Corpus Statistics and Fine Tuning Latent Dirichlet Allocation for Transfer Learning: Dissertation Proposal

Tommy Jones*

Abstract

Language is one of the most information rich and abundant data sources available. Rigorous statistical study of linguistic phenomena can elevate the digital humanities, linguistic applications to computational social science, and statistics itself. As yet, statistical applications to language, whether in linguistics, computation, or otherwise, are largely ad-hoc. Performance gains in modeling have largely been due to two factors: fine tuning transfer learning and increasing the size and complexity of models. Due to the statistical nature of human language—governed by several power law phenomena—fine tuning transfer learning may be advantageous for corpus analyses, not just artificial intelligence applications. Yet state of the art "transformer" models are expensive and opaque. I propose we turn revisit Latent Dirichlet Allocation (LDA) for corpus statistics. As a parametric statistical model of a data generating process, it has the potential to be used in statistically rigorous ways to study language. And I have implemented an algorithm for fine tuning transfer learning using LDA. In this dissertation proposal, I state my case for "corpus statistics", a more statistically rigorous take on analyzing text data, review the literature around LDA, propose 3 studies, and 1 piece of software to fulfill the requirements of my dissertation.

Contents

1 Introduction and Motivation **3**

1.1 Corpus Statistics 3

1.2 Fine Tuning Transfer Learning 5

1.3 Summary of Proposed Research 7

*PhD Student, George Mason University Dept. of Computational and Data Sciences, tjones42@gmu.edu

2 Latent Dirichlet Allocation and Related Models **7**

2.1 Vector Space Model . . . 8

2.2 Latent Semantic Indexing . . . 8

2.3 Probabilistic Latent Semantic Indexing . . . 9

2.4 Latent Dirichlet Allocation . . . 9

2.5 Related Topic Models . . . 12

2.6 Text Embeddings and Distributional Semantics . . . 13

2.7 LDA for Corpus Statistics . . . 14

3 Fine Tuning Latent Dirichlet Allocation for Transfer Learning **16**

3.1 Related Work . . . 17

3.2 Proposed Contributions . . . 18

4 Studying the LDA-DGP **22**

4.1 Related Work . . . 23

4.2 Proposed Contributions . . . 27

5 Coefficient of Determination for Topic Models **28**

5.1 Related Work . . . 29

5.2 Preliminary Results and Proposed Contributions . . . 30

6 *tidylda*, an R Package **32**

6.1 Related Work . . . 32

6.2 Preliminary Results and Proposed Contributions . . . 35

7 Approach and Timeline **36**

8 Appendix 1: Projection Matrix for Probabilistic Embedding Models **38**

9 Appendix 2: Expected Term Frequency of the LDA-DGP **40**

10 Appendix 3: Reproduction of Table S3 from Shi et al., 2019 **44**

References **45**

1 Introduction and Motivation

Human language is one of the most information rich sources of data that exists. Language is literally the medium humans use to communicate information to each other. And in an increasingly digitally connected world, the amount of text available for analysis has exploded. Improvements in computing power and algorithmic advances have driven staggering progress in machine learning tasks for natural language, including machine translation, question answering, automatic summarization, information extraction and more.

Such advances have lead an increase in textual analysis in two relatively new, interdisciplinary fields—the "*digital humanities*" and "*computational social science*". The digital humanities represents a quantification of traditionally qualitative fields such as history, literature, communications, and the arts. Computational social science emphasizes computationally-intensive methods for such disciplines as economics, political science, psychology, etc. Examples of textual analyses from these two new fields include using text data include statistical modeling of language in the *Pennsylvania Gazette* from 1728 to 1800 [1], tracking the evolving use of the Greek word *kosmos* from 700 BC onward [2], using language from the Federal Reserve Board's public press releases to predict the Fed's non-public economic forecasts [3] and more. Applications to economics and social science were presented in recent Association for Computational Linguistics workshops [4] [5]

1.1 Corpus Statistics

The tools used for text analyses in these new fields derive from linguistics, computing, and statistics. Linguistics has the sub-field of *corpus linguistics*, the study of language as it appears in samples of "real world" text (corpora). Computing has the sub-field of *natural language processing* (NLP) which concerns itself with the interactions of people and computers. A particularly relevant sub-field of NLP is *distributional semantics*, which quantifies semantic similarities in language in terms of relative frequencies of linguistic items (such as words). Yet in statistics no named sub-field exists.

Statistics as a field is waking up to its role regarding linguistic data, however. The American Statistical Association (ASA) recently formed an interest group[1] for text analysis in 2019 [6]. A year later, the Joint

[1] I have been involved since the beginning and am the group's web master.

Statistics Meetings—the annual conference of the ASA and 11 other statistical associations—had over 20 sessions featuring text analysis research [7]. This isn't to say the interest group caused the volume of text analysis research. Note that many of the references in this proposal publish in statistics journals—such research is happening anyway. But only recently has there been effort to organize a community of practice for text analyses under a statistics umbrella.

Rigorous statistical study of linguistic phenomena can elevate the digital humanities, linguistic applications to computational social science, and statistics itself. As yet, statistical applications to language, whether in linguistics, computation, or otherwise, are largely ad-hoc. Save a handful of empirical laws, there is little statistical theory guiding the modeling of textual data. What theory does exist generally does not inform specification or use of statistical or machine learning models of text. Instead, the field has relied on increasingly complex models, requiring tremendous computational power, to drive these advances.

To envision the art of the possible, consider the state of statistical theory in linear regression. Linear regression—and its related statistical theory—dates back to the early 1800's from the works of Legendre [8], Gauss [9], and Galton [10]. We have formal statements of the assumptions required for valid statistical inference using linear regression. We have multitudes of diagnostic statistics to assess the degree these assumptions are violated in the data or with model specifications. There are a plethora of remediations to apply when key assumptions are violated so that valid statistical inference may still be made. And there are an abundance of statistical inferential methods built on top regression models used to detect structural breaks [11], calculate individual variable's contribution to the coefficient of determination [12], and on and on. We have been studying linear regression for a long time. As a result, we have an extremely powerful kit of statistical tools that are so easy to use that in 2020, we mostly take them for granted. Imagine what we could do if we brought this level of rigor to models and applications using language data in all of its abundance.

Again, statistics does not have a named sub-field dedicated to such study. Since naming a concept can give it power, I humbly submit "*corpus statistics*", so named because of similarity with corpus linguistics as the study of "samples" of real-world language in linguistics. Corpus linguistics uses statistics and other tools but its primary concern is linguistics itself. Corpus statistics may focus on "traditional" statistical concerns such as constructing appropriate random samples, quantifying uncertainty about claims learned from data, developing rigorous evaluation metrics for models, and so on where language is the topic of study. Corpus statistics can build off of work already begun in linguistics, machine learning, and complexity theory which concerns itself with the study of complex phenomena and power laws.

Language is saturated with the statistics of power laws. Two empirical laws of language, Zipf's law and

Heaps's law—covered in more detail later in this proposal—are two manifestations of power laws in language data. Power laws are extreme distributions, with significant mass in their tails. Power laws have extremely large variance. In fact, for many parameterizations, the second moment does not exist, making the variance effectively infinite [13]. Because of power laws in language, any finite sample (i.e., corpus) will miss key information.

Power laws in language might imply, then, that one needs external information for a thorough analysis of any one corpus. In fact, this is actually how humans learn from language! Consider the following thought experiment where one wants to learn about chemistry from an English-language textbook. Presumably, this person has good grasp on the English language already, having a vocabulary and covering subject matter greater than the book itself contains. This person then reads the textbook, learns about chemistry, and in the process updates and expands their knowledge of the English language.[2] So, in addition to traditional statistical concerns like representative samples and reasonable model specification, perhaps corpus statistics needs large base models of language to be updated for analyzing finite corpora.

1.2 Fine Tuning Transfer Learning

In fact such models are in wide use in machine learning. This is called the *fine tuning paradigm of transfer learning*. Transfer learning is when a model is developed for one task—on one data set—and then re-used for another task and data set. In the fine tuning paradigm, the base model is modified for the new task, allowed to update based on the new data, or both modified to the task and updated with new data.

Neural networks lend themselves naturally to fine tuning transfer learning. They are trained (or fit) using the iterative back propagation algorithm. Instead of initializing the model parameters—often called "weights"—at random, they are initialized at the same values they had in the base model. Then back propagation is resumed using the new data and all or some of the weights are allowed to update. This lets the analyst leverage information from a very large data set encoded in base model and adjust the parameters to fold in information in their new data set. Fine tuning transfer learning has been used for many years in deep learning models for computer vision, but it has recently gained widespread adoption in deep learning models for language.

Current state of the art natural language processing models belong to a class of deep neural networks called "transformers."[3] Famous examples of transformers include BERT [15], XLM-R [16], GPT-2 [17], GPT-3 [18],

[2]This sounds philosophically Bayesian to me. Yet I see no reason why one needs to limit their study to the use of Bayesian statistical models.

[3]Transformers get their name from the "transformer" sub-architecture for deep neural networks [14]. Technically, this is

and more. In terms of raw accuracy for benchmark task-specific objectives, transformers reign supreme in Natural Language Processing [19].

Transformers are extremely accurate on pre-defined natural language processing tasks, but they are not without problems. First, and most famously, these models are huge and expensive. BERT has 110 million parameters, XLM-R has 550 million parameters, and GPT-3 has a whopping 175 billion parameters.[4] These base models can cost from hundreds to tens-of-thousands of dollars in compute costs for a single run [20]. GPT-3 is rumored to have cost $4.6 million in compute [21]. These figures exclude trial and error runs inherent in any model development process.

Second, the data sets used to build these base models affect results, but we don't know what the "right" data set looks like. That models inherit biases from the data on which they are developed is not controversial. Yet some of the issues in these large language models are particularly jarring. They can encode—then reproduce—racist or otherwise extreme language [22]. And they may exclude language we wish was included, perhaps those of underrepresented groups. Yet the data sets used to construct transformers are so large, researchers cannot truly audit what they do or do not include. And as Timnit Gebru et al. point out, the investment required to construct large language models comes at an opportunity cost of learning how to strategically construct data sets without these downsides [20].[5]

Third, transformers are built for supervised tasks for artificial intelligence, not corpus analysis. Mostly these are sequence to sequence models used for question answering, machine translation, text generation, document summarization, and so on. While some of these tasks may be useful for corpus analysis, they really are distinct from a statistical analysis of a corpus.

Fourth, and most obviously, transformers are deep neural networks. Deep neural networks are spectacular in their flexibility and accuracy for many supervised learning tasks. Yet this flexibility and accuracy has come at the cost of complexity, often antithetical to understanding.

In sum, it seems that the fine-tuning approach of modeling would be beneficial to develop for corpus statistics. It reflects how we intuitively understand that humans use language to learn themselves. Yet we would need simpler, more transparent models upon which to apply and build on statistical theory. Transformers are too big, complex, and opaque for this task. Boyd-Graber and Mimno make this point explicitly on p. 116 of *Applications of Topic Models*, "Deep learning has a reputation for inscrutable parameters but state-of-the-art

distinct from fine tuning transfer learning. But as of this writing, the two approaches are used hand-in-hand for natural language processing models.

[4] If each parameter is stored as a 4 byte float, then GPT-3 is 700 Gb on disk, larger than most data sets.

[5] To me, this sounds like a task that the statistics community is well suited for, adapting sampling theory and statistical design for use in constructing data sets of language.

performance. One of the strengths of probabilistic models is their interpretability and grounded generative processes" [23].

1.3 Summary of Proposed Research

I propose reexamining a model that has become less popular in machine learning circles, Latent Dirichlet Allocation (LDA) [24]. Why? With the above comments in mind, LDA has some desirable properties. It models a data generating process which may be linked to the empirical laws of language. This property makes LDA, and related models, candidates for helping to develop a more robust statistical theory for modeling language. And while LDA may be less popular at the cutting edge of machine learning, it and its variants are still popular in fields such as computational social science [25] and the digital humanities [26].

To complete the requirements of my dissertation, I propose making four contributions—three studies and one software library. The first study introduces a method for fine tuning transfer learning applicable to MCMC algorithms for LDA. The second investigates the data generating process modeled by LDA, linking it to empirical language laws and using it for simulation studies concerned with model specification. The third study introduces a new (yet old) evaluation metric for topic models, a generalized coefficient of determination. Finally, this sort of research is not practical if people cannot use it. I am developing a software package for the R language [27] that draws on my research and a framework known as the "tidyverse" [28] to make a principled, flexible, performant, and user-friendly interface for training and using LDA models.

The remainder of this proposal is organized as follows: Section 2 reviews the foundations of LDA and related models. Section 3 outlines my proposed study of transfer learning for LDA. Section 4 outlines my proposed study of the LDA data generation process (LDA-DGP). Section 5 outlines my proposed study developing a coefficient of determination for topic models. Section 6 introduces *tidylda* an in-development R package for LDA. Finally section 7 offers a timeline for completing the proposed dissertation.

2 Latent Dirichlet Allocation and Related Models

In this section I summarize LDA, select models that came before, and select models that came after. Then I make a case for why LDA continues to be a worthy model of study in machine learning and why it is an advantageous place to start for corpus statistics.

2.1 Vector Space Model

Most modern approaches to language modeling can trace their origin to the vector space model [29]. The vector space model represents contexts (usually documents) in multidimensional space whose coordinates are given by the frequencies of words within that context. These frequencies may be explicit counts, or they may be re-weighted. The vector space model makes the *bag of words* assumption. Within a context, word order and proximity have no meaning. When one says a document is a "bag of words", they mean that word order is discarded and the document is only the relative frequencies of words within it.

More formally, let $\boldsymbol{X}$ be a $D \times V$ matrix of contexts. Each row represents a context, and each column a word.[6] The key to the vector space model is that if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are close in the vector space given by $\boldsymbol{X}$, then the contexts represented by $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ must be semantically similar.

The key limitation of the vector space model is that $\boldsymbol{X}$ is large and sparse. This leads to the "curse of dimensionality" where meaningful comparisons can be different because most dimensions have little signal [30]. The bag of words assumption also disregards the use of synonyms or the fact that the same words can have multiple meanings—known as "polysemy".

2.2 Latent Semantic Indexing

Latent Semantic Indexing (LSI) reduces the dimensions of $\boldsymbol{X}$ through a single value decomposition (SVD) [31]. Since $\boldsymbol{X}$ is large and sparse, one can approximate it by $\boldsymbol{X}_{(K)}$ which is of rank K and corresponds to the K largest eigenvalues of $\boldsymbol{X}$. This projects the data matrix $\boldsymbol{X}$ into a K-dimensional Euclidean "semantic" space. Formally,

$$\boldsymbol{X} \approx \boldsymbol{X}_{(K)} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{1}$$

$\boldsymbol{U}$ and $\boldsymbol{V}^T$ are orthonormal matrices and $\boldsymbol{\Sigma}$ is a diagonal matrix of singular values. New documents, $\boldsymbol{X}'$ may be embedded by right multiplying them by the projection matrix, $\boldsymbol{\Lambda} = [\boldsymbol{\Sigma}\boldsymbol{V}^T]^{-1}$.

