}_{i-1}$ [Doucet et al., 2000].

Over time the particle weights tend to diverge. To compensate for this inefficiency, after every state transition I estimate the effective sample size (n_{eff}) of the particle weights as $\|\omega_i\|_2^{-2}$ [Liu and Chen, 1998] and resample the particles when that estimate drops below a pre-specified threshold. Several resampling strategies have been proposed [Doucet et al., 2000]; I perform multinomial resampling as in Pitt and Shephard [1999] and Ahmed et al. [2011], treating the weights as unnormalized probability masses on the particles.

After resampling we are likely to have several copies of the same particle, yielding a degenerate approximation to the posterior. To reintroduce diversity to the particle cloud I take MCMC steps over a sequence of states from the history [Doucet et al., 2000, Gilks and Berzuini, 2001]. I call the indices of these states the rejuvenation sequence, denoted $\mathcal{R}(i)$ [Canini et al., 2009]. The transition probability for a state $j \in \mathcal{R}(i)$ is given by

$$\mathrm{P}\left[z_j \middle| \mathbf{z}_{N\setminus j}, \mathbf{w}_N\right] \propto \frac{n^{(w_j)}_{z_j,N\setminus j} + \beta_{w_j}}{n^{(\cdot)}_{z_j,N\setminus j} + \sum_{t=1}^{W}\beta_t} \cdot \frac{n^{(d_j)}_{z_j,N\setminus j} + \alpha_{d_j}}{n^{(d_j)}_{\cdot,N\setminus j} + \sum_{k=1}^{K}\alpha_k}.$$

The rejuvenation sequence can be chosen by the practitioner. Choosing a long sequence (large $|\mathcal{R}(i)|$) may result in a more accurate posterior approximation but also increases runtime and storage requirements. The tokens in $\mathcal{R}(i)$ may be chosen

uniformly at random from the history or under a biased scheme that favors recent observations. The particle filter studied empirically by Canini et al. [2009] stored the entire history, incurring linear storage complexity in the size of the stream. Ahmed et al. [2011] instead sampled ten documents from the most recent 1000, achieving constant storage complexity at the cost of a recency bias. If we want to fit a model to a long non-i.i.d. (where i.i.d. stands for independent and identically distributed) stream, we require an unbiased rejuvenation sequence as well as sub-linear storage complexity.

3.5 Reservoir Sampling

Reservoir sampling is a widely-used family of algorithms for choosing an array ("reservoir") of m items. The most common example, presented in Vitter [1985] as Algorithm R, chooses m elements of a stream S such that all subsets of m elements are equally likely. This effects sampling m items uniformly without replacement, using runtime $O(|S|)$ (linear in the stream length, constant per update) and storage $O(m)$.

To ensure constant space over an unbounded stream, I draw the rejuvenation sequence $\mathcal{R}(i)$ uniformly from a reservoir. As each token of the training data is ingested by the particle filter, we decide to insert that token into the reservoir, or not, independent of the other tokens in the current document. Thus, at the end of step i of the particle filter, each of the i tokens seen so far in the training sequence has

```
Initialize m-element array R ;
Stream S ;
for i = 1, ..., m do
    R[i] ← S[i] ;
for i = m + 1, ..., |S| do
    Sample j uniformly from {1, 2, ..., i};
    if j ≤ m then
        R ← S[i] ;
```

Algorithm 2: Algorithm R for reservoir sampling

an equal probability of being in the reservoir, hence being selected for rejuvenation.

3.6 Experiments

I evaluate the particle filter on three data sets studied in Canini et al. [2009]: `diff3`, `rel3`, and `sim3`. Each of these data sets is a collection of posts under three categories from the 20 Newsgroups data set.[6] I use a 60% training/40% testing split of this data available online.[7]

I preprocess the data by splitting each line on non-alphabet characters, converting the resulting tokens to lower-case, and filtering out any tokens that appear in a list of common English stop words. In addition, I remove the header of every file and filter every line that does not contain a non-trailing space (which removes embedded ASCII-encoded attachments). Finally, I shuffle the order of the documents. After these steps, I compute the vocabulary for each data set as the set of all non-singleton

[6] `diff3`: `{rec.sport.baseball, sci.space, alt.atheism}`; `rel3`: `talk.politics.{misc, guns, mideast}`; and `sim3`: `comp.{graphics, os.ms-windows.misc, windows.x}`.

[7] `http://qwone.com/~jason/20Newsgroups/20news-bydate.tar.gz`

types in the training data augmented with a special out-of-vocabulary symbol.

After each step of particle filter training, I report the similarity between the inferred topics and gold-standard topics. Following Canini et al. [2009], I compute similarity using out-of-sample normalized mutual information (NMI), holding the word proportions $\boldsymbol{\phi}$ fixed, running five sweeps of collapsed Gibbs sampling on the test set, and inferring the topic proportions for each document as during training. Two Gibbs sweeps have been shown to yield good performance in practice [Yao et al., 2009]; I increase the number of sweeps to five after inspecting the stability on my data set. The variance of the particle filter is often large, so for each experiment I perform 30 runs and plot the mean NMI inside bands spanning one sample standard deviation in either direction.

3.6.1 Fixed Initialization

My first set of experiments has a similar parametrization to the experiments of Canini et al. [2009] except I draw the rejuvenation sequence from a reservoir.[8] I initialize the particle filter with 200 Gibbs sweeps on the first 10% of each data set. Then, for each data set, for rejuvenation disabled, rejuvenation based on a reservoir of size 1000, and rejuvenation based on the entire history (in turn), I perform 30 runs of the particle filter from that fixed initial model. My results (Figure 3.2) resemble those of Canini et al. [2009]; I believe the discrepancies are mostly attributable to

[8]The parametrization I use consists of $K = 3$ topics, symmetric priors $\alpha_\cdot = \beta_\cdot = 0.1$, $\mathcal{P} = 100$ particles, an effective sample size n_{eff}, and a rejuvenation sequence of size $|\mathcal{R}| = 30$ tokens.

differences in preprocessing.

In these experiments, the initial model was not chosen arbitrarily. Rather, the initial model was tuned: An initial model that yielded out-of-sample NMI close to the corresponding out-of-sample NMI score reported in the previous study was chosen from a set of 100 candidates. Note that out-of-sample NMI involves comparison to a set of gold-standard topics (and document-topic assignments), so these experiments do not represent an unsupervised setting.

3.6.2 Variable Initialization

I now investigate the significance of the initial model selection step used in the previous experiments. I run a new set of experiments in which the reservoir size is held fixed and the size of the initialization sample is varied. Specifically, I vary the size of the initialization sample, in documents, between zero (corresponding to no Gibbs initialization), 30, 100, and 300, and also perform a run of batch Gibbs sampling (with no particle filter). In each case, 200 Gibbs sweeps over the initialization sample are performed. In these experiments, the initial models are not held fixed; for each of the 30 runs for each data set, the initial model was generated by a different Gibbs chain. The results for these experiments, depicted in Figure 3.3, indicate that the size of the initialization sample improves mean NMI and reduces the variance, and that the variance of the particle filter itself is dominated by the variance introduced by the initialization sample.

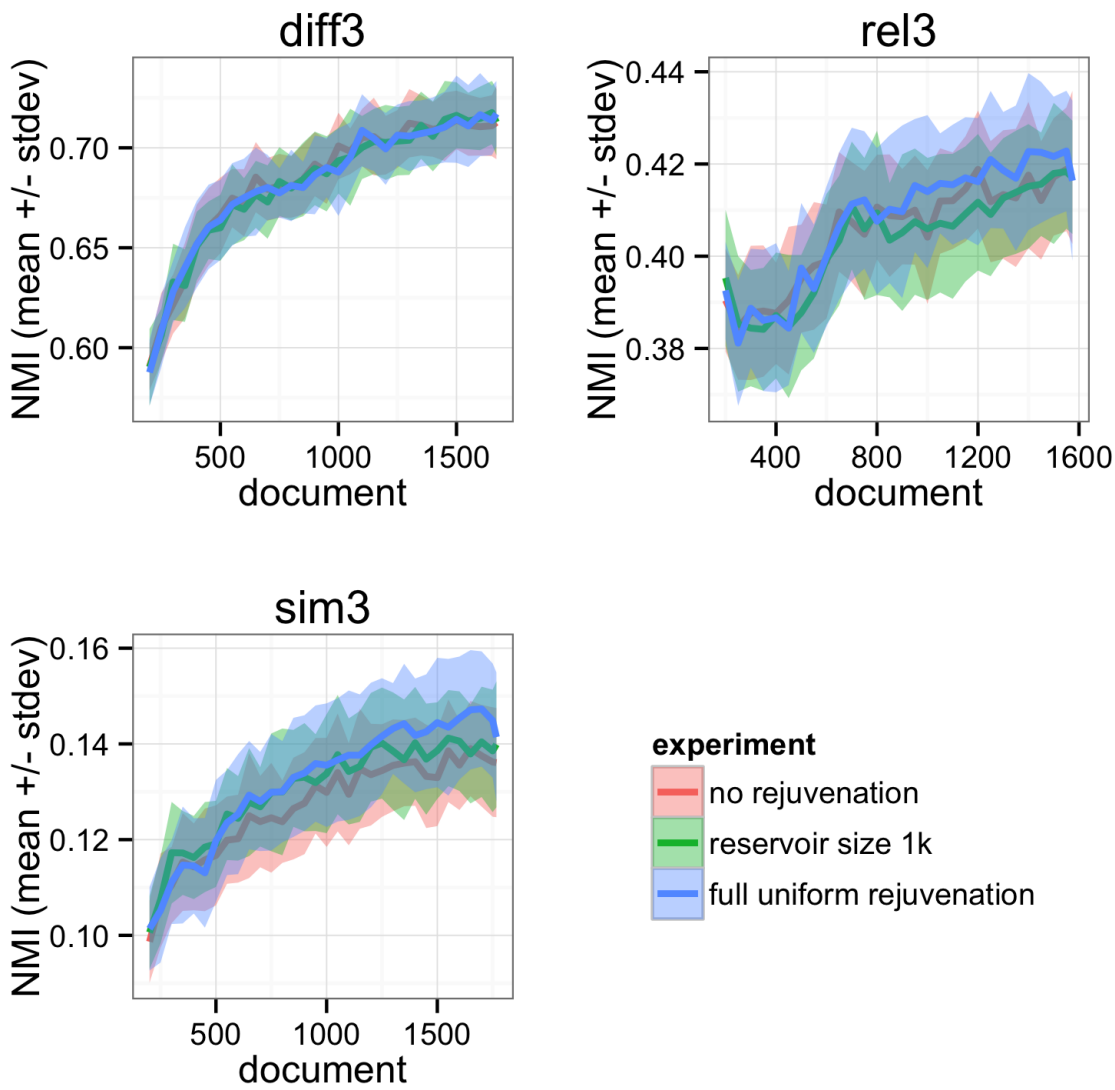


Figure 3.2: NMI over the course of training using fixed initialization with no rejuvenation, rejuvenation using a reservoir of 1000 tokens, and rejuvenation over the entire history.

This is the only initialization setting I study that does not require some form of supervised data.

3.6.3 Tuned Initialization

I observed previously that variance in the Gibbs initialization of the model contributes significantly to variance of the overall algorithm, as measured by NMI. With this in mind, I consider whether we can reduce variance in the initialization by tuning the initial model based on NMI. Thus I perform a set of experiments in which we perform Gibbs initialization 20 times on the initialization set, setting the particle filter's initial model to the model out of these 20 with the highest in-sample NMI. This procedure is performed independently for each run of the particle filter. We may not always have labeled data for initialization, so I also consider a variation in which Gibbs initialization is performed 20 times on the first 80% of the initialization sample, held-out perplexity (per word) is estimated on the remaining 20% each time, using a first-moment particle learning approximation [Scott and Baldridge, 2013], and the particle filter's initial model is set to the model out of these 20 with the lowest held-out perplexity. Results, shown in Figure 3.4, suggest we can mitigate the variance due to initialization by tuning the initial model to either NMI or perplexity.

As in the fixed initialization experiments, however, note that comparison to NMI requires gold-standard topics and document-topic assignments, and so breaches the constraints of strictly unsupervised learning.

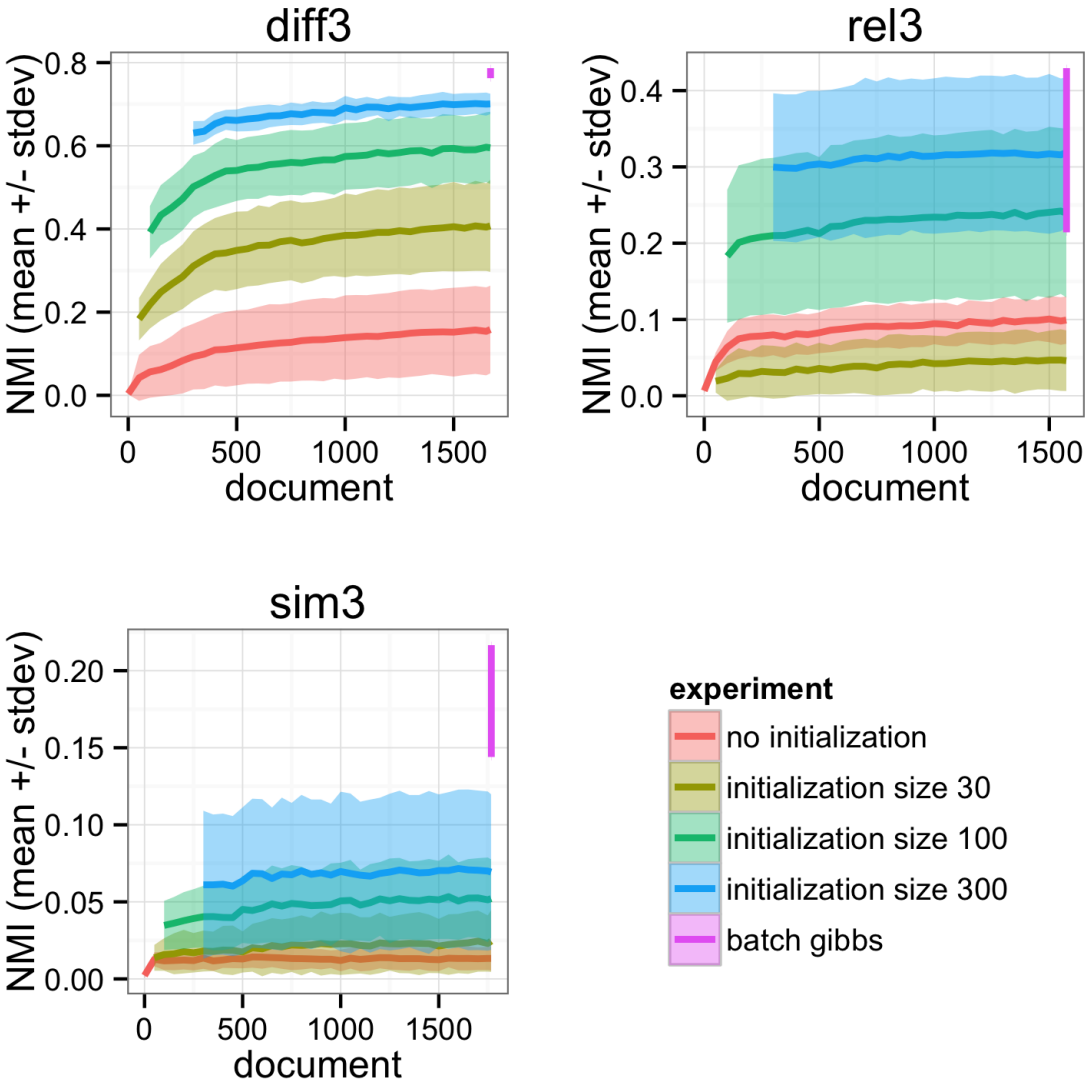


Figure 3.3: NMI over the course of training using variable initialization with initialization sample sizes of zero documents, 30 documents, 100 documents, 300 documents, and the size of the training corpus (equivalent to batch Gibbs sampling).

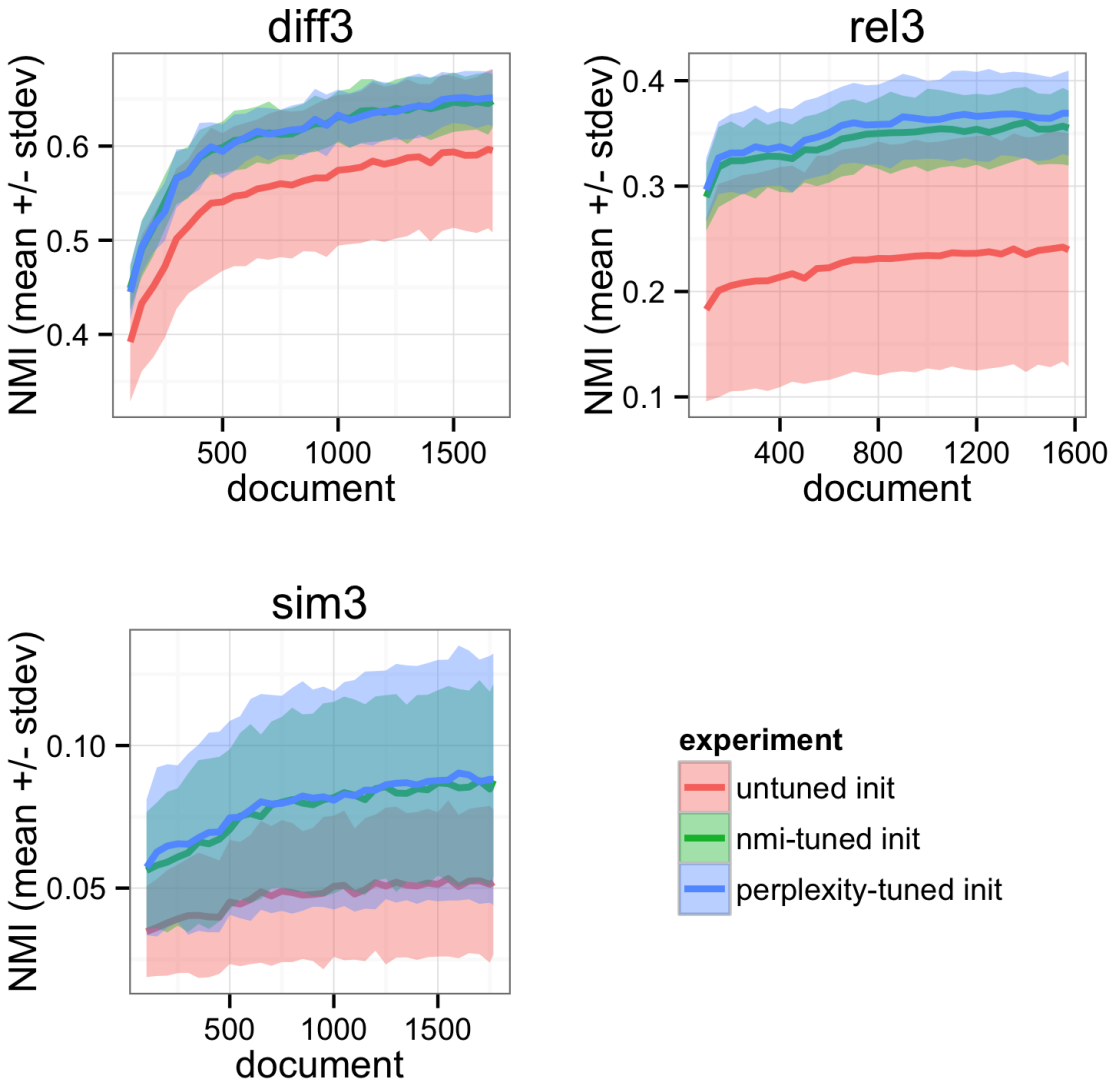


Figure 3.4: NMI over the course of training using variable initialization without tuning, with NMI-based tuning, and with perplexity-based tuning.

3.6.4 Large rejuvenation

Finally, I investigate the impact of the rejuvenation sequence size, $|\mathcal{R}|$, allowing it to vary across the large values of 1, 3, 10, 30, 100, or 300 *documents* (contrasting the previous setting of 30 *tokens*). In the "fixed initialization" experiment, I varied the size of the reservoir underpinning the rejuvenation sequence, and hence the storage complexity; in this experiment, I vary the size of the rejuvenation sequence itself, and hence the runtime complexity. Additionally, in this experiment, I control for other effects by omitting initialization altogether, using $\mathcal{P} = 1$ particle, and drawing the rejuvenation sample from the entire history (rather than a reservoir sample). Results, shown in Figure 3.5, show that the particle filter is sensitive to the rejuvenation sequence size at large scales, and the effect of initialization can be overcome when rejuvenation sequence is the same scale as the training data itself. However, at this scale, the algorithm's runtime is effectively quadratic, as the particle filter is revisiting most of its past observations after each new one.

3.7 Discussion

Overall, my results indicate that the particle filter for LDA on this data set is sensitive to initialization and only sensitive to rejuvenation at large scales. Accordingly, at moderate rejuvenation sequence sizes, reservoir sampling neither improves nor detracts from performance.

