Memorable Title

Tommy Jones[*]

1 Introduction

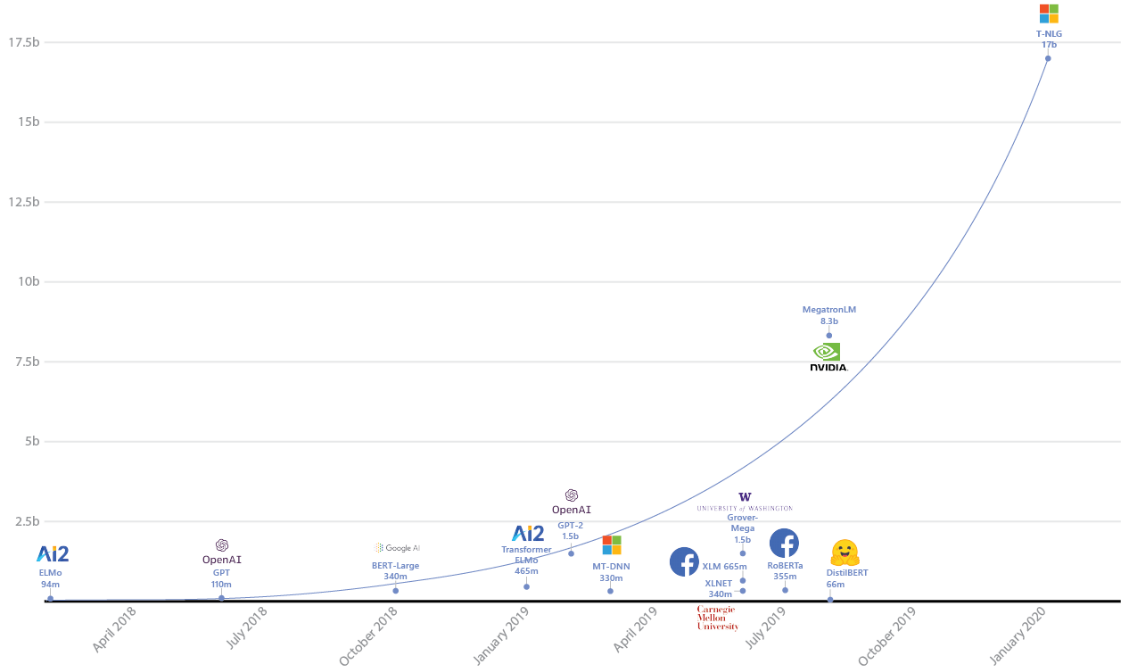
Human language is one of the most information rich sources of data that exists. Language is literally the medium humans use to communicate information to each other. And in an increasingly digitally connected world, the amount of text available for analyis has exploded. Improvements in computing power 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.

Current state of the art natural language processing (NLP) models belong to a class of deep neural networks called “transformers”. Famous examples of tranformers include ERNIE (cite), BERT (cite), XLNet (cite), GPT-2 (cite), GPT-3 (cite), and more. Transformers operate on a transfer learning paradigm called “pre-train then fine tune”. In the pre-training step, a “base” model is trained on an unsupervised or self-supervised (e.g. predict the next word) task with a very large volume of textual data. The fine tuning step involves replacing the output layer of the base model with a task specific output layer. Then a much smaller set of task-specific training data is used to update the weights of interior layers and lern new weights for the output layer. This approach has obvious advantages. In terms of raw accuracy for benchmark task-specific objectives, transformers reign supreme. It seems, also, that one can get acceptable results with fewer labeled examples when starting with a base model than using traditional end-to-end models. (cite)

Yet in spite of these advances, machine learning for NLP remains a largely ad-hoc field. 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 icreasingly complex models, requiring tremendous computational power, to drive these advances.

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

Figure 1 depicts the number of parameters in several famous examples of transformers. These deep neural network models are not only complex, they are expensive to train. GPT-3 (not pictured), the latest and greatest member of this class has approxmiately 175 billion parameters and cost an estimated $4.6 million to train.[1]



https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/

In addition to increasing complexity and cost, there is some evidence that the marginal returns to research in several subfields of machine learning are declining. (cite) When progress in one direction begins to slow, it may be time to push in another direction. Perhaps it is time to revisit statisitcal theory?