A key limitation LSI brings is that there is no obvious way to choose K, the embedding dimension. This problem plagues nearly all models that follow it.

[6] While there are distinct differences in the definitions of "word" and "token", for the purposes of this work I will use the two terms interchangeably for simplicity.

2.3 Probabilistic Latent Semantic Indexing

Probabilistic Latent Semantic Indexing (pLSI) brings a probabilistic approach to LSI [32]. pLSI models a generative process for $\boldsymbol{X}$ where $\boldsymbol{X}$ is explicitly a matrix of integer counts of word occurrences. Assuming there are D contexts, K topics, V unique words, N total words and Nd words in the d-th document, the process is as follows. For each word, n, in context d:

1. Draw topic $z_{d,n}$ from Multinomial($\boldsymbol{\theta}_d$)
2. Draw word $w_{d,n}$ from Multinomial($\boldsymbol{\beta}_{z_{d,n}}$)
3. Repeat 1. and 2. N_d times.

From the above, it's clear that $P(z_k|d) = \theta_{d,k}$ and $P(w_v|z_k) = \beta_{k,v}$.

pLSI is fit using the EM algorithm to find the values of parameters in $\boldsymbol{\Theta}$ and $\boldsymbol{B}$ that maximize the joint likelihood of $\boldsymbol{X}$. The likelihood of a single word in a single document is $P(w_v, d) = P(d)P(w_v|d)$. Taking $P(d) = \frac{N_d}{N}$ and noting that $P(w_v|d) = \sum_{k=1}^{K} \theta_{d,k} \cdot \beta_{k,v}$ we can derive the joint likelihood

$$\mathcal{L} = \prod_{d=1}^{D} \prod_{v=1}^{V} \left(\frac{N_d}{N} \sum_{k=1}^{K} \theta_{d,k} \cdot \beta_{k,v} \right) \tag{2}$$

It is stated that pLSI cannot be extended to unseen contexts. "It is not clear how to assign probability to a document outside of the training set" [24]. This does not make sense to me. One can use Bayes's rule to derive a projection matrix, $\boldsymbol{\Lambda}$, to embed new contexts into the probability space fit by pLSI. A derivation is in Appendix 1. pLSI is alleged to habitually over fit its training data [33]. And, as with LSI, there's no clear guidance for selecting the number of topics, K.

2.4 Latent Dirichlet Allocation

LDA is a Bayesian version of pLSI. It was developed by David Blei, Andrew Ng, and Michael Jordan to address perceived shortcomings of pLSI [24]. LDA adds Dirichlet priors to the parameters $\boldsymbol{\Theta}$ and $\boldsymbol{B}$. This modifies the data generating process as

1. Generate $\boldsymbol{B}$ by sampling K topics $\boldsymbol{\beta}_k \sim \text{Dirichlet}(\boldsymbol{\eta}), \forall k \in \{1, 2, ..., K\}$
2. Generate $\boldsymbol{\Theta}$ by sampling D documents $\boldsymbol{\theta}_d \sim \text{Dirichlet}(\boldsymbol{\alpha}), \forall d \in \{1, 2, ..., D\}$
3. Then for each document, d

1. Draw topic $z_{d,n}$ from Multinomial($\boldsymbol{\theta}_d$)
2. Draw word $w_{d,n}$ from Multinomial($\boldsymbol{\beta}_{z_{d,n}}$)
3. Repeat 1. and 2. N_d times.

For ease of notation, I refer to the above process as the LDA-DGP throughout this proposal.[7]

The above process has a joint posterior of

$$P(\mathbf{W}, \mathbf{Z}, \boldsymbol{\Theta}, \boldsymbol{B}|\boldsymbol{\alpha}, \boldsymbol{\eta}) \propto \left[\prod_{d=1}^{D}\prod_{n=1}^{n_d} P(w_{d,n}|\boldsymbol{\beta}_{z_{d,n}})P(z_{d,n}|\boldsymbol{\theta}_d)P(\boldsymbol{\theta}_d|\boldsymbol{\alpha})\right]\left[\prod_{k=1}^{K} P(\boldsymbol{\beta}_k|\boldsymbol{\eta})\right] \tag{3}$$

The above posterior does not have an analytical closed form when formulas for the multinomial and Dirichlet distributions are plugged in. So a variety of Bayesian estimation methods have been employed to estimate the model. Blei et. al used variational expectation maximization (VEM). Shortly thereafter, Griffiths and Steyvers developed a collapsed Gibbs sampler for LDA [34]. This sampler is "collapsed" in that the parameters of interest—$\boldsymbol{\Theta}$ and $\boldsymbol{B}$—are integrated out for faster computation. After iteration is complete, $\boldsymbol{\Theta}$ and $\boldsymbol{B}$ can be easily calculated. Others have since developed many other MCMC algorithms, discussed more in the next section.

The priors $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ may be either asymmetric or symmetric. Asymmetric priors are those where the vector hyper parameter (e.g., $\boldsymbol{\alpha}$) has different values in each slot. Symmetric priors are those where the vector hyper parameter has the same value in each slot. In the latter case, researchers will often represent the hyper parameter as a scalar rather than a vector. The magnitudes of $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ affect the average concentration of topics within documents and words within topics, respectively. This is a property of the Dirichlet Distribution.

2.4.1 Persistent Issues with LDA (and Most Other Topic Models)

In spite of its being almost 20 years old, LDA has several persistent issues. Many of these are shared by the models that came before and after. Given its continued popularity (and the primary focus of my proposed research) it is worth some detailed discussion.

[7] The original specification of the LDA-DGP [24] specified that each document's length is drawn from a Poisson random variable. This specification has been dropped from most subsequent work on LDA. Likely this is because in practical applications document lengths are given by the data and do not need to be modeled. It's just as well since specifying a distribution of document lengths for any real-world corpus is likely overly prescriptive.

There is no generally accepted method for choosing hyper parameters—i.e., model specification—for an LDA model [35]. Anecdotally (and backed by a quick search of Stack Overflow) most concern is in choosing K, the number of topics. There has been less concern with choosing the prior parameters, $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$. Yet, I have some preliminary evidence that all three parameters need to work in concert, at least to get a semantically coherent model [36]. A rigorous examination of how all three hyper parameters interact in the LDA-DGP is warranted.

There are two philosophies for selecting K, the number of topics or dimensions of embedding for a topic model. The first is to select a value—perhaps including different values for $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ as well—to optimize a metric. This metric may be the log likelihood [34], perplexity [35], a coherence metric[8] [41], or something else [42]. Chen et al. use a fully-Bayesian approach and put a prior on the number of topics [43]. The second philosophy holds that optimizing for such metrics leads to topics that are too specific for human interpretability [44]. Yet, my experience has not been consistent with Chang and Boyd-Graber. They constructed a topic model on a general news corpus, then had people using Amazon's Mechanical Turk evaluate topic quality. The subjects covered by the model were broad and the audience were not subject matter experts. Meanwhile, I started out modeling scientific research documents for program directors in federal science agencies. These experts were adept at interpreting highly specific topics in their domain.

There has been less attention paid to selecting $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$. A notable—and influential—exception is Wallach, Mimno, and McCallum's *Rethinking LDA: Why Priors Matter*. They find that $\boldsymbol{\alpha}$ should be asymmetric and that $\boldsymbol{\eta}$ should be symmetric and small [45]. Yet this is counter to a proof I wrote in 2014—included in Appendix 2. I find that the expected term frequency of a collection of documents generated by the LDA-DGP is proportional to $\boldsymbol{\eta}$. Given Zipf's law [46]—described in more detail in Section 4—it would appear that a symmetric $\boldsymbol{\eta}$ constitutes a prior that is impossible in any real corpus. More recently, George and Doss explore an MCMC approach to help select $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ [47]. Unfortunately, they too assume a symmetric $\boldsymbol{\eta}$.

There is still no consensus on how to evaluate topic models. Shi et al. divide approaches to evaluation into three categories: manual inspection, intrinsic evaluation, and extrinsic evaluation [48]. Manual inspection—such as the "intruder test" used in [44]—is subjective and labor intensive. Intrinsic evaluation evaluates the model directly against the data on which it was developed. This is measured in goodness of fit metrics such as log likelihood and perplexity, with a coherence metric[9], or other metrics—such as comparing to baseline distributions [49]. Intrinsic evaluation may be measured in-sample or out-of-sample. Extrinsic methods evaluate the performance of topic models against some external task, such as identifying document class.

[8]Not all coherence [37] metrics are created equal. Evaluation of several coherence metrics has found considerable variation in correlation with human judgement [38], [39]. In particular, the "UMASS" metric [40] is widely used, but does not correlate highly with human judgement [38], [39].

[9]Coherence metrics are intrinsic measures that are designed to approximate rigorous manual inspection.

In both his dissertation [33] and related publication [48], Shi argues that comparing topic models to "gold standard" synthetic data is a more principled approach. I agree, yet seek to link the process that generated such "gold standard" data—the LDA-DGP—to both laws of language and intrinsic methods to reduce trial and error when developing LDA models.

Algorithms to fit LDA models have scaleability issues. Gibbs is a naturally sequential algorithm. When the data set is large, it can be prohibitively slow. Gibbs scales quadratically with the number of documents, topics, vocabulary size, and total number of tokens. Newman et al. developed a distributed "approximate" Gibbs sampler for LDA [50]. It is implemented in MALLET and the in-development version of *tidylda*. Yet quality of the model—by any intrinsic evaluation metric—decreases when used. This is due to the approximation voiding theoretical guarantees of convergence inherent to MCMC samplers, a point that Newman et al. concede [50]. VEM has more-easily distributed computations, yet anecdotally VEM gives less coherent topics than MCMC algorithms, especially on smaller data sets [51]. A host of other MCMC algorithms have been developed to try to scale LDA to very large corpora [52], [53], [54], [55]. One approach uses a variational autoencoder to approximate VEM in a neural network [56]. Even so, the most approachable implementations of LDA still use VEM or collapsed Gibbs.

The LDA-DGP makes strong independence assumptions. Human language is said to have a property called "burstiness" [57]. Burstiness is the idea that words cannot be independent draws from a distribution since they occur in clusters. i.e., Seeing a word once, greatly increases the probability you will see it again. Mimno and Blei developed posterior predictive checks for LDA to detect the degree to which burstiness affects LDA [58].

2.5 Related Topic Models

Researchers have developed many topic models intended to improve upon LDA. This section covers some notable examples.

Teh et al. used a hierarchical Dirichlet process (HDP) to determine the number of topics automatically [59]. They show that HDP has performance advantages over LDA in terms of perplexity. Yet Chen et al. point out that HDP is both computationally intensive, that it is actually a fundamentally different model from LDA, and using it to select the number of topics is not well defined [43].

Dynamic topic models (DTM) add a time series component to topic modeling [60]. By incorporating date of publication, DTMs allow the linguistic mixture of topics to change dynamically over time. Consider an intuitive example; the amount of writing about a given topic changes over time, but also *how* writers write

about that topic changes as well. DTMs require explicit metadata on when a context appeared.

Topic models are often used to embed contexts into a vector space to aid a supervised task downstream. Supervised LDA (sLDA) incorporates this outcome in the modeling process [61]. McAuliffe and Blei find that sLDA improves accuracy of the predictive task and gives improved topic quality over a two-step of LDA then using a separate supervised algorithm.

Two closely-related algorithms are the correlated topic model (CTM) [62] and structural topic model (STM) [63]. CTMs modify LDA by placing a log-normal prior on $\mathbf{\Theta}$ to allow modeling the correlations between documents. STMs extend CTMs by allowing the user to declare context-level co-variates, rather than simply trying to estimate them. If no such co-variates are provided, then STM collapses to CTM [63].

The above models—and many more—encompass a large landscape, but they are not so different. Each of these models effectively embeds their contexts into a probability space. (See Section 2.7.) As a result, I hypothesize that much of my proposed research should extend—or can be modified to extend—to these models. Perhaps LDA is not the "best" model long term. But its simplicity and widespread use make it a good place to start.

2.6 Text Embeddings and Distributional Semantics

Word embeddings were developed in parallel with topic models, beginning with the neural language model of Bengio et al. [64]. The idea is to represent words as distributions embedded into a vector space such that proximity corresponds to semantic similarity. The idea of distance in vector space corresponding to semantic distance is known as *the distributional hypothesis* and forms the basis of distributional semantics [65, Ch. 14]. The distributional hypothesis states that "you shall know a word by the company it keeps" [66]. In other words, a word's meaning may be inferred from the context in which it appears.[10] Since 2003, word embeddings have been extended to cover larger linguistic units such as sentences and documents [67] and have taken on the more general moniker "text embeddings". Notable models include word2vec [68], GloVe [69], ELMo [70], and more.

One popular method for constructing this "context" is a term co-occurrence matrix of *skip grams*. Skip grams count the number of times each word in the vocabulary occurs in a window around a target word [71]. For example, a skip gram window of 5 counts the number of times each word appears within five words to the left or five words to the right of the target word. The result is a symmetric $V \times V$ matrix of integers,

[10] I have heretofore been using the general word "context" instead of "document" to discuss topic models. This is intentionally related to text embeddings and discussed more in Section 2.7, *LDA for Corpus Statistics*.

with each row and column representing a word. Levy and Goldberg show that neural word embeddings may be viewed as a factorization of this matrix [72].

Text embedding research brings exciting possibilities to distributional semantics. Researchers have found that comparisons and compositions of embeddings may be linked to semantic meaning. For example, Mikolov et al. found similarities between the vectors representing countries and their capitals [68]. Compositions between word vectors may also capture semantic meaning as demonstrated by the oft-used "king $-$ man $+$ woman $\approx$ queen" example [73]. And alignment between vector spaces across languages promises the ability to compare meaning of words and phrases across languages [74].

As with topic models, there is no principled way to choose K, the number of embedding dimensions. Anecdotally, this seems to concern the distributional semantics community far less than the topic modeling community. Yet, as I describe below, I believe the two communities are actually working on the same class of models, in spite of their disjoint lineage.