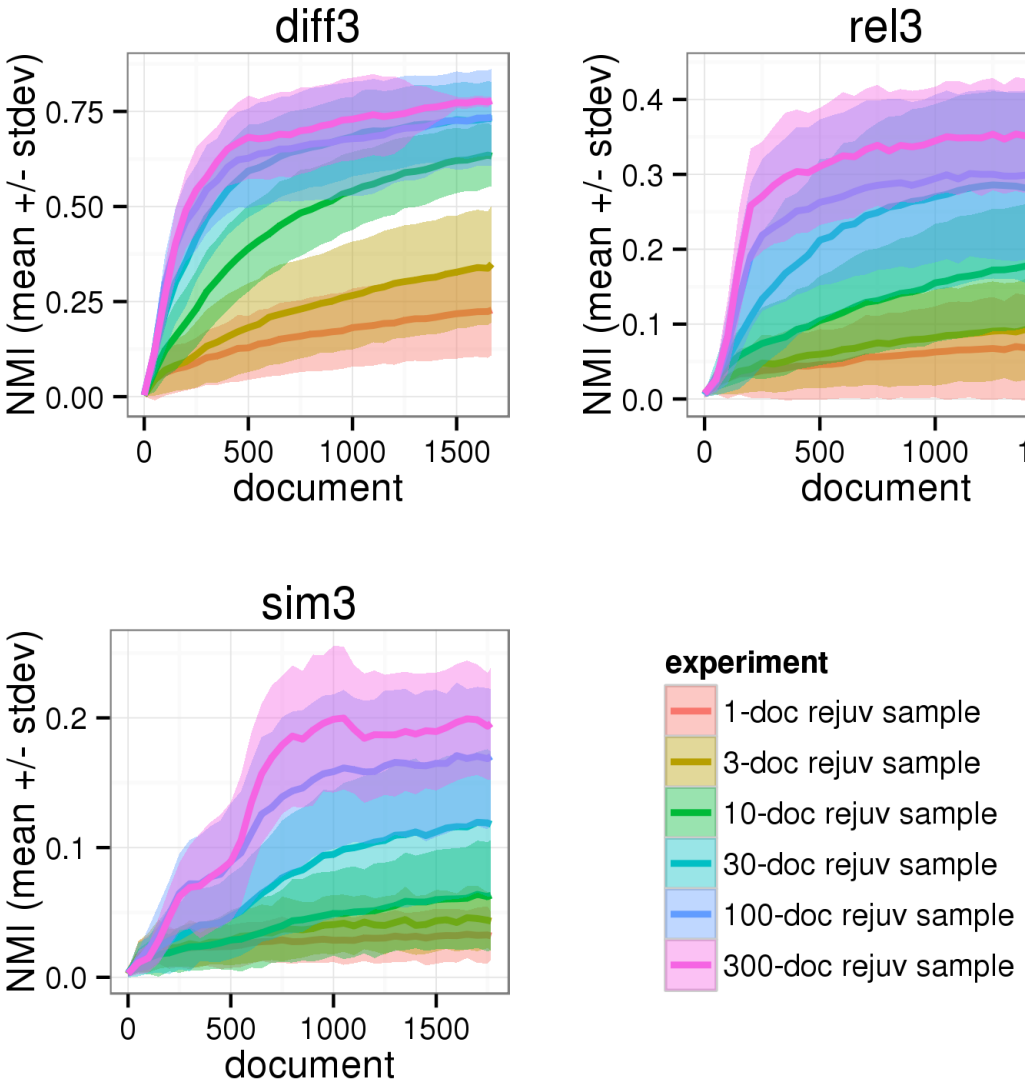


Figure 3.5: NMI over the course of training using large rejuvenation samples. Whereas rejuvenation samples of previous experiments were measured in *tokens*, these samples are measured in *documents* and typically consist of a much larger number of tokens.

On a different task, Börschinger and Johnson [2012] found rejuvenation to significantly improve performance, measured by token F-score, of a Bayesian word segmentation model. The best performance in that study was obtained by a degenerate (single-particle) particle filter with a rejuvenation sequence containing almost one-sixth of the training data; in this configuration, rejuvenation MCMC steps occur much more often than the particle state transitions, allowing the particle filter to revisit the word assignments of most of the characters in a given utterance several times. In my large-rejuvenation experiment (Section 3.6.4), I consider rejuvenation sequences of one or more documents, mirroring this setting, and I found that the particle filter was sensitive to the rejuvenation sequence size at these scales. However, in this setting, the runtime of the algorithm appears effectively quadratic because the particle filter revisits a large proportion of its past observations each time it makes a new one, such that the algorithm virtually becomes an iterative Gibbs sampler. Thus, in this setting, the algorithm does not satisfy the streaming criteria in practice.

I have also shown that tuning the initial model using in-sample NMI or held-out perplexity can improve mean NMI and reduce variance. Perplexity (or likelihood) is often used to estimate model performance in LDA [Blei et al., 2003b, Griffiths and Steyvers, 2004, Wallach et al., 2009b, Hoffman et al., 2010] and does not compare the inferred model against gold-standard labels, yet it appears to be a good proxy for NMI in my tuning experiment. Thus, if the particle filter's performance continues to exhibit sensitivity to initialization, at least we may have the flexibility of performing

that initialization without gold-standard labels. On the other hand, the data sets I have studied are small and approximately stationary due to randomization. I expect initialization to provide a smaller improvement for larger, less stationary data sets. Accordingly, I believe extending this study to larger data sets should be a priority for future work.

I have focused on NMI as my evaluation metric for the sake of comparison with Canini et al. [2009]. However, evaluation of topic models is a subject of considerable debate [Wallach et al., 2009b, Yao et al., 2009, Newman et al., 2010, Mimno et al., 2011] and it may be informative to investigate the effects of initialization and rejuvenation under a different metric, such as held-out perplexity or semantic coherence.

Perhaps the main difference between the use of particle filtering for LDA training proposed by Canini et al. [2009] and the typical uses of particle filtering is the unbounded state space in the LDA training application. In the case studied here, the state space grow with each SMC step and old states cannot be revised in general unless the memory usage of the algorithm is also allowed to grow with each step. Moreover, each dimension of the state space (topic assignment) is weighted equally, so the influence of each SMC step decreases, converging to zero, as the algorithm processes more and more data. Accordingly, I hypothesize that using a technique like exponential smoothing, as stochastic variational inference does in the global variational parameter update [Hoffman et al., 2013]—perhaps coupled with mini-batching to reduce variance, as stochastic variational inference does [Hoffman et al., 2013]—

could improve the performance of the particle filter.

3.8 Conclusion

I proposed reservoir sampling as a way to reduce the storage complexity of a particle filter from linear to constant, making it a streaming algorithm. However, in the process of reproducing the findings of prior work, I discovered that rejuvenation does not play a significant role in the experiments of Canini et al. [2009] in the first place. I also found that performance of the particle filter was largely determined by the initialization of the model, and I suggested a simple approach to reduce this sensitivity without using additional data. Ultimately, I found that the particle filter yields little improvement to the initial model except when the rejuvenation sequence size is very large, but at that scale, the runtime of the algorithm is effectively quadratic and no longer satisfies the streaming criteria. Therefore, while rejuvenation allows an LDA model of a certain quality to be learned via streaming particle filtering *in theory*, I found that the particle filter failed to produce an acceptable model in the streaming setting *in practice*.

Chapter 4

Features for Topic Identification

4.1 Preface

This work originally appeared in May et al. [2015a], a collaboration between local text-processing and speech-processing researchers. I was the primary author for the paper excepting sections 3 (the description of input representations), 4.2 (the description of multinomial i-vectors), and 5.2 (the topic discovery experiments using v-measure). Observing that the text and speech processing research communities both ultimately studied language but rarely collaborated, we sought to cross-pollinate the communities: to find techniques from the speech processing community that could benefit the text processing community and vice versa. We performed this cross-pollination by applying learned, lower-dimensional representations from the text and speech processing communities—notably including the multinomial i-vector model

from speech processing, which is omitted in the current presentation—to data from the text and speech communities for the common task of topic identification. In the course of this project, I discovered that topic model representations were not as well-suited to topic identification as I had expected. I will now elaborate.

Topic models allow us to infer a set of underlying topics from a corpus of documents, a task I call *topic discovery*. However, in some settings—for example, when performing a larger information extraction task—we may already have a collection of topics that we believe applies to the corpus, and what we need is to determine which of those topics apply to which documents. If those topics were originally produced by a topic model and are represented accordingly (typically a probability distribution over words from a fixed vocabulary), we can use the model to estimate the proportions of the topics latent in each document. However, in practice, the topic set is often defined by human domain experts and takes a less structured form. For example, a topic might be described by a paragraph, a sentence, or even just a word or phrase, precluding its direct application in a topic model. In these settings, supervised learning comes in handy: We can use examples of documents labeled with their respective topics to train a topic classifier. I call this task *topic identification*. The literature suggests that in some settings where supervised data is scarce, we can boost topic identification performance by first performing topic discovery on all documents and then using the resultant topic proportions as features for topic identification. Indeed, topic model features were applied to topic identification in the very paper that in-

troduced LDA, where LDA features were found to improve performance especially when labeled training data was scarce [Blei et al., 2003b]. Topic model features have also been applied to topic identification and related tasks in more recent work [Griffiths and Steyvers, 2004, Newman et al., 2008, 2009, Wintrode, 2011, Harwath and Hazen, 2012, Morchid et al., 2014b]. I test the applicability of topic discovery features to topic identification on conversational text, making the following contributions to science:

- This study is the first of its time to provide *cross-community* evaluations of SAGE and other models on both text and speech data.
- This study also uses low-resource triphone state cluster soft counts as speech data for topic ID, following May et al. [2015a]. The low-resource setting reflects constraints often faced in real-world applications, and I report topic ID performance under limited supervision to better illuminate the practical strengths and weaknesses of the learned representations.
- Finally, I believe that the comparison herein of several prominent learned representations on two complementary tasks on both text and speech, presented together in the same study, will provide a useful point of reference for future research.

While researching this project, I gleaned the following implicit theory:

Theory. *Under limited supervision, topic model features aid topic identification.*

However, in my experiments on conversational text, I find that several feature representations, including potentially less interpretable and higher-dimensional representations like tf-idf, yield higher classification accuracy than topic proportions, even when as few as two labeled examples are available per topic during training [May et al., 2015a]. Thus, although we are led to expect that topic discovery representations would be effective features for topic identification under limited supervision, that convergence is not borne out in practice.

4.2 Introduction

In this chapter, I assess the use of inferred topics from topic discovery as features for topic identification. To do so, I compare features from topic discovery with features from other dimensionality reduction techniques as well as high-dimensional baseline feature representations. I perform this comparison on two kinds of multinomial language data, one derived from text and another derived from speech. As dimensionality reductions, I consider the sparse additive generative (SAGE) [Eisenstein et al., 2011] and latent Dirichlet allocation (LDA) [Blei et al., 2003b] topic models as well as latent semantic analysis (LSA) [Deerwester et al., 1990]. The SAGE topic model represents a multinomial parameter vector as the softmax of a sum of vectors, one of which is a background vector representing overall word usage in the corpus; LDA is a more established topic model with no background vector; and LSA is a class of methods

based on the singular value decomposition (SVD). I evaluate all three learned representations on the supervised task of topic identification (topic ID). To perform this task, raw text or speech data is processed into multinomial counts, which are then transformed using one of the dimensionality reduction methods; a logistic regression classifier then predicts the topic of each document based on its representation. This pipeline is depicted in Figure 4.1.

Topic ID is the task of assigning topics to documents with a known topic set.[1] The two topic models, SAGE and LDA, were developed for the related task of topic discovery, in which an unknown topic set is inferred (discovered) from a set of documents, assigning topics to documents in the process. Intuitively, these two tasks concern the same information about a set of documents (the information encoded by "topics"), and we might reasonably expect that a set of topic-document assignments produced by topic discovery would be effective features for topic identification.

More concretely, Blei et al. [2003b] demonstrate the utility of LDA by using it to produce features for topic ID, observing that "in almost all cases the performance is improved with the LDA features" [Blei et al., 2003b]. In another foundational paper, Griffiths and Steyvers [2004] infer the topics of a corpus of journal article abstracts using LDA and find "strong diagnostic topics for almost all of the minor categories" [Griffiths and Steyvers, 2004], where those categories were created and assigned by experts. Thus, there is an implicit theory that topic identification benefits

[1] For an introduction to text classification, including topic classification, see Manning et al. [2008, Ch. 13].

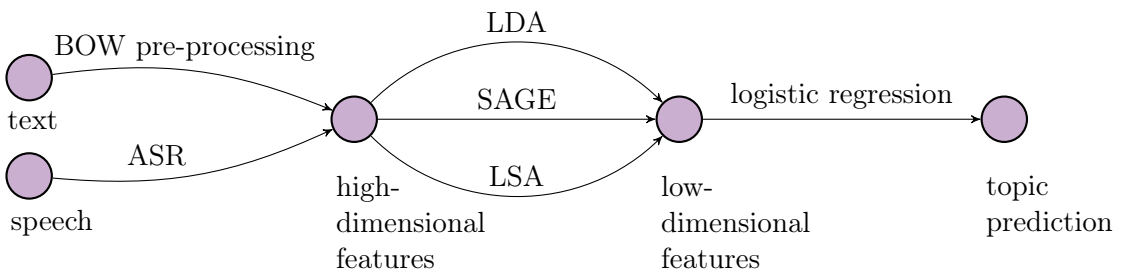


Figure 4.1: Depiction of the topic ID pipeline.

from features derived by topic modeling (topic discovery). In this chapter, I test that theory.

I use the *bag-of-words* multinomial representation of text data; that is, each document is represented by a vector of counts over the word vocabulary. For speech data, I use a modern automatic speech recognition (ASR) system to produce frame-wise triphone state cluster posteriors and I take the sum of these posteriors across all frames in a document to obtain a document-level vector of triphone state cluster soft counts. This choice marks a break from convention: Modern topic ID systems for speech use ASR output instead of a lower-resource representation like triphone state cluster soft counts in order to improve performance [Hazen et al., 2007]. ASR word counts are high-resource and can be viewed as a noisy version of word counts from text. However, I wish to assess the relative performance of my learned representations, not the quality of the data pre-processing scheme, and I desire to strengthen my results by evaluating performance on two distinct views of a corpus. Thus, I use triphone state cluster soft counts to represent speech data.

4.3 Background

Previous work has compared LDA and other dimensionality reduction techniques. Chen et al. [2014] compared a multinomial i-vector language model against LDA and other models on the task of spoken document retrieval, and found the multinomial i-vector model to significantly outperform the other models on words, but not on sub-words (syllable pairs), derived from ASR. The syllable pairs are similar in granularity to the triphone state clusters used as multinomial speech data in the current work.

Morchid et al. [2014a] improved conversation theme identification by employing LDA and a Gaussian i-vector model in a pipeline. They learn LDA models of varying dimensions (numbers of topics) on ASR output and use them to generate a suite of feature vectors. The feature vector for each document-dimension pair is created by marginalizing over topics according to the document's inferred topic proportions. A Gaussian i-vector model is then learned on those feature vectors; the i-vectors are normalized and used to identify document themes via the Bayes decision rule.

In this study, I have a fundamentally different approach from that of Morchid et al. [2014a]. First, I treat topic models as representations themselves, directly comparing SAGE and LDA, while Morchid et al. use LDA as a pre-processing step for computing Gaussian i-vectors. Second, I use triphone state cluster soft counts instead of ASR word counts, hence my representation of speech data is significantly lower-resource. Third, I also evaluate performance on text data, and where Morchid et al. limit their vocabulary (from ASR) to 166 task-specific words, I use all 26 606 words present in

my training data.

4.3.1 Input Representations

I use the speech and text data set from May et al. [2015a]. This data comes from a subset of the Fisher English corpus [Cieri et al., 2004c] that has been annotated for topic ID [Hazen et al., 2007]. It consists of audio recordings of telephone conversations [Cieri et al., 2004a] and manual transcriptions of those recordings [Cieri et al., 2004b]. In this study, I use speech data consisting of triphone state cluster posteriors inferred from the recordings and text data consisting of bag-of-words representations of the transcripts. The topic ID annotations comprise 40 topics, some of which are used more than others. Further details about the data set are provided in May et al. [2015a].

4.4 Learned Representations

I consider three main dimensionality reduction models: the SAGE and LDA topic models and LSA. The learned representations I consider explain which words appear in a document d via a latent, lower-dimensional representation $\boldsymbol{\theta}^{(d)}$. All representations operate under a bag-of-words assumption. To compare topic models and LSA, I find it useful to formulate each learned representation as operating on different *contexts* (subsets) c of a document; such a formulation does not negate the fundamental

bag-of-words assumption. The three models represent the words that appear in context c—either the entire document or each token—via multinomial-style parameters $\boldsymbol{\phi}^{(c)}$.[2,] Each model consists of K *components* (e.g., a K-dimensional affine subspace), and shared parameters $H_{k,w}$ prescribe the amount of weight each component k places on each vocabulary word w. The models construct $\boldsymbol{\phi}^{(c)}$ by combining $\boldsymbol{H}$ and $\boldsymbol{\theta}^{(d)}$; in some cases empirical word statistics $\mathbf{m}$ are also used to stabilize the representations.

4.4.1 LSA

LSA [Deerwester et al., 1990] factorizes a term-document matrix by truncated SVD, learning the projection of the data onto a linear subspace of fixed rank such that the approximation error of the reconstructed term-document matrix (as measured by the Frobenius norm) is minimized. In the basic version of LSA, SVD is applied to the raw term counts, giving the low-dimensional representation

$$\boldsymbol{\phi}^{(d)} = \boldsymbol{H}\boldsymbol{\theta}^{(d)},$$

where $\boldsymbol{\phi}^{(d)}$ is the vector of observed multinomial counts in document d, $\boldsymbol{H}$ is the matrix of left singular vectors of the term-document count matrix, and $\boldsymbol{\theta}^{(d)}$ is the inferred representation of $\boldsymbol{\phi}^{(d)}$. In practice, LSA is often applied instead to the term-document matrix weighted by term frequency–inverse document frequency (tf-idf) in

[2]Other efforts have modeled documents with intermediate granularity, e.g., sentence-level [Titov and McDonald, 2008] or entity-level [Newman et al., 2006] granularity.

order to normalize terms by importance. We can also apply further pre-processing steps, such as term-wise centering by subtracting the column-wise mean $\mathbf{m}$ of the data, in which case LSA finds an affine subspace that approximates the data.

4.4.2 Bayesian Discrete Topic Models

Bayesian topic models explain word occurrences via K latent components $\boldsymbol{H}_k$ (topics) each drawn from some prior distribution G. Unlike LSA, multinomial topic models are admixture models: Each token w_i is drawn from a multinomial distribution parametrized by $\boldsymbol{H}_{z_i}$. Latent token assignment variables z_i, taking integral values between 1 and K (indexing $\boldsymbol{H}$), dictate the token's topic choice. The document d_i controls how often each topic is chosen via the K-dimension multinomial distribution parametrized by $\boldsymbol{\theta}^{(d_i)}$. In the parametric settings I consider, Dirichlet priors are often placed on the topic proportions $\boldsymbol{\theta}^{(d)}$, allowing experimentation with the topic representation $\boldsymbol{H}$.[3] Topic-specific word distributions $\boldsymbol{\phi}^{(k)}$ are formed by a mapping $Q(\boldsymbol{H}_k)$, possibly the identity, ensuring $\boldsymbol{\phi}^{(k)}$ are probability vectors. A

[3]There have been many efforts to provide or induce latent structure among the topics [Blei et al., 2003a, Li and McCallum, 2006, Wallach et al., 2009a, Paul and Girju, 2010], but most models ground out to Dirichlet and discrete random variables.

general formulation is

$$
\begin{aligned}
w_i|z_i, \boldsymbol{\phi}^{(z_i)} &\sim \text{Multinomial}\left(\boldsymbol{\phi}^{(z_i)}\right) \\
\boldsymbol{\phi}^{(k)} &= Q\left(\boldsymbol{H}_k\right) \\
\boldsymbol{H}_k &\sim G(\boldsymbol{\eta}) \\
z_i|\boldsymbol{\theta}^{(d_i)} &\sim \text{Categorical}\left(\boldsymbol{\theta}^{(d_i)}\right) \\
\boldsymbol{\theta}^{(d)} &\sim \text{Dirichlet}(\boldsymbol{\alpha}).
\end{aligned}
$$

The hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ dictate the informativeness of the priors over $\boldsymbol{H}_k$ and $\boldsymbol{\theta}^{(d)}$: Often (empirically optimized) symmetric hyperparameters are employed, resulting in a form of Laplace smoothing during topic estimation. In the current work, I follow this strategy, noting that there have been concerted efforts to encode domain or expert knowledge via the hyperparameters [Gormley et al., 2012, Paul and Dredze, 2015].

4.4.2.1 SAGE Topic Model

The Sparse Additive Generative (SAGE) model [Eisenstein et al., 2011] is a generative Bayesian modeling framework in which the word distribution $\boldsymbol{\phi}$ for each token is formed by summing a background vector and one or more sparse vectors generated from appropriate priors. Those additive components can reflect the contributions of documents, aspects, topics, or other factors chosen by the modeler. A basic

SAGE topic model sets $\boldsymbol{\phi}^{(k)} = \text{softmax}\left(\mathbf{m} + \boldsymbol{H}_k\right)$ and draws $\boldsymbol{H}_k$ from some sparsity-inducing distribution G, for example, the Laplace distribution. As $\mathbf{m}$ is a shared background frequency vector, $\boldsymbol{H}_k$ is the learned residual frequency vector of topic k.

Replacing the assigned topic in SAGE by its conditional expectation gives

$$\begin{aligned}\tilde{\boldsymbol{\phi}}^{(z_i,d_i)} &= \text{softmax}\left(\mathbf{m} + \text{E}_{z_i}\left[\boldsymbol{H}_{z_i}\middle|\boldsymbol{\theta}^{(d_i)}, \boldsymbol{H}\right]\right) \\ &= \text{softmax}\left(\mathbf{m} + \boldsymbol{H}\boldsymbol{\theta}^{(d_i)}\right).\end{aligned}$$

This "marginal SAGE" model could be useful in future work: The marginalization may mitigate the problem of topic-switching, yielding a more identifiable (but perhaps less interpretable) model and lending to downstream tasks such as topic ID.

