Domain-specific Topic Model for Knowledge Discovery through Conversational Agents in Data Intensive Scientific Communities

Yuanxun Zhang, Prasad Calyam, Trupti Joshi, Satish Nair, Dong Xu
Department of Electrical Engineering and Computer Science, University of Missouri-Columbia, USA
yzd3b@mail.missouri.edu, {calyamp, nairs, xudong}@missouri.edu, joshitr@health.missouri.edu

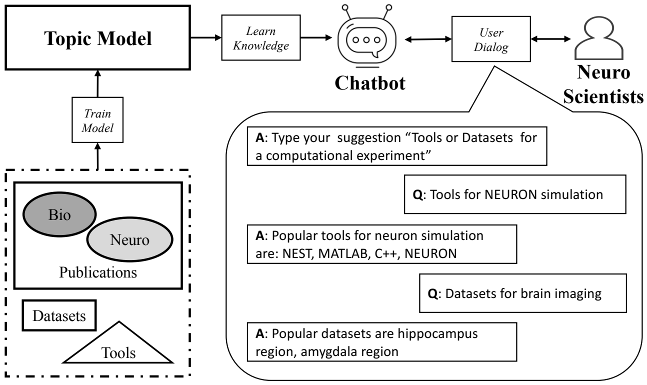
***Abstract*—Machine learning techniques underlying Big Data analytics have the potential to benefit data intensive communities in e.g., bioinformatics and neuroscience domain sciences. Today's innovative advances in these domain communities are increasingly built upon multi-disciplinary knowledge discovery and cross-domain collaborations. Consequently, shortened time to knowledge discovery is a challenge when investigating new methods, developing new tools, or integrating datasets. The challenge for a domain scientist particularly lies in the actions to obtain guidance through query of massive information from diverse text corpus comprising of a wide-ranging set of topics. In this paper, we propose a novel "domain-specific topic model" (DSTM) that can drive conversational agents for users to discover latent knowledge patterns about relationships among research topics, tools and datasets from exemplar scientific domains. The goal of DSTM is to perform data mining to obtain meaningful guidance via a chatbot for domain scientists to choose the relevant tools or datasets pertinent to solving a computational and data intensive research problem at hand. Our DSTM is a Bayesian hierarchical model that extends the Latent Dirichlet Allocation (LDA) model and uses a Markov chain Monte Carlo algorithm to infer latent patterns within a specific domain in an unsupervised manner. We apply our DSTM to large collections of data from bioinformatics and neuroscience domains that include hundreds of papers from reputed journal archives, hundreds of tools and datasets. Through evaluation experiments with a perplexity metric, we show that our model has better generalization performance within a domain for discovering highly specific latent topics.**

***Keywords*-Topic Model, Theoretical Model for Big Data, Latent Dirichlet Allocation, Multi-disciplinary Knowledge Discovery, Computation and Data Intensive Applications**

I. Introduction

Scientific domains such as bioinformatics and neuroscience have the potential to benefit from Big Data analytics that use underlying machine learning techniques for solving computational and data intensive research problems. Bold innovations will increasingly emerge from processing large amount of datasets or recognizing complex knowledge patterns using e.g., artificial neural networks. Moreover, the bold innovations will occur from solving multi-disciplinary research problems that require knowledge discovery across disciplines and from cross-domain scientist collaborations. To enable rapid pace of innovation, domain scientists are continuously seeking to investigate new methods, develop new tools or integrate structured/unstructured data sets.

However, finding relevant knowledge patterns featuring tools, methods and datasets amongst vast information archives to obtain timely guidance to solve multi-disciplinary research problems can be challenging for domain scientists. For example, biologists may need to use relevant machine learning and statistics methods for protein structure or gene sequence predictions; machine learning studies may need to extend new algorithms/tools to solve unique problems in personalized medicine; and data-intensive neuroscience efforts could adopt cyberinfrastructure integration best practices from bioinformatics [1] for building workflows across distributed computing resources.



A: Type your suggestion "Tools or Datasets for a computational experiment"

Q: Tools for NEURON simulation

A: Popular tools for neuron simulation are: NEST, MATLAB, C++, NEURON

Q: Datasets for brain imaging

A: Popular datasets are hippocampus region, amygdala region

Figure 1: Framework of incorporating conversational agents (i.e., chatbot in a science gateway portal) with a topic model for providing helpful guidance for scientific users.