2.7 LDA for Corpus Statistics

Latent Dirichlet Allocation has lost favor in machine learning circles. A colleague recently asked me without any irony, "who still uses LDA?!?!" This sentiment is unfortunately widespread in the machine learning community. Jacob Eisenstein—a research scientist at Google—recently published *Introduction to Natural Language Processing*—one of the first comprehensive textbooks on the subject since deep learning models came to dominate in NLP [65]. He does not mention LDA at all in the chapter on distributional semantics (pages 309–332) and only mentions it in a footnote in the chapter on discourse (on page 358).

Yet LDA is still in widespread use in the digital humanities and computational social science. Machine learning may have moved on, but applications of LDA are widespread in these less explicitly technical disciplines. This points to a need to make LDA more accessible [23, Ch. 10.2]. Accessibility means computational tools that are easy to use as well as a deeper understanding of *how* these models may be deployed to answer questions these disciplines have.

Moreover, I argue that the distinction between "topic models"—of which LDA is a member—and "text embeddings" made in the machine learning community is meaningless. Viewed through the lens of topology, LDA and newer text embeddings belong in the same class of models. In topology, an embedding, f, is a mapping between topological spaces. $f : \boldsymbol{X} \rightarrow \boldsymbol{Y}$ means that f maps a point in topological space $\boldsymbol{X}$ to a point in topological space $\boldsymbol{Y}$. From this perspective, LDA maps $\boldsymbol{X}$—points in a V-dimensional integer space—to $\boldsymbol{\Theta}$—points in a K-dimensional probability space. Newer text embeddings map from $\boldsymbol{X}$ to (usually)

Euclidean space $\boldsymbol{Y}$.[11]

LDA is also criticized for its over reliance of the bag of words assumption. This conflates data pre-processing with the model itself. For this reason, I prefer to refer to the rows of $\boldsymbol{X}$—the data being modeled—as "contexts" rather than "documents". A context may be a whole document, a chapter, a paragraph, a sentence, etc. Or a context may incorporate proximity or word order. For example a context could be the count of times each word in a corpus appears around a target word, as with skip-grams. A context could also be a count of each word appearing after or before a target word.[12] It's true that within a context, the bag-of-words assumption still holds. So, LDA cannot be a full sequence to sequence model. However, the way a context is constructed may encode proximity and order. And that construction affects the interpretations of probabilities resulting from the model. I have done this explicitly with LDA when writing the vignettes for a previously-released R package, *textmineR* [75]. Just this year, Adji Deng co-authored a paper with David Blei constructing the "embedding topic model" in a similar fashion [42].[13] Panigrahi et al. use LDA for word embeddings and find evidence that the "topic" dimensions encode different word senses, potentially addressing polysemy [76].

I also argue that LDA and other probabilistic topic models have an advantage *because* they embed into a probability space. As we know from traditional statistics, probability spaces are well-suited to quantify uncertainty around claims and statistical inferences made from data—with or without linguistic origins. And while I have no plans to address this in my dissertation research, I hypothesize that embedding to probability spaces may aid tasks in distributional semantics such as uncovering analogies and cross-lingual embedding alignment. Probability spaces have well-defined relationships, transformations, and methods for composition. Euclidean spaces have fewer constraints on operations within them, leading to greater researcher degrees of freedom.

With this in mind, I believe that LDA is a good candidate of study for corpus statistics for three reasons. First, LDA is a parametric Bayesian statistical model, nothing more. This allows for uncertainty quantification, model diagnostics, etc. in line with established statistical practices. Second, as stated above, LDA embeds into probability spaces with the benefits they bring. And lastly, LDA is a generative model of language. This allows us to use simulation studies of the LDA-DGP to help develop methods to help build models and diagnose model misspecification, as described in section 4 below.

[11] We can extend this logic to encompass Transformers as well. A multil layer neural network may be viewed through this lens as a collection of embeddings $\{f_0, f_1, ..., f_O\}$ such that $f_0 : \boldsymbol{X} \rightarrow \boldsymbol{H_1}$, $f_1 : \boldsymbol{H_1} \rightarrow \boldsymbol{H_2}$, and so on until $f_{o-1} : \boldsymbol{H}_{o-1} \rightarrow \boldsymbol{O}$. $\boldsymbol{X}$ is the input layer; $\boldsymbol{H}_i$ are hidden layers; and $\boldsymbol{O}$ is the output layer. One might then consider relationship between contexts at any layer $\boldsymbol{H}_i$ or $\boldsymbol{O}$.

[12] I'd call these lead-grams and lag-grams, respectively.

[13] I published the vignette doing this with LDA in 2017 and had made the connection years earlier. Coulda shoulda woulda published it then. But if I'm going to get scooped in topic modeling, I'm glad it was on a paper co-authored with David Blei.

LDA is a good candidate for corpus statistics, but corpus statistics can also be good for LDA. In my opinion, the tension over intrinsic evaluation versus human interpretability—as highlighted by Chang et al. [44]—is premature. LDA has an identification problem. If one cannot tell if a model is pathologically misspecified, how can one rely on any interpretation of it? If a linear regression model's residuals are clearly not Gaussian distributed with mean zero, one does not attempt to interpret its coefficients. We should expect the same from LDA and related models.

For LDA, corpus statistics should focus on three tasks: detect pathological model misspecification[14] (e.g., choice of K, $\boldsymbol{\alpha}$, and $\boldsymbol{\eta}$), develop metrics to aid researchers in justifying a model's specification, and quantifying uncertainty around claims a researcher might want to make using a model. Relating empirical laws of language to the LDA-DGP is a first step for the former two tasks. The latter task relates closely to interpretation and depends on a model being statistically valid.

3 Fine Tuning Latent Dirichlet Allocation for Transfer Learning

As stated in Section 1, properties of language make the fine tuning paradigm of transfer learning attractive for statistical analyses of corpora. The pervasiveness of power laws in language—the most famous example of which is Zipf's law [46]—mean that we can expect just about any corpus of language to not contain information relevant to the analysis. (i.e., Linguistic information necessary for understanding the corpus of study would be contained in a super set of language around—but not contained in—said corpus.) Intuitively, humans approach corpus analytics in a way that on its surface appears consistent with fine tuning transfer learning. Humans have general competence in a language before consulting a corpus to learn a new subject—or even just to have human analysis of the corpus. Also as stated in Section 1, LDA has attractive properties as a model for statistical analyses of corpora. Its very nature allows us to use probability theory to guide model specification and quantify uncertainty around claims made with the model.

I have developed an algorithm for fine tuning transfer learning with LDA models. The algorithm is currently implemented in a collapsed Gibbs sampler, but should generalize to any MCMC sampler for LDA. This approach enables 5 related use cases with LDA models.

1. Pre-training a base model on a large corpus and then fine-tuning it on a smaller corpus for specific analyses,

[14] One might argue that humans do not write using the LDA-DGP and thus no "correct" specification exists. They would be right! Yet, "this model is not right" is a much narrower statement than "this model is right". To use linear regression as an example again, having Gaussian distributed residuals does not mean that one specified the "right" model. But having non-Gaussian residuals means the model is wrong.

2. Time-series analyses of corpora where both the topic prevalence and distribution of words within topics changes over time,
3. Developing models on streaming corpora, where new documents[15] are continually being added,
4. Allowing subject matter experts to seed relationships between words to topics prior to model development, and
5. Allowing a researcher to begin training a model, stop prematurely to inspect the posterior, then resume training almost exactly where they left off if, for example, the model has not yet converged.

3.1 Related Work

Related work from LDA and other probabilistic topic models falls into three categories. The first category contains topic models that explicitly model a topic's evolution over time [60], [77]. These models differ from transfer learning in that time is explicitly part of the model, rather than being updated post-hoc. The second category contains models that allow external information to guide the development of topics. External information may be in the form of supervised outcomes [61] [78] [79], seeded by model structure [80], seeded in the prior [81], or constructed interactively with subject matter experts [82]. The third category contains models designed for on-line training of streaming document collections [83] [84].

My algorithm may be viewed as an extension of AlSumait, Barbará, and Domeniconi [83]. Their approach encodes the topics from the previous time slices into the prior, $\boldsymbol{\eta}$, for the next time slice. More formally

$$\boldsymbol{\eta}_k^{(t)} = \boldsymbol{\omega}^{(\delta)} \cdot \boldsymbol{\beta}_k^{(t-1)} \tag{4}$$

where $\boldsymbol{\eta}_k^{(t)}$ is the prior for the k-th topic for the current model, $\boldsymbol{\omega}^{(t)}$ is a vector of weights with one entry for the previous δ models, and $\boldsymbol{\beta}_k^{(t-1)}$ is a *matrix* whose rows correspond to the posterior of the k-th topic for the previous δ models. In essence, $\boldsymbol{\eta}_k^{(t)}$ is a weighted linear combination of the last δ posterior distributions for the k-th topic. The weights sum to 1 such that $\sum_{j=t-\delta-1}^{t-1} \omega_j = 1$. An implication of this is that $\boldsymbol{\eta}^{(t)}$ is a matrix, not a vector as in the vanilla version of LDA. And by including information from the previous δ periods—as opposed to just the last period—this implementation loses the Markov property of stochastic processes.

[15] Here, I do mean documents. Yet there is no barrier to researchers using them to re-compute contexts and updating their models that way.

3.2 Proposed Contributions

The fine tuning paradigm of transfer learning requires two mechanisms: a mechanism to regulate between the base model's influence and tuning data set's influence over the resulting model and a mechanism to initialize "weights" where the base model left off—rather than at random. My algorithm addresses both mechanisms and two more: the ability to dynamically add new topics not in the base model and a method to add additional vocabulary words.

3.2.1 Brief Description of My Algorithm

3.2.1.1 Mechanism to regulate influence between base model and data

I use a very similar mechanism as AlSumait, Barbará, and Domeniconi's [83] to encode prior information and tune between the base model's and current data set's influence. The prior for words over topics, $\boldsymbol{\eta}$, is a the base model's $\boldsymbol{B}$, weighted to control its influence. Formally

$$\boldsymbol{\eta}^{(t)} = \boldsymbol{\omega}^{(t)} \odot \boldsymbol{B}^{(t-1)} \tag{5}$$

This specification is similar to AlSumait et al.'s—in that $\boldsymbol{\eta}^{(t)}$ is a weighted matrix—but it is not identical. Rather than encoding a linear combination of past posteriors into the prior, it encodes only the previous run's posterior. My specification is simpler and retains the Markov property. Yet if one were to daisy chain many models together—as AlSumait et al. do—the previous $t - \delta - 1$ models still influence model t through the posterior of model $t - 1$.[16]

The k-th entry of $\boldsymbol{\omega}^{(t)}$ is a weight that controls the concentration of topic k based on the previous model. $\boldsymbol{\omega}^{(t)}$ is set a-priori by the researcher. This allows the researcher to give certain topics more or less weight in development of the fine tuned model. The exact influence tuned by $\boldsymbol{\omega}^{(t)}$ is something I still need to determine.[17]

This mechanism is also how one can seed expert knowledge into a model. In this case, there is no base model. But if one wants to encode lexical information in g topics, they can do so in g rows of $\boldsymbol{\eta}$; the remaining $K - g$ rows remain identical, a default prior for other topics.

[16] I've not done so, but I hypothesize one could derive equivalence or near equivalence between the two approaches.

[17] Conjugate priors can be interpreted as adding data. My intuition is that there may be some $\boldsymbol{\omega}^{(t)*}$ that will weight the prior proportionally to the volume of data used to train the base model. This value may be useful as a default as well as providing context for a researcher who wants to choose their own weight.

3.2.1.2 Mechanism to initialize where the base model left off

When fine tuning a pre-trained neural network, one initializes the weights of a model to be the values from the base model, rather than initializing at random. LDA does not have "weights" but MCMC samplers have analogues, matrices of counts: $\boldsymbol{Cd}$ and $\boldsymbol{Cv}$. These count the number of times each topic was sampled in each context and for each word.

At each iteration (i), the sampler loops over each context, d, and each instance, n, of each word in that context, $w_{d,n}$. For each word, a topic is sampled according to the below probability

$$P(z_{d,n}^{(i)} = k) \propto \frac{Cv_{k,v}^{(i)} + \eta_{k,v}}{\sum_{v=1}^{V} Cv_{k,v}^{(i)} + \eta_{k,v}} \cdot \frac{Cd_{d,k}^{(i)} + \alpha_k}{\left(\sum_{k=1}^{K} Cd_{d,k}^{(i)} + \alpha_k\right) - 1} \tag{6}$$

Once sampled, $\boldsymbol{Cd}$ and $\boldsymbol{Cv}$ are re-calculated reflecting the current number of times each topic has been sampled for each context and word. At the end of I iterations, the final posteriors[18] for $\boldsymbol{\Theta}$ and $\boldsymbol{B}$ are calculated with

$$E[\theta_{d,k}] = \frac{Cd_{d,k}^{(I)} + \alpha_k}{\sum_{k=1}^{K} Cd_{d,k}^{(I)} + \alpha_k} \tag{7}$$

$$E[\beta_{k,v}] = \frac{Cv_{k,v}^{(I)} + \eta_{k,v}}{\sum_{v=1}^{V} Cv_{k,v}^{(I)} + \eta_{k,v}} \tag{8}$$

My algorithm allocates $\boldsymbol{Cd}^{(t)}$ and $\boldsymbol{Cv}^{(t)}$ in proportion to $\boldsymbol{Cd}^{(t-1)}$ and $\boldsymbol{Cv}^{(t-1)}$. In standard training for LDA, $\boldsymbol{Cd}$ and $\boldsymbol{Cv}$ are initialized at random. For transfer learning, one must initialize $\boldsymbol{Cd}^{(t)}$ and $\boldsymbol{Cv}^{(t)}$ to the same state as $\boldsymbol{Cd}^{(t-1)}$ and $\boldsymbol{Cv}^{(t-1)}$ respectively. The challenge lies in the fact that $\sum_{d,k} \boldsymbol{Cd}_{d,k}^{(t-1)} = \sum_{k,v} \boldsymbol{Cv}_{k,v}^{(t-1)} = \sum_{d,v} \boldsymbol{X}_{d,v}^{(t-1)}$ yet $\sum_{d,v} \boldsymbol{X}_{d,v}^{(t-1)} \neq \sum_{d,v} \boldsymbol{X}_{d,v}^{(t)}$. My algorithm[19] addresses this constraint as described below:

1. Project $\hat{\boldsymbol{\Theta}}^{(t)}$ by using $\boldsymbol{\Lambda}^{(t-1)}$ as described in Appendix 1,
2. Adding new vocabulary words from $\boldsymbol{X}^{(t)}$ by appending them to $\boldsymbol{B}^{(t-1)}$ and allocating uniform mass for these new terms proportional to the median of $\boldsymbol{B}^{(t-1)}$ and then normalizing the rows of this new $\hat{\boldsymbol{B}}^{(t)}$,

[18] Technically, these are expected values as the true posterior distributions are given by $\boldsymbol{\beta}_k \sim \text{Dirichlet}(\boldsymbol{Cv}_k + \boldsymbol{\eta}_k)$ and $\boldsymbol{\theta}_d \sim \text{Dirichlet}(\boldsymbol{Cd}_d + \boldsymbol{\alpha})$

[19] Source code for the algorithm may be found at https://github.com/TommyJones/tidylda/blob/main/R/refit.tidylda.R

3. Solving for the approximate initial values of $\boldsymbol{Cd}^{(t)}$ and $\boldsymbol{Cv}^{(t)}$ using equations (7) and (8),
4. Rounding and re-allocating counts in $\boldsymbol{Cd}^{(t)}$ and $\boldsymbol{Cv}^{(t)}$ so that they are both integer matrices and ensure
    - the row sums of $\boldsymbol{Cd}^{(t)}$ equal the row sums of $\boldsymbol{X}^{(t)}$
    - the column sums of $\boldsymbol{Cd}^{(t)}$ equal the row sums of $\boldsymbol{Cv}^{(t)}$

There is more detail to be described. For brevity in this document, I will save that detail for the final research paper.