4.4.2.2 LDA

Latent Dirichlet Allocation (LDA) [Blei et al., 2003b] is a generative Bayesian topic model similar to SAGE, but in which each topic is drawn from a Dirichlet prior G rather than a sparsity-inducing distribution. LDA does not explicitly account for the background distribution; to account for this, it is common practice to threshold the vocabulary *a priori* to remove very common and very rare words (though in my experiments, I do not do this). For LDA, $\boldsymbol{\phi}^{(z_i)} = \boldsymbol{H}_{z_i}$, and $\boldsymbol{H}_k \sim \text{Dirichlet}\left(\boldsymbol{\eta}\right)$ with $\eta = \beta$.

4.5 Experiments

I compare these three models of learned representations empirically on the task of topic ID. I use a C++ implementation of **SAGE**[4] that uses approximate mean-field variational inference as in Eisenstein et al. [2011]. I learn the **LDA** model using the MALLET implementation of Gibbs sampling [McCallum, 2002].[5] I perform **LSA** using centered tf-idf–weighted word counts and centered l_2-normalized triphone state cluster soft counts. I implement tf-idf by scaling the raw term count by the log inverse document frequency. I apply l_2 normalization rather than tf-idf weighting to the speech data because it is dense and tf-idf is thus inappropriate. On both text and speech, mean-centering is performed *after* the respective normalization, as this pre-processing recipe performed best of all the variants I tried.

For each of the three models, the low-dimensional real vector $\boldsymbol{\theta}^{(d)}$ represents a given document d in my experiments. I also consider two high-dimensional baseline representations: **raw** (soft) counts on both the text and speech data, and, only on the text data, **tf-idf**–weighted word counts.[6] These tf-idf weights constitute a high-dimensional *learned* representation.

In my first topic ID experiment I evaluate topic ID error on raw multinomial views

[4] `https://github.com/fmof/sagepp`

[5] For Gibbs sampling, fractional counts are truncated.

[6] I do not apply tf-idf to the speech data because tf-idf requires hard (discrete, sparse) counts in practice and my speech data is represented by soft (continuous, dense) counts. If I were to apply tf-idf to the speech data, the document frequencies would be similar (close to the number of documents in the corpus) for most triphone state clusters because of the low sparsity of that representation [May et al., 2015a].

of the data. In subsequent experiments I explore the interaction of representation dimensionality with each model and dataset and evaluate the classifier when it is only given a fraction of the available data for training. This latter configuration is the most interesting as it reflects the cost of obtaining supervised data in practice.

Given feature vectors for some representation of the documents in a corpus, topic ID is performed in a one-versus-all framework. I use logistic regression as the per-class binary classifier, implemented using LIBLINEAR [Fan et al., 2008]. Results were similar when logistic regression was replaced by support vector machines. All document representations are length-normalized (divided by their l_2 norm) before they are input to the classifier. Performance is measured by topic ID error, the error of multi-class prediction where the class predicted for each document is that of the per-class classifier that gave it the highest weight. Baseline performance on the test set (where the baseline classifier chooses the most prevalent topic in the training set for all test examples) is 96.2% error. Note that this error rate differs from the uniform-at-random classification error rate of 97.5% because of the uneven distribution of topics.

4.5.1 Document Construction

Prior work [Hazen et al., 2007, Wintrode and Khudanpur, 2014] treated whole conversations as documents in addition to separating each conversation into its two sides. I perform a small topic ID experiment in this configuration to probe the impact of this

design choice. Ten-fold cross-validation (CV) is used to tune the logistic regression regularizers. On the test set, the classifier achieves topic ID error of 12.4% and 15.6% for whole-conversation and individual-side text data, respectively, and 20.1% and 29.5% for whole-conversation and individual-side speech data, respectively. These results correspond roughly to results listed in Table 3 of Hazen et al. [2007], specifically, the topic ID error of 8.2% and 12.4% for whole-conversation and individual-side transcriptions, respectively, and 22.9% and 35.3% for whole-conversation and individual-side triphones derived from ASR lattices, respectively [Hazen et al., 2007]. However, I use logistic regression without feature selection instead of Naïve Bayes with feature selection, and I apply my classifier to triphone state cluster soft counts inferred by a DNN instead of triphone counts from ASR lattices. I believe that the discrepancies in performance with respect to prior work are due to these differences in experimental configuration. My results and those of prior work show that using whole-conversation documents instead of individual-side documents make the topic ID task easier. As a result, I expect that differences in performance between the different learned representations will be more clearly pronounced on individual conversation sides and I restrict the rest of my study to that setting.

4.5.2 Dimensionality Study

I perform topic ID on learned representations at dimensions of $K = 10$, 50, 100, 200, 300, and 600 on individual conversation sides, using ten-fold cross-validation to

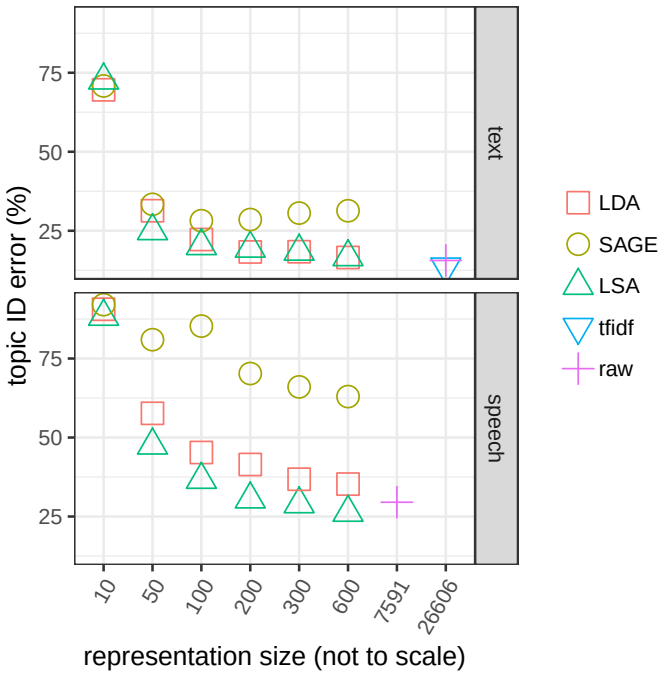


Figure 4.2: Topic ID error (%) on the test set for raw and tf-idf representations and lower-dimensional learned representations at dimensions of $K = 10$, 50, 100, 200, 300, and 600.

tune the logistic regression regularizers. Figure 4.2 gives topic ID error results on the test set, varying K; selected values are listed in Table 4.1. In both datasets, as the dimension K of a learned representations increases, topic ID error decreases, approaching (approximately) the raw baseline. On text, tf-idf performs slightly better than the raw representation. LSA is marginally the best-performing lower-dimensional learned representation; LDA performs well at some representation sizes, depending on the data source, but is less consistent. SAGE performs poorly overall.

view	model	dimension	error
text	LDA	600	16.5
text	SAGE	600	31.3
text	LSA	600	16.7
text	tf-idf	26 606	13.6
text	raw	26 606	15.6
speech	LDA	600	35.3
speech	SAGE	600	63.0
speech	LSA	600	26.2
speech	raw	7591	29.5

Table 4.1: Selected topic ID error (%) values from Figure 4.2.

4.5.3 Limited Data Study

The raw text and speech representations (multinomial observations) are very high-dimensional, and the classifier is likely to overfit to specific components (words or triphone state clusters) in these representations. To measure this effect and attempt to separate the predictive power of logistic regression from the quality of the learned representations in my analysis, I experiment with reducing the number of *labeled* training examples the *classifier* can use; I still learn representations on the full (unlabeled) training set. This experiment represents the limited-supervision setting in which supervised data is costly to obtain but unlabeled data abounds.

I run this experiment twice, using $\ell = 2$ and $\ell = 6$ labeled examples per topic, for a total of 80 and 240 classifier training examples, respectively. Ten-fold cross-validation is used to fit the regularizer; per-class loss coefficients are set according to the class prior in the original training set in order to counteract the artificial balancing of the classes in the limited-supervision dataset. I report cross-validation

estimates of the topic ID error on the training set for $K = 10$ (Figure 4.3), $K = 100$ (Figure 4.4), and $K = 600$ (Figure 4.5). For $K = 100$ and $K = 600$, LSA dominates in the limited-supervision setting. SAGE performs poorly overall;[7] LDA performs significantly better than SAGE but worse than LSA. Finally, tf-idf–weighted word counts perform very well on text, often achieving the best performance of all representations even under limited supervision.

4.6 Discussion

I have theoretically and empirically compared several content-bearing representations from prior work, measuring their relative performance as features for topic ID. In the full-supervision setting, the lower-dimensional learned representations converge in performance to the raw representation as the dimension K increases. However, if only a couple of labeled examples per class are available—which reflects the expense of obtaining labels in practice—then learned representations generally outperform the raw representation, which is more prone to overfitting. Among learned representations, LSA consistently outperforms the topic discovery (topic model) representations.

[7] I believe that approximately sparse posterior $\boldsymbol{\theta}^{(d)}$ values result in a kind of topic switching, contributing to the poor performance of SAGE. To test this hypothesis, I "tested on train" and analyzed the top topics inferred for each document: While the highest-weighted topic tended to be consistent, SAGE infers approximately sparse $\boldsymbol{\theta}^{(d)}$ with large variation in the next four highest-weighted topics (the remaining topics are assigned trace mass). Second, a phenomenon known as conversation drift, described in May et al. [2015b], is so pronounced in Fisher that the first 25% percent of words of each conversation side are nearly as predictive as the entire document [Wintrode, 2013]. All representations must contend with this drift, but SAGE may be particularly susceptible due to sparsity in $\boldsymbol{\theta}^{(d)}$. These two issues may make the classification I use much less robust.

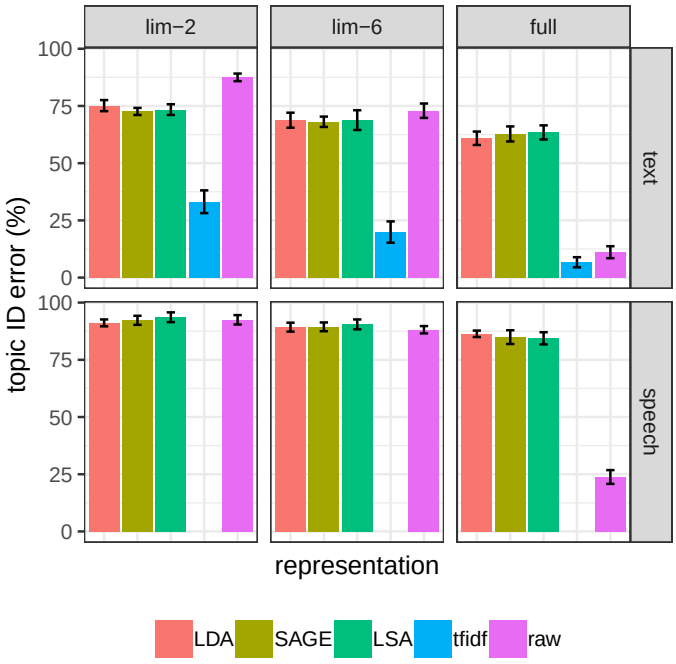


Figure 4.3: CV topic ID error (%) for raw and tf-idf representations and lower-dimensional learned representations of size $K = 10$. Error bars denote plus and minus one standard deviation according to the CV empirical distribution.

Tf-idf performs surprisingly well in the limited supervision setting; it is learned from the data, but it should be prone to overfitting due to its high dimensionality. Moreover, tf-idf outperforms the learned representations by the widest margin for $K = 10$, which is relatively close to typical dimensionalities of topic models. While LDA's relative performance improves for higher K, those dimensionalities are rare in topic modeling practice, as a very large number of topics may be intractable for human analysts. It is also surprising that SAGE performance on text degrades significantly at high dimensions; I suspect this is due to topic switching, but further investigation is warranted.

Word counts and triphone state cluster soft counts provide only one view of text

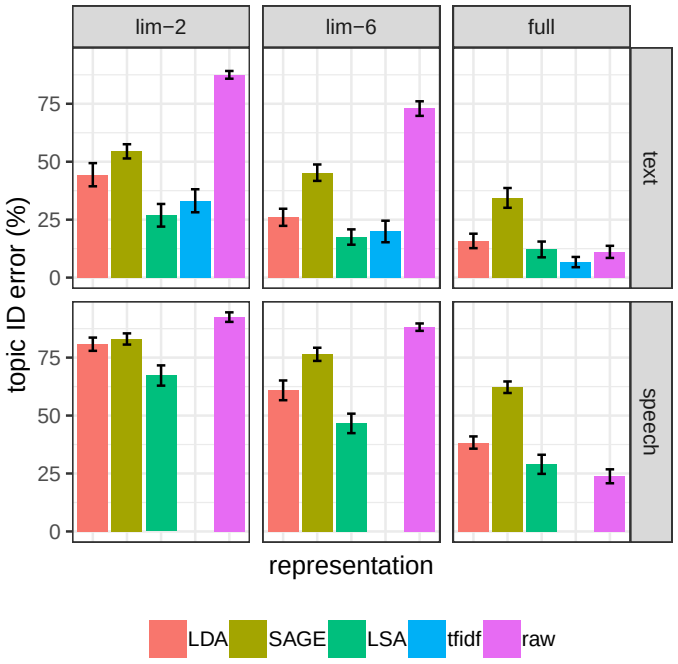


Figure 4.4: CV topic ID error (%) for raw and tf-idf representations and lower-dimensional learned representations of size $K = 100$. Error bars denote plus and minus one standard deviation according to the CV empirical distribution.

and speech (respectively), and other input representations may yield different conclusions. The particular LSA approach I used for text, based on tf-idf weighting, is not as appropriate for my speech data, which is dense. Future work could evaluate other implementations of LSA or use a higher-level view of speech, such as triphone state cluster n-grams, that more naturally exhibits sparsity and lends to tf-idf weighting. In particular, weighting by a likelihood ratio test statistic and applying a log transform has generated better performance in several other tasks [Lapesa and Evert, 2014]. Future work could also test my conclusions on higher-resource views of speech such as ASR word counts or lower-resource views such as mel-frequency cepstral coefficients (MFCCs).

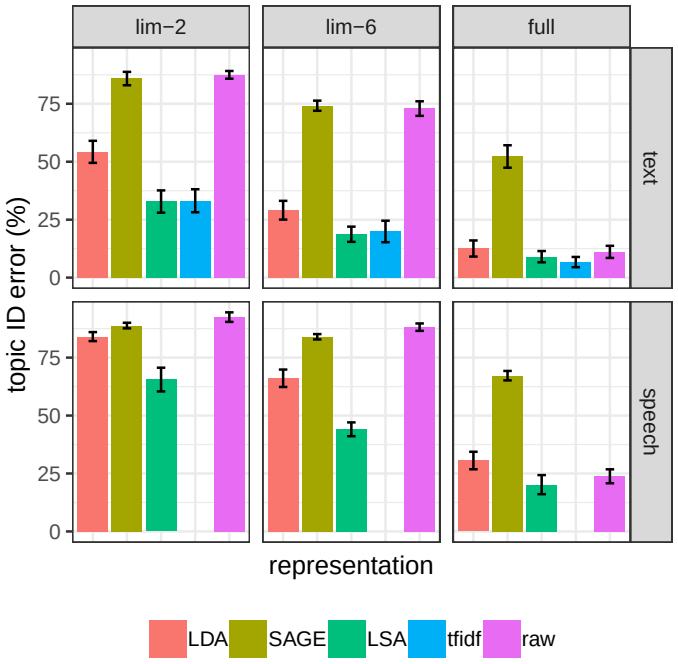


Figure 4.5: CV topic ID error (%) for raw and tf-idf representations and lower-dimensional learned representations of size $K = 600$. Error bars denote plus and minus one standard deviation according to the CV empirical distribution.

I have provided a brief evaluation of learned representations on multinomial text and speech data. Prior work has evaluated related learned representations on text data alone, surveying parameters and tasks at greater breadth [Lapesa and Evert, 2014, Levy et al., 2015]. A similarly comprehensive evaluation spanning the text and speech research communities would demand great effort but serve as a proportionately large and versatile resource. In complement, a detailed, case-by-case analysis of errors made by the models in my study could illuminate future modeling efforts by exposing exactly how and why each model errs or excels in each setting.

I suspect the reason that LDA does not perform as well as other feature representations is simply that the effective definition of "topic" differs between the tasks

of topic discovery and topic identification. Indeed, the earnest evaluation of topic models is difficult in part because the notion of "topic" is ill-defined. In any case, I hypothesize that the poor performance of topic discovery features for topic identification is at least partly due to mismatched objective. To improve the performance of such features, I would propose applying a semi-supervised topic model such as LDA with Dirichlet Forest priors [Andrzejewski et al., 2009] or the anchored Correlation Explanation model [Gallagher et al., 2017] to incorporate the (limited) supervision directly into the topic modeling objective. I would also suggest a qualitative comparison of the tf-idf and LDA features used in topic identification and the gold-standard topics, as that analysis may reveal specific shortcomings of the LDA representation.

4.7 Conclusion

The literature suggests that topic discovery and topic ID rely on the same basic information in a collection of documents. I've conducted preliminary experiments to test this hypothesis in a controlled setting. On both speech and text transcripts derived from topic-labeled conversations, I find that representations derived from topic models are consistently outperformed by common non–topic-model representations as features for topic ID. Therefore, while topic models and topic identification may share a common underlying representation *in theory*, I find that congruence is not borne out in practice.

Chapter 5

Language Independence

5.1 Preface

This chapter is split into two parts: a study using a manual (human-in-the-loop) evaluation and a larger study using an automated evaluation. The original manual study was a collaboration [May et al., 2016], and in the following, I reproduce without further notice those parts of the study for which I was primary author, citing the original study for work done primarily by my collaborators. This study was originally performed and submitted for publication prior to the publication of Schofield and Mimno [2016], which was performed independently and ultimately "scooped" our study. That said, Schofield and Mimno [2016] analyzed a different language, took a different methodological approach, and provided a much more comprehensive analysis altogether. In the second part of this chapter, I build on the foundation laid by

Schofield and Mimno [2016] to extend my previous study.

Natural language processing research has long focused almost exclusively on English-language text. Topic modeling research is no exception, and because most topic models treat each document as a "bag of words," they are sometimes framed as language-independent technologies or applied uniformly across languages [Mimno et al., 2009]. However, effective use of a topic model is inevitably contingent on the application of various preprocessing and/or postprocessing steps. Indeed, some authors argue that stemming or lemmatizing the text before training may be necessary: "For languages with a richer morphology, [stemming] is particularly critical . . . . However, for English, this is more a matter of taste. When topics are designed for human inspection, many users prefer not to see stemmed words" [Boyd-Graber et al., 2014]. Similarly, Schofield and Mimno write: "Although stemmers are commonly used in topic models . . . we find no empirical benefits for the practice" [Schofield and Mimno, 2016]. Hence, I inferred the following implicit theory (coexisting with a contradicting theory):

Theory. *The effective usage of LDA is similar across languages.*

Using both human-in-the-loop and fully automated procedures, I evaluated topic models trained on lemmatized and unlemmatized corpora across several languages, providing several contributions to science:

- This chapter is one of the first published studies of the impact of lemmatization on topic modeling.

- I have performed a human-in-the-loop analysis of topic model interpretability and and an automated analysis using proxy methods on a similar data set, increasing collective knowledge about the practical relationship between costly human-in-the-loop evaluation methods and automated proxy methods.

- I have analyzed the impact of lemmatization on topic models in several languages by re-using methods from Schofield and Mimno [2016], increasing the comparability of our studies and facilitating future work.

- I have publicly released my experiment code for the benefit of future research.[1]

Ultimately, I found that topic model quality improved with lemmatization in some cases I studied but not in most. Thus, although LDA is sometimes presented as a language-independent technology in theory, its effective application in practice benefits from a language-dependent treatment.

5.2 Introduction

Topic modeling is often portrayed as a tool for the unsupervised analysis of text corpora, regardless of their language or domain. In fact, Blei et al. [2003b] introduce LDA as "a generative probabilistic model for collections of discrete data such as text corpora" [Blei et al., 2003b], suggesting that LDA not only applies across languages, but beyond language altogether. The generality of LDA is a theme reproduced in

[1] `https://github.com/ccmaymay/lda-lemmas`

later work; for example, in an extension of LDA that models multiple languages in parallel, Mimno et al. [2009] treat data from different languages uniformly. However, while topic modeling discourse supports a theory that the effective usage of LDA is language-independent, it also seems to support the opposing theory, that the usage of LDA is language-*dependent*: For example, Boyd-Graber et al. [2014] write that "for languages with a richer morphology, [stemming] is particularly critical" [Boyd-Graber et al., 2014]. These two theories appear to exist *simultaneously*, such that the theory encountered when reading the literature (for example) depends on context. In this chapter, I investigate the former theory, that effective topic modeling is language-*independent*.

Topic models pick up on the co-occurrence signal between different words in a corpus, such that words that occur often in the same document are likely to belong to the same latent topic. In languages that exhibit rich inflectional morphology, the signal becomes weaker because of the proliferation of different word tokens with similar meanings. While lemmatization (or stemming) is often used to preempt this problem, its effects on a topic model are studied privately, or perhaps assumed; they are not studied in published work.[2] In this study, I provide some of the first published measurements of the effect of token-based lemmatization on topic models across languages of varying morphological richness.

Syntactic information is not typically considered to exert a strong force on the

[2]For a more detailed discussion of topic modeling practice including stemming in different languages, see Boyd-Graber et al. [2014].

thematic nature of a document. Indeed, for this reason topic models often make a bag-of-words assumption, discarding the order of words within a document. In morphologically rich languages, however, syntactic information is often encoded in the word form itself. This form of syntactic variation is a nuisance variable in many topic modeling applications, "polluting" topic representations learned from data [Boyd-Graber et al., 2014]. As May et al. [2016] explain, while there is only form of the Russian name *Putin* in the English language, there are many forms in the Russian language, including Путин, Путина, Путину, Путине, and Путином. The choice of which form to use depends on the surrounding sentence's syntactic structure. Additionally, relations involving prepositions and other *stop words* in English are often marked by inflectional suffixes in Russian [May et al., 2016]. Topic models are generally sensitive to the presence of prepositions and other stop words in English [Wallach et al., 2009a, Blei et al., 2010, Eisenstein et al., 2011], so we might expect them to be sensitive to morphological variation in languages like Russian.