I propose reexamining a model that has become less popular in machine learning circles, Latent Dirichlet Allocation (LDA). Why? With the above comments in mind, LDA has some desireable 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. Akin to what we have for linear regression, statistical theory helps guide modeling decisions. This often results in models that are accurate, parsimonious, and interpretable. Modern NLP models acheive only the first. 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 (Roberts, Stewart, and Airoldi 2016) and the digital humanities (Erlin 2017). Additionally, I believe that I have developed method of transfer learning for LDA,

[1] https://bdtechtalks.com/2020/08/17/openai-gpt-3-commercial-ai

allowing it to be used in a pre-train then fine tune paradigm similar to that which is employed in transformer models.

To complete the requirements of my dissertation, I propose making three contributions. The first is a theoretical study of the LDA data generating process (LDA-DGP). The second is a study exploring transfer learning for LDA. The third is a software package for the R language that draws on my research and a framework known as the "tidyverse" (cite) 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. Section 3 outlines my proposed study of the LDA-DGP. Section 4 outlines my proposed study of transfer learning for LDA. Section 5 outlines a new (old) performance metric for topic modeling, the coefficient of determination for multidimensional outcomes. Section 6 introduces *tidylda* an in-develpment R package for LDA. Finally section 7 offers a timeline for completing the proposed dissertation.

2 Background: Latent Dirichlet Allocation

Probabilistic topic models are widely used latent variable models of language. Popularized in 2002 by latent Dirichlet allocation (LDA) (Blei, Ng, and Jordan 2003) many related models have been developed, for example (Blei and Lafferty 2007), (Roberts et al. 2013), (Nguyen et al. 2015), and more. These models share common characteristics. Probabilistic topic models estimate the probability that any word token[2] was sampled from a topic given the context and the probability of sampling each specific token given the topic, respectively.

Latent Dirichlet Allocation (LDA) is a Bayesian model of language. It models an idealized stochastic process for how words get on the page. Instead of writing full, syntatictically-coherent, sentences, the author samples a topic from a multinomial distribution and then given the topic samples a word. I call this process the LDA data generating process (LDA-DGP). Succintly it is

- Sample K topics from a V-dimensional Dirichelt s.t. $\boldsymbol{\beta}_k \sim Dirichlet_V(\boldsymbol{\eta})$
- Sample D documents from a K-dimensional Dirichlet s.t. $\boldsymbol{\beta}_k \sim Dirichlet_V(\boldsymbol{\eta})$
- For each document, d, and each word instance, $n \in \{1, 2, ..., n_d\}$ perform the following two-step sampling

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

1. Sample a topic $z_{d,n} \sim Multinomial(1, \boldsymbol{\theta}_d)$
2. Given topic $z_{d,n}$, sample a word $w_{d,n} \sim Multinomial(1, \boldsymbol{\beta}_{z_{d,n}})$

The original specification of the LDA-DGP (Blei, Ng, and Jordan 2003) specified that each document's length is a draw 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.

Though a generative model, LDA was designed to infer latent topics from a corpus. For a corpus of D documents, V unique tokens, and K latent topics, the goal is to estimate two matrices: $\boldsymbol{\Theta}$ and $\boldsymbol{B}$. The d-th row of $\boldsymbol{\Theta}$ comprises $\boldsymbol{\theta}_d$, above. And the k-th row of $\boldsymbol{B}$ comprises $\boldsymbol{\beta}_k$, above. LDA estimates these parameters by placing Dirichlet priors $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ on $\boldsymbol{\theta}_d$ and $\boldsymbol{\beta}_k$ respectively.

Loosely, this translates to a joint posterior of

$$P(w, z, \beta, \theta|\alpha, \eta) = P(w|z, \beta)P(z|\theta)P(\theta|\alpha)P(\beta|\eta) \tag{1}$$

A proper specification of the full posterior for a single document is

$$\begin{aligned} P(\mathbf{w}_d, \mathbf{z}_d, \boldsymbol{\theta}_d, \boldsymbol{B}|\boldsymbol{\alpha}, \boldsymbol{\eta}) &= \prod_{n=1}^{n_d} P(w_{d,n}|\boldsymbol{\beta}_{z_{d,n}})P(z_{d,n}|\boldsymbol{\theta}_d)P(\boldsymbol{\theta}_d|\boldsymbol{\alpha})P(\boldsymbol{B}|\boldsymbol{\eta}) & (2) \\ &= \left[\prod_{n=1}^{n_d} P(w_{d,n}|\boldsymbol{\beta}_{z_{d,n}})P(z_{d,n}|\boldsymbol{\theta}_d)P(\boldsymbol{\theta}_d|\boldsymbol{\alpha})\right]\left[\prod_{k=1}^{K} P(\boldsymbol{\beta}_k|\boldsymbol{\eta})\right] & (3) \end{aligned}$$

Note that the term $P(w_{d,n}|\boldsymbol{\beta}_{z_{d,n}})$ is equivalent to $P(w_{d,n}|z_{d,n}, \boldsymbol{\beta}_{z_{d,n}})$ because of the index on $\boldsymbol{\beta}$ is $z_{d,n}$.