3.2.1.3 Mechanism to dynamically add topics

New topics are added by appending rows to $\boldsymbol{\eta}^{(t)}$. For example if $\boldsymbol{\eta}^{(t-1)}$ has K topics and I wish to add 3 topics during fine tuning, $\boldsymbol{\eta}^{(t)}$ has $K+3$ rows. Similar to seeding topics, these new rows are initialized with a standard prior, proportional to the column means of $\boldsymbol{B}^{(t-1)}$. If a researcher wants to retire topics, they may remove those rows from $\boldsymbol{\eta}^{(t-1)}$ before fine tuning.

3.2.1.4 Mechanism to add vocabulary

New vocabulary is added by appending columns to $\boldsymbol{\eta}^{(t)}$. I initialize values for new words as the median of $\boldsymbol{\eta}^{(t-1)}$, a flat prior. This, admittedly, is not consistent with Zipf's law. I am open to exploring new heuristics.

3.2.2 What Needs to Be Done

Below are the tasks I propose putting into my dissertation for this study.

1. A more thorough literature review,
2. A formal statement of my transfer learning algorithm,
3. Derivation of a new log likelihood for an LDA model where $\boldsymbol{\eta}$ is a matrix, rather than a vector,
4. Analytical exploration of the effect $\boldsymbol{\omega}^{(t)}$ has for tuning trade off between the base model and new data,
5. A simulation experiment to isolate the effects of
    - Using $\boldsymbol{\eta}^{(t)} = \boldsymbol{\omega}^{(t)} \odot \boldsymbol{B}^{(t-1)}$ as a prior and initializing $\boldsymbol{Cd}^{(t)}$ and $\boldsymbol{Cv}^{(t)}$ at random,
    - Using a standard $\boldsymbol{\eta}$ prior and initializing $\boldsymbol{Cd}^{(t)}$ and $\boldsymbol{Cv}^{(t)}$ based on $\boldsymbol{Cd}^{(t-1)}$ and $\boldsymbol{Cv}^{(t-1)}$
    - Using both $\boldsymbol{\eta}^{(t)} = \boldsymbol{\omega}^{(t)} \odot \boldsymbol{B}^{(t-1)}$ as a prior and initializing $\boldsymbol{Cd}^{(t)}$ and $\boldsymbol{Cv}^{(t)}$ based on $\boldsymbol{Cd}^{(t-1)}$ and $\boldsymbol{Cv}^{(t-1)}$

- Exploration of different values of $\boldsymbol{\omega}^{(t)}$

6. An experiment using real data (corpora TBD)

    a. Choose a base data set (for example, English language Wikipedia)
    b. Fine tune on a specific subject area (for example, abstracts of research presented at the ACL conference)
    c. Analysis of topic presence, absence, and linguistic change in the latter data set
    d. Daisy chaining multiple years together and letting topics evolve in a time series (e.g., multiple years of abstracts presented at the ACL conference)

3.2.3 Prerequisites and Extensions

Acknowledging that I have already implemented the algorithm in question[20], I have identified two prerequisites necessary for me to accomplish the proposed research, above. First, the Gibbs algorithm—where this is already implemented—is too slow to scale to large corpora. I want to implement the *WarpLDA* algorithm [55]—which is natively parallel while retaining Metropolis Hastings convergence guarantees—so that I may train a base model on a large corpus, such as the English language Wikipedia. The full promise of pre-train then fine-tune cannot be realized if my implementation cannot scale to use corpora on the scale of state of the art NLP models. Second, I would like to complete my study of the LDA-DGP so that (a) my simulation study is justified—producing simulations that reproduce empirical laws of language—and (b) in the hopes that it will inform choices of hyper parameter values so that the study using real data has models that are reasonably well-specified.

I am considering an extension to study how to quantify uncertainty in claims one might make with a fine-tuned model. A fine tuned model chains together two data sets—with two different sample sizes—with a base model that introduces error. When quantifying uncertainty to a claim—e.g., "This topic is more prevalent in corpus (t) than in corpus $(t-1)$"—one must consider the relative weights of the two data sets and error introduced by both the base model and fine tuned model. I do believe that exploring this topic can have major impact in corpus statistics, the digital humanities, and perhaps other fields. I also believe that this would greatly expand the scope of my dissertation research. Moreover, this extension is also dependent on my study of the LDA-DGP. If that study is inconclusive, then I'm not sure I can obtain valid uncertainty quantification.

[20] Available in the in-development *tidylda* package at https://github.com/TommyJones/tidylda/.

I am also considering an extension to make $\boldsymbol{\alpha}$ a matrix as well. For use cases where the context itself updates, one might wish to initialize $\boldsymbol{Cd}^{(t)}$ from $\boldsymbol{Cd}^{(t-1)}$.

4 Studying the LDA-DGP

LDA is a latent variable model, as a result it can be challenging to study. We do not observe "ground" truth topics in real data, against which to compare the correctness of a topic model. Extrinsic evaluation method compare a topic model's results against a different ground truth—one that we do observe, such as a document's class. Yet if a researcher's concern is document classification, better to build a supervised classifier. LDA's strength is in its unsupervised nature, enabling us to discover that which we don't already know. It appears we have a catch 22; we want to use LDA as a tool of discovery, yet without ground truth, how do we know if our discovery is true or a statistical fluke?

Fortunately, LDA models a data generating process, the LDA-DGP. Researchers can, and do, generate data sets by sampling from the LDA-DGP where they have chosen K, $\boldsymbol{\alpha}$, and $\boldsymbol{\eta}$ themselves. They then may use these simulated data sets to compare a model against a synthetic ground truth. Such "simulation studies" have a lengthy history in the statistical literature, going back to 1975 or further [85].

Yet one cannot choose arbitrary parameters in the LDA-DGP and expect the results to reflect the statistical properties of language. In 2014, when conducting my own simulation study, I discovered that common values used for $\boldsymbol{\eta}$ cannot produce corpora consistent with Zipf's law [46]. I include a proof (derived then) in Appendix 2.[21] Figure 1 plots data simulated from the LDA-DGP against an actual corpus of NSF grant abstracts [86]. The same holds for an asymmetric—but not proportional to a power law—$\boldsymbol{\eta}$. The simulation corresponding to symmetric $\boldsymbol{\eta}$—by far the most common specification since Wallach et al. in 2009 [45]—does not conform to a power law, which is linear in log-log space. Yet setting $\boldsymbol{\eta}$ proportional to a power law does produce power law distributed data, similar to the actual NSF abstracts data.[22] Language has such stark statistical properties—i.e. power law distributions—that the validity of a simulation that cannot produce such properties is suspect.

Yet, I believe that producing data with statistical properties of human language from the LDA-DGP means more than the low bar of one's simulation study not being invalid. I believe that if the LDA-DGP can produce data with that shares the statistical properties of human language, then LDA is a valid model

[21] Small as it was, this is the discovery that prompted me to seek a PhD. I wanted to complete this research, but knew that I could not do it without the structure and support of a formal program.

[22] Pardon the use of β rather than η in the figure. The figure is old and does not conform to my current notation scheme. I made the switch so that *tidylda* would be more consistent with popular text analysis packages in the R ecosystem.

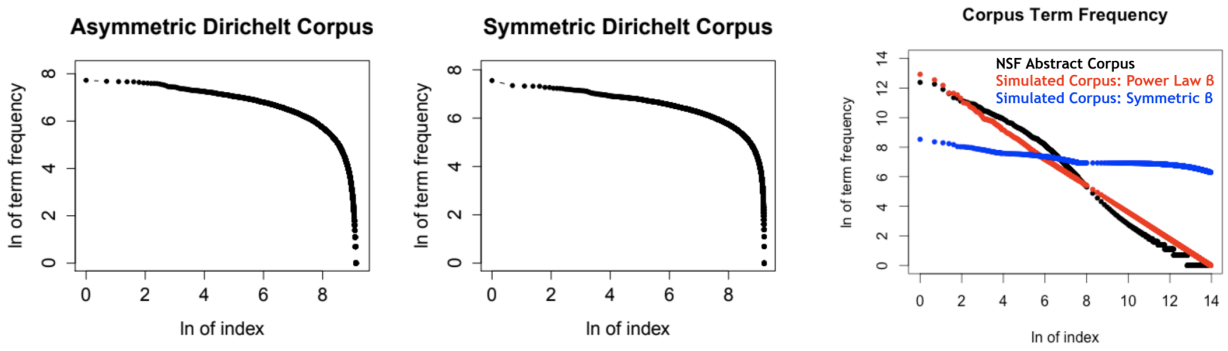

Figure 1: Comparing two simulated data with different Dirichlet priors. The leftmost figure uses an asymmetric, but not proportional to a power law, parameter. The center figure uses a symmetric parameter. The rightmost figure compares simulations to word frequencies in an actual corpus of NSF grant abstracts. The simulation generated with a symmetric prior for words over topics—as is commonly used—is not consistent with Zipf's law. It represents an impossible prior. Only the simulation made with a prior proportional to a power law produces word frequencies similar to the actual corpus. From Jones and St. Thomas, 2014.

for analyzing corpora of human language. Put another way: When conducting a simulation study, Shi et al. state "our analysis is grounded on the assumption that a hidden topic structure exists in the texts" [48]. I go further: If the LDA-DGP can produce data consistent with statistical laws of language, then *this is the correct assumption to make.*

I don't want to over state my case. Techniques—whether analytical of empirical—that discover the "right" model on simulated data can guide the researcher on real data sets. Yet it is unlikely that there is a "right" model for any real corpus. This is why I refer to "detecting pathologically misspecified models", rather than "finding the right model", throughout this document. It may also be that the LDA-DGP cannot perfectly reproduce relevant statistical laws of language. If that is the case, researchers must decide whether LDA is "good enough" or if a different topic model—for example CTMs [62] or STMs [87]—is a better choice.

My objectives with this research are as follows: I wish to analytically link the LDA-DGP to relevant statistical laws of language—as I have begun with Appendix 1, discover rules and heuristics from statistics on $\boldsymbol{X}$ to guide LDA model specification, and discover diagnostic statistics to discover whether an LDA model is pathologically misspecified. The remainder of this section is organized as follows: Section 4.1 summarizes related work from complex systems theory, stochastic simulation, and—of course—topic modeling. Section 4.2 outlines the approach I propose for this study.

4.1 Related Work

Work related to studying the LDA-DGP pulls from two—seemingly disparate—fields: complex systems theory and—of course—topic modeling. Complex systems theory deals in part with the emergence of power

law distributions–as exemplified by some empirical laws of language—from complex systems [88]. Topic modeling concerns itself with the study of LDA and LDA-like models.

4.1.1 Simulation Studies for LDA

Synthetic corpora appear commonly in topic modeling research. The table in Appendix 3—a copy of Shi et al.'s supplementary materials, Table S3 [48]—lists 14 works using synthetic corpora to study topic models. Most use some flavor of the LDA-DGP and focus on only one or two aspects of comparison between model and "ground truth" in the synthetic data set. I am unaware of any simulation study for LDA that explicitly links the data generating process to any of the empirical laws of language except for [33], [48]. Even so, they do not use the LDA-DGP and compare only to Zipf's law (described below). Boyd-Graber, Hu, and Mimno suggest the use of semi-synthetic data—i.e., drawing simulations from the posterior—to capture properties of natural language [23].[23]

Shi et al. go further than others and argue that use of synthetic corpora is a "principled approach" to evaluating topic models [33], [48]. Their approach does reproduce Zipf's law of language. Yet it does not use the LDA-DGP. So, it may represent a principled approach to studying probabilistic topic models with simulated data, but it is a parallel path to the one I propose. Using the LDA-DGP specifically allows for stronger statements related to pathological misspecification of LDA models. e.g., "Under the true model, then I would expect to see outcome A. Instead I see outcome B. Therefore, my model must be misspecified." An analogue from linear regression might be, "under the true model, residuals are distributed *i.i.d.* Gaussian with mean zero. The residuals of my model are not *i.i.d.* Gaussian with mean zero. Therefore, my model must be misspecified."

4.1.2 Empirical Language Laws

Altmann and Gerlach describe 9 universal laws purported to describe statistical regularities in human language [89]. These laws are Zipf's [46], Heaps's [90], Taylor's [91], Menzerath-Altmann [92], Recurrence [46], Long-range correlation [93], Entropy scaling [89], Information content [46], and various network topology laws [89].

Of these laws, I believe three are most relevant: Zipf's, Heaps's, and Taylor's laws. The laws of Menzerath-Altmann, Recurrence, Long-range correlation, and Information content refer to properties of sub-words

[23] They also refer to stochastic simulation studies for LDA as "toy problems". As you may have guessed, I strongly disagree. But their position is understandable insomuch as it is based on the assumption that LDA cannot reproduce properties of natural language. I have already demonstrated that it can reproduce at least one such property [86].

(e.g. length to information content), order of words, or proximity between words. None of these does the LDA-DGP purport to model. The law of Entropy scaling links entropy—in the information theoretic sense [94]—to the number of words in a block of text. The LDA-DGP may or may not be able to re-produce this law. Similarly, some network views of a corpus may or may not apply to the LDA-DGP. Both may warrant future exploration but are less directly macroscopic properties of a corpus than Zipf's, Heaps's, and Taylor's laws.

4.1.2.1 Zipf's law

Zipf's law states that the frequency of a word is inversely proportional to the power of its frequency-rank. Zipf's law is not unique to any language; it appears to apply to all of them [95]. Zipf's law has also been applied to the sizes of cities [96], casualties in armed conflict [97], and more. Zipf's law is a statement of a word's frequency and its rank. Yet if word frequencies in a corpus are plotted as a histogram, the power law relationship holds [98].