In this chapter, I use a variety of methods to test the language-independence theory of topic modeling. First, in a small preliminary study, I train topic models on a corpus of Wikipedia articles with and without preprocessing by lemmatization. Using a human-in-the-loop procedure with Russian language experts, I evaluate the effect of lemmatization on a model's interpretability. I then extend my study to multiple languages and lemmatizers, switching to automated evaluation procedures to account for the increase in scale. Specifically, I evaluate topic models using several

automated evaluation procedures on English, Farsi, Korean, and Russian Wikipedia articles subject to no lemmatization, lemmatization using the same lemmatizer as before, and lemmatization using a different lemmatizer. Finally, I summarize my findings across these studies to argue that effective topic modeling in practice is language-dependent in general.

5.3 Background

Following May et al. [2016], I focus on *inflectional morphology*, a kind of variation in word structure that marks syntactic properties like number and gender. English has relatively little inflectional morphology, including variation in number (*dancer* and *dancers*) and degree of comparison (*fancier* and *fanciest*). However, languages like Russian use inflectional morphology more prominently, with Russian nouns having twelve word forms and Russian verbs having more than thirty [May et al., 2016]. Many natural language processing techniques, including topic models, generally treat words as discrete units, so the proliferation of word forms for the same underlying word type in languages like Russian creates sparsity in the data. To mitigate that increase in sparsity, we can *lemmatize* the data, applying an approximation to map each word form to a *lemma* representing the underlying word type. In this study I use the TreeTagger lemmatizer [Schmid, 1994] of May et al. [2016].

There are other kinds of morphology that one might consider in topic modeling.

Inflectional morphology refers to variation within a *lexeme*, a set of word forms sharing a lemma but referring to the same underlying concept. There is also variation in word formation between lexemes, including derivation and compounding forms. *Derivation* refers to the formation of a word (lexeme) from another word by adding affixes that are not themselves independent words. For example, the word *dancer* is derived by affixing the non-word *r* to the word *dance*. While *dancer* and *dancers* refer to different numbers of the same underlying concept, *dance* refers to a slightly different concept altogether, namely, the thing that dancers do. *Compounding*, on the other hand, refers to the formation of a word from two or more other independent words. For example, the word *dancewear* is formed from the independent words *dance* and *wear* and represents a new concept altogether.

Given that topic models represent corpora by collections of topics, or concepts, I focus on the role of inflectional morphology in topic modeling. Because word formation creates words representing different concepts from their components, normalizing variation from derivation and compounding (for example) is susceptible to change the meaning of some word tokens. In contrast, controlling for inflectional variation does not change the underlying meaning of a token in principle.

To study the effect of lemmatization on topic models in practice, we must operationalize the concept of "topic models." In this study, for comparability with other work, I restrict my attention to latent Dirichlet allocation (LDA) [Blei et al., 2003b], the foundational Bayesian graphical topic model. I measure the performance of a

topic model with metrics representing its interpretability, as topic models are most fit for discovering human-interpretable decompositions of the data [May et al., 2015a].

5.3.1 Related Work

There are more modern but less widely-used topic models than LDA, such as the sparse additive generative (SAGE) topic model, which explicitly models the background word distribution and encourages sparse topics [Eisenstein et al., 2011], or the nested hierarchical Dirichlet process (nHDP) topic model, which represents topics in a hierarchy and automatically infers its effective size [Paisley et al., 2015]. These models may be more interpretable by some measures but are less widely used and accessible. Separately, the infinite-vocabulary LDA model has a prior similar to an n-gram model [Zhai and Boyd-Graber, 2013], which could be viewed as loosely encoding beliefs of a concatenative morphology, but the effect of that prior has not been analyzed in isolation. I seek to measure the impact of lemmatization on a topic model and would like my results to be applicable to research and industry, so I leave these alternative topic models as considerations for future work.

Though stemming and lemmatization have long been applied in topic modeling studies [Deerwester et al., 1990, Hofmann, 1999, Mei et al., 2007, Nallapati et al., 2008, Lin and He, 2009], their effect on a topic model was publicly investigated only recently, in a comparison of rule-based and context-based stemmers in LDA topic models on four English corpora [Schofield and Mimno, 2016]. Overall, stemming was

found to reduce model fit, negligibly affect topic coherence, and negligibly or negatively affect model consistency across random initializations. In light of these results, Schofield and Mimno [2016] recommended refraining from stemming the corpus as a pre-processing step and instead stemming the topic keys as a post-processing step, as needed. The primary distinction between this work and Schofield and Mimno [2016] is my consideration of languages other than English. While my preliminary study using a human-in-the-loop evaluation on Russian Wikipedia was performed prior to the publication of Schofield and Mimno [2016], I performed the larger, multilingual follow-up study more recently and reused many of the methods of Schofield and Mimno [2016].

As observed in May et al. [2016], morphology has been studied relatively little in the context of topic modeling; it has received more substantial attention, however, in the context of related but more recent word embedding models [Bian et al., 2014, Soricut and Och, 2015, dos Santos and Zadrozny, 2014, Ling et al., 2015].

5.4 Manual Evaluation

In my first study, I use a human-in-the-loop procedure to evaluate the effect of lemmatization on LDA topic models trained on Russian Wikipedia articles.

Recall that for some pre-specified number of topics K and Dirichlet concentration hyperparameters $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$, the LDA topic model represents a corpus by a set of K

model	topic keys
lemmatized	деревня* сельский поселение пункт сельсовет
none	деревня* деревни* деревне* жителей волости
lemmatized	клетка лечение* заболевание препарат действие
none	лечения* течение лечение* крови заболевания
lemmatized	японский* япония корея префектура смотреть
none	считается японии японский* посёлок японской*
lemmatized	художник* искусство художественный* картина выставка†
none	искусства музея картины выставки† выставка†

Table 5.1: Topic pairs from my coauthored paper [May et al., 2016, Table 2], reproduced with permission. We manually aligned four topics from the lemmatized model with four topics (respectively) from the unlemmatized ("none") model based on their content. We used the symbols * and † to mark words sharing a lemma; for example, in the first topic pair, the words from the unlemmatized model деревня, деревни, and деревне are marked with the * symbol to indicate they have the same lemma, деревня (the Russian word for *village*). In this reproduction, I have color-coded the words by lemma to aid interpretation. The topic keys from the unlemmatized model are less informative, as they contain largely redundant information in the form of multiple forms of the same underlying word type.

i.i.d. topics $\boldsymbol{\phi}^{(k)}$, represents each document d as an i.i.d. mixture over those topics (with mixture weights $\boldsymbol{\theta}^{(d)}$), and specifies that each token in a document is generated by sampling a word type from the document's topic mixture. Meaningful evaluation of topic models is notoriously difficult and has received considerable attention in the literature [Chang et al., 2009, Wallach et al., 2009b, Newman et al., 2010, Mimno et al., 2011, Lau et al., 2014]. Given common applications of topic models, an evaluation metric that correlates with a human's ability to use the model to explore or filter a large dataset, hence the interpretability of the model, is preferable. In this study, I moreover require an evaluation metric that is comparable across different treatments of the same corpus, specifically an unlemmatized treatment and a lemmatized

treatment.

With those concerns in mind I choose a *word intrusion* evaluation: A human expert is shown one topic at a time, represented by its m topic keys (for some small number m) in random order, as well as an additional word (called the *intruder*) randomly placed among the topic keys [Chang et al., 2009]. The intruder is randomly selected from the set of high-probability words from other topics in the model. The expert is tasked with identifying the intruder in each list of $m+1$ words. As in prior work [Chang et al., 2009], I instruct the expert to ignore syntactic and morphological patterns.

If the model is interpretable, the topic keys will be internally coherent whereas the intruder word is likely to stand out. Thus a model's interpretability can be quantified by the fraction of topics for which the expert correctly identifies the intruder. I call this value the *mean model precision (MMP)*:

$$\text{MMP} = \frac{1}{K} \sum_{k=1}^{K} [i_k = \omega_k]$$

where K is the number of topics in the model, i_k is the index of the intruder in the randomized word list generated from topic k, and ω_k is the index of the word the expert identified as the intruder. This is just the mean (over topics) of the *model precision* metric from prior work [Chang et al., 2009] in the special case where there is only one expert (and the model is left implicit).

I use the lemmatized and unlemmatized treatments of a Russian Wikipedia corpus, as described in May et al. [2016], as my data set. I consider two additional preprocessing schemes to account for stop words, which also play a role in interpretability. First, I compute the vocabulary as the top 10 000 words by document frequency,[3] separately for the lemmatized and unlemmatized data, and specify an asymmetric prior on each document d's topic proportions $\boldsymbol{\theta}^{(d)}$. I refer to this preprocessing scheme as the *unfiltered-asymmetric* approach. The second preprocessing scheme I consider uses a vocabulary with high-frequency words filtered out and a uniform prior on the document-wise topic proportions. I refer to this approach as *filtered-symmetric*. Specifically, a 10 000 word vocabulary is formed from the lemmatized data by removing the top 100 words by document frequency over the corpus and taking the next 10 000. To determine the unlemmatized vocabulary, I map the filtered lemmatized vocabulary onto all word forms that produce one of those lemmas in the data. Finally, observing that some of the uninformative high-frequency words reappear in this projection, I remove any of the top 100 words from the lemmatized and unlemmatized corpora from this list, producing a unlemmatized vocabulary of 72 641 words. While the large size of this vocabulary slows learning, I do not believe it impacts the results negatively; my priority is retaining the information captured by the lemmatized vocabulary to provide a fair comparison.

In addition to exploring different choices of vocabulary, I also consider truncating

[3]Due to implementation concerns, the lemmatized and unlemmatized vocabularies consist of the top 9387 and 9531 words (respectively) by document frequency.

the documents to their first 50 tokens.[4] This augmentation simulates data sparsity by reducing the amount of content-bearing signal in each document, so I might expect the truncated documents to more greatly benefit from lemmatization (which can be cast as a dimensionality reduction method).

I train LDA by stochastic variational inference [Hoffman et al., 2013], initializing the models randomly and using fixed priors.[5] I specify $K = 100$ topics to all models. Uniform priors with $\eta_v = 0.1$ and $\alpha_k = 5/K$ were given to filtered-symmetric models; non-uniform priors with $\eta_v = 0.1$, $\alpha_1 = 5$, and $\alpha_k = 5/(K-1)$ for $k > 1$ were given to unfiltered-asymmetric models. The local hyperparameters $\boldsymbol{\alpha}$ are informed by mean document word usage and document length; in particular, I believe approximately 50% of the word tokens in the corpus are uninformative.

The mean model precision for all four configurations (filtered-symmetric or unfiltered-asymmetric vocabulary and full-length or truncated documents), and the p-values for one-sided MMP differences (testing my hypothesis that the lemmatized models yield higher MMP than the unlemmatized models), are reported in Table 5.2. Word intrusion performance benefits significantly from lemmatization on a filtered vocabulary and a symmetric prior. Truncated documents exhibit lower performance overall and are helped less by lemmatization. Further, I observe differences between use of an asymmetric prior on an unfiltered vocabulary and use of a symmetric prior on a

[4] As the vocabulary does not contain rare words, the number of tokens per document seen by the model is less than 50.

[5] In preliminary experiments, Gibbs sampling with hyper-parameter optimization did not improve interpretability.

vocabulary	prior	documents	MMP		p-value
			unlemmatized	lemmatized	
unfiltered	symmetric	full	0.54	0.52	0.61
filtered	**asymmetric**	**full**	**0.50**	**0.65**	**0.02**
unfiltered	symmetric	truncated	0.37	0.37	0.50
filtered	asymmetric	truncated	0.43	0.47	0.28

Table 5.2: Mean model precision for the unlemmatized and lemmatized models and p-values for the one-sided MMP difference tests. The MMP when using a filtered vocabulary coupled with an asymmetric prior on full-length documents benefits significantly from lemmatization (row highlighted in bold).

vocabulary with stop words filtered out.

As observed in May et al. [2016], stop words were highly ranked in many topics from the unfiltered-asymmetric models despite the asymmetric priors encouraging frequent words into each model's first topic. Additionally, a manual alignment of topics from the lemmatized and unlemmatized filtered-symmetric models 5.1 (trained on full documents) revealed that the keys of many unlemmatized topics contained redundant word forms corresponding to a single lemma in the topic keys of a similar lemmatized topic [May et al., 2016]. Thus, in many cases, the lemmatized model lends more easily to human interpretation.

5.5 Automatic Evaluation

In the previous section, I studied the interpretability of topic models on Russian text using a human-in-the-loop evaluation process. In this section, I study topic models on several different languages using an automated evaluation process. While

the automated evaluation is more cost-effective, I expect it exhibits a lower correlation with most notions of *interpretability* than the manual evaluation.

In this study, in addition to using different evaluation metrics, I use different data sets and learning algorithms. In this study, I use the preprocessed Wikipedia data sets prepared by Al-Rfou et al. [2013] largely out of practical concerns, as I no longer have access to the Russian data set used in the previous study, and preprocessing Wikipedia data sets in other languages would add substantially to the overall computational cost. I use the Gibbs sampling LDA training procedure implemented in MALLET [McCallum, 2002] partly for practical concerns (as it is more readily accessible) but largely out of hindsight: While I expect the hyperparameter optimization provided by MALLET to filter stop words at least as well as the fixed asymmetric priors considered in the prior study, MALLET is also far more accessible to other researchers and practitioners and better represents the *practice* of topic modeling.[6]

Specifically, I perform an evaluation of LDA topic models trained on the preprocessed English, Farsi (Persian), Korean, and Russian Wikipedia data subsets of the Polyglot multilingual Wikipedia data set [Al-Rfou et al., 2013].[7] These data sets vary in size by over an order of magnitude, so before performing any other processing, I subsample all data sets to 200 000 documents. I consider three different lemmatization treatments applied to each data set: No lemmatization treatment, lemmatization

[6]MALLET is accessible online as a compiled Java program or source code at `https://mimno.github.io/Mallet/index`. Additionally, the MALLET GitHub repository (`https://github.com/mimno/Mallet`) has over 30 contributors and 300 forks at the time of writing.

[7]The Polyglot data set is accessible online at `https://sites.google.com/site/rmyeid/projects/polyglot`.

using TreeTagger [Schmid, 1994],[8] and lemmatization using UDPipe models trained on Universal Dependencies 2.5 treebanks [Straka and Straková, 2017]. I then lowercase each treatment of each data set and train ten topic models (using ten different random seeds) on each one.[9] I evaluate each data set using the variation of information (VOI) and modified topic coherence procedures of Schofield and Mimno [2016],[10] using $m = 5$ topic keys for topic coherence and filtering the top 200 words in each corpus (by document frequency) out of the untreated topic keys as stop words.

5.5.1 Topic Coherence

Recall that topic coherence measures the degree to which a topic's keys co-occur in the corpus (occur in the same documents) [Mimno et al., 2011]. If $D(w, w')$ is the number of documents in which words w and w' co-occur, $D(w)$ is the number of documents in which w occurs (its document frequency), and $v_i^{(k)}$ is the i-th topic key (the i-th most probable word) of topic k, then define the topic coherence $TC'(k)$ of topic k as

$$TC'(k) = \sum_{i=2}^{m} \sum_{j=1}^{i-1} \log \frac{D\left(v_i^{(k)}, v_j^{(k)}\right) + \beta}{D\left(v_j^{(k)}\right) + \beta}$$

[8] TreeTagger and the requisite parameter files for each language are currently available online from `https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/`.

[9] UDPipe and the requisite parameter files for each language are currently available online from `https://ufal.mff.cuni.cz/udpipe/1`.

[10] I set the modified topic coherence smoothing parameter β to the optimized value of the Dirichlet hyperparameter β of the corresponding topic model.

where m is the number of keys used to represent each topic. This definition differs slightly from that of Mimno et al. [2011]: For my experiments, following Schofield and Mimno [2016], I smooth the counts by a parameter β to imitate the smoothing induced by the Dirichlet priors in LDA.

To control for differences in vocabulary size between different treatments, I use the modified version of topic coherence proposed by Schofield and Mimno [2016]. In addition to smoothing with a parameter β, in the modified formulation of topic coherence, topic keys are computed by taking the token assignments in the model to be evaluated, replacing the word at each token location by the corresponding word in the untreated corpus,[11] and computing the most frequent words in the resultant list. Accordingly, the modified topic coherence is computed using the document frequencies and co-document frequencies from the untreated corpus. This procedure allows us to compare the coherence of a topic model trained on a lemmatized version of a corpus to one trained on the untreated corpus without favoring the lemmatized model *a priori* just because its data set exhibits less sparsity.

Both the modified and unmodified versions of topic coherence only represent the coherence of a single topic. I compute a model's overall topic coherence by taking the mean of the topic coherence scores for its individual topics.

Figure 5.1 shows the distribution of the (modified) negative topic coherence for each treatment on each language corpus. Overall, lemmatization treatments increased

[11] This operation requires that lemmatization maps each word to exactly one lemma. In particular, a lemmatizater cannot map a word form to multiple "lemmas."

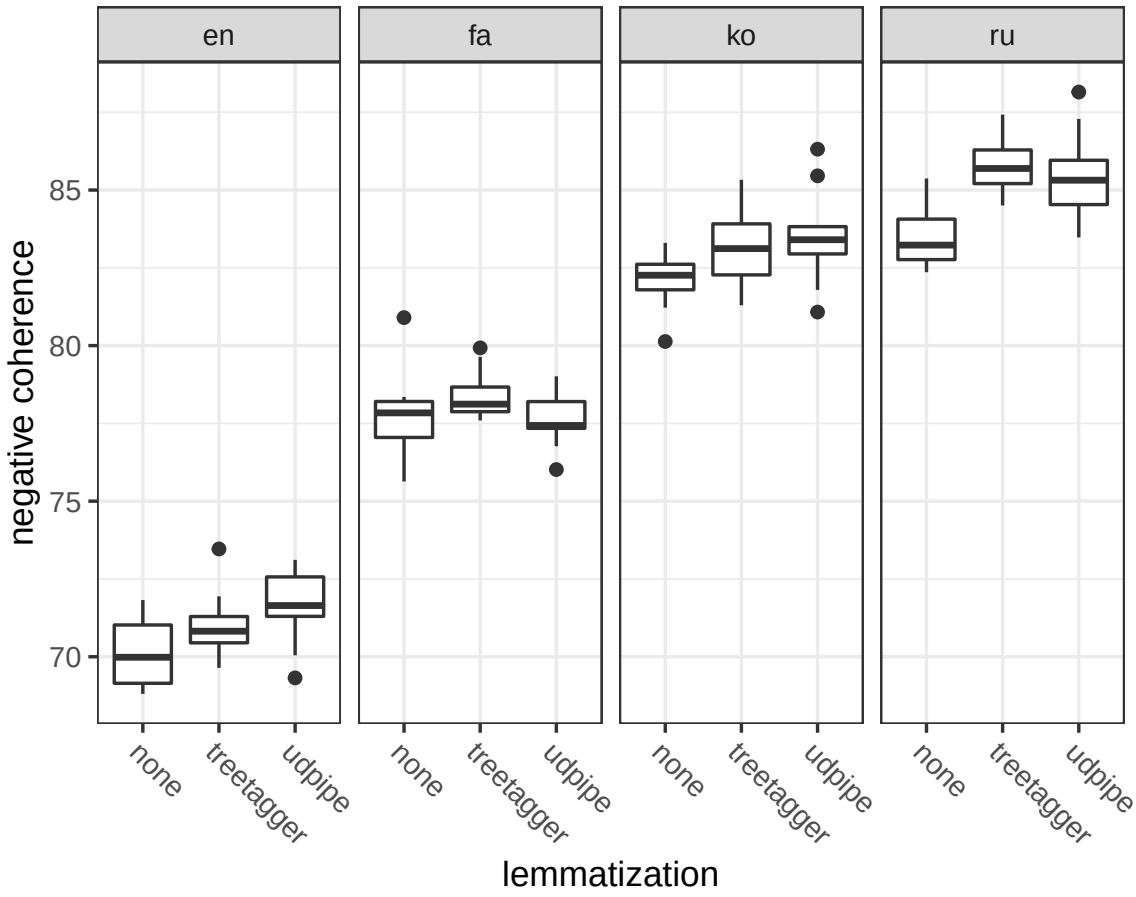


Figure 5.1: Distribution of modified negative topic coherence for topic models trained on each treatment of each corpus. Lower is better.

negative topic coherence, meaning they made the topics less coherent. However, UDPipe on Farsi and Korean yielded small (and likely insignificant) increases in median coherence. Lemmatization with TreeTagger had a detrimental effect on Farsi and the largest detrimental effect on Russian, while UDPipe had the largest detrimental effect on English and Korean.

Unlike the human evaluation, this evaluation controls for the vocabulary size. Thus, while TreeTagger appears to lower the topic quality according to these results and raise the topic quality according to the human evaluation results, this discrepancy might be explained by the different vocabulary sizes between treatments in the human evaluation.

To investigate this hypothesis, I first compute the ratio of word types to word tokens for each treatment of each corpus to measure the reduction in vocabulary effected by each lemmatizer. These ratios are depicted in Figure 5.2. On all languages but English, UDPipe yields a larger reduction in vocabulary than TreeTagger. The differences between ratios are especially stark in Korean, on which UDPipe almost halves the vocabulary.

I also compute the unmodified topic coherence, which does not control for vocabulary size. The distributions of unmodified negative topic coherence for each treatment of each language data set are shown in Figure 5.3. Unmodified topic coherence is improved by both lemmatization treatments on all languages. The improvements provided by TreeTagger and UDPipe on Korean and Russian are particularly stark.