Because of the exchangeability of documents, the full posterior of the model is

$$P(\mathbf{W}, \mathbf{Z}, \boldsymbol{\Theta}, \boldsymbol{B}|\boldsymbol{\alpha}, \boldsymbol{\eta}) = \left[\prod_{d=1}^{D}\prod_{n=1}^{n_d} P(w_{d,n}|\boldsymbol{\beta}_{z_{d,n}})P(z_{d,n}|\boldsymbol{\theta}_d)P(\boldsymbol{\theta}_d|\boldsymbol{\alpha})\right]\left[\prod_{k=1}^{K} P(\boldsymbol{\beta}_k|\boldsymbol{\eta})\right] \tag{4}$$

LDA implementations have three hyper parameters, parameters that must be set prior to modeling rather than being fit from data. These hyper parameters are K, $\boldsymbol{\alpha}$, and $\boldsymbol{\eta}$ described above. I discuss the roll of these hyper parameters in more detail in section 3.3.2.

Many algorithms for fitting LDA models exist. The most popular are variational Bayesian expectation maximization (VB) (Blei, Ng, and Jordan 2003) and collapsed[3] Gibbs sampling (Griffiths and Steyvers 2004). In response to limitations of these a host of other MCMC algorithms have been derived, mostly to address speed and scalability. Examples include (Teh, Newman, and Welling 2007), (Newman et al. 2008), (Yao, Mimno, and McCallum 2009), and (Chen et al. 2015).

3 Study 1: Examining the LDA-DGP

LDA models a process that generates language data. The LDA-DGP is clearly not how people write. Yet the degree to which the LDA-DGP can—or cannot—generate data that share the same statistical properties of human language has received little study. If the LDA-DGP can generate data that reasonably approximates human language data, then studying a collection of synthetic corpora drawn from the LDA-DGP can inform systematic strategies and rules for selecting hyperparameters and diagnosting model misspecification.

[Describe LDA-DGP]

Paragraph on history of research: 2011 goldwater et al 2015 taylor law thing 2019 guy dissertation

The study I propose has three components, described below. The first component is analytical, linking the three statistical laws of language to the LDA-DGP. The second component will generate a set of synthetic corpora using the LDA-DGP. [use statistical design to define a sample space covering any corpus likely to be fit with LDA and generate synthetic data to cover it] The third component, will analyze these corpora and the models that generated them to develop strategies for picking hyperparameters and diagnosing pathological model misspecification when using LDA on real corpora.

3.1 Linking LDA to Empirical Laws of Language

3.1.1 Thesis

If a data generating process purports to simulate the process generating human language, then the resulting data should display statistical properties consistent with empirical laws of language. The degree to which the LDA-DGP can or cannot capture these statistical properties is a measure of how well lessons derived from studying the LDA-DGP may extend to real-world corpora.

[3] This formulation is called "collapsed" because it integrates out the parameters of interest, $\boldsymbol{\Theta}$ and $\boldsymbol{B}$, for computational efficiency. The parameters of interest can be reconstructed after sampling completes.

This portion of the study is largely analytical, as opposed to statistical. I will mathematically examine the LDA-DGP and attempt to link it to relevant statistical laws of language. If successful, this will form a principled link between LDA and natural language.

3.1.2 Empirical Laws of Language

The gross statistical properties of language are captured in a set of empirical laws. The most famous is Zipf's law [cite] which captures the relationship between a word's frequency in a corpus and its rank when words are ordered from most to least frequent. Zipf's law is not the only such law. Altman and Gerlach describe 9 such laws, depicted in Table [X]. [cite]

[TABLE X ABOUT HERE]

Of these laws, three are relevant to LDA. They are Zipf's, Heap's, and Taylor's laws. LDA itself is principally a "bag of words" model, where word order within a document does not matter.[4] Nor does LDA concern itself with subword components such as phonemes. For these reasons, the other 6 laws listed in Table [X] do not apply.