Empirical distributions of Zipf's law for large corpora have demonstrate a relationship somewhat inconsistent with that predicted by Zipf's law. Some have proposed that the frequency-to-rank relationship is actually a set of broken power laws with one parametarization for the head of the distribution, another for the body, and a third parametarization for the tail. Yet, Ha et al. find that this is a trick of tokenization. "Language is not made of individual words but also consists of phrases of 2, 3 and more words, usually called n-grams for n=2, 3, etc." When including n-grams in both English and Chinese corpora, Ha et al. find that Zipf's law holds through the tail [99]. Mandelbrot developed a generalization of Zipf's law that accounts for behavior at the head of the distribution [100].

Formally, Zipf's law is

$$F(r) \propto r^{-\gamma} \text{ for } \gamma \geq 1, r > 1 \tag{9}$$

where r is a word's rank, $F(r)$ is the frequency of a word's rank, and γ is a parameter to be estimated. For human language $\gamma \approx 1$ [95] and can be found through maximum likelihood estimation [97]. Altmann and Gerlach find $1.03 \leq \gamma \leq 1.58$ depending on the corpus and estimation method [89]

Goldwater et al. explore the relationship between Zipf's law and then standard statistical models of language [101]. They develop a framework for producing power law word frequencies in two stages. Critically, they

link this framework to several models closely related to LDA but do not extend it to the LDA-DGP itself.

4.1.2.2 Heaps's law

Heaps's law states that the number of unique words in a corpus, V, scales sub-linearly with the total number of words in a corpus, N [102]. Heaps's law is another power law. Unlike Zipf's law, it is an increasing power law.

$$V \propto N^{\delta} \text{ for } N > 1, 0 < \delta < 1 \tag{10}$$

There are several ways to compute Heaps's law for a single corpus. Altmann and Gerlach use two methods: compute V and N for each document in the corpus and progress over each word in a corpus calculating new values of V and N as each word is added. The latter approach works for single documents of sufficient length as well, such as a book [89].

4.1.2.3 Taylor's law

Taylor's law is a third power law relationship, originally posed in the context of ecology by Lionel Roy Taylor [103]. In linguistics, Taylor's law states that the standard deviation of the total number of words is proportional to the power of the mean of the total number of words. Formally,

$$\sigma(N) \propto \mu(N)^{\epsilon} \text{ for } \mu(N) > 1 \tag{11}$$

where $\sigma(N) = \sqrt{\mathbb{V}(N)}$, $\mu(N) = \mathbb{E}(N)$, and ϵ is to be fit from data.

4.1.2.4 Relationship between laws

Gerlach and Altmann explore the relationships between Zipf's, Heaps's laws [91]. They model word frequencies as resulting from a Poisson process. The "null" Poisson model links Zipf's and Heaps's laws. Yet it fits the data under Taylor's law poorly. Yet they find that incorporating a topic model—where relative frequencies of words differ by topic—fits data for all three laws reasonably well.

Gerlach and Altmann use LDA as the topic model, yet they do not use the LDA-DGP directly. Instead, they use the Poisson process model. Asyptotically, the Poisson process model and the LDA-DGP should be equivalent; the Poisson process is the limiting distribution of repeated Bernoulli trials (as in the LDA-DGP). Yet in the face of power laws and pre-asyptotics of finite samples (i.e., corpora), I think this choice warrants more exploration. Moreover, they used a "quenched average"—a concept from physics—rather than true expected value for this analysis. Under some conditions the two concepts may be equivalent; in other conditions not.

Even so, Gerlach and Altmann's exploration is a crucial link between Zip's law, Heaps's law, Taylor's law, and LDA, albeit in the asyptote.

4.2 Proposed Contributions

For this portion of my dissertation research, I propose making the following contributions.

1. Producing formal mathematical statements linking the LDA-DGP to Zipf's, Heaps's, and Taylor's laws of language as well as other corpus statistics such as correlation between words in two contexts,
2. Use principles of statistical design to plan and produce many synthetic corpora using the LDA-DGP that conform to statistical laws of human language and comprise a representative sample of the population of corpora that researchers may study with LDA,
3. Using (1) and (2) above, attempt to find rules or heuristics for specifying K, $\boldsymbol{\alpha}$, and $\boldsymbol{\eta}$ based on properties of the corpus, and
4. Using (1) and (2) above, diagnose the effects of model misspecification and attempt to develop diagnostic statistics or tests one may apply to an LDA model to detect pathological misspecification.

For (2), I intend to use principles from statistical design—recommended for such studies [104]. Considerations for producing a collection of simulated data sets include varying observable corpus variables—the number of documents, document lengths, vocabulary sizes, correlations between documents, and possibly more—and varying latent LDA-DGP parameters—K, $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$. Another relationship that I may need to derive is the expected correlation between the words in any two documents produced by the LDA-DGP.[24]

For (3), I intend to consider the same variables. The goal is to use the observable corpus variables to predict the latent LDA-DGP parameters.

[24]The covariance between words in any two documents is a function of the interaction between the covariance given by independent draws from the Dirichlet distribution that produced the documents—$\boldsymbol{\theta}_d \sim \text{Dirichlet}(\boldsymbol{\alpha})$—and the covariance given by independent draws from the Dirichlet distributions that produce the words—$\boldsymbol{\beta}_k \sim \text{Dirichlet}(\boldsymbol{\eta})$.

Diagnostic statistics that I am considering for (4) are: coherence metrics, likelihood metrics, R^2 (see below), and parameters from Zipf's, Heaps's, and Taylor's laws observed in the data compared to those predicted by drawing from the posterior of a fit LDA model. Pathological misspecifications I am considering are: too many or too few topics and misspecifications in shape or magnitude of $\boldsymbol{\alpha}$ or $\boldsymbol{\eta}$. I am also interested in exploring the effects of the procedure for optimizing $\boldsymbol{\alpha}$ [105] employed in MALLET [106].

5 Coefficient of Determination for Topic Models

According to an often-quoted but never cited definition, "the goodness of fit of a statistical model describes how well it fits a set of observations. Measures of goodness of fit typically summarize the discrepancy between observed values and the values expected under the model in question."[25] Goodness of fit measures vary with the goals of those constructing the statistical model. Inferential goals may emphasize in-sample fit while predictive goals may emphasize out-of-sample fit. Prior information may be included in the goodness of fit measure for Bayesian models, or it may not. Goodness of fit measures may include methods to correct for model over fitting. In short, goodness of fit measures the performance of a statistical model against the ground truth of observed data. Fitting the data well is generally a necessary—though not sufficient—condition for trust in a statistical model, whatever its goals.

Yet many researchers have eschewed goodness of fit measures as metrics of evaluating topic models. This is unfortunate. Goodness of fit is often the first line of defense against a pathologically misspecified model. If the model does not fit the data well, can it be relied on for inference or interpretation? Of course goodness of fit is not the only, perhaps not even the most important, measure of a good model. In fact a model that fits its training data too well is itself problematic. Nevertheless, a consistent and easily-interpreted measure of goodness of fit for topic models can serve to demystify the dark art of topic modeling to otherwise statistically literate audiences.

Several years ago, I derived a version of the coefficient of determination, R-squared, for topic models. I have written this up and posted it on arXiv [107], written and published an R package for it [108], but not yet published it in a peer-reviewed setting. I am currently working on a separate, related, paper with Mark Meyer presenting this R-squared as a generalization of the traditional R-squared for a range of statistical models. For my dissertation, I propose taking the additional step of re-stating this metric for topic models, including LDA and its derivatives as well as non-probabilistic models such as LSA.

The remainder of this section is organized as follows. Section 5.1 summarizes some relevant research related

[25] This quote appears verbatim on Wikipedia and countless books, papers, and websites. I cannot find its original source.

to evaluating topic models. Section 5.2 summarizes my preliminary results and proposal for contributions to my dissertation.

5.1 Related Work

Evaluation methods for topic models can be broken down into three categories: manual inspection, intrinsic evaluation, and extrinsic evaluation [48]. Manual inspection involves human judgement upon examining the outputs of a topic model. Intrinsic evaluation measures performance based on model internals and its relation to training data. R-squared is an intrinsic evaluation method. Extrinsic methods compare model outputs to external information not explicitly modeled, such as document class.

Manual inspection is subjective but closely tracks how many topic models are used in real-world applications. Most research on topic model evaluation has focused on presenting ordered lists of words that meet human judgement about words that belong together. For each topic, words are ordered from the highest value of $\boldsymbol{\beta}_k$ to the lowest (i.e. the most to least frequent in each topic). In [44] the authors introduce the "intruder test." Judges are shown a few high-probability words in a topic, with one low-probability word mixed in. Judges must find the low-probability word, the intruder. They then repeat the procedure with documents instead of words. A good topic model should allow judges to easily detect the intruders.

One class of intrinsic evaluation methods attempt to approximate human judgment. These metrics are called "coherence" metrics. Coherence metrics attempt to approximate the results of intruder tests in an automated fashion. Researchers have put forward several coherence measures. These typically compare pairs of highly-ranked words within topics. Röder et al. evaluate several of these [38]. They have human evaluators rank topics by quality and then compare rankings based on various coherence measures to the ranking of the evaluators. They express skepticism that existing coherence measures are sufficient to assess topic quality. In an ACL paper, [109] find that normalized pointwise mutual information (NPMI) is a coherence metric that closely resembles human judgement.

Other popular intrinsic methods are types of goodness of fit. The primary goodness of fit measures in topic modeling are likelihood metrics. Likelihoods, generally the log likelihood, are naturally obtained from probabilistic topic models. Likelihoods may contain prior information, as is often the case with Bayesian models. If prior information is unknown or undesired, researchers may calculate the likelihood using only estimated parameters. Researchers have used likelihoods to select the number of topics [34], compare priors [45], or otherwise evaluate the efficacy of different modeling procedures [110] [111]. A popular likelihood method for evaluating out-of-sample fit is called perplexity. Perplexity measures a transformation of the

likelihood of the held-out words conditioned on the trained model.

The most common extrinsic evaluation method is to compare topic distributions to known document classes. These evaluations employ precision and recall or the area under a receiver operator characteristic (ROC) curve (AUC) on topically-tagged corpora [110]. The most prevalent topic in each document is taken as a document's topical classification.

Though useful, prevalent evaluation metrics in topic modeling are difficult to interpret, are inappropriate for use in topic modeling, or cannot be produced easily. Intruder tests are time-consuming and costly, making intruder tests infeasible to conduct regularly. Coherence is not primarily a goodness of fit measure. AUC, precision, and recall metrics mis-represent topic models as binary classifiers. This misrepresentation ignores one fundamental motivation for using topic models: allowing documents to contain multiple topics. This approach also requires substantial subjective judgement. Researchers must examine the high-probability words in a topic and decide whether it corresponds to the corpus topic tags or not.

Likelihoods have an intuitive definition: they represent the probability of observing the training data if the model is true. Yet properties of the underlying corpus influence the scale of the likelihood function. Adding more documents, having a larger vocabulary, and even having longer documents all reduce the likelihood. Likelihoods of multiple models on the same corpus can be compared. (Researchers often do this to help select the number of topics for a final model [34].) Topic models on different corpora cannot be compared, however.[26] One corpus may have 1,000 documents and 5,000 tokens, while another may have 10,000 documents and 25,000 tokens. The likelihood of a model on the latter corpus will be much smaller than a model on the former. Yet this does not indicate the model on the latter corpus is a worse fit; the likelihood function is simply on a different scale. Perplexity is a transformation of the likelihood often used for out-of-sample documents. The transformation makes perplexity less intuitive than a raw likelihood. Perplexity's scale is influenced by the same factors as the likelihood.

5.2 Preliminary Results and Proposed Contributions

Goodness of fit manifests itself in topic modeling through word frequencies. It is a common misconception that topic models are fully-unsupervised methods. If true, this would mean that no observations exist upon which to compare a model's fitted values. However, probabilistic topic models are ultimately generative models of word frequencies [24]. The expected value of word frequencies in a document under a topic

[26] Actually, I am cautiously hopeful that my work in transfer learning can enable such comparisons based on changes of the same topic from the same base model fine tuned to two different corpora. The scale issue related to comparison via log likelihood will still remain, however.

model is given by the expected value of a multinomial random variable. The that can be compared to the predictions, then, are the word frequencies themselves. Most goodness of fit measures in topic modeling are restricted to in-sample fit. Yet some out-of-sample measures have been developed [112].

For the sake of brevity, I will state the key to R-squared for topic models here. A fuller justification and derivation is in my arXiv pre-print [107] and will be included in my dissertation. The key to this metric lies in two observations:

1. $E(\boldsymbol{X}|\boldsymbol{\Theta}, \boldsymbol{B}) = \boldsymbol{n} \odot \boldsymbol{\Theta} \cdot \boldsymbol{B}$, where $\boldsymbol{n}$ is a d-length vector of document lengths. In other words, we can compare the observed word frequencies in the data ($\boldsymbol{X}$) to the expected word frequencies under the model ($E(\boldsymbol{X}|\boldsymbol{\Theta}, \boldsymbol{B})$).
2. The various sums of squares used to calculate the coefficient of determination may be interpreted as sums of squared Euclidean distances in 1 space. If generalized to n-space, we can then compare the actual and expected word frequencies in a calculation that follows the definition of the coefficient of determination.

This interpretation of R-squared has most of the same properties as the commonly used R-squared. It is interpretable as the proportion of variation in the data explained by the model. An R-squared of 1 means that your model perfectly predicts the data. An R-squared of zero means that your model is no better than just guessing the mean of the data, i.e. a vector of word frequencies averaged across all documents. Yet we lose the lower bound of zero. Negative values of this new R-squared are computationally possible but this isn't a problem. It just means that one's model is worse than just guessing the mean of the data, quite the feat if one were to achieve it.

I propose performing the following for this section of my dissertation:

1. Update the paper on arXiv to reflect the current state of research in topic model evaluation
2. Expand on the simulation study used for the paper based on the simulation method I will propose in the LDA-DGP study
3. Expand on the real world corpora study to include more corpora. I would like to use more commonly used corpora so that readers familiar with the literature will have a more intuitive understanding of how the metric works.