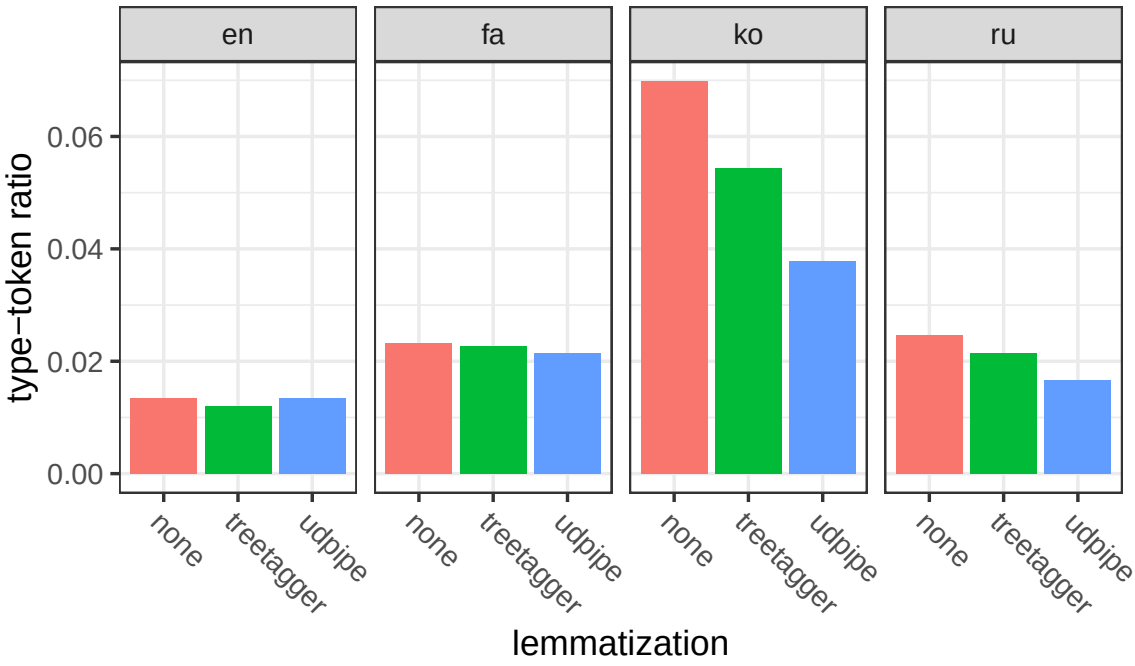


Figure 5.2: Type-token (word type/word token) ratio for each treatment of each corpus. Because both lemmatizers map each unlemmatized token to exactly one lemmatized token, the ratios for each language are directly proportional to the vocabulary sizes of that language's untreated and treated corpora.

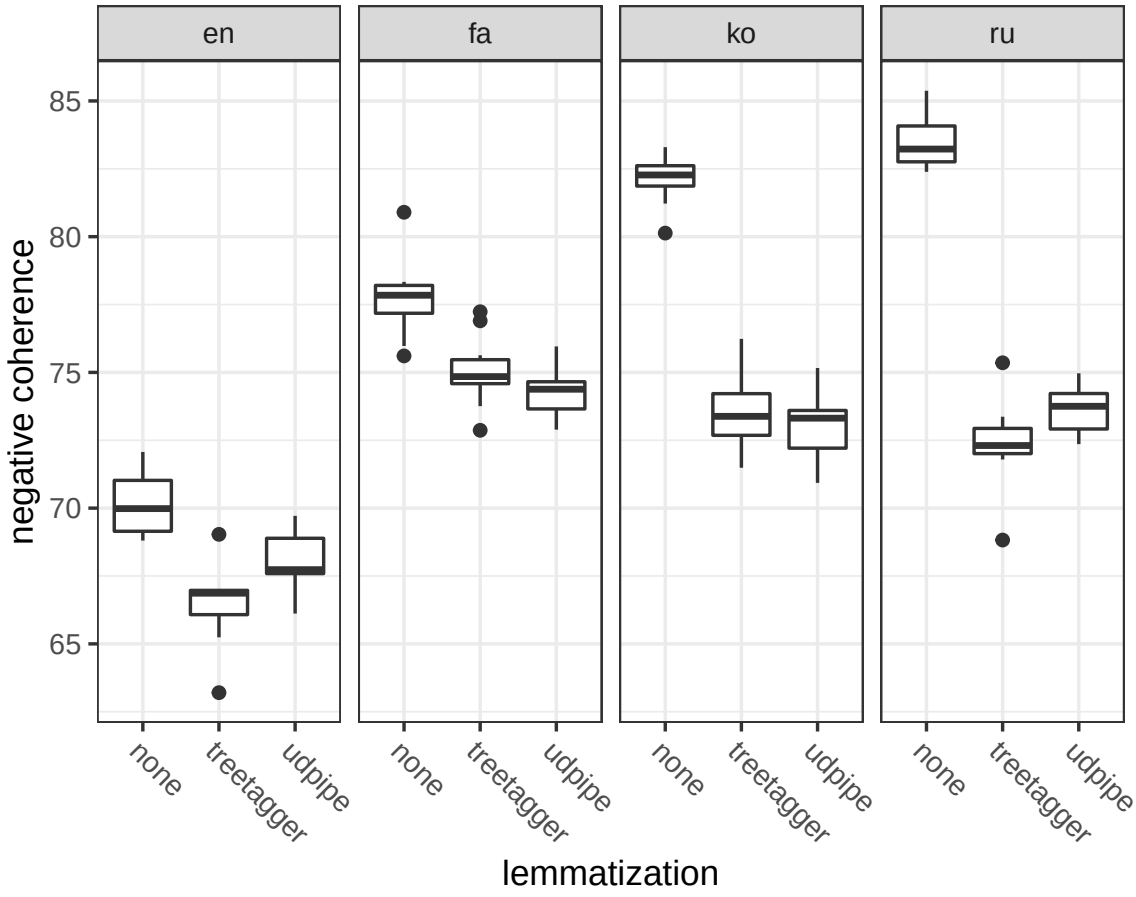


Figure 5.3: Distribution of unmodified negative topic coherence for topic models trained on each treatment of each corpus. This variant of topic coherence does not control for differing vocabulary sizes between treatments. Lower is better.

Thus, the improvement in topic quality indicated under the human evaluation might be explained largely by the reduction in vocabulary size as observed through the topic keys, and may not represent an improvement in the underlying fit of the topic model. However, recall that I use a different data set in this evaluation than in the human evaluation, likely reflecting a different subset of Wikipedia in a different year preprocessed with a different text normalization procedure. Hence, comparison of the modified and unmodified variants of topic coherence affords only suggestive evidence for explaining the discrepancy between the human and automated evaluations.

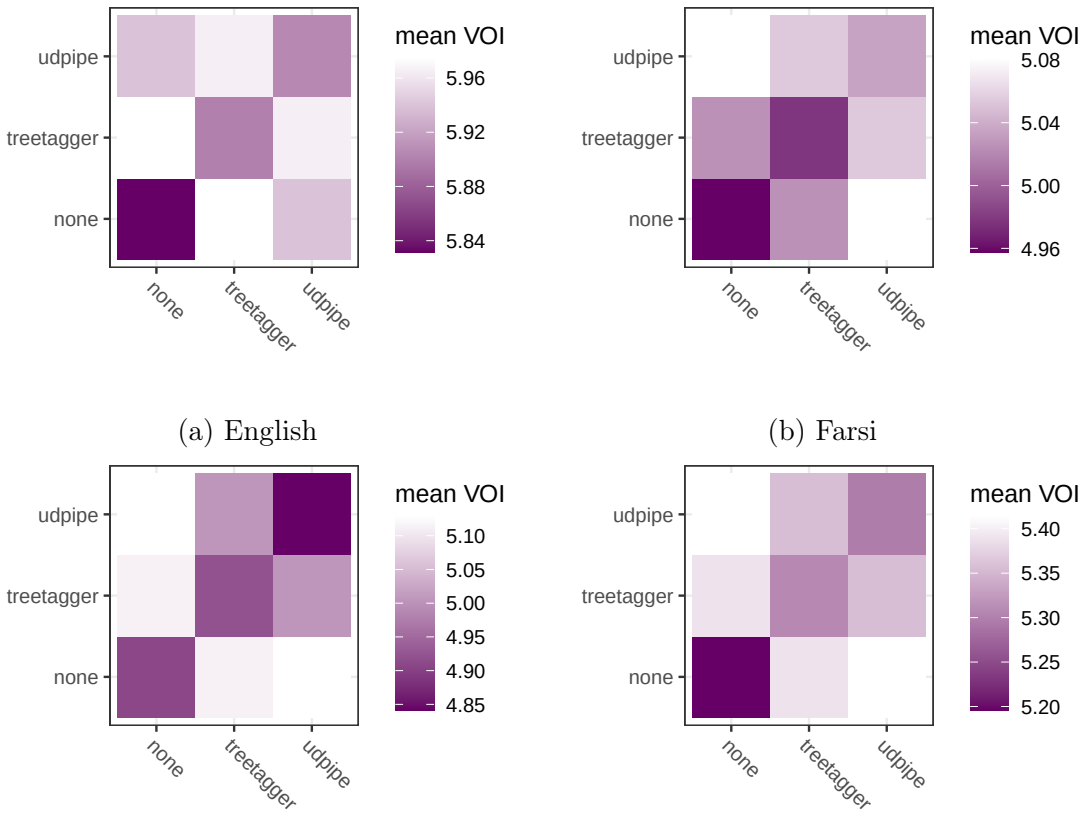
5.5.2 Variation of Information

Recall that the variation of information (VOI) is an information theoretic metric of how much information is lost when moving between two clusterings C_1 and C_2. VOI is defined [Meilă, 2007] as

$$VI(C_1, C_2) = H(C_1) + H(C_2) - 2I(C_1, C_2)$$

or equivalently

$$VI(C_1, C_2) = H(C_1|C_2) + H(C_2|C_1).$$



(a) English

(b) Farsi

(c) Korean

(d) Russian

Figure 5.4: Mean variation of information (VOI) between topic models trained on each treatment of each corpus. The diagonal in each plot shows intra-treatment VOI and the off-diagonal shows inter-treatment VOI. Darker (lower VOI) is better.

Following Schofield and Mimno [2016], I apply the variation of information to two topic models by taking the tokens in a corpus as the data points $\mathcal{X}$ and the token assignments under the two models as the clusterings $\mathcal{C}_1$ and $\mathcal{C}_2$ of those tokens.[12]

Figure 5.4 shows the mean variation of information for each setting. The tiles on the diagonals depict mean intra-treatment VOI (VOI between topic models trained

[12]Within a treatment (intra-treatment), I compute VOI for all two-sets of trials (distinct unordered pairs of distinct trials), yielding $\binom{10}{2} = 45$ samples. Across treatments (inter-treatment), I compute VOI for all pairs of trials, yielding $10^2 = 100$ samples.

on the same data with different random seeds) while the tiles off the diagonals depict mean inter-treatment VOI (VOI between topic models trained on different treatments of the data and different random seeds). The mean intra-treatment VOI for each language and treatment is less than the mean inter-treatment VOI with each of the other two treatments at a significance level of $p < 10^{-5}$ (using a two sample t-test) in all settings except TreeTagger on Farsi, for which $p < 10^{-2}$.

On Korean, the mean VOI between the TreeTagger and UDPipe treatments is significantly less than that between TreeTagger and the untreated corpus ($p < 10^{-15}$) and that between UDPipe and the untreated corpus ($p < 10^{-15}$), suggesting that TreeTagger and UDPipe lemmatization treatments yield topic models more similar to each other than to the unlemmatized corpus. Hence, TreeTagger and UDPipe may have similar effects on a corpus on Korean. On Russian, the mean VOI between the TreeTagger and UDPipe treatments is significantly less than that between TreeTagger and the untreated corpus ($p < 10^{-15}$) and that between UDPipe and the untreated corpus ($p < 10^{-15}$), so the same pattern holds. English and Farsi do not exhibit this pattern. In fact, on English, the mean VOI between TreeTagger and UDPipe is significantly *greater* than that between UDPipe and the untreated corpus ($p < 10^{-6}$), and on Farsi, the mean VOI between TreeTagger and UDPipe is significantly greater than that between TreeTagger and the untreated corpus ($p < 10^{-4}$).

On English and Russian, the mean intra-treatment VOI is significantly greater for both TreeTagger and UDPipe than for the unlemmatized corpus ($p < 10^{-15}$).

On Farsi, mean intra-treatment VOI is somewhat significantly greater for TreeTagger than for the unlemmatized corpus ($p < 0.02$ and significantly greater for UDPipe than for the unlemmatized corpus ($p < 10^{-11}$). On Korean, mean intra-treatment VOI is somewhat significantly greater for TreeTagger than for the unlemmatized corpus ($p < 0.04$). These results indicate that both lemmatization treatments tend to detract from the stability of the topic model, with the notable exception of UDPipe on Korean, which significantly increases stability (mean intra-treatment VOI is significantly less for UDPipe than for the unlemmatized corpus with $p < 10^{-15}$).

Finally, the mean intra-treatment VOI on Farsi is significantly greater for UDPipe than it is for TreeTagger ($p < 10^{-6}$), indicating that UDPipe detracts from topic model stability on Farsi significantly more than TreeTagger does.

5.5.3 Automatic Translation

To supplement the automated quantitative analysis of the models, I also perform a small, partially automated qualitative analysis using machine translation. Specifically, I use the Translator Dictionary Lookup method from Microsoft Azure Cognitive Services to translate the topic keys of each model to English. In contrast with other cloud machine translation interfaces, Translator Dictionary Lookup provides translations of individual words independent of context, allowing translation of topic keys without requiring potentially confounding heuristics to ground them in appropriate contexts. The translated topic keys are then filtered by a set of stop words developed

for the English corpora by computing the top 200 words in a corpus by document frequency.

However, while filtering topic keys by these stop words produces satisfactory keys for the English models, this procedure leaves a significant amount of punctuation and uninformative words in the translated results from other languages. For example, the translated word "pm" appeared at or near the top of almost every topic on Russian. Novel word forms containing lemmatizer-specific syntax, specifically lemmatized words containing # in Farsi and lemmatized words containing + in Korean, also present an obstacle to automatic translation (hence interpretation). Those machine-readable "lemmas" may be difficult to interpret even in their untranslated forms by native speakers. Thus, to facilitate interpretation of the translated topic keys, I also filter out words with three characters or less and words containing characters outside of the Latin alphabet.

In the following results, I pick five topics uniformly at random from a TreeTagger-lemmatized model and show their topic keys alongside manually aligned topics from the respective UDPipe-lemmatized and unlemmatized models.[13] Manual alignment of topics was performed to approximately control for the effect of interpretability varying between topics. Thus, while I report only five keys per topic for the purpose

[13]Specifically, I take the first five topics of the first trial of the TreeTagger-lemmatized model and align them with topics from the first trial of the UDPipe-lemmatized model and the first trial of the unlemmatized model. Because of symmetry, this is virtually equivalent to selecting a TreeTagger trial uniformly at random and then selecting five topics uniformly at random from that model. Although topic model initialization and other implementation choices may break symmetry, hence biasing the results of this sampling procedure, I did not see evidence of bias when inspecting the models and aligning their topics.

model	topic keys
treetagger	german germany berlin dutch austria
udpipe	german germany berlin dutch swedish
none	german germany berlin dutch netherlands
treetagger	system* signal light power device
udpipe	power energy system* signal current
none	energy light surface system* field
none	system* digital systems* technology computer
treetagger	virginia florida carolina ohio illinois
udpipe	york virginia jersey washington carolina
none	york virginia jersey washington pennsylvania
treetagger	match cricket test score england
udpipe	australia australian zealand cricket match
none	match cricket wrestling championship test
treetagger	company* business product sell market
udpipe	company* business product sell market
none	company* business companies* products founded

Table 5.3: Five random topics from a TreeTagger-lemmatized model on English, manually aligned with topics from models subject to other lemmatization treatments. Words that appear in multiple forms in the unlemmatized model are annotated with the symbol * and color-coded. The second set of topics includes two topics from the unlemmatized model ("none") because the content of the topics in the lemmatized models appeared to span multiple topics in the unlemmatized model.

of judging interpretability, I allowed a larger set of topic keys when aligning topics: One hundred topic keys were output for each topic, translated, and filtered, and I used the resultant lists to determine whether two topics aligned. This greater allowance of information during alignment enables better approximation of an "oracle" topic alignment, if one exists.

The (untranslated) topic keys of randomly sampled topics from the English models are presented in Table 5.3. While most of the topics appear to be equally informative in all three models, in two cases, the unlemmatized model's topic keys contain multiple

model	topic keys
treetagger	party sccr election boss representative
udpipe	iran islamic election republic revolution
none	iran islamic revolution republic tehran
treetagger	village* water dara agricultural abad
udpipe	village* water dara agricultural mountain
none	village* village* water rated agricultural
treetagger	disease treatment patient person blood
udpipe	disease treatment medication patient person
none	disease treatment consumption blood body
treetagger	game china japan japanese chinese
udpipe	game series company version published
udpipe	china country province population source
none	game series company published harry
none	japan station japanese korea tokyo
treetagger	iran tehran iranian volume activity
udpipe	iran tehran seyed mirza shah
none	iran islamic revolution republic tehran

Table 5.4: Five random topics from a TreeTagger-lemmatized model on Farsi, manually aligned with topics from models subject to other lemmatization treatments. Words that appear in multiple forms in the unlemmatized model are annotated with the symbol * and color-coded. The fourth set of topics includes two topics each from the UDPipe-lemmatized ("udpipe") and unlemmatized ("none") model because the content of the topics in the TreeTagger-lemmatized model appeared to span multiple topics in the other models.

forms of a lemma. However, in my judgment, these topics are still highly interpretable, so lemmatization would not seem to have a substantial effect overall.

The translated topic keys of randomly sampled topics from the Farsi models are presented in Table 5.4. While most of the topics appear to be equally informative in all three models, in one case, the unlemmatized model's topic keys contain multiple forms of a lemma. Because Farsi exhibits greater morphological variation than English, the two word forms in Farsi translate to the same word form in English. However, in

model	topic keys
treetagger	having routes seoul sectors current
udpipe	having routes railway train sectors
none	station subway platform guide railway
treetagger	having person yourself days week
udpipe	having yourself person days pick
none	–
treetagger	japan emperor* meiji cabinet kyoto
udpipe	japan tokyo prefecture having upon
udpipe	castle times emperor* shogunate having
none	emperor* king china japan emperor*
treetagger	mind select having mean buddhist
udpipe	select having mind mean status
none	select mind says should buddhist
treetagger	language character english* english* kanji
udpipe	language character pronunciations notation word
none	character language english* english* hangul

Table 5.5: Five random topics from a TreeTagger-lemmatized model on Korean, manually aligned with topics from models subject to other lemmatization treatments. Words that appear in multiple forms in the unlemmatized model are annotated with the symbol * and color-coded. The second set of topics does not contain an unlemmatized topic because one could not be found that aligned with the TreeTagger-lemmatized model's topic. The third set of topics contains two UDPipe-lemmatized topics because the content of the TreeTagger-lemmatized model appeared to span multiple topics in that model.

my judgment, these topics are still highly interpretable, so lemmatization would not seem to have a substantial effect overall.

The translated topic keys of randomly sampled topics from the Korean models are presented in Table 5.5. While many of the topics appear similarly informative in all three models, the unlemmatized topic in the first set is more interpretable than the other two, and the UDPipe-lemmatized topic is slightly more interpretable than the TreeTagger-lemmatized topic. The lemmatized topics in the second set largely

evade interpretation and do not appear to have an analogue in the unlemmatized model. Additionally, the relatively uninformative word "having" appears in three out of five of the TreeTagger-lemmatized topics' keys and five out of six of the UDPipe-lemmatized topics' keys and is not prevalent in the unlemmatized model's topic keys. In two cases, the unlemmatized model's topic keys contain multiple forms of a lemma. Because Korean exhibits greater morphological variation than English, the two word forms in a topic in Korean translate to the same word form in English. However, in my judgment, this redundancy does not impair interpretability, so lemmatization would not seem to have a substantial effect overall.

The translated topic keys of randomly sampled topics from the Russian models are presented in Table 5.6. While many of the topics appear to be equally informative in all three models, the TreeTagger-lemmatized topic in the first set is not as interpretable as the other two topics, and the second TreeTagger-lemmatized topic and the unlemmatized topic in the second set have lower interpretability as well. In two cases, the unlemmatized model's topic keys contain multiple forms of a lemma. Because Russian exhibits greater morphological variation than English, some of those word forms in Korean translate to the same word form in English. In my judgment, this redundancy does not impair interpretability, so lemmatization would not seem to have a substantial effect overall. However, these topic sets exhibit more redundancy in word forms in the unlemmatized model than those for the other languages studied; based on these results only, we might expect lemmatization to have a greater (al-

model	topic keys
treetagger	star mark save thumb edition
udpipe	button book* writer magazine novel
none	books* book* novel author button
treetagger	community kazakhstan bulgaria enter elections
udpipe	party political government president minister
udpipe	save surname watch community enter
none	star bulgaria community enters results
treetagger	device signal system frequency digital
udpipe	system energy signal field current
none	prior image signal device communication
treetagger	linear peak kitt spacewatch mart
udpipe	linear peak kitt spacewatch mart
none	linear peak kitt spacewatch martha
treetagger	church* bishop diocese orthodox holy
udpipe	church* holy monastery† diocese bishop
none	church* church* monastery† orthodox monastery†

Table 5.6: Five random topics from a TreeTagger-lemmatized model on Russian, manually aligned with topics from models subject to other lemmatization treatments. Words that appear in multiple forms in the unlemmatized model are annotated with the symbols * or † and color-coded. The second set of topics includes two topics from the UDPipe-lemmatized ("udpipe") model because the content of the topics in the TreeTagger-lemmatized model appeared to span multiple topics in the UDPipe-lemmatized model. The fourth set of topics is conspicuous for its virtually perfect alignment between the topic keys. These topics pertain to U.S. near-earth object discovery programs such as LINEAR at MIT Lincoln Laboratory and Spacewatch at Kitt Peak National Observatory. Based on inspection of the data set, I hypothesize that the stability of this topic across models results from the existence of long lists of near-earth objects and the programs that discovered them, which gives rise to large word frequencies that directly reflect ratios in the real world (specifically, ratios between the numbers of near-earth objects discovered by each program) in a subset of the documents.

beit still somewhat small) effect on interpretability in Russian than it does in other languages.