3.1.2.1 Zipf's Law

Zipf's law describes the relationship between a word's frequency and its frequency-rank as a power law. It is typically parameterized by

$$f_r \propto r^{-a} \tag{5}$$

where f_r is a word's frequency, r is its frequency-rank, and b is a free parameter to be fit from data. An alternative, statistical parameterization is

$$P(f_r|b) \propto f_r^{-b} \tag{6}$$

which states that the probability of observing a given word frequency is subject to a decreasing power law.

[4] The bag of words assumption can be relaxed in LDA with certain preprocessing steps. For example, LDA may be used to find topics in a skip-gram term co-occurrence matrix. Skip-grams inherently capture some properties of word order. LDA itself needs only count data.

The two parameterizations may be mapped to each other where $b = 1 + a^{-1}$. See Figure [X] (a) for a graphical depiction of Zipf's law.

Zipf's law has also been extended into the Zipf-Mandlebrodt law to provide a better fit for high frequency words. The Zipf-Mandlebrodt law is often parameterized by a discrete probability distribution.

$$P(f_r|N, q, s) \propto (f_r + q)^{-s} \tag{7}$$

3.1.2.2 Heap's Law

Heap's law describes the relationship between the number of unique words in a corpus and the total number of words in a corpus. It is typically parameterized by

$$V \propto N^c \tag{8}$$

where V is the number of unique words (vocabulary size), N is the total number of words, and c is a free parameter to be fit from data. Typically c is between 0.4 and 0.6.

3.1.2.3 Taylor's Law

3.1.2.4 Relationship Between Laws

3.1.3 Approach and Preliminary Results

3.2 Using the LDA-DGP to Simulate Corpora

3.2.1 Thesis

3.2.2 Simulation Studies and Topic Models

3.2.3 Approach and Preliminary Results

3.3 Lessons for LDA Model Specification

3.3.1 Thesis

3.3.2 Folklore and Heuristics

3.3.3 Approach and Preliminary Results

4 Study 2: A Coefficient of Determination for Topic Models

4.1 Thesis

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."[5] 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 overfitting. 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 most 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

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

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 (cite), written and published an R package for it (cite), but have never formally submitted it for peer review. I am currently working on a separate, related, paper 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 derivates as well as non-probabilistic models such as LSA.

The remainder of this section is organized as follows. Section 4.2 summarizes some relevant research related to evaluating topic models. Section 4.3 summarizes my preliminary results and proposal for research that I propose.

4.2 Background

Evaluation methods for topic models can be broken down into three categories: manual inspection, intrinsic evaluation, and extrinsic evalutaion (**???**). Manual inspection involves human judgement upon examinging 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 modeld, 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 (**???**) 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. (**???**) evaluate several of these. 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, (???) 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. he 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 (???), compare priors (Wallach, Mimno, and McCallum 2009), or otherwise evaluate the efficacy of different modeling procedures. (???) (???) 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. However, (???), researchers have eschewed such goodness of fit metrics.

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 (???). 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.)(???) Topic models on different corpora cannot be compared, however. 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 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.

4.3 Approach and Preliminary Results

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 (**???**). The expected value of word frequencies in a document under a topic 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 (**???**).

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 (cite) and will be included in my dissertation. The key to this metric lies in two observations:

1. $E(\boldsymbol{X}|\boldsymbol{\Theta}, \boldsymbol{B}) = \boldsymbol{n} \odot \boldsymbol{\Theta} \cdot \boldsymbol{B}$ 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 a sum 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 a key difference is that we lose the lower bound of zero. While negative values of this new R-squared are computationally possible, this is not problematic. It just means that ones model is worse than just guessing the mean of the data, quite the feat if one were to acheive 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 “benchmark” corpora so that readers familiar with the literature will have a more intuitive understanding of how the metric works.

5 Study 3: Transfer Learning for LDA

5.1 Thesis

One of the advantages transformers bring is the pre-train then fine tune paradigm of transfer learning. This allows users to start with a model that “fits” a language reasonably well then improve it for a target corpus or task. Intuitively, this is how humans use langage. For example, I have a general mastery of English first, then read a textbook written in English to learn about chemistry. I do not try to learn English and chemistry at the same time from a chemistry textbook.