6 *tidylda*, an R Package

Off the shelf implementations of LDA are plentiful. Why do we need one more? My research will be far more impactful if it is easy for researchers to use it. Putting it in a package and hosting it on a well used package repository like CRAN [113] makes it more accessible. I also strongly desire to support the philosophies of the *tidyverse* [28] movement in the R ecosystem, which focuses on making data analysis and computing tools work for humans. Having topic modeling software compatible with the *tidyverse* ecosystem is in line with Boyd-Graber et al.'s cry for "automatic text analysis for the people" [23, Ch. 10.3].

tidylda [114] is a natural extension of my own work in developing *textmineR* [115]. In 2014, I grew frustrated with the state of software for natural language processing in general, and topic modeling specifically. NLP software was esoteric in both syntax and data structures, concealing the fact that NLP workflows were not so different from others in statistics and machine learning. I developed *textmineR* with a mind to bring text analyses in R into the mainstream with respect to syntax, workflows, and data structures. In parallel—and unbeknownst to me at the time—Julia Silge and David Robinson were developing *tidytext* [116] with the same goals in mind (and frankly better execution), but tied into a popular framework and philosophy, the *tidyverse*. The syntactic differences between *textmineR* and *tidytext* are large. Though the user base is smaller than many other text mining packages for R, *textmineR* has enough users that it would be unkind to radically alter its user interface. *tidylda* is built from the ground up to conform to these tidy principles, houses a unique Gibbs sampler for LDA, and is narrower in scope than *textmineR*. When *tidylda* is ready to be hosted on CRAN, I will replace *textmineR*'s native LDA functionality with calls to *tidylda*.

The remainder of this section is organized as follows: Section 6.1 summarizes existing implementations of LDA that are commonly used and the "tidyverse" framework used by *tidylda*. Section 6.2 outlines the novel contributions of *tidylda* and the work remaining that I propose to include in my dissertation.

6.1 Related Work

6.1.1 The "tidyverse" and "tidy" text mining

tidylda takes its syntactic cues from an ecosystem of R packages known as *the tidyverse*. The tidyverse's goal is to "facilitate a conversation between a human and computer about data" [28]. Packages in—and adjacent to—the tidyverse share a common design philosophy and syntax based on "tidy data" principles [117]. Tidy data has each variable in a column, each observation in a row, and each observational unit in a table. Extensions include the *broom* package [118] for "tidying" up outputs from statistical models and the

in-development *tidymodels* ecosystem [119] which extends the tidyverse philosophy to statistical modeling and machine learning workflows.

Recently, Silge et al. articulated a "tidy data" framework for text analyses—the *tidytext* package [116]. Their approach has "one row per document per token". The *tidytext* package provides functionality to tokenize a corpus, transform it into this "tidy" format, and manipulate it in various ways, including preparing data for input into some of R's many topic modeling packages. The *tidytext* package also provides tidying functions in the style of *broom* to harmonize outputs from some of R's topic modeling packages. *tidylda* manages inputs and outputs in the flavor of *tidytext* but in one self contained package.

I am still frustrated by topic modeling software in R. Pretty much all topic modeling packages I have seen—with the exceptions of *textmineR* and *tidylda*—eschew R's conventional methods like *predict* for working with statistical models. Some provide that functionality, but approach it from unconventional directions. Others don't provide the ability to predict topic distributions for contexts not in the training sample at all![27]

6.1.2 Topic modeling software in R

R has many packages for topic modeling. As far as I am aware, none are natively "tidy" though some have wrapper functions available in *tidytext* for interoperability. In all cases—at least for R packages—these models support only symmetric $\boldsymbol{\eta}$ priors, though some support asymmetric $\boldsymbol{\alpha}$.

The *topicmodels* package [120] supports fitting models for LDA and correlated topic models [62] with both a collapsed Gibbs sampler and VEM. When using VEM, $\boldsymbol{\alpha}$ may be treated as a free parameter and estimated during fitting. It is designed to be interoperable with the *tm* package [121], the oldest framework for text analysis in R.[28] *tidytext* provides "tidier" functions to make the *topicmodels* package interoperable with other frameworks, such as *quanteda* [122], *text2vec* [123], and more.

The *lda* package [124] provides a collapsed Gibbs sampler for LDA, supervised LDA [125], and other less well-known models. It allows users to specify only symmetric $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$. Its syntax is esoteric and it requires text documents as input, but does not offer much flexibility in the way of pre-processing. It is generally not interoperable with other packages without significant programming on the part of its users. This is not to say that *lda* is a bad package. To the contrary, its sequential Gibbs sampler is one of the fastest. *textmineR* wrapped *lda* for its topic modeling capabilities until I implemented my own Gibbs sampler in 2018.[29]

[27] If they do, I can't figure out how to get it to work.

[28] From private conversation, I can tell you that both *textmineR* and *tidytext* grew out of an immense frustration with working with *tm*.

[29] This was the result of an independent study with Bill Kennedy. If not for that course, neither *tidylda* nor any of my work on transfer learning would be where they are now. Thanks, Bill!

The *text2vec* package [123] is a framework for very fast text pre-processing and modeling. *textmineR* wraps *text2vec* for its pre-processing functions. *text2vec* offers LDA implemented using the WarpLDA [55] algorithm, only allowing for symmetric priors. *text2vec* also offers other models related to distributional semantics. Its syntax is also esoteric using objects that reach back to actively running C++ code for performance reasons. One of *text2vec*'s novel features is that it implements many different coherence calculations; most packages implement one or none.

The *STM* package [63] implements VEM algorithms for structural topic models [87] and correlated topic models [62]. *STM* receives much of its interoperability through interfaces provided in *tidytext*. It offers unique capabilities for model initialization somewhat analogous to transfer learning. Models may be initialized at random or from an LDA model that has run for a few iterations. *STM* does not offer this as a fully-fledged "transfer learning" paradigm. Instead it is a flag the user sets at run time. *STM* then produces the LDA model to hand off to the STM model internally. STM has several unique methods for setting priors but inspecting the documentation makes me believe that they are still symmetric, where applicable.

6.1.3 Topic modeling software in other languages

There are many (many) programs to implement topic models in many languages. Anecdotally, the two most common are also two of the oldest: *MALLET* and *Gensim*.

MALLET [106] stands for "Machine Learning for LanguagE Toolkit." It is a Java program that implements LDA and many other models for working with text. Its LDA capabilities have a wrapper available in R—the *mallet* package—which launches a JVM in the background while users interact with it from R. MALLET allows users to set symmetric priors, but has an option to estimate an asymmetric $\boldsymbol{\alpha}$ based on [105]. Input must be text files, as it does all of its own pre-processing. MALLET's LDA implementation offers both a collapsed Gibbs sampler and distributed approximate Gibbs in the flavor of Newman et al. [50]. *tidytext* has *broom*-like functions to format the outputs of the *mallet* package.

Gensim [126] is a Python package that implements many models for working with text data, including LDA, LSI, and HDP [59]. Its LDA implementation is based on VEM and can be distributed across many compute nodes. Users can set flags for both $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ to be estimated during fitting. $\boldsymbol{\eta}$ can be a scalar, vector, or matrix. Gensim is the only other program that I am aware of that offers this option—other than *tidylda*. Gensim uses other Python packages for pre-processing.

6.2 Preliminary Results and Proposed Contributions

My goal for *tidylda*—as it pertains to my dissertation—is to publish it in a peer-reviewed journal. The two I am considering are the *Journal of Open Source Software* or the *Journal of Statistical Software*. *tidylda* in its current state is the most sophisticated piece of software I've written; it's available for download; and has 96% test coverage. Yet it is not ready for publication.

The current version of *tidylda* has the following novel features and capabilities.

1. It has a novel Gibbs sampler for fitting LDA models
    - Sequential "true" Gibbs or parallel "approximate" Gibbs as in [50]
    - Implements transfer learning as described above
    - Burn-in aggregates posterior samples over final iterations. Consistent with common practice in Bayesian stats and demonstrated improvements for LDA by as in [111]
    - Asymmetric prior hyper parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$. Per transfer learning: $\boldsymbol{\eta}$ can be a matrix as well.
    - Predict methods for easy topic distributions on new documents ($\hat{\boldsymbol{\Theta}}$)
2. Implements "tidy" methods
    - augment: calculates $P(\text{topic}|\text{word}, \text{document})$ and appends the value to each "document-token" observation, as in *tidytext*.
    - tidy: cleans up the output of posteriors—-$\boldsymbol{\Theta}$ and $\boldsymbol{B}$—to make them compatible with *tidyverse* syntax
    - print and summarize: banal, but expected methods for any model in the tidy ecosystem
3. Implements posterior derivations useful for analysis and—so far as I can tell—unique to packages I have developed
    - $\boldsymbol{\Lambda} = P(\text{topics}|\text{words})$ as described in Appendix 1.
    - $P(\text{topic}|\text{document}, \text{word})$—in the augment method—aggregates to $\boldsymbol{\Lambda}$ or $\hat{\boldsymbol{\Theta}}$ as projected by $\boldsymbol{\Lambda}$
4. Incorporates diagnostic statistics derived elsewhere: probabilistic coherence and R^2 for topic models.

As I stated, I believe *tidylda* needs more work before publication. The following is what I would like to achieve before a write up.

1. Substitute Gibbs for WarpLDA Metropolis Hastings sampler [55]

- WarpLDA is embarrassingly parallel at the token instance level while retaining MCMC theoretical guarantees of convergence
- I will definitely implement a CPU version of this for the dissertation. I want to extend this to a GPU for scalability that parallels what is available for deep neural nets. This scalability is necessary to achieve the future I see for fine tuned models in corpus statistics.

2. Implement the method to optimize $\boldsymbol{\alpha}$ as MALLET does, based on [105]
3. Write a series of vignettes demonstrating use of the package for various tasks including and perhaps extending the detail of what I have for *textmineR*.[30]
4. Formal write up of the package for publication in the Journal of Open Source Software (JOSS) or Journal of Statistical Software

7 Approach and Timeline

I am imagining the following chapters in my dissertation:

1. Introduction and Motivation - Essentially Section 1, above
2. Latent Dirichlet Allocation and Related Models - Essentially Section 2, above
3. Studying the LDA-DGP - Placed as the first study as its lessons will propagate to others
4. A Coefficient of Determination for Topic Models
5. Fine Tuning LDA for Transfer Learning
6. *tidylda*: Extended Latent Dirichlet Allocation using "Tidyverse" Conventions
7. Discussion and Conclusions

Because I already have chunks of my proposed research done, and because of interdependencies in the research, I plan to proceed in a nonlinear fashion. I believe I can accomplish the below in 2 years, by April 2023. In priority order:

1. I'd like to begin a draft write up of the work that has already been done in sufficient detail for the dissertation. This includes the separate (more general) R-squared paper I am working on with Dr. Meyer.
2. Implement WarpLDA in parallel on a CPU and integrate it into *tidylda*. This includes the transfer learning algorithm. (The good news is that the code is mostly modular and I can switch out the C++ parts keeping inputs and outputs the same.) Once this is done, I will submit *tidylda* to CRAN.

[30] See the 6 vignettes I have here https://CRAN.R-project.org/package=textmineR

3. Link the LDA-DGP to Zipf's, Heaps's and Taylor's laws
4. Design a series of simulations using the LDA-DGP for study
    - These data sets may be re-used for all three studies that require simulated data
5. Study the data sets in (4) to see if there are any rules or heuristics to guide hyper parameter selection before modeling and to diagnose pathological misspecification after modeling
6. Conduct the simulation portions of the R-squared and Transfer Learning studies
7. Collect real-world data sets to be used for the empirical sections of R-squared and Transfer Learning research
8. Conclude writing the three studies
9. Write *tidylda* up for JOSS or JSS
10. Re-review literature review (in case any late breaking research needs to be included in my dissertation for context) and conclude write up of the dissertation

8 Appendix 1: Projection Matrix for Probabilistic Embedding Models

We can use Bayes's rule to derive a projection matrix, $\mathbf{\Lambda}$, for any model that embeds text into a probability space such as pLSI, LDA, and similar.

Given a model developed on a matrix $\boldsymbol{X}$ with D contexts, indexed by $d \in \{1, 2, ..., D\}$; V unique words in the vocabulary, indexed by $v \in \{1, 2, ..., V\}$; N total word instances in the vocabulary and N_d word instances in the d-th document such that $N = \sum_{d=1}^{D} N_d$ and $N_d = \sum_{v=1}^{V} x_{d,v}$. The model itelf has; K topics, indexed by $k \in \{1, 2, ..., K\}$.

In the resulting model, $P(z_k|d) = \theta_{d,k}$ is the probability that a token in document d was sampled from topic k and $P(w_v|z_k) = \beta_{k,v}$ is the probability of sampling token v from topic k.

We need to derive $P(z_k|w_v) = \lambda_{v,k}$ using Bayes's rule.

$$\lambda_{v,k} = P(z_k|w_v) \tag{12}$$

$$= \frac{P(w_v|z_k) \cdot P(z_k)}{P(w_v)} \tag{13}$$

$$= \frac{\beta_{k,v} \cdot P(z_k)}{P(w_v)} \tag{14}$$

We can use the law of total probability to find $P(z_k)$.

$$P(z_k) = \sum_{d=1}^{D} P(z_k|d) \cdot P(d) \tag{15}$$

$$= \sum_{d=1}^{D} \theta_{d,k} \cdot P(d) \tag{16}$$

We can take $P(w_v)$ and $P(d)$ from $\boldsymbol{X}$ with

$$P(w_v) = \frac{1}{N}\sum_{d=1}^{D} x_{d,v} \tag{17}$$

$$P(d) = \frac{1}{N}\sum_{v=1}^{V} x_{d,v} \tag{18}$$

This gives us our final form

$$\lambda_{v,k} = \frac{\beta_{k,v} \cdot (\sum_{d=1}^{D} \theta_{d,k}) \cdot (\frac{1}{N}\sum_{v=1}^{V} x_{d,v})}{\frac{1}{N}\sum_{d=1}^{D} x_{d,v}} \tag{19}$$

Each of the above values represents the v, k entry of $\mathbf{\Lambda}$. We can then project a new data set, $\boldsymbol{X}'$ into the embedding space in two steps:

1. Normalize the rows of $\boldsymbol{X}'$ so that each row sums to 1; call this $\boldsymbol{X}'_n$
2. Right multiply $\boldsymbol{X}'_n$ by $\mathbf{\Lambda}^T$ such that $\mathbf{\Theta}' = \boldsymbol{X}'_n \cdot \mathbf{\Lambda}^T$

The above is valid for any probabilistic topic model, though it is a purely frequentist approach. My experience has been that this leads to noisier projections than, for example, "folding in" new data with a Gibbs sampler when using LDA.

9 Appendix 2: Expected Term Frequency of the LDA-DGP

Below derives the expected term frequency of a corpus whose terms are generated by the stochastic process modeled by Latent Dirichlet Allocation, the LDA-DGP. The expected term frequencies of a corpus generated with the above process are proportional to $\boldsymbol{\eta}$—the parameter for the Dirichlet prior for terms over topics. This implies that for a simulated corpus to follow Zipf's law, then $\boldsymbol{\eta}$ must be proportional to a power law.