5.6 Discussion

In a preliminary study, I used a human-in-the-loop evaluation procedure to measure the effect of pre-processing by lemmatization on the interpretability of topic models on Russian language. The results of this study suggest that topic models on Russian text benefit from lemmatization in some cases, contrasting a related study that found topic models on English text did not benefit from lemmatization.

In a larger follow-up study, I used automated evaluation metrics to study the effect of pre-processing by lemmatization on the interpretability of topic models in several additional languages and one additional lemmatizer. First, I used a modified topic coherence metric to assess model fit while controlling for the size of the vocabulary. Since any non-trivial lemmatization reduces the size of the vocabulary, many metrics that are used to assess model fit, like the original unmodified version of topic coherence, are biased toward models on lemmatized data simply because their dimensionality is reduced. The modified version of topic coherence controls for that dimensionality reduction. And the results using that version of the metric indicate that the lemmatization treatments I considered actually *reduced* (worsened) model fit to some extent. That is, after controlling for vocabulary size, LDA models fit un-

lemmatized data better than lemmatized data in the cases I studied. This finding is surprising given my claim that inflectional morphological variation has little impact on the underlying concept represented by a word. In hindsight, we might hypothesize that that variation correlates with variation in word sense.

I next applied the original, unmodified topic coherence metric to test whether the differences in vocabulary size could help explain why the human-in-the-loop evaluation suggested that lemmatization benefited interpretability while the automated evaluation using topic coherence suggested that lemmatization harmed the fit of the model (hence, I expect, interpretability). In this case, the lemmatization treatments noticeably improved the model fit (hence, I expect, interpretability), supporting the hypothesis that reduction in vocabulary largely explains any improvement in interpretability when using lemmatized data.

Perhaps the most readily generalizable results of the automated evaluation are the vocabulary reduction and intra-treatment variation of information statistics. A smaller vocabulary allows more iterations of the training algorithm given the same computational resources and reduces the degree of data sparsity that the algorithm must overcome. The lemmatizers provide a much more noticeable vocabulary reduction in Korean (and, to a lesser degree, Russian) than in English or Farsi, so all else equal, we might expect lemmatization to provide a more noticeable impact on topic model interpretability on Korean (and Russian). As for the other metric, UDPipe yields lower intra-treatment variation of information than either TreeTagger or no

lemmatization, so we can expect the *reliability* of a topic model on Korean to improve when the data is lemmatized by UDPipe. While both lemmatizers increase intra-treatment variation of information (over the no-lemmatization treatment) on the other three languages, TreeTagger yields a relatively small increase on Farsi, so we might expect the reliability of a topic model on Farsi to be somewhat higher when preprocessed with TreeTagger compared to UDPipe.

While the automated evaluation does not require human judges who can read the languages under study, that decrease in cost and increase in reliability come at the expense of *construct validity*. That is, while this automated evaluation protocol eliminates the cost of soliciting "interpretability" judgments from people who can read the languages in question,[14] I argue it produces measurements that are less representative of the construct of "interpretability." For example, while the unmodified topic coherence statistics suggest that lemmatization does significantly improve interpretability across languages, that metric only tells us about distributions of words in the corpus and topic model, not their contents, and the words that represent each topic may still be difficult to interpret. Moreover, while automated translation enables a glimpse of the contents of those words, English exhibits less morphological variation

[14]I do not believe this protocol ended up reducing the total number of person-hours expended in my study, nor do I believe it would provide a reduction in most related studies; designing and implementing the automated evaluation protocol took considerable time on my part, and I suspect that experience is typical. I also do not believe that developing unifying software frameworks for NLP obviates this problem, at least not in the general case. Research is evaluated on its novelty, among other aspects, and while the availability and accessibility of software automating individual steps of the research process continues to improve, the requirement of novelty effectively prevents the use of more holistic NLP software frameworks directly "off the shelf" in my experience.

than many other languages, so some amount of information about the topic keys is inevitably lost in translation; additionally, translation fails on a large number of topic keys, and words that confound the translation algorithm are plausibly less likely to be interpretable. That is, filtering out words that were not translated into the Latin alphabet may unintentionally also filter out many words that are not interpretable by themselves.

In any case, to reduce the amount of language-dependent processing needed for effective topic modeling, we might consider applying topic models to language units other than words. In recent years, learned subword representations like byte pair encoding (BPE) have become popular in many NLP tasks. Unlike lemmatizers, these representations are computed directly from corpus statistics and do not require further language-specific resources. While topic modeling on BPE would produce relatively uninterpretable topic keys using standard methods, I argue we should also consider revisiting the construction of these keys (or the design of human–topic-model interfaces altogether). By sampling words or phrases containing those keys from the corpus, we might be able to regain interpretability in the topic representations. Additionally, a tremendous amount of research has extended LDA and proposed alternative topic models and algorithms, but many approaches still culminate in simple top-m word lists. Usage of an alternative representation that balances the estimated probability of each word with the diversity of words presented, like FREX [Bischof and Airoldi, 2012], might improve topic modeling more than improvements in the underlying model

or learning algorithm.

5.7 Conclusion

If nothing else, the results of both evaluation protocols suggest that preprocessing Russian language data by lemmatization yields a modest improvement in topic model interpretability, while preprocessing English or Farsi by similar lemmatization technology yields less improvement, if any at all. Thus, while topic models are often framed as relatively language-independent techniques *in theory*, I have found evidence that effective topic modeling benefits from a language-dependent approach in practice.

Chapter 6

Hierarchical Modeling for Stop Word Filtering

6.1 Preface

This chapter began with an attempt to re-implement the algorithm and reproduce the results of a recent (at the time) topic modeling paper [Paisley et al., 2012]. Upon reading that paper, I became excited about the potential of its model not only to induce topic hierarchies but also to filter out stop words automatically.[1] In the following, I describe an investigation emanating from that reproduction attempt.

[1]Not finding a public software implementation linked in the manuscript or appearing in a cursory web search, I began re-implementing the algorithm. I found, many months later, that the first author had released code implementing the algorithm on his website. I also designed my implementation for scalability and generality, starry eyed over potential applications—applications that were never able to materialize. I hope future researchers will approach their work more cautiously in order to better avoid such lost investments.

While the foundational LDA topic model learns an unstructured collection of topics to describe a corpus, some more recent models, like the nested hierarchical Dirichlet process (nHDP), learn topic hierarchies [Paisley et al., 2015]. Because the nHDP's hierarchy has an optional root node (topic), we might expect that the root topic will automatically collect stop words, obviating the need for removing stop words as a preprocessing or postprocessing step. Indeed, in a footnote, the researchers who developed the nHDP wrote that the hierarchy "includes a root node topic, which is shared by all documents and is intended to collect stop words" [Paisley et al., 2015]. However, in that study, stop-word filtering was used for all models that were studied qualitatively or presented as examples using the topic keys. Nonetheless, I argue that an optimistic reader would expect an active root node to filter out stop words:

Theory. *The nested hierarchical Dirichlet process obviates the need for stop-word filtering.*

However, while I was able to approximately reproduce the results of Paisley et al. [2015], I found that an active root node did not prevent stop words from proliferating the descendant topics.

In this chapter, I make the following contributions to science:

- I have performed a focused analysis of the stop word filtering behavior of the nHDP, complementing the original study.
- I have produced a detailed analysis of the software implementation of the nHDP

provided by Paisley et al., comparing and contrasting it with the published algorithm.

- I have publicly released my nHDP training and experiment code for the benefit of future research.[2] This code is closely based on Paisley et al.'s code, amending its discrepancies with the algorithm from the paper.

Although it is impossible to prove that the nHDP cannot be configured to automatically filter stop words (as the number of possible configurations is infinite), the negative results of this study further support my thesis that theories of topic models suggested in the literature do not generally hold in practice. In particular, the nHDP does not filter stop words in practice with the level of ease the literature suggests.

6.2 Background

After the success of LDA, there were a number of extensions of the model and basic learning algorithms to new contexts. Informally, there seemed to be a common belief—a theory—that any problem could be solved (within reason) with the right generative story. For example, LDA requires the number of topics to be specified before learning, but what if we don't have a good idea of how many topics there should be? What if we want both coarse-grained *and* fine-grained representations of the data?

[2] `https://github.com/ccmaymay/nhdp`

One approach taken to address those issues was Bayesian nonparametric modeling. Bayesian nonparametric models such as the hierarchical Dirichlet process (HDP) topic model [Teh et al., 2006], sparse topic model [Wang and Blei, 2009], and focused topic model [Williamson et al., 2010] allow the number of topics to be inferred from the data, while models like nonparametric Bayes Pachinko allocation [Li et al., 2007], the nested Chinese restaurant process (nCRP) topic model [Blei et al., 2010], and the nested hierarchical Dirichlet process (nHDP) topic model [Paisley et al., 2015] further allow a hierarchical topic *structure* to be inferred. In this chapter, I study the latter model, the nHDP, which relaxes the assumption of prior knowledge of the number of topics and topic structure. I provide a brief description of the nHDP and the HDP on which it is based; for a more thorough and accessible introduction to Bayesian nonparametric models, see Gershman and Blei [2012].

The hierarchical Dirichlet process (HDP) topic model represents a corpus with a theoretically unbounded set of topics, and a finite topic representation can then be inferred from that representation as needed [Teh et al., 2006]. Specifically, the HDP "explains" the occurrence of each word by postulating that a document was generated by repeatedly: (1) Sampling a topic parameter vector $\boldsymbol{\phi}^{(i)}$ from a document-specific mixture over topics, G, and (2) sampling a word w_i from a topic-specific probability distribution over a vocabulary of W words, Categorical($\boldsymbol{\phi}^{(i)}$). The document-wise topic distribution G is modeled as a Dirichlet process over a global topic distribution G_0, which is in turn modeled as a Dirichlet process over a prior Dirichlet distribution

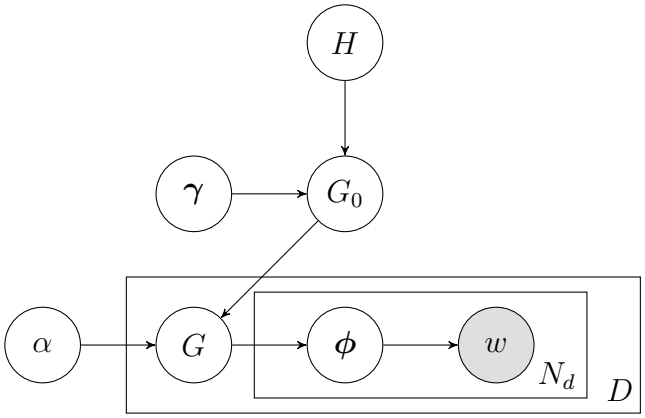


Figure 6.1: Plate diagram for hierarchical Dirichlet process topic model.

H [Teh et al., 2006]. Altogether, the model is as follows:

$$
\begin{aligned}
w_i|\boldsymbol{\phi}^{(i)} &\sim \text{Categorical}\left(\boldsymbol{\phi}^{(i)}\right), \\
\boldsymbol{\phi}^{(i)}|G &\sim G, \\
G|G_0 &\sim \text{DP}(\alpha, G_0), \\
G_0 &\sim \text{DP}(\gamma, H).
\end{aligned}
$$

Additionally, a plate diagram for the model is provided in Figure 6.1.

The HDP uses Dirichlet processes (DPs) to model the document-specific topic mixtures G as well as a global topic mixture G_0. While a Dirichlet distribution is a probability distribution over (probability) *vectors*, a Dirichlet process is a probability distribution over probability *distributions*. Consider a Dirichlet process $\text{DP}(\alpha', G_0')$ parametrized by a *concentration parameter* α_0' and a *base distribution* G_0'. Following Teh et al. [2006], who in turn use a result from Sethuraman [1994], every draw $G' \sim$

$\mathrm{DP}(\alpha', G_0')$ has the following form:

$$G' = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k'},$$

$$\phi_k' \sim G_0',$$

where δ_x is the Dirac delta distribution,

$$\delta_x(x') = \begin{cases} 1, & x' = x, \\ 0, & \text{otherwise}, \end{cases}$$

and where $\{\pi_k\}_k$ is a sequence of non-negative, monotonically non-increasing random variables that sum to one:

$$\pi_k = \pi_k' \prod_{\ell=1}^{k-1} (1 - \pi_\ell'),$$

$$\pi_k' \sim \mathrm{Beta}(1, \alpha').$$

This representation of G' is called the *stick-breaking* representation. Note that a higher concentration parameter α' tends to lend distributions G' that are concentrated on the first few atoms ϕ_k', whereas a lower α' tends to lend distributions G' that are more uniform.

The Chinese restaurant process (CRP) is a stochastic process closely related to

the Dirichlet process and frequently appears alongside it in the literature. The CRP produces distributions over partitions (clusterings) of the integers and is commonly described by the following metaphor. Consider a restaurant with an infinite number of tables and infinite seating at each table. When the first customer arrives, they sit at the first table. Then, at any point in the process, let n_k be the number of customers currently sitting at each occupied table k. When customer n arrives, they sit at any occupied table k with probability

$$\frac{n_k}{\alpha' + n - 1}$$

or at the next unoccupied table with probability

$$\frac{\alpha'}{\alpha' + n - 1}$$

for some positive real number α'. As Blei et al. [2010] summarize, the CRP induces the same distribution on partitions of the integers as draws from a stick-breaking representation,

$$Z_n \sim \sum_{k=1}^{\infty} \pi_k \delta_k,$$

where Z_n indicates the cluster to which integer n is assigned and the weights π_k are defined as previously. The CRP can be used as a mixture model by associating each

cluster with a draw from a base distribution, that is, by serving a single "dish" to each table when it is first occupied.

Coming back to the HDP topic model, then, the global topic distribution G_0 is a mixture over countably many draws from a Dirichlet distribution H provided as a prior. For each document, the local topic distribution G is a mixture over countably many draws from G_0. Put another way, each G is essentially a permutation of the topics in the stick-breaking representation of G_0, albeit with different weights. Because all local topic distributions G are draws from $DP(\alpha, G)$, they are likely to share topics unless the global concentration parameter γ is very small (close to zero). Because we only have a finite amount of time in practice, Teh et al. [2006] proposes learning the HDP using a *truncated* representation, meaning that the global and local topic distributions are truncated to a finite number of topics; the truncation level mediates a trade-off between computational cost and approximation error.[3] After a model is learned, an effective number of (and distribution over) topics can be computed by, for example, taking the minimal set of topics that cover 95% of the global topic distribution's posterior probability mass.

While many Bayesian nonparametric models like the HDP promise to alleviate the problem of determining an appropriate topic structure beforehand, models like the sparse additive generative topic model [Eisenstein et al., 2011] relax the assumption

[3]Truncation is essentially the imposition of a finite upper bound on the number of components in an otherwise unbounded model. Bryant and Sudderth [2012] proposes an online variational inference algorithm for the HDP that does not rely on such a fixed truncation, allowing the number of components to grow until the available memory is exhausted.

that stop words are filtered out separately from the main learning algorithm. In this chapter, I focus on the nHDP, a model that appears to combine the inference of topic structure with explicit modeling of (hence alleviation of the need to filter out) stop words. The nHDP is a Bayesian nonparametric tree-structured topic model [Paisley et al., 2015].[4] It consists of a global tree-structured random process constructed as recursively nested CRPs [Blei et al., 2010] and a set of local (per-document) tree-structured random processes constructed as recursively nested HDPs [Teh et al., 2006]. A document is thus modeled as a node-wise permutation of the global topic tree: The immediate child topics of a node in the local tree are a random permutation of the child topics of the corresponding node in the global tree. A topic assignment for a word token in the document is drawn by first sampling a path from the local tree and then sampling a node (topic) along that path from a stick-breaking distribution specific to that path. Accordingly, the nHDP topic model can be thought of as an extension of the HDP from a flat topic structure to a hierarchical one.

Based only on the tree-like structure of the nHDP, we might hope that it obviates the need to filter out stop words by assigning them predominantly to the root node, leaving the topic keys of other topics free of stop words. Indeed, in setting up an initial batch learning experiment, Paisley et al. [2015, p. 265] write that "because these three data sets contain stop words, we follow [2] and [4] by including a root node shared by all documents for this batch problem only." This statement suggests that a root node

[4] A similar model called the nested Chinese restaurant franchise was independently proposed in 2013 [Ahmed et al., 2013]; note that a manuscript introducing the nHDP appeared on arXiv in 2012 [Paisley et al., 2012].

is necessary for successful training of the nHDP when stop words are present in the data, and because topic models are often evaluated by the interpretability of their topic keys, it is reasonable to suspect that the root node filters out stop words. In a footnote, the authors appear to confirm this suspicion, writing that the root node "is shared by all documents and is intended to collect stop words" Paisley et al. [2015, p. 259]. For the aforementioned batch learning experiment on three data sets with stop words, the authors do not show topic keys or otherwise report any inspection of the contents of the topics, except indirectly via the predictive log-likelihood of the models. However, for the stochastic learning experiment on the New York Times and Wikipedia data sets, the authors "remove stop words and rare words." [Paisley et al., 2015, p. 265] and *do* present the topic keys of several topic subtrees below the root. These diagrams, which together take up over a page of space and are perhaps the focal point of the paper, appear free of stop words. Altogether, Paisley et al. [2015] convey a theory that the nHDP's root node filters stop words. I set out to test this theory.

6.3 Experiments

To facilitate reproduction of my work and reduce the computational expense, I did not use the exact data sets of Paisley et al. [2015]. While they used two large, unspecified corpuses of several million articles from The New York Times and Wikipedia (re-

spectively) for their main experiment, I used the smaller, publicly released WikiText-103 corpus [Merity et al., 2016]. Specifically, I used the tokenized training data from WikiText-103, splitting into documents along title headers, splitting each document into tokens along whitespace, and lower-casing the tokens, but otherwise performing no text normalization or filtering. I retained all title and section headers, including markup, in each document's text. This preprocessing procedure yielded a data set of 28 474 documents spanning 229 463 unique word types, including the pre-existing placeholder word for rare word types, `<unk>`.

I learned a tree of size (1, 10, 7, 5), that is, an active root node, ten nodes immediately below the root, seven nodes below each of those, and five nodes below each of those. I used an initialization sample size of 1000 documents and a mini-batch size of 1000 documents, stopping training at 1500 iterations (1.5 million documents seen). Informally, the topic keys appeared to be largely stable well before the 1500-iteration mark. For other hyperparameters I used $\lambda_0 = 0.1$, $\alpha = 5$, $\beta = 1$, $\gamma_1 = 1/3$, $\gamma_2 = 2/3$, and $\kappa = 0.5$, following Paisley et al. [2015]. While Paisley et al. [2015] used the adaptive learning rate of Ranganath et al. [2013], I used the non-adaptive learning rate of $\rho_s = (1 + s)^{-0.75}$ (where s is the iteration number starting at one) implemented in Paisley's software.

I used my own software implementation,[5] a modification of the MATLAB nHDP code released by Paisley,[6] for training. Beyond the choice of learning rate, I followed

[5] `https://github.com/ccmaymay/nhdp`

[6] `http://www.columbia.edu/~jwp2128/code/nHDP.zip`

the algorithm described by Paisley et al. [2015] as closely as possible, keeping implementation choices of the original software where implementation was not specified in the paper (for example, using three iterations of k-means in initialization). To follow the Paisley et al. [2015] algorithm, I amended the following discrepancies in the original software:

- In both the initialization and the global parameter update, the parameter update was scaled by 100 000 instead of $D/|C_s|$.
- In the first local parameter update (after subtree selection), prior terms are ignored.
- The global parameter update was smoothed by taking a convex combination of the update with a uniform term, where the uniform term had weight $\rho_s/10$.
- Subtree selection was limited to a maximum of 20 iterations (nodes).
- The local parameter update (after subtree selection) was limited to a maximum of 50 iterations.
- There was a numerical error in subtree selection that caused some nodes early in the tree to be selected repeatedly, potentially making selected subtrees smaller (on average) than they should be. The subtree selection iteration limit in the original implementation prevented this bug from creating an infinite loop.