As widely used as LDA is, it lacks this capability. Transfer learning aside, many users of LDA often need to update previously trained models with new or revised data. This can leave users with an unpleasant tradeoff. Either models go stale or they must be retrained from scratch. In the former case, innacuracies and biases creep in over time. In the latter case, topics are reinitialized at random, breaking continuity with the old model. There has been little research on transfer learning for LDA. And as far as I know, there is no off-the-shelf software implementation available except for *tidylda* described in Section 5.

I have developed a method enabling full transfer learning for LDA models. The method involves generalizing the LDA model to allow the $\boldsymbol{\eta}$ hyper parameter to be a matrix, rather than a vector, and an algorithm for initializing the tokens in the new corpus allocated to topics proportionally to the token allocations for the base model were when training iterations stopped. The method also provides a means to add new topics to the transfer model that were not available in the base model. This method is implemented in the forthcoming *tidylda* package for R, described in Section 5.

The work I propose for my dissertation has three components. First is a formal statement of the method, both mathematically and algorithmically. Second, I need to derive a new likelihood function for a generalized LDA with matrix $\boldsymbol{\eta}$. (Sampling with a matrix $\boldsymbol{\eta}$ is complete. Yet the liklihood that one might calculate to assess convergence is not valid in my current implementation.) Finally, I propose two empirical studies, one with synthetic data and one with a real corpus.

The remainder of this section is organized as follows. Section 4.2 summarizes prior work related to transfer learning for LDA and reviews the mechanics of MCMC algorithms for LDA to set up my approach. Section 4.3 outlines the method I have developed and states in more detail the research approach I propose.

5.2 Background

5.2.1 Related Work

5.2.2 MCMC Mechanics for LDA

When fine-tuning a neural network, one introduces new data to a network whose weights start at values learned from training on a larger dataset. The weights update using backpropagation and whichever optimization algorithm (e.g. ADAM) is appropriate. The optimization algorithm may contain a learning rate, which tunes how quickly the neural network incorporates information from the new training data set.

Yet the approach for LDA is not as straightforward. MCMC algorithms do not have weights in the same way that neural networks do. And for LDA, they involve several arrays of counts that must be tightly calibrated to each other and to the counts of tokens in the training data set. Changing the data set, essential for transfer learning, disrupts this calibration. This section describes how most MCMC samplers work for LDA to set up understanding of my contribution, described in the next section.

MCMC algorithms for LDA must track several arrays[6] of integers.

- X is the original matrix of integer data, generally a document term matrix. It has D rows and V columns. $x_{d,v}$ represents the number of times word v appeared in document d.
- Cd is a D by K matrix. $cd_{d,k}$ represents the number of times topic k was sampled for a word in document d.
- Cv is a K by V matrix. $cv_{k,v}$ represents the number of times topic k was sampled for word v across all documents.
- Ck is a vector of length K. ck_k represents the number of times topic k was sampled for any token in X.

These four arrays must be kept in sync for every iteration of the sampler. To do this, the following constraints apply.

[6] In practice, these arrays &mdash or transformations of them &mdash may be represented in different ways for computational reasons. The description I use in this paper is for simplicity, rather than pure technical accuracy.

$$\sum_{d,v} x_{d,v} = \sum_{d,k} cd_{d,k} = \sum_{k,v} cv_{k,v} = \sum_{k} ck_k \tag{9}$$

$$\sum_{v} x_{d,v} = \sum_{k} cd_{d,k} \forall d \tag{10}$$

$$\sum_{d} xd, v = \sum_{k} cv_{k,v} \forall v \tag{11}$$

$$\sum_{d} cd_{d,k} = \sum_{v} cv_{k,v} = ck_k \forall k \tag{12}$$

Iteration proceeds as follows. Cd, Cv, and Ck are initialized by assigning each instance of a word in X to a topic at random. This is a random initialization while still keeping all four arrays syncronized. Then for each iteration, the sampler loops over each word instance sampling a topic. Once a topic has been assigned for the given word instance, Cd, Cv, and Ck are updated. The probability of sampling a topic is a function of the current state of Cd, Cv, and Ck and incorporates prior information through $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$. When iteration is complete, the posterior probabilities given in $\boldsymbol{\Theta}$ and $\boldsymbol{B}$ are calculated as follows.

$$\theta_{d,k} = \frac{cd_{d_k} + \alpha_k}{\sum_k cd_{d_k} + \alpha_k} \tag{13}$$

$$\beta_{k,v} = \frac{cv_{k,v} + \eta_v}{\sum_v cv_{k,v} + \eta_v} \tag{14}$$

So for transfer learning with LDA, one must address 2 issues:

1. Recover Cd, Cv, and Ck from the posteriors above, for a new dataset, X^*, to re-initialize the sampler and
2. Control the amount that X and X^* contribute to the fine-tuned model.