Assuming there are D contexts, K topics, V unique words, N total words and N_d words in the d-th context, the LDA-DGP is as follows. For each word, n, in context d:

1. Generate $\boldsymbol{B}$ by sampling K topics $\boldsymbol{\beta}_k \sim \text{Dirichlet}(\boldsymbol{\eta}), \forall k \in \{1, 2, ..., K\}$
2. Generate $\boldsymbol{\Theta}$ by sampling D documents $\boldsymbol{\theta}_d \sim \text{Dirichlet}(\boldsymbol{\alpha}), \forall d \in \{1, 2, ..., D\}$
3. Then for each context, d
    1. Draw topic $z_{d,n}$ from Multinomial($\boldsymbol{\theta}_d$)
    2. Draw word $w_{d,n}$ from Multinomial($\boldsymbol{\beta}_{z_{d,n}}$)
    3. Repeat 1. and 2. N_d times.

Note that context d has N_d words $\forall d \in \{1, 2, ..., D\}$ and that the total number of words in the corpus is $N = \sum_{d=1}^{D} N_d$.

Under the model, the expected term frequency of a single context is

$$\mathbb{E}(\mathbf{w}_d | \boldsymbol{\theta}_d, \boldsymbol{B}) = n_d \odot \boldsymbol{\theta}_d \cdot \boldsymbol{B} \tag{20}$$

Using the law of total expectation, we have

$$
\begin{aligned}
\mathbb{E}(\mathbf{w}_d) &= \mathbb{E}(\mathbb{E}(\mathbf{w}_d|\boldsymbol{\theta}_d, \boldsymbol{B})) \\
&= \mathbb{E}(n_d \odot \boldsymbol{\theta}_d \cdot \boldsymbol{B}) \\
&= \mathbb{E}\left(n_d \odot \boldsymbol{\theta}_d \cdot \boldsymbol{\beta}\right) \\
&= \mathbb{E}\left(\begin{array}{c} n_d \begin{pmatrix} \sum_{k=1}^{K} \theta_{d,k} \cdot \beta_{k,1} \\ \sum_{k=1}^{K} \theta_{d,k} \cdot \beta_{k,2} \\ \cdots \\ \sum_{k=1}^{K} \theta_{d,k} \cdot \beta_{k,V} \end{pmatrix} \end{array}\right) \\
&= \mathbb{E}(n_d) \begin{pmatrix} \mathbb{E}(\sum_{k=1}^{K} \theta_{d,k} \cdot \beta_{k,1}) \\ \mathbb{E}(\sum_{k=1}^{K} \theta_{d,k} \cdot \beta_{k,2}) \\ \cdots \\ \mathbb{E}(\sum_{k=1}^{K} \theta_{d,k} \cdot \beta_{k,V}) \end{pmatrix} \\
&= \mathbb{E}(n_d) \begin{pmatrix} \sum_{k=1}^{K} \mathbb{E}(\theta_{d,k})\mathbb{E}(\beta_{k,1}) \\ \sum_{k=1}^{K} \mathbb{E}(\theta_{d,k})\mathbb{E}(\beta_{k,2}) \\ \cdots \\ \sum_{k=1}^{K} \mathbb{E}(\theta_{d,k})\mathbb{E}(\beta_{k,V}) \end{pmatrix}
\end{aligned}
$$

The last step, above, is due to indpendence of $\boldsymbol{\theta}_d$ and $\boldsymbol{\beta}_k$ $\forall d, k$.

Before carrying on, note two more relationships:

1. $\boldsymbol{\beta}_k \sim$ i.i.d. Dirichlet($\boldsymbol{\eta}$) means that $\mathbb{E}(\boldsymbol{\beta}_i) = \mathbb{E}(\boldsymbol{\beta}_j)$ $\forall i, j \in \{1, 2, .., K\}$
2. The expected value of a Dirchlet random variable—$\mathbf{X}$—with parameter $\boldsymbol{\delta}$ is $\mathbb{E}(\mathbf{X}) = \frac{1}{\sum_{m=1}^{M} \delta_m} \cdot \boldsymbol{\delta}$

From number 1., above, we can pull $\mathbb{E}(\beta_{k,\cdot})$ outside of the summation. And we can carry through the expected values using number 2., above.

$$
\begin{aligned}
\mathbb{E}(\mathbf{w}_d) &= \mathbb{E}(n_d) \begin{pmatrix} \mathbb{E}(\beta_{k,1}) \sum_{k=1}^{K} \mathbb{E}(\theta_{d,k}) \\ \mathbb{E}(\beta_{k,2}) \sum_{k=1}^{K} \mathbb{E}(\theta_{d,k}) \\ \cdots \\ \mathbb{E}(\beta_{k,V}) \sum_{k=1}^{K} \mathbb{E}(\theta_{d,k}) \end{pmatrix} \\
&= n_d \begin{pmatrix} \frac{\eta_1}{\sum_{v=1}^{V} \eta_v} \sum_{k=1}^{K} \frac{\alpha_k}{\sum_{k=1}^{K} \alpha_k} \\ \frac{\eta_2}{\sum_{v=1}^{V} \eta_v} \sum_{k=1}^{K} \frac{\alpha_k}{\sum_{k=1}^{K} \alpha_k} \\ \cdots \\ \frac{\eta_V}{\sum_{v=1}^{V} \eta_v} \sum_{k=1}^{K} \frac{\alpha_k}{\sum_{k=1}^{K} \alpha_k} \end{pmatrix} \\
&= n_d \begin{pmatrix} \frac{\eta_1}{\sum_{v=1}^{V} \eta_v} \frac{\sum_{k=1}^{K} \alpha_k}{\sum_{k=1}^{K} \alpha_k} \\ \frac{\eta_2}{\sum_{v=1}^{V} \eta_v} \frac{\sum_{k=1}^{K} \alpha_k}{\sum_{k=1}^{K} \alpha_k} \\ \cdots \\ \frac{\eta_V}{\sum_{v=1}^{V} \eta_v} \frac{\sum_{k=1}^{K} \alpha_k}{\sum_{k=1}^{K} \alpha_k} \end{pmatrix} \\
&= n_d \begin{pmatrix} \frac{\eta_1}{\sum_{v=1}^{V} \eta_v} \cdot 1 \\ \frac{\eta_2}{\sum_{v=1}^{V} \eta_v} \cdot 1 \\ \cdots \\ \frac{\eta_V}{\sum_{v=1}^{V} \eta_v} \cdot 1 \end{pmatrix} \\
&= \frac{n_d}{\sum_{v=1}^{V} \eta_v} \boldsymbol{\eta} \\
&\propto \boldsymbol{\eta}
\end{aligned}
$$

The end result is that the expected term frequency of a single document is proportional to $\boldsymbol{\eta}$—the Dirichlet parameter for terms over topics.

The term frequency for the whole corpus is the sum of the term frequencies for each document. Specifically

$$
\mathbf{w} = \sum_{d=1}^{D} \mathbf{w}_d
$$

The expected value under the model, then, can be carried through.

$$
\begin{aligned}
\mathbb{E}(\mathbf{w}) &= \mathbb{E}\left(\sum_{d=1}^{D} \mathbf{w}_d\right) \\
&= \sum_{d=1}^{D} \mathbb{E}(\mathbf{w}_d) \\
&= \sum_{d=1}^{D} \frac{n_d}{\sum_{v=1}^{V} \eta_v} \boldsymbol{\eta} \\
&= \frac{\sum_{d=1}^{D} n_d}{\sum_{v=1}^{V} \eta_v} \boldsymbol{\eta} \\
&\propto \boldsymbol{\eta}
\end{aligned}
$$

10 Appendix 3: Reproduction of Table S3 from Shi et al., 2019

Table S3: **Usage of Synthetic Corpora in Previous studies.**

Reference	Synthetic Corpora	Corresponding evaluation metric
Mukherjee and Blei (2009)	Generated from LDA	Likelihood; Variational free energy
Newman et al. (2009)	Generated from LDA	L1-norm between true and inferred word-topic distribution
Wallach et al. (2009)	Generated from LDA	Held-out likelihood
Mimno and Blei (2011)	Generated from LDA	Check hypothesis that words and documents are independent given the topic using mutual information
Taddy (2012)	Generated from LDA	Compare entries (and residuals) in θ_k (defined as the distribution over words for each topic) between true topics and inferred topics
Tang et al. (2014)	Generated from LDA	Posterior contraction analysis of the topic polytope
Hsu and Poupart (2016)	Generated from LDA	Use the synthetic corpora to test inferred number of topics
Minka and Lafferty (2002)	Multinomial with 5 equiprobable words	Likelihood; Classification
Griffiths and Steyvers (2004)	Bar-data (5x5 grid)	Visual comparison
Andrzejewski et al. (2009)	Small size synthetic corpora based on their proposed topic model (LDA with Dirichlet Forest Priors)	Visual inspection of the word-document matrix
AlSumait et al. (2009)	Small size synthetic corpora: 6 samples of 16 documents from three static equally weighted topic distributions. On average, the document size was 16 words.	Topic signficance score (similar to topic coherence)
Arora et al. (2013, 2016)	Semi-synthetic data (train parameters of a model on a real corpus, then use the model to generate synthetic data)	Training time; L1-error between true and inferred matrix A (defined as the word-topic matrix).
Lancichinetti et al. (2015)	Language data with non-overlapping topics	L1-norm between true and inferred $p(d\|t)$
This work	A flexible framework that could include a range of topic structure and realistic features	Measure the overlap between the planted and the inferred topic labels on the token level

Figure 2: A selection of topic modeling studies that use synthetic corpora. Reproduced from Shi et al., 2019, supplimentary materials, Table S3. "This work." refers to Shi et al., 2019.

References

[1] D. J. Newman and S. Block, "Probabilistic topic decomposition of an eighteenth-century american newspaper," *Journal of the American Society for Information Science and Technology*, vol. 57, no. 6, pp. 753–767, 2006.

[2] V. Perrone, M. Palma, S. Hengchen, A. Vatri, J. Q. Smith, and B. McGillivray, "GASC: Genre-aware semantic change for ancient greek," *arXiv preprint arXiv:1903.05587*, 2019.

[3] N. R. Ericsson, "Predicting fed forecasts," *Journal of Reviews on Global Economics*, vol. 6, pp. 175–180, 2017.

[4] U. Hahn, V. Hoste, and Z. Zhang, Eds., *Proceedings of the second workshop on economics and natural language processing*. Hong Kong: Association for Computational Linguistics, 2019.

[5] S. Volkova, D. Jurgens, D. Hovy, D. Bamman, and O. Tsur, Eds., *Proceedings of the third workshop on natural language processing and computational social science*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019.

[6] AMSTAT News, "New asa interest group: Text analysis," 2020. https://magazine.amstat.org/blog/2020/07/01/new-asa-interest-group-text-analysis/ (accessed Dec. 25, 2020).

[7] *Joint statistics meetings*. 2020.

[8] A. Legendre, *Nouvelles méthodes pour la détermination des orbites des comètes*. F. Didot, 1805.

[9] J. Angrist and J. Pischke, *Mostly harmless econometrics: An empiricist's companion*. Princeton University Press, 2008.

[10] J. M. Stanton, "Galton, pearson, and the peas: A brief history of linear regression for statistics instructors," *Journal of Statistics Education*, vol. 9, no. 3, 2001.

[11] G. C. Chow, "Tests of equality between sets of coefficients in two linear regressions," *Econometrica: Journal of the Econometric Society*, pp. 591–605, 1960.

[12] R. Anderson-Sprecher, "Model comparisons and r^2," *The American Statistician*, vol. 48, no. 2, pp. 113–117, 1994.

[13] N. N. Taleb, *Statistical consequences of fat tails*. New York, Ny: STEM Academic Press, 2020.

[14] A. Vaswani *et al.*, "Attention Is All You Need," *arXiv*, 2017.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv*, 2018.

[16] A. Conneau *et al.*, "Unsupervised Cross-lingual Representation Learning at Scale," *arXiv*, 2019.

[17] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[18] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," *arXiv*, 2020.

[19] T. Wolf *et al.*, "Transformers: State-of-the-Art Natural Language Processing," in *Proceedings of the 2020 conference on empirical methods in natural language processing: System demonstrations*, 2020, p. 38—45.

[20] K. Hao, "We read the paper that forced timnit gebru out of google. Here's what it says." 2020. https://www.technologyreview.com/2020/12/04/1013294/google-ai-ethics-research-paper-forced-out-timnit-gebru/ (accessed Dec. 24, 2020).

[21] C. Li, "Demystifying gpt-3," 2020. https://lambdalabs.com/blog/demystifying-gpt-3/ (accessed Dec. 24, 2020).

[22] D. Raji, "How our data encodes systematic racism," 2020. https://www.technologyreview.com/2020/12/10/1013617/racism-data-science-artificial-intelligence-ai-opinion/ (accessed Jan. 01, 2021).

[23] J. Boyd-Graber, Y. Hu, and D. Mimno, *Applications of Topic Models*, vol. 11. now Publishers Incorporated, 2017.

[24] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, 2003.

[25] M. E. Roberts, B. M. Stewart, and E. M. Airoldi, "A Model of Text for Experimentation in the Social Sciences," *Journal of the American Statistical Association*, vol. 111, no. 515, pp. 988–1003, 2016, doi: 10.1080/01621459.2016.1141684.

[26] M. Erlin, "Topic modeling, epistemology, and the english and german novel," 2018.

[27] R. C. Team, *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2013.

[28] H. Wickham *et al.*, "Welcome to the tidyverse," *Journal of Open Source Software*, vol. 4, no. 43, p. 1686, 2019, doi: 10.21105/joss.01686.

[29] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[30] D. L. Lee, H. Chuang, and K. Seamons, "Document ranking and the vector-space model," *IEEE software*, vol. 14, no. 2, pp. 67–75, 1997.

[31] 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*, vol. 41, no. 6, pp. 391–407, 1990, doi: 10.1002/(sici)1097-4571(199009)41:6<391::aid-asi1>3.0.co;2-9.

[32] T. Hofmann, "Probabilistic latent semantic analysis," in *Proceedings of the fifteenth conference on uncertainty in artificial intelligence*, 1999, pp. 289–296.

[33] H. Shi, "A Principled Approach to the Evaluation of Topic Modeling Algorithms," PhD thesis, 2019.

[34] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. Supplement 1, pp. 5228–5235, 2004, doi: 10.1073/pnas.0307752101.

[35] W. Zhao *et al.*, "A heuristic approach to determine an appropriate number of topics in topic modeling," *BMC Bioinformatics*, vol. 16, no. Suppl 13, p. S8, 2015, doi: 10.1186/1471-2105-16-s13-s8.