I list these discrepancies for completeness and for the potential benefit of future

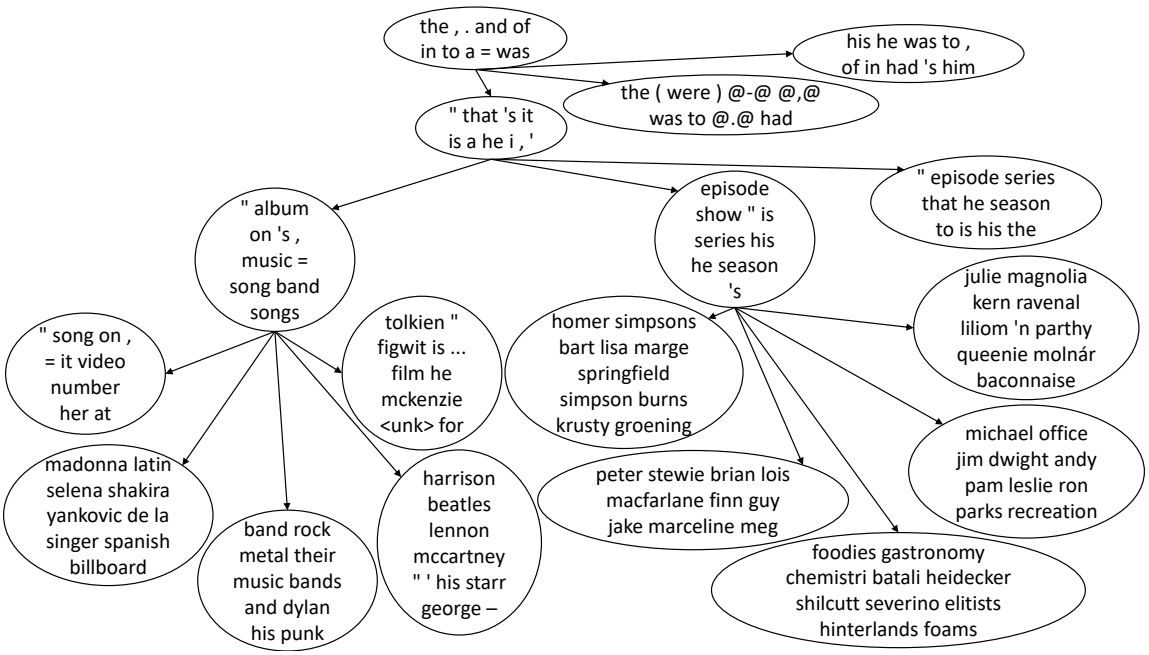


Figure 6.2: Topic keys for topics in subtree of nHDP trained on WikiText. At each level of the tree, the first m topics (for varying m, chosen to fit the screen) are shown in order of highest prior probability to lowest from left to right. The root node is at the top.

work, but preliminary experiments using the original software did not yield different conclusions; that is, the discrepancies do not explain my findings.

Following Paisley et al. [2015], I report results from a single learned model, that is, from one run of the training algorithm. However, repeated runs performed during development suggest that the effect of random initialization does not explain my findings. Also following Paisley et al. [2015], we investigate the model qualitatively by inspecting subtrees of the global topic tree, as there are 431 topics (nodes) in our configuration in total, and analyzing all of them is impractical. Moreover, the tree we learn is a truncated, finite approximation of an infinite tree, so we are already confining our analysis to an arbitrary subtree of the model.

topic keys
" that 's it is a he i , '
the (were) @-@ @,@ was to @.@ had
his he was to , of in had 's him
= on , @-@ his in he for a 's
is of are = , or <unk> . in and
the ( ) = is was city @.@ . @,@
" his he 's , a in that as and
of the <unk> ( ) and is in century ,
to of that for " in , = the were
the . , of ( ) on storm to in

Table 6.1: Topic keys of each topic immediately below the root for nHDP trained on WikiText. Topics have the same order as they do in the tree, starting with the first (leftmost) topic of the tree level at the top of the table.

Figure 6.2 shows the topic keys of a subtree formed by selecting the topics at each tree level (depth) of the learned model with highest prior probability. Consulting the figure, both the root node and the nodes in the level immediately below the root have topic keys dominated by stop words, and it is virtually impossible to interpret these topics solely based on the topic keys. In the third level, there is a mix of stop words and more content-bearing words, and it is much easier to understand what each topic is about; however, the second two topics both appear to concern TV shows, and one might wonder whether the difference between them might be illuminated by the next few content-bearing words in their lists (which are crowded out of the topic keys by stop words). In the fourth level, there are still stop words in many topics, but the ratio of content-bearing words to stop words is higher.

Figure 6.2 shows some of the most probable topics at each level (according to the global prior), illustrating that stop words are only gradually filtered as one descends

topic keys (filtered)
people wrote stated way felt described don good best think
m 000 ft crew km sea 20 metres water 30
british family death years died london father war george son
2011 2010 2008 2009 2012 2007 2013 announced year team
found form body example type water study human even animals
city park bridge river area construction street building $ west
book work published life wrote writing books story death own
century period island modern early people south east western history
united states public people u.s. american government national women 000
storm tropical km winds h mph cyclone damage hurricane depression

Table 6.2: Topic keys of each topic immediately below the root for nHDP trained on WikiText, in the same order as in Table 6.1, but after stop words have been filtered out in postprocessing. This view of the topics is more interpretable: We might start with labels of "attribution," "geography," and "history" for the first three topics (from top to bottom), respectively. For context, the root's topic keys in this configuration are: "well part year years day end early following number long."

the tree. In Table 6.1, I list the $k = 10$ topic keys of all topics in the first level below the root. These word lists are predominated by stop words, illustrating that stop words are not filtered as one traverses a single level of the tree. In Table 6.2, I list the $k = 10$ topic keys of those same topics after postprocessing the topics by filtering stop words from them [Schofield et al., 2017]. This view of each topic is more coherent and lends more readily to topic labeling, demonstrating that the topics are not inherently uninterpretable.[7]

[7]The stop word list used for filtering was developed on models from the WikiText-2 data set, but it was augmented (added to) substantially based on the WikiText-103 model's topic keys. I developed a basic interactive script with a read-eval-print loop to create the initial stop word list, and I would recommend a similarly interactive process for implementing the postprocessing approach of Schofield et al. [2017] in general (as stop words are domain-dependent in practice). Note that this is not a shortcoming of the postprocessing approach, but a feature: Interactive stop word filtering based on the resulting topic contents is impractical when filtering is performed during preprocessing.

6.4 Discussion

My results show that the proposed training algorithm for the nHDP does not filter stop words into the root node. The behavior of the nHDP with respect to stop words is presented more as a passing mention in Paisley et al. [2015] than as a result in itself; it is scarcely given any explicit attention in the paper. However, the authors draw a connection between the root node of the model and stop words in the data at multiple points in the paper, and state that the root node is at the very least *designed* ("intended") to filter ("collect") stop words. Additionally, by including topic keys in the paper at all and by allocating so much space on the page to them, the authors show that they (or the reviewers) believe inspection of those topic keys to have some value in topic modeling research; the explanation of my results is not that Paisley et al. [2015] don't consider the topic keys meaningful or relevant. One might propose that the root node's "collection" of stop words does not imply that stop words will not appear prominently in other topics as well; perhaps the collection is only a partial one, or happens gradually down the levels of the tree. But I argue a focus on the root node in this case would be misplaced, as the topics immediately below the root are virtually as full of stop words as the root (Figure 6.2 and Table 6.1), and the gradual filtering of stop words as the levels of the tree are descended is much more apparent in *subsequent* (lower) levels. Moreover, if a user was only interested in the lowest (most fine-grained) level of the tree, a simpler, flat model like the HDP would seem more appropriate; if we are interested in stop word filtering primarily for its effect on

the topic keys (interpretability), gradual filtering of stop words down the tree would not be useful.

While I maintain that Paisley et al. [2015] suggest a behavior that is not borne out in practice, I do not mean to suggest they *purposefully* misled the reader. It is plausible that none of the authors realized the implication created in the final version of the paper. It is also plausible that none of the authors even realized that the root node does not filter out stop words, as text data is commonly preprocessed by converting word types into integer array indices to improve performance of a text processing algorithm, and the software implementation of the nHDP released by Paisley indeed uses this approach, meaning any trained model would require postprocessing before the topic keys could be inspected. Further supporting this explanation is that the models learned in the batch experiment were evaluated only by predictive log-likelihood and not through any other lens, so the omission of topic keys is not conspicuous for that experiment in isolation.

In this chapter, I have presented a single model resulting from one run of the training algorithm using one setting of the model hyperparameters and algorithm parameters. I chose those parameters to reproduce the results of the original study as well as possible within some constraints. However, I spent months, perhaps even a year or more developing and testing a C implementation of the training algorithm before I found the existing MATLAB implementation, and I devoted a large part of a summer to applying the nHDP to different data sets using different hyperparameters

and algorithm parameters.[8] I did not find any results (models) that filtered stop words substantially better than the model I have presented here. Although results of these experiments are scattered among my archived e-mails, I did not catalog the settings that gave rose to those results, so I am unable to present those experiments in a way that I think would provide further insight. I hope it will suffice to say: I tried a *lot* of settings.

One potential limitation of the nHDP in the context of stop word filtering is its lack of sparsity. The nHDP represents each topic as a distribution that has positive probability mass on each word in the vocabulary, so there is not a strong penalty on a topic giving weight to a word that already has weight in another topic. Accordingly, descendant topics in the nHDP can model slight variations in the stop word distribution (relative to the root topic's stop word distribution) that correlate with their more content-bearing words. This could be tested, for example, by comparing the word distributions of each child of the root topic to the empirical stop word distributions of documents that select those topics during inference. Alternatively, the nHDP could be trained on a corpus in which proportions of stop words are artificially fixed in all documents, and the resultant model could be inspected for stop word filtering.

[8] This was almost seven years prior to the time of writing—time flies!

6.5 Conclusion

I have implemented a learning algorithm introduced in a previous paper, building on a partial implementation released by the authors. I was approximately able to reproduce their finding that the algorithm was capable of learning interpretable topic trees. However, while their paper suggested the algorithm would automatically collect stop words in the topic tree's root node, I found this not to be the case, illustrating that even theories of language technology created largely within a single paper may fail to hold up in practice.

Chapter 7

Conclusion

I have used case studies to show that four theories of topic modeling, spanning algorithms, applications, and models, are not generally borne out in practice. In Chapter 3, I showed that the theory of an online learning algorithm developed in a prior study failed to hold beyond the precise (yet unspecified) initial conditions of their experiments, and I made the following additional contributions to science:

- I perform a parameter study of a particle filtering training algorithm for LDA introduced in prior work [Canini et al., 2009], providing a more comprehensive analysis of the algorithm.

- I show that tuning an LDA model by perplexity is just as good as tuning it by the evaluation metric (using gold-standard forum labels) on several qualitatively different subsets of the common 20 newsgroups data set, suggesting that there may be little room for improvement in topic modeling on that data set from an

information theoretic perspective.

- I publicly released my LDA particle filter training and experiment code for the benefit of future research.[1]

In Chapter 4, I demonstrated that the tasks of topic identification and discovery are less complementary than the literature suggests, with topic features failing to improve topic identification performance even under low supervision; I made the following additional contributions to science:

- This study is the first of its time to provide *cross-community* evaluations of SAGE and other models on both text and speech data.

- This study also uses low-resource triphone state cluster soft counts as speech data for topic ID, following May et al. [2015a]. The low-resource setting reflects constraints often faced in real-world applications, and I report topic ID performance under limited supervision to better illuminate the practical strengths and weaknesses of the learned representations.

- Finally, I believe that this comparison of several prominent learned representations on two complementary tasks on both text and speech, presented together in the same study, will provide a useful point of reference for future research.

In Chapter 5, I studied the purported language independence of topic modeling, illustrating that the end-to-end topic modeling algorithm does bear a significant language

[1] `https://github.com/ccmaymay/pflda`

dependence in practice; I also made the following additional contributions to science:

- This chapter is one of the first published studies of the impact of lemmatization on topic modeling.
- I performed a human-in-the-loop analysis of topic model interpretability and and an automated analysis using proxy methods on a similar data set, increasing collective knowledge about the practical relationship between costly human-in-the-loop evaluation methods and automated proxy methods.
- I analyzed the impact of lemmatization on topic models in several languages by re-using methods from Schofield and Mimno [2016], increasing the comparability of our studies and facilitating future work.
- I publicly released my experiment code for the benefit of future research.[2]

Finally, in Chapter 6, I showed that a hierarchical topic model theorized as capable of filtering stop words does not achieve that goal in practice, making the following additional contributions to science:

- I performed a focused analysis of the stop word filtering behavior of the nHDP, complementing the original study.
- I produced a detailed analysis of the software implementation of the nHDP provided by Paisley et al., comparing and contrasting it with the published algorithm.

[2] `https://github.com/ccmaymay/lda-lemmas`

- I publicly released my nHDP training and experiment code for the benefit of future research.[3] This code is closely based on Paisley et al.'s code, amending its discrepancies with the algorithm from the paper.

In each chapter, I moreover proposed specific further research for investigating methodological explanations of the gap between theory and practice.

While the case studies I have presented represent a wide sample of topic modeling research, they are also limited in several ways. For one, they are almost exclusively based on sources from the academic literature, but topic modeling discourse spans many channels, including: journal articles; conference presentations, question-and-answer sessions, panels, and proceedings; funding proposals; book chapters; classroom lectures and homework assignments, electronic pre-prints; peer reviews and author responses; discussions in meeting rooms, hallways; and social media, software source code, documentation, and issue trackers. I have chosen to provide a broad investigation of implicit theories in topic modeling, but concerns about scope preclude a more thorough understanding of how topic modeling knowledge is communicated more generally. A complementary, more narrow investigation of a single theory's communication across channels could help contextualize or challenge my findings.

In fact, these studies are all retrospective: Each one arose from an original research project with different objectives. The case studies on streaming learning (Chapter 3), language independence (Chapter 5), and stop word filtering (Chapter 6) went awry

[3] `https://github.com/ccmaymay/nhdp`

in various ways, while the study on features for topic identification (Chapter 4) produced surprising results. The prevalence of implicit theories that don't generally hold in practice only became apparent in hindsight, and this dissertation represents a reanalysis of the original projects through a novel lens. Had I known then what I know now, I may have begun by reviewing the literature, enumerating common implicit theories, and testing those in a more controlled fashion. For example, in the manual evaluation of the language independence study (Section 5.4), I used stochastic variational inference to train LDA models because I had believed that setting a fixed asymmetric prior was the best way to automatically filter out stop words and my implementation of stochastic variational inference facilitated that. However, in hindsight, I believe that Gibbs sampling would have produced better results. Additionally, using Gibbs sampling would have eliminated a potential confounding variable by bringing the methods of the language independence study more in line with the studies on streaming learning and features for topic identification, which use sampling algorithms for training.

7.1 Recent Developments

In the time I have been a Ph.D. student, NLP research has changed dramatically. The year I began my Ph.D. program was the year the word2vec paper came out [Mikolov et al., 2013], and over the next several years, virtually all of NLP seemed

to transition from traditional probabilistic graphical models to neural networks as the technology of choice. While I can't comment on the state of neural topic modeling directly, I don't expect the transition to neural models to have addressed the specific knowledge gaps I've discussed. One of the main challenges of topic modeling is its lack of clear objectives and evaluation metrics; there is currently no topic modeling equivalent of accuracy or F-score. This presents a challenge to neural modeling, which largely does away with modeling the internal workings of natural processes and instead uses flexible classes of functions and powerful learning techniques to model those processes' observable behavior. The desired behavior of topic models is generally only specified abstractly, making neural modeling more difficult than in most cases.

However, I argue that the evaluation of topic models is not fundamentally more complex than that of other NLP tasks. Rather, the difference is that the complexity of topic model evaluation is more obvious than the complexity of evaluation in other tasks. The existence of a widely accepted evaluation metric for a task masks the complexity of that task's evaluation by abstracting from it to a relatively straightforward algorithm. For example, the F-score is a widely accepted metric for classification tasks like topic identification. In addition to aggregating a topic identification system's predictions in a certain way (and not in others), the F-score relies on a set of "gold" or "ground truth" *labels* indicating which topic each document belongs to, and this set of labels replaces a complex, even fraught *problem* with a fixed, definitive set

of *answers*.[4] For instance, the problem of whether to assign the label "crime" or the label "terrorism" to a news article about hate crime is masked behind a metric and set of labels amenable to grad student descent,[5] leading to potentially dangerous biases in downstream applications. Thus, the evaluation of topic modeling may ultimately be no more complex than that of other tasks; its complexity may just be harder to overlook.

Additionally, the culture of neural NLP gives me no reason to expect that the implicit theories of today's research are more often made explicit and tested empirically. Indeed, in neural NLP generally, there seems to be much more attention given to held-out estimates of evaluation metrics, with many data sets accompanied by "leaderboards" of the best-scoring models and algorithms. This focus on direct comparison of approaches using automated evaluation metrics draws attention away from inspecting and investigating the assumptions built into those metrics, where the implicit theories I've encountered seem to lurk. Research into the assumptions and limitations of evaluation methods continues, but I have no reason to believe it has accelerated, nor do I think the transition to neural methods has improved the implicit theory situation otherwise.

Another shift that has happened during my time as a student is the growth of research focusing on *reproducibility* in machine learning. A relatively early example

[4] This framing mirrors that of a comment I heard at a conference (or perhaps on Twitter or in a book; regrettably, I can't find the source) regarding the use of gender as an explanatory variable in computational linguistics and social science: Gender is not a *solution*, it is a *problem*.

[5]

of this work in NLP is Fokkens et al. [2013], who analyze the reproducibility of results in two NLP tasks and find five general sources of implicit variation that hinder it: preprocessing, experimental setup (including train/dev/test splits), versions of software and resources, differences in intermediate outputs, and random variation. These types of variation are implicit in the sense that they are not reported in the original papers. To the extent we can assume researchers write papers with the intention that their results should be reproducible, the problematic sources of variation identified by Fokkens et al. [2013] may point to implicit theories about which details are important for reproduction and which details are not. For example, the omission of preprocessing details in a paper could be taken to suggest that those preprocessing details are not essential to reproducing the results.

Crane [2018] performs a similar analysis of reproducibility in question answering that focuses on deep learning papers and variation in results that persists even after releasing source code. In addition to finding similar sources of variation to those of Fokkens et al. [2013], Crane [2018] finds additional sources of variation that are created or exacerbated by the complexity of deep learning algorithms and software.

While my analysis is novel, the individual observations I've made are just points on a spectrum of negative results arising from reproductions and extensions of prior work. The "reproducibility" research community has coalesced to study some of the more challenging parts of that spectrum, and even before the growth of that community, many researchers individually encountered such problems using topic models.

Research on the *interpretability* of NLP (and more generally, machine learning) models has also blossomed in recent years. Topic modeling may be considered a precursor of this research thrust, as topic models are often developed to be interpreted and used by humans. "Interpretability" research aims to facilitate the interpretation of NLP models generally, such that the user of a question answering system, for example, can understand how that system predicts the answers it does. While the concept of understanding a prediction may be intuitive, Lipton [2016] shows that interpretability research is motivated by a variety of concerns: the desire for a model to gain the trust of its users, so that it can be deployed in high-stakes situations; the desire to be able to infer causal relationships from a model's predictions, so that it can help generate scientific hypotheses; the desire to be able to judge a model's transferability to unseen data and new use cases; the desire for a model to be informative, so that human decision-makers can use it to aid their work; and the desire for a model's use to be ethical, so that the application of machine learning does not continue to subject minorities to stereotyping or perpetuate other injustices [Lipton, 2016].

Similarly, the methods of interpretability research may be designed to satisfy a variety of properties: simulatability, the ability of a user to be able to reasonably understand and simulate the entire model; decomposability, the ability to decompose a larger model into individually understandable parts; algorithmic transparency, the ability of a researcher to be able to understand and reason about the learning algorithm that produced the model; text explanations and visualization, the *post-hoc*

construction of text or visual explanations of a model; local explanations, the *post-hoc* explanation of what is happening in a model locally (for example, when making a specific prediction); and explanation by example, the *post-hoc* explanation of prediction on a given example by listing other examples or feature representations that yield similar predictions [Lipton, 2016]. Accordingly, research on interpretability comprises not a set of efforts to solve a single task but a collection of efforts to solve a diverse set of related tasks.

7.2 Extensions

Though my work focuses on topic modeling research, I do not believe gaps between implicit theory and practice are limited to topic modeling or even to computer science. Practice not living up to theory is a pattern that extends far beyond the case studies presented here; after all, theory itself is defined by assumptions and simplifications. And I have only shown that these theories do not hold in general, that a good-faith effort to reproduce or extend research building on such implicit theories often fails; however, I suspect there are common settings (special cases) in which those theories do hold. I also do not wish to malign the intentions of researchers who promote these theories or do not publish their empirical results about them: I suspect most researchers with knowledge about such implicit theories take that knowledge for granted. On the other hand, those who realize they *could* publish such knowledge

may be unable to. Nguyen et al. [2016] finds that the field of computational linguistics primarily values creativity and predictive accuracy, values that would likely be flouted by a study of an implicit theory many researchers take for granted. Additionally, in a study of reviews for submissions to the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, Gao et al. [2019] found that the overall submission score most correlated with the sub-scores for the submission's "soundness" and "substance," followed closely by "originality." While empirical studies of implicit theories would seem well-aligned with the value of "soundness," the prioritization of "substance" and "originality" over the remaining sub-scores reflects the findings of Nguyen et al. [2016]. Thus, if one seeks a career in computational linguistics research, spending time to study and publish an investigation of implicit theory may be imprudent. The belief that every researcher must "publish or perish" is widespread and largely justified, and attempting to publish work of little value to the research community may not prevent one from perishing.

Furthermore, while I have sometimes referred to a singular topic modeling community, there are actually multiple, variously overlapping communities, and the knowledge and values held by these communities may differ. For example, "topic identification" is a term that seems to be used most prominently in speech processing communities, so the different histories and values between speech-based topic identification and text-based topic modeling may help explain my findings in Chapter 4. More broadly, much of the prior work underpinning my arguments was published in

general machine learning and artificial intelligence communities [Blei et al., 2003b, Griffiths and Steyvers, 2004, Canini et al., 2009, Paisley et al., 2015], not natural language processing or computational linguistics communities that might be more interested in the nuances of language data. At the same time, a large amount of topic modeling research is published by the Association for Computational Linguistics, so the comparison of implicit theories across research publishers and communities would be an important next step in validating my conclusions.