5.3 Approach and Preliminary Results

Contributions

- Allocate token counts from a new corpus proportional to counts at the last iteration of training from a pre-trained model
- Incorporate new vocabulary words

- Use $\boldsymbol{B}$ from a base model as a sort of learning rate and guard against "catastrophic forgetting"
- Add additional topics to a base model while fine tuning

Current state

- Exists in beta format now
- Missing capability to tune magnitude rows of $\boldsymbol{B}$ in the learning rate as a new $\boldsymbol{\eta}$ prior

Need to do

- Formal write up of algorithm
- Ability to tune magnitude of rows of $\boldsymbol{B}$ as new $\boldsymbol{\eta}$ in fine tuning
- Derivation of valid likelihood for matrix $\boldsymbol{\eta}$.
- Experimentation on sensitivity of tuning parameters and new data
- Possible experimentation using synthetic data to evaluate model converging to "correct" answer when DGP changes

6 Software: *tidylda*, an R Package

6.1 Thesis

Off the shelf implementations of LDA are plentiful. Why do we need one more? I have four answers to this question. First is usability: many implementations lack the ability to do basic tasks or it is challenging to execute those tasks. Examples of this include predicting topic distributions for new documents or even calculating the posterior parameters of the model from what is spit out of the sampler.[7] Second is interoperability: many implementations rely on esoteric data objects and workflows, not conforming to existing data analysis frameworks available in the wider ecosystem in which the implementation is written. Third is speed: algorithms to fit LDA models are notoriously slow. The most readily available algorithms for parallel implementations rely on approximations that may affect the quality of results.[8] Finally there is flexibility: most implementations are restrictive in the type of data accepted as input and have inflexibility in specification of the prior hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$. As I will demonstrate in my study of the LDA-DGP

[7] The latter is especially frustrating. This requires the user to have the mathematical sophistication to know an implement the final calculation. This is not user friendly and easily fixed by the package developer.

[8] The current implementation of *tidylda* does this too. However, as I describe below, I have plans to implement an algorithm that is both embarissingly parallel and retains the theoretical guarantees associated with MCMC algorithms.

(above), $\boldsymbol{\eta}$ must be asymmetric and proportional to a power law to incorporate a prior conforming to Zipf's law.

To address the above, I present *tidylda*, an R package implementing LDA that conforms to the "tidyverse" (cite) framework. *tidylda* is still in development, but it is the most flexible implementation of LDA available in any language. With future work, I also intend for it to be the fastest.

The remainder of this section is organized as follows: Section 6.2 summarizes existing implementations of LDA that are commonly used and the "tidyverse" framework used by *tidylda*. Section 6.3 outlines the novel contributions of *tidylda* and the work remaining that I propose to include in my dissertation.

6.2 Background

- What other LDA packages exist and why do we need a new one
  - User friendly, speed, principled defaults
- "Tidyverse" and "tidy text mining": theoretical framework

6.3 Approach and Preliminary Results

6.3.1 Contributions

1. tidyverse: augment, tidy, print, summarize
2. Posterior derivations useful for analyses
   - $\boldsymbol{\Lambda} = P(\text{topics}|\text{words})$
   - Augment method: $P(\text{topic}, \text{word})$ aggregates to $\boldsymbol{\Lambda}$ or "dot" method of prediction $\hat{\boldsymbol{\Theta}}$
3. Novel Gibbs sampler (soon to be WarpLDA MH sampler)
   - Sequential "true" Gibbs or parallel "approximate" Gibbs
   - 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 Boyd Graber et al. (cite)
   - 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 ($\hat{\boldsymbol{\Theta}}$)
4. Incorporates diagnostic statistics derived elsewhere: probabilistic coherence and R^2 for topic models.