[36] T. Jones, "Optimizing topic models for classification tasks," 2019. https://www.jonesingfordata.com/talk/2019_11_09_dcr/ (accessed Dec. 27, 2020).

[37] I. Douven and W. Meijs, "Measuring coherence," *Synthese*, vol. 156, no. 3, pp. 405–425, 2007, doi: 10.1007/s11229-006-9131-z.

[38] M. Röder, A. Both, and A. Hinneburg, "Exploring the Space of Topic Coherence Measures," in *Proceedings of the eighth acm international conference on web search and data mining*, 2015, pp. 399–408, doi: 10.1145/2684822.2685324.

[39] P. Pietilainen, "Properties of Semantic Coherence Measures - Case of Topic Models," 2020.

[40] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum, "Optimizing Semantic Coherence in Topic Models," 2011.

[41] K. Stevens, P. Kegelmeyer, D. Andrzejewski, and D. Buttler, "Exploring Topic Coherence over many models and many topics," in *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, 2012, p. 952—961.

[42] A. B. Dieng, F. J. R. Ruiz, and D. M. Blei, "Topic Modeling in Embedding Spaces," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 439–453, 2020, doi: 10.1162/tacl_a_00325.

[43] Z. Chen and H. Doss, "Inference for the Number of Topics in the Latent Dirichlet Allocation Model via Bayesian Mixture Modeling," *Journal of Computational and Graphical Statistics*, 2019.

[44] J. Chang and J. Boyd-Graber, "Reading Tea Leaves: How Humans Interpret Topic Models," 2009.

[45] H. M. Wallach, D. Mimno, and A. McCallum, "Rethinking LDA: Why Priors Matter," 2009.

[46] G. K. Zipf, *Human behavior and the principle of least effort.* Oxford, England: Addison-Wesley Press, 1949.

[47] C. P. George and H. Doss, "Principled Selection of Hyperparameters in the Latent Dirichlet Allocation Model," *Journal of Machine Learnign Research*, vol. 18, no. 1, pp. 5937–5974, 2018.

[48] H. Shi, M. Gerlach, I. Diersen, D. Downey, and L. A. N. Amaral, "A new evaluation framework for topic modeling algorithms based on synthetic corpora," in *Proceedings of machine learning research*, 2019, vol. 89, p. 816—826, [Online]. Available: http://proceedings.mlr.press/v89/shi19a.html.

[49] L. AlSumait, D. Barbará, J. Gentle, and C. Domeniconi, "Topic Significance Ranking of LDA Generative Models," in *Joint european conference on machine learning and knowledge discovery in databases*, 2009, pp. 67–82.

[50] D. Newman, A. Asuncion, P. Smyth, and M. Welling, "Distributed Algorithms for Topic Models," *Journal of Machine Learning Research*, vol. 10, no. 8, 2009.

[51] M. Antoniak, "Tweet," 2020. https://twitter.com/maria_antoniak/status/1338155254756462592 (accessed Dec. 25, 2020).

[52] L. Yao, D. Mimno, and A. McCallum, "Efficient methods for topic model inference on streaming document collections," pp. 937–946, 2009, doi: 10.1145/1557019.1557121.

[53] J. Yuan *et al.*, "LightLDA: Big Topic Models on Modest Computer Clusters," in *Proceedings of the 24th international conference on world wide web*, 2015, p. 1351—1361.

[54] A. Ahmed, M. Aly, J. Gonzalez, S. Narayanamurthy, and A. J. Smola, "Scalable inference in latent variable models," pp. 123–132, 2012, doi: 10.1145/2124295.2124312.

[55] J. Chen, K. Li, J. Zhu, and W. Chen, "WarpLDA: a Cache Efficient O(1) Algorithm for Latent Dirichlet Allocation," *arXiv*, 2015.

[56] A. Srivastava and C. Sutton, "Autoencoding variational inference for topic models," *arXiv preprint arXiv:1703.01488*, 2017.

[57] P. Rychly', "Words' burstiness in language models." in *RASLAN*, 2011, pp. 131–137.

[58] D. Mimno and D. Blei, "Bayesian Checking for Topic Models," 2011.

[59] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet Processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2012, doi: 10.1198/016214506000000302.

[60] D. M. Blei and J. D. Lafferty, "Dynamic topic models," pp. 113–120, 2006, doi: 10.1145/1143844.1143859.

[61] J. McAuliffe and D. Blei, "Supervised topic models," *Advances in neural information processing systems*, vol. 20, p. 121—128, 2007.

[62] D. M. Blei and J. D. Lafferty, "A correlated topic model of Science," *The Annals of Applied Statistics*, vol. 1, no. 1, pp. 17–35, 2007, doi: 10.1214/07-aoas114.

[63] M. E. Roberts, B. M. Stewart, and D. Tingley, "stm : An R Package for Structural Topic Models," *Journal of Statistical Software*, vol. 91, no. 2, 2019, doi: 10.18637/jss.v091.i02.

[64] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[65] J. Eisenstein, *Introduction to natural language processing*. MIT Press, 2019.

[66] J. R. Firth, "A synopsis of linguistic theory, 1930-1955," *Studies in linguistic analysis*, 1957.

[67] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, 2014, pp. 1188–1196.

[68] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, pp. 3111–3119, 2013.

[69] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)*, 2014, pp. 1532–1543.

[70] M. E. Peters *et al.*, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[71] C. McCormick, "Word2vec tutorial-the skip-gram model." Retrieved, 2016.

[72] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," *Advances in neural information processing systems*, vol. 27, pp. 2177–2185, 2014.

[73] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 2013, pp. 746–751.

[74] A. Søgaard, I. Vulić, S. Ruder, and M. Faruqui, "Cross-lingual word embeddings," *Synthesis Lectures on Human Language Technologies*, vol. 12, no. 2, pp. 1–132, 2019.

[75] T. Jones, "Text embeddings," 2017. https://cran.r-project.org/web/packages/textmineR/vignettes/d_text_embeddings.html (accessed Dec. 25, 2020).

[76] A. Panigrahi, H. V. Simhadri, and C. Bhattacharyya, "Word2Sense : Sparse Interpretable Word Embeddings," in *Proceedings of the 57th annual meeting of the association for computational linguistics*, 2019, p. 5692—5705.

[77] C. Wang, D. Blei, and D. Heckerman, "Continuous Time Dynamic Topic Models," *arXiv*, 2012.

[78] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora," in *Proceedings of the 2009 conference on empirical methods in natural language processing*, 2009, p. 248—256.

[79] D. Andrzejewski and X. Zhu, "Latent Dirichlet Allocation with Topic-in-Set Knowledge," in *Proceedings of the naacl hlt 2009 workshop on semi-supervised learning for natural language processing*, 2009, p. 43—48.

[80] J. Jagarlamudi, R. Udupa, and H. D. III, "Incorporating Lexical Priors into Topic Models," 2012.

[81] D. Andrzejewski, X. Zhu, and M. Craven, "Incorporating domain knowledge into topic modeling via Dirichlet Forest priors," pp. 25–32, 2009, doi: 10.1145/1553374.1553378.

[82] Y. Hu, J. Boyd-Graber, B. Satinoff, and A. Smith, "Interactive topic modeling," *Machine Learning*, vol. 95, no. 3, pp. 423–469, 2014, doi: 10.1007/s10994-013-5413-0.

[83] L. AlSumait, D. Barbará, and C. Domeniconi, "On-Line LDA: Adaptive Topic Models for Mining Text Streams with Applications to Topic Detection and Tracking," *2008 Eighth IEEE International Conference on Data Mining*, pp. 3–12, 2008, doi: 10.1109/icdm.2008.140.

[84] M. D. Hoffman, D. M. Blei, and F. Bach, "Online Learning for Latent Dirichlet Allocation," *advances in neural information processing systems*, vol. 23, p. 856—864, 2010.

[85] D. C. Hoaglin and D. F. Andrews, "The Reporting of Computation-Based Results in Statistics," *The American Statistician*, vol. 29, no. 3, pp. 122–126, 1975, doi: 10.1080/00031305.1975.10477393.

[86] T. Jones and B. St. Thomas, "Zipf's law and latent dirichlet allocation," 2014.

[87] M. E. Roberts, B. M. Stewart, D. Tingley, and E. M. Airoldi, "The Structural Topic Model and Applied Social Science," 2013.

[88] C. Cioffi-Revilla, "Power laws and non-equilibrium distributions of complexity in the social sciences," 2008.

[89] E. G. Altmann and M. Gerlach, "Statistical laws in linguistics," *arXiv*, 2015, doi: 10.1007/978-3-319-24403-7_2.

[90] L. Egghe, "Untangling herdan's law and heaps' law: Mathematical and informetric arguments," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 5, pp. 702–709, 2007.

[91] M. Gerlach and E. G. Altmann, "Scaling laws and fluctuations in the statistics of word frequencies," *New Journal of Physics*, vol. 16, no. 11, p. 113010, 2014, doi: 10.1088/1367-2630/16/11/113010.

[92] G. Altmann, "Prolegomena to menzerath's law," *Glottometrika*, vol. 2, no. 2, pp. 1–10, 1980.

[93] F. J. Damerau and B. B. Mandelbrot, "Tests of the degree of word clustering in samples of written english," *Linguistics*, vol. 11, no. 102, pp. 58–75, 1973.

[94] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE mobile computing and communications review*, vol. 5, no. 1, pp. 3–55, 2001.

[95] R. F. i Cancho and R. V. Sole, "Least effort and the origins of scaling in human language," *Proceedings of the National Academy of Sciences*, vol. 100, no. 3, p. 788—791, 2003.

[96] S. Arshad, S. Hu, and B. N. Ashraf, "Zipf's law and city size distribution: A survey of the literature and future research agenda," *Physica A: Statistical Mechanics and its Applications*, vol. 492, pp. 75–92, 2018.

[97] C. S. Gillespie, "Fitting Heavy Tailed Distributions: The {poweRlaw} Package," *Journal of Statistical Software*, vol. 64, no. 2, p. 1—16, 2015, [Online]. Available: http://www.jstatsoft.org/v64/i02/.

[98] R. V. Sole, "Scaling laws in language evolution," in *Power laws and non-equilibrium distributions of complexity in the social sciences*, C. Cioffi-Revilla, Ed. Unpublished, 2008.

[99] L. Q. Ha, E. I. Sicilia-Garcia, J. Ming, and F. J. Smith, "Extension of Zipf's law to words and phrases," 2002, doi: 10.3115/1072228.1072345.

[100] B. Mandelbrot, "Information theory and psycholinguistics," *BB Wolman and E*, 1965.

[101] S. Goldwater, T. L. Griffiths, and M. Johnson, "Producing Power-Law Distributions and Damping Word Frequencies with Two-Stage Language Models," *Journal of Machine Learning Research*, vol. 12, 2011.

[102] H. S. Heaps, *Information retrieval, computational and theoretical aspects.* Academic Press, 1978.

[103] L. R. Taylor, "Aggregation, variance and the mean," *Nature*, vol. 189, no. 4766, pp. 732–735, 1961.

[104] T. P. Morris, I. R. White, and M. J. Crowther, "Using simulation studies to evaluate statistical methods," *Statistics in Medicine*, vol. 38, no. 11, pp. 2074–2102, 2019, doi: 10.1002/sim.8086.

[105] T. Minka, "Estimating a dirichlet distribution." Technical report, MIT, 2000.

[106] A. K. McCallum, "MALLET: A Machine Learning for Language Toolkit," 2002.

[107] T. Jones, "A coefficient of determination for probabilistic topic models," *arXiv preprint arXiv:1911.11061*, 2019.

[108] T. Jones, "Mvrsquared," 2020. https://CRAN.R-project.org/package=mvrsquared (accessed Dec. 30, 2020).

[109] 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*, 2014, pp. 530–539, doi: 10.3115/v1/e14-1056.

[110] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh, "On Smoothing and Inference for Topic Models," *arXiv preprint arXiv:1205.2662*, 2012.

[111] V.-A. Nguyen, J. Boyd-Graber, and P. Resnik, "Sometimes Average is Best: The Importance of Averaging for Prediction using MCMC Inference in Topic Modeling," in *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)*, 2014, pp. 1752–1757, doi: 10.3115/v1/d14-1182.

[112] W. Buntine, "Lecture Notes in Computer Science," pp. 51–64, 2009, doi: 10.1007/978-3-642-05224-8_6.

[113] T. C. Team, "The Comprehensive R Archive Network." Accessed: Jan. 01, 2020. [Online]. Available: https://cran.r-project.org/.

[114] T. Jones, "tidylda." 2020, [Online]. Available: https://github.com/TommyJones/tidylda.

[115] T. Jones, "textmineR: Functions for Text Mining and Topic Modeling." 2015, [Online]. Available: https://CRAN.R-project.org/package=textmineR.

[116] J. Silge and D. Robinson, "tidytext: Text Mining and Analysis Using Tidy Data Principles in R," *The Journal of Open Source Software*, vol. 1, no. 3, p. 37, 2016, doi: 10.21105/joss.00037.

[117] H. Wickham and others, "Tidy data," *Journal of Statistical Software*, vol. 59, no. 10, pp. 1–23, 2014.

[118] D. Robinson, "broom: An R Package for Converting Statistical Analysis Objects Into Tidy Data Frames," *arXiv*, 2014.

[119] M. Khun and H. Wickham, "Tidymodels," 2018. https://www.tidymodels.org/ (accessed Jan. 01, 2021).

[120] B. Grün and K. Hornik, "topicmodels: An R Package for Fitting Topic Models," *Journal of Statistical Software*, vol. 40, no. 13, 2011.

[121] I. Feinerer, K. Hornik, and D. Meyer, "Text Mining Infrastructure in R," *Journal of Statistical Software*, vol. 25, no. 5, 2008.

[122] K. Benoit *et al.*, "quanteda: An R package for the quantitative analysis of textual data," *Journal of Open Source Software*, vol. 3, no. 30, p. 774, 2018, doi: 10.21105/joss.00774.

[123] D. Selivanov, M. Bickel, and Q. Wang, "text2vec." CRAN, 2020, [Online]. Available: https://CRAN.R-project.org/package=text2vec.

[124] J. Chang, "lda." 2015, [Online]. Available: https://CRAN.R-project.org/package=lda.

[125] T. Nguyen, J. Boyd-Graber, J. Lund, K. Seppi, and E. Ringger, "Is Your Anchor Going Up or Down? Fast and Accurate Supervised Topic Models," in *Human language technologies: The 2015 annual conference of the north american chapter of the acl*, 2015, pp. 746–755, doi: 10.3115/v1/n15-1076.

[126] R. R. u rek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the lrec 2010 workshop on new challenges for nlp frameworks*, 2010, p. 45—50.