Although the theories I study are not formally stated in the literature, I argue they are nonetheless supported and conveyed such that the reader—especially a wide-eyed, optimistic graduate student like I once was—is prone to detect and adopt them. Much like the patterns that a machine learning algorithm discovers, these theories are often hidden, but that does not mean they do not exist or do not matter. I hypothesize that these implicit theories (and the gaps that often separate them from practice) not only impede research progress in the near term, but function as gatekeeping devices and limit the diversity of the research community. Specifically, if we assume people encounter these theories and learn about their relation to practice as they become established in topic modeling communities, then established researchers—and those connected to them—will have an advantage. Although having an experienced mentor is surely to improve one's chances for success in any vocation, if we want research to function as a meritocracy, we should be wary of any secrets—intentionally kept or otherwise—that are needed to successfully publish research. The existence of

general, significant, yet unpublished topic modeling knowledge potentially enables well-connected groups of researchers to out-publish others on average; thus, implicit theories are positioned to help researchers who are upper-class, white, straight, able, cisgender, and/or otherwise privileged succeed, while presenting yet another obstacle to the less privileged.

If we do take my conclusions at face value, though, there is hope of ameliorating the problem. While my original research was taking place, an interdisciplinary interest in improving the reproducibility of research was growing rapidly. Improvements to reproducibility would likely translate to expositions of some implicit theories like those I've considered, as the steps required to reproduce a paper's results would become more transparent, making any idiosyncrasies more visible.

However, the philosopher W. V. O. Quine argued that theory—and knowledge claims in general—are underdetermined by evidence; that is, the available empirical evidence is generally insufficient to support a given theory by itself, and that gap between evidence and theory is bridged by our biases, our values. Moreover, from a feminist empiricist view, these values play a legitimate role in scientific inquiry.[6] perhaps, then, the solution to "implicit theory" lies not in changing our methods, but in changing our culture.

[6]See Anderson [2020] for an overview of feminist empiricism and underdetermination.

Bibliography

A. Ahmed, Q. Ho, J. Eisenstein, E. P. Xing, A. J. Smola, and C. H. Teo. Unified analysis of streaming news. In *Proceedings of the 20th International World Wide Web Conference (WWW)*, pages 267–276, Mar 2011.

A. Ahmed, L. Hong, and A. Smola. Nested chinese restaurant franchise process: Applications to user tracking and document modeling. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1426–1434, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL `http://proceedings.mlr.press/v28/ahmed13.html`.

R. Al-Rfou, B. Perozzi, and S. Skiena. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W13-3520`.

BIBLIOGRAPHY

E. Anderson. Feminist epistemology and philosophy of science. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Spring 2020 edition, 2020. URL `https://plato.stanford.edu/archives/spr2020/entries/feminism-epistemology/`.

D. Andrzejewski, X. Zhu, and M. Craven. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 25–32, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553378. URL `https://doi.org/10.1145/1553374.1553378`.

S. Arora, R. Ge, Y. Halpern, D. Mimno, A. Moitra, D. Sontag, Y. Wu, and M. Zhu. A practical algorithm for topic modeling with provable guarantees. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 280–288, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL `http://proceedings.mlr.press/v28/arora13.html`.

A. Banerjee and S. Basu. Topic models over text streams: A study of batch and online unsupervised learning. In *Proceedings of the 7th SIAM International Conference on Data Mining (SDM)*, pages 431–436, Apr 2007.

H. Becker. *Identification and Characterization of Events in Social Media*. PhD thesis, Columbia University, 2011.

J. Bian, B. Gao, and T. Liu. Knowledge-powered deep learning for word embedding.

In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, ECML PKDD 2014, pages 132–148, New York, NY, USA, 2014. Springer-Verlag New York, Inc. ISBN 978-3-662-44847-2. doi: 10.1007/978-3-662-44848-9_9. URL http://dx.doi.org/10.1007/978-3-662-44848-9_9.

J. M. Bischof and E. M. Airoldi. Summarizing topical content with word frequency and exclusivity. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 9–16, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

D. M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, Apr. 2012. ISSN 0001-0782. doi: 10.1145/2133806.2133826. URL https://doi.org/10.1145/2133806.2133826.

D. M. Blei, T. L. Griffiths, M. I. Jordan, and J. B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems 16 (NIPS)*, 2003a.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, Mar. 2003b. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=944919.944937.

D. M. Blei, T. L. Griffiths, and M. I. Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*,

57(2):1–30, Feb. 2010. ISSN 0004-5411. doi: 10.1145/1667053.1667056. URL http://doi.acm.org/10.1145/1667053.1667056.

J. Bohr and R. E. Dunlap. Key topics in environmental sociology, 1990–2014: results from a computational text analysis. *Environmental Sociology*, 4(2):181–195, 2018. doi: 10.1080/23251042.2017.1393863. URL https://doi.org/10.1080/23251042.2017.1393863.

B. Börschinger and M. Johnson. Using rejuvenation to improve particle filtering for Bayesian word segmentation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 85–89, Jul 2012.

J. Boyd-Graber, D. Mimno, and D. Newman. Care and feeding of topic models: Problems, diagnostics, and improvements. In E. M. Airoldi, D. Blei, E. A. Erosheva, and S. E. Fienberg, editors, *Handbook of Mixed Membership Models and Their Applications*, CRC Handbooks of Modern Statistical Methods. CRC Press, Boca Raton, Florida, 2014. URL https://home.cs.colorado.edu/~jbg/docs/2014_book_chapter_care_and_feeding.pdf.

J. Boyd-Graber, Y. Hu, and D. Mimno. Applications of topic models. *Foundations and Trends©in Information Retrieval*, 11(2-3):143–296, 2017. ISSN 1554-0669. doi: 10.1561/1500000030. URL http://dx.doi.org/10.1561/1500000030.

T. Broderick, N. Boyd, A. Wibisono, A. C. Wilson, and M. I. Jordan. Streaming vari-

ational Bayes. In *Advances in Neural Information Processing Systems 26 (NIPS)*, Dec 2013.

M. Bryant and E. Sudderth. Truly nonparametric online variational inference for hierarchical dirichlet processes. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/838e8afb1ca34354ac209f53d90c3a43-Paper.pdf.

M. Bunge. Technology as applied science. *Technology and Culture*, 7(3):329–347, 1966. doi: 10.2307/3101932.

R. S. Buurma. The fictionality of topic modeling: Machine reading anthony trollope's barsetshire series. *Big Data & Society*, 2(2):2053951715610591, 2015. doi: 10.1177/2053951715610591. URL https://doi.org/10.1177/2053951715610591.

K. Canini, L. Shi, and T. Griffiths. Online inference of topics with latent Dirichlet allocation. In D. van Dyk and M. Welling, editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 65–72, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR. URL http://proceedings.mlr.press/v5/canini09a.html.

J. Chang, J. L. Boyd-Graber, S. Gerrish, C. Wang, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In Y. Bengio,

D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 288–296. Curran Associates, Inc., 2009. URL `http://papers.nips.cc/paper/3700-reading-tea-leaves-how-humans-interpret-topic-models.pdf`.

K.-Y. Chen, H.-S. Lee, H.-M. Wang, B. Chen, and H.-H. Chen. I-vector based language modeling for spoken document retrieval. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 7083–7088, 2014.

C. Cieri, D. Graff, O. Kimball, D. Miller, and K. Walker. Fisher english training speech part 1 speech LDC2004S13. DVD, 2004a.

C. Cieri, D. Graff, O. Kimball, D. Miller, and K. Walker. Fisher english training speech part 1 transcripts LDC2004T19. Web Download, 2004b.

C. Cieri, D. Miller, and K. Walker. The fisher corpus: a resource for the next generations of speech-to-text. In *International Conference on Language Resources and Evaluation (LREC)*, pages 69–71, 2004c.

M. Crane. Questionable Answers in Question Answering Research: Reproducibility and Variability of Published Results. *Transactions of the Association for Computational Linguistics*, 6:241–252, 04 2018. ISSN 2307-387X. doi: 10.1162/tacl_a_00018. URL `https://doi.org/10.1162/tacl_a_00018`.

BIBLIOGRAPHY

T. A. Curry and M. P. Fix. May it please the twitterverse: The use of twitter by state high court judges. *Journal of Information Technology & Politics*, 16(4): 379–393, 2019. doi: 10.1080/19331681.2019.1657048. URL https://doi.org/10.1080/19331681.2019.1657048.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990. ISSN 1097-4571. doi: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9. URL http://dx.doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

C. N. dos Santos and B. Zadrozny. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st Annual International Conference on Machine Learning*, pages 1818–1826, 2014.

A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 176–183, Jun 2000.

A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.

J. Eisenstein, A. Ahmed, and E. P. Xing. Sparse additive generative models of text. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1041–1048, Bellevue, WA, US, Jun 2011. doi: 10.5555/3104482.3104613.

BIBLIOGRAPHY

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9: 1871–1874, 2008.

A. Fokkens, M. van Erp, M. Postma, T. Pedersen, P. Vossen, and N. Freire. Offspring from reproduction problems: What replication failure teaches us. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1691–1701, Sofia, Bulgaria, Aug. 2013. Association for Computational Linguistics. URL `https://aclanthology.org/P13-1166`.

M. Franssen, G.-J. Lokhorst, and I. van de Poel. Philosophy of technology. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2018 edition, 2018. URL `https://plato.stanford.edu/archives/fall2018/entries/technology/`.

R. J. Gallagher, K. Reing, D. Kale, and G. Ver Steeg. Anchored correlation explanation: Topic modeling with minimal domain knowledge. *Transactions of the Association for Computational Linguistics*, 5:529–542, 2017. doi: 10.1162/tacl_a_00078. URL `https://aclanthology.org/Q17-1037`.

Y. Gao, S. Eger, I. Kuznetsov, I. Gurevych, and Y. Miyao. Does my rebuttal matter? insights from a major NLP conference. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1274–1290, Min-

neapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1129. URL https://aclanthology.org/N19-1129.

S. J. Gershman and D. M. Blei. A tutorial on Bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1–12, 2012. ISSN 0022-2496. doi: https://doi.org/10.1016/j.jmp.2011.08.004. URL https://www.sciencedirect.com/science/article/pii/S002224961100071X.

W. R. Gilks and C. Berzuini. Following a moving target—Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society*, 63(1):127–146, 2001.

S. Godsill. Particle filtering: the first 25 years and beyond. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7760–7764, 2019. doi: 10.1109/ICASSP.2019.8683411.

M. R. Gormley, M. Dredze, B. Van Durme, and J. Eisner. Shared components topic models. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, pages 783–792, 2012.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, 2004. ISSN 0027-8424. doi: 10.1073/pnas.0307752101. URL https://www.pnas.org/content/101/suppl_1/5228.

BIBLIOGRAPHY

Z. Harris. Distributional structure. *Word*, 10(23):146–162, 1954. doi: 10.1080/00437956.1954.11659520.

D. Harwath and T. J. Hazen. Topic identification based extrinsic evaluation of summarization techniques applied to conversational speech. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5073–5076, 2012. doi: 10.1109/ICASSP.2012.6289061.

T. J. Hazen, F. Richardson, and A. Margolis. Topic identification from audio recordings using word and phone recognition lattices. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 659–664, 2007. doi: 10.1109/ASRU.2007.4430190.

M. Hoffman, D. M. Blei, and F. R. Bach. Online learning for latent Dirichlet allocation. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 856–864. Curran Associates, Inc., 2010. URL `http://papers.nips.cc/paper/3902-online-learning-for-latent-dirichlet-allocation.pdf`.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(4):1303–1347, 2013. URL `http://jmlr.org/papers/v14/hoffman13a.html`.

T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth*

Conference on Uncertainty in Artificial Intelligence, UAI'99, page 289–296, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1558606149.

G. Lapesa and S. Evert. A large scale evaluation of distributional semantic models: Parameters, interactions and model selection. *Transactions of the Association for Computational Linguistics*, 2:531–545, 2014.

J. H. Lau, D. Newman, and T. Baldwin. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/E14-1056`.

O. Levy, Y. Goldberg, and I. Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.

W. Li and A. McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23th International Conference on Machine Learning (ICML)*, pages 577–584, 2006.

W. Li, D. Blei, and A. McCallum. Nonparametric Bayes pachinko allocation. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*, UAI'07, page 243–250, Arlington, Virginia, USA, 2007. AUAI Press. ISBN 0974903930.

BIBLIOGRAPHY

C. Lin and Y. He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 375–384, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-512-3. doi: 10.1145/1645953.1646003. URL http://doi.acm.org/10.1145/1645953.1646003.

W. Ling, T. Luís, L. Marujo, R. F. Astudillo, S. Amir, C. Dyer, A. W. Black, and I. Trancoso. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL https://aclweb.org/anthology/D/D15/D15-1176.

Z. C. Lipton. The mythos of model interpretability. *CoRR*, abs/1606.03490, 2016. URL http://arxiv.org/abs/1606.03490.

J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, Sep 1998.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, 2008. ISBN 9780521865715.

C. May, A. Clemmer, and B. Van Durme. Particle filter rejuvenation and Latent Dirichlet Allocation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 446–451,

Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-2073. URL https://www.aclweb.org/anthology/P14-2073.

C. May, F. Ferraro, A. McCree, J. Wintrode, D. Garcia-Romero, and B. Van Durme. Topic identification and discovery on text and speech. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2377–2387, Lisbon, Portugal, Sept. 2015a. Association for Computational Linguistics. doi: 10.18653/v1/D15-1285. URL https://www.aclweb.org/anthology/D15-1285.

C. May, F. Ferraro, A. McCree, J. Wintrode, D. Garcia-Romero, and B. Van Durme. Supplementary material for: Topic identification and discovery on text and speech. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (Suppl.)*, Lisbon, Portugal, Sept. 2015b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D15-1285.

C. May, R. Cotterell, and B. Van Durme. An Analysis of Lemmatization on Topic Models of Morphologically Rich Language. *arXiv e-prints*, art. arXiv:1608.03995, Aug. 2016. URL https://arxiv.org/abs/1608.03995.

A. K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: Modeling facets and opinions in weblogs. In *Proceedings of the 16th International Conference*

on World Wide Web, WWW '07, pages 171–180, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. doi: 10.1145/1242572.1242596. URL http://doi.acm.org/10.1145/1242572.1242596.

M. Meilă. Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007. ISSN 0047-259X. doi: https://doi.org/10.1016/j.jmva.2006.11.013. URL https://www.sciencedirect.com/science/article/pii/S0047259X06002016.

S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models, 2016.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

D. Mimno, H. M. Wallach, J. Naradowsky, D. A. Smith, and A. McCallum. Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 880–889, Singapore, Aug. 2009. Associa-

tion for Computational Linguistics. URL https://www.aclweb.org/anthology/D09-1092.

D. Mimno, H. Wallach, E. Talley, M. Leenders, and A. McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 262–272, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D11-1024.

T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 352–359, Aug 2002.

M. Morchid, M. Bouallegue, R. Dufour, G. Linarès, D. Matrouf, and R. de Mori. An I-vector based approach to compact multi-granularity topic spaces representation of textual documents. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 443–454, Doha, Qatar, Oct. 2014a. Association for Computational Linguistics. doi: 10.3115/v1/D14-1051. URL https://aclanthology.org/D14-1051.

M. Morchid, R. Dufour, and G. Linarès. A LDA-based topic classification approach from highly imperfect automatic transcriptions. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1309–1314, Reykjavik, Iceland, May 2014b. European Language Resources Associ-

ation (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/8_Paper.pdf.

R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen. Joint latent topic models for text and citations. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 542–550, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. doi: 10.1145/1401890.1401957. URL http://doi.acm.org/10.1145/1401890.1401957.

D. Newman, C. Chemudugunta, and P. Smyth. Statistical entity-topic models. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 680–686, 2006.

D. Newman, P. Smyth, M. Welling, and A. Asuncion. Distributed inference for latent dirichlet allocation. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper/2007/file/2dea61eed4bceec564a00115c4d21334-Paper.pdf.

D. Newman, A. Asuncion, P. Smyth, and M. Welling. Distributed algorithms for topic models. *Journal of Machine Learning Research*, 10(62):1801–1828, 2009. URL http://jmlr.org/papers/v10/newman09a.html.

D. Newman, J. H. Lau, K. Grieser, and T. Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the*

North American Chapter of the Association for Computational Linguistics, HLT '10, pages 100–108, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5. URL http://dl.acm.org/citation.cfm?id=1857999.1858011.

D. Nguyen, A. S. Doğruöz, C. P. Rosé, and F. de Jong. Computational Sociolinguistics: A Survey. *Computational Linguistics*, 42(3):537–593, 09 2016. ISSN 0891-2017. doi: 10.1162/COLI_a_00258. URL https://doi.org/10.1162/COLI_a_00258.

J. Nothman, H. Qin, and R. Yurchak. Stop word lists in free open-source software packages. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 7–12, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-2502. URL https://aclanthology.org/W18-2502.

J. Paisley, C. Wang, D. M. Blei, and M. I. Jordan. Nested hierarchical Dirichlet processes, 2012. URL https://arxiv.org/abs/1210.6738v1.

J. Paisley, C. Wang, D. M. Blei, and M. I. Jordan. Nested hierarchical Dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37 (2):256–70, Feb 2015. doi: 10.1109/TPAMI.2014.2318728.

M. Paul and M. Dredze. Sprite: Generalizing topic models with structured priors. *Transactions of the Association for Computational Linguistics*, 3:43–57, 2015.

ISSN 2307-387X. URL https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/403.

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

M. K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, Jun 1999.

R. Ranganath, C. Wang, B. David, and E. Xing. An adaptive learning rate for stochastic variational inference. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 298–306, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL http://proceedings.mlr.press/v28/ranganath13.html.

R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

J. Reich, D. Tingley, J. Leder-Luis, M. E. Roberts, and B. Stewart. Computer-assisted reading and discovery for student generated text in massive open online courses. *Journal of Learning Analytics*, 2(1):156–184, Nov. 2014. doi: 10.18608/jla.2015.

21.8. URL https://learning-analytics.info/index.php/JLA/article/view/4138.

M. Roberts, B. Stewart, D. Tingley, and E. Airoldi. The structural topic model and applied social science. In *Advances in Neural Information Processing Systems Workshop on Topic Models: Computation, Application, and Evaluation*, 2013. URL http://mimno.infosci.cornell.edu/nips2013ws/nips2013tm_submission_24.pdf.

M. E. Roberts, B. M. Stewart, D. Tingley, C. Lucas, J. Leder-Luis, S. K. Gadarian, B. Albertson, and D. G. Rand. Structural topic models for open-ended survey responses. *American Journal of Political Science*, 58(4):1064–1082, 2014. doi: https://doi.org/10.1111/ajps.12103. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/ajps.12103.

A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/D07-1043.

H. Rubenstein and J. B. Goodenough. Contextual correlates of synonymy. *Commun.*

ACM, 8(10):627–633, Oct. 1965. ISSN 0001-0782. doi: 10.1145/365628.365657. URL https://doi.org/10.1145/365628.365657.

H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK, 1994.

A. Schofield and D. Mimno. Comparing apples to apple: The effects of stemmers on topic models. *Transactions of the Association for Computational Linguistics*, 4: 287–300, 2016. ISSN 2307-387X. URL https://transacl.org/ojs/index.php/tacl/article/view/868.

A. Schofield, M. Magnusson, and D. Mimno. Pulling out the stops: Rethinking stopword removal for topic models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 432–436, Valencia, Spain, April 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E17-2069.

C. Schwemmer and S. Jungkunz. Whose ideas are worth spreading? the representation of women and ethnic groups in ted talks. *Political Research Exchange*, 1(1):1–23, 2019. doi: 10.1080/2474736X.2019.1646102. URL https://doi.org/10.1080/2474736X.2019.1646102.

J. Scott and J. Baldridge. A recursive estimate for the predictive likelihood in a topic model. In C. M. Carvalho and P. Ravikumar, editors, *Proceedings of the Six-*

teenth International Conference on Artificial Intelligence and Statistics, volume 31 of *Proceedings of Machine Learning Research*, pages 527–535, Scottsdale, Arizona, USA, 29 Apr–01 May 2013. PMLR. URL http://proceedings.mlr.press/v31/scott13a.html.

J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994. URL http://www3.stat.sinica.edu.tw/statistica/j4n2/j4n216/j4n216.htm.

R. Soricut and F. Och. Unsupervised morphology induction using word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1627–1637, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N15-1186.

M. Straka and J. Straková. Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada, Aug. 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-3009. URL https://aclanthology.org/K17-3009.

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

1581, 2006. doi: 10.1198/016214506000000302. URL https://doi.org/10.1198/016214506000000302.

I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th International World Wide Web Conference (WWW)*, pages 111–120, 2008.

J. S. Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, 11(1):37–57, Mar 1985.

H. M. Wallach, D. M. Mimno, and A. McCallum. Rethinking lda: Why priors matter. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1973–1981. Curran Associates, Inc., 2009a. URL http://papers.nips.cc/paper/3854-rethinking-lda-why-priors-matter.pdf.

H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1105–1112, New York, NY, USA, 2009b. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553515. URL http://doi.acm.org/10.1145/1553374.1553515.

C. Wang and D. Blei. Decoupling sparsity and smoothness in the discrete hierarchical Dirichlet process. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams,

and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper/2009/file/3b8a614226a953a8cd9526fca6fe9ba5-Paper.pdf.

S. Williamson, C. Wang, K. A. Heller, and D. M. Blei. The ibp compound Dirichlet process and its application to focused topic modeling. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 1151–1158, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

J. Wintrode. Using latent topic features to improve binary classification of spoken documents. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5544–5547, 2011. doi: 10.1109/ICASSP.2011.5947615.

J. Wintrode. Leveraging locality for topic identification of conversational speech. In *Interspeech*, pages 1579–1583, 2013.

J. Wintrode and S. Khudanpur. Limited resource term detection for effective topic identification of speech. In *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 7118–7122, 2014.

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

BIBLIOGRAPHY

K. Zhai and J. Boyd-Graber. Online latent Dirichlet allocation with infinite vocabulary. In *Proceedings of the 30th Annual International Conference on Machine Learning*, 2013.