6.3.2 Future work to be completed as part of dissertation

1. Substitute Gibbs for WarpLDA Metropolis Hastings sampler
   - Embarassingly parallel at the token instance level while retaining MCMC theoretical guarantees
   - 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.
2. Formal mathematical derivations of $P(topics|words)$ and $P(topic, word)$
3. Implement a method to optimize $\boldsymbol{\alpha}$ as in MALLET for Java
4. Write a series of vignettes demonstrating use of the package for various tasks
5. Formal write up of the package for publication in the Journal of Open Source Software (JOSS)

7 Timeline

Best guess is a semester per study + 1 semester for lit review and to button up citations, format, etc. So… 5 semesters $\approx$ 1.5 years? LDA-DGP may take 2 semesters which extends me by 6 months or so.

Order:

I plan to proceed in a nonlinear fashion. First, I want to document all the work that has already been done as its outcome isn't uncertain. Second, I want to do the LDA-DGP study as it forms the basis for simulated data in the empirical sections of the R-squared work and transfer learning work. Then, I will fill out the empirical sections of both the R-Squared work and transfer learning work. Finally, I will formalize my literature review, citations, and errata in the dissertation.

I am leaving the literature review for last because changes in the literature may have downstream implications for the context of the rest of my work. I'm generally familiar with the literature as demonstrated (will demonstrate) in this dissertation proposal. But I want the dissertation to be as up to date on the relevant literature as possible at the time of publication.

In summary:

1. *tidylda* for JOSS (least work, may or may not have WarpLDA at the time of JOSS publication)
2. R-squared: this paper is mostly written and available on arXiv. I plan to break it into two papers, one for publication in a statistical journal as a genralization of R-squared. The second will be included in my dissertation and focus on it as it applies to topic models (any probabilistic one and LSA).

3. WarpLDA in *tidylda* if I didn't get it done in (1)
4. Transfer learning for LDA - eyeballing the WarpLDA algorithm, the fundamental approach will not change switching from Gibbs to WarpLDA. All mathematical derivations should be the same.
5. LDA-DGP - Theoretical
6. LDA-DGP - Simulations
7. LDA-DGP - Heuristics for choosing hyper parameters
8. Simulation study for R-squared
9. Simulation study for transfer learning
10. Literature review and finalize dissertation write up

References

Blei, David M., and John D. Lafferty. 2007. "A correlated topic model of Science." *The Annals of Applied Statistics* 1 (1): 17–35. https://doi.org/10.1214/07-aoas114.

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

Chen, Jianfei, Kaiwei Li, Jun Zhu, and Wenguang Chen. 2015. "WarpLDA: a Cache Efficient O(1) Algorithm for Latent Dirichlet Allocation." *arXiv*.

Erlin, Matt. 2017. "Topic Modeling, Epistemology, and the English and German Novel." *Journal of Cultural Analytics*. https://doi.org/10.22148/16.014.

Griffiths, T L, and M Steyvers. 2004. "Finding scientific topics." *Proceedings of the National Academy of Sciences* 101 (Supplement 1): 5228–35. https://doi.org/10.1073/pnas.0307752101.

Newman, David, Arthur Asuncion, Padhraic Smyth, and Max Welling. 2008. "Distributed Inference for Latent Dirichlet Allocation." In *Advances in Neural Information Processing Systems*.

Nguyen, Thang, Jordan Boyd-Graber, Jeffrey Lund, Kevin Seppi, and Eric Ringger. 2015. "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*, 746–55. https://doi.org/10.3115/v1/n15-1076.

Roberts, Margaret E, Brandon M Stewart, and Edoardo M Airoldi. 2016. "A Model of Text for Experimentation in the Social Sciences." *Journal of the American Statistical Association* 111 (515): 988–1003. https://doi.org/10.1080/01621459.2016.1141684.

Roberts, Margaret E., Brandon M. Stewart, Dustin Tingley, and Edoardo M. Airoldi. 2013. "The Structural Topic Model and Applied Social Science." In.

Teh, Yee Whye, David Newman, and Max Welling. 2007. "A Collapsed Variational Bayesian Inference A lgorithm for Latent Dirichlet Allocation." In *Advances in Neural Information Processing Systems*.

Wallach, Hanna M., David Mimno, and Andrew McCallum. 2009. "Rethinking LDA: Why Priors Matter." In *Advances in Neural Information Processing Systems*.

Yao, Limin, David Mimno, and Andrew McCallum. 2009. "Efficient methods for topic model inference on streaming document collections," 937–46. https://doi.org/10.1145/1557019.1557121.