One major challenge in obtaining useful guidance through query of massive information is to discover knowledge pattern digests amongst diverse text corpus comprising of a wide-ranging set of topics. With the access to such knowledge pattern digests that feature a list of topics, tools and data sets, it is possible to design conversational agents to help domain scientists as shown in Figure 1. For instance, a chatbot integrated with a science gateway portal can leverage a relevant topic model in answering researcher questions such as e.g., What are the best tools to handle particular modeling problems with high accuracy?; Which types of datasets have been used previously to evaluate a certain kind of hypothesis? Given that computational and data intensive science is expensive and time consuming, availability of topic models to provide useful guidance through data mining of massive open information within a domain or across domains can significantly benefit domain scientists.

In our preliminary experiments that involved manually querying/surveying the publications from neuroscience, and bioinformatics domains, we found that common knowledge patterns within as well as across the domains can be useful to domain scientists. We found by observing novice/expert

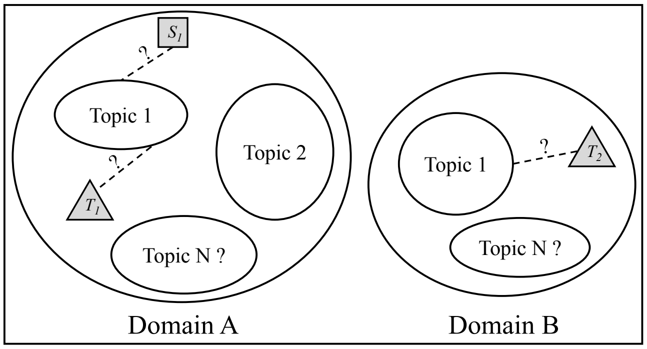


Figure 2: Discovery of relationships among research topics, tools (with "T" notation) and datasets (with "S" notation) for scientific domains.

researchers that significant text corpus relating to popular tools (e.g., Pegasus [2] in bioinformatics, and NEURON [3] in neuroscience) and datasets (e.g., RNA, Interneuron) are frequently used as guidance using a manual (slow/inefficient) approach. Also, latest computational and data intensive research problems in neuroscience tend to be influenced by efforts in prior bioinformatics literature that took leadership in successfully investigating related problems with relevant combinations of topic sets. Exemplar topic sets include e.g., integration of data sets with community-wide standards, and sustainable toolkits in distributed computing environments.

In order to be efficient and effective (i.e., to obtain quick and meaningful guidance), we further found that ideal topic models need to handle several uncertain factors. Uncertainty can be caused by changing/evolving relationships among topics, tools, and datasets as a domain matures and its text corpus increases in size/variety. For instance, observing Figure 2, uncertainty can affect the effective discovery of the purpose of tool T_1 with dataset S_1, while seeking guidance for appropriate tools and dataset(s) to solve a problem related to Topic 1 in a scientific domain A. Alternately, uncertainty can occur when efficiently determining whether the tool T_2 is an appropriate tool for solving a problem related Topic 1 in a scientific domain B. Therefore, design of an ideal topic model should be scalable and flexible to deal with daily/monthly/yearly changing Big Data "volume", "variety" and "value" within scientific domains, and satisfy query needs on state-of-the-art problems for domain scientists.

In this paper, we propose a novel "domain-specific topic model" (DSTM) that can drive conversational agents for users to discover latent knowledge patterns in scientific domains that rely on multi-disciplinary knowledge and cross-domain collaborations. DSTM is fundamentally a generative model that extends the popular Latent Dirichlet Allocation (LDA) model [4] and the Author-Topic model [5], which is a LDA variant. LDA is applied to find sets of topics among large collections of text corpus. Our DSTM can not only detect sets of topics, but more importantly it can discover the relationships among topics, tools and datasets for a specific scientific domain. DSTM assumes each topic can be represented as a distributions over words, and each tool or dataset is modeled as an individual distribution over topics. Such distributions or parameters can be learned through unsupervised learning from collections of text corpus that reflect the patterns of tools or datasets that are more likely to be used for domain research problems by using Markov chain Monte Carlo inference algorithm for a specific domain. Our DSTM is designed to be integrated within chatbots in science gateway portals to provide helpful guidance for domain scientists to choose appropriate tools or data sets in their research. As illustrated in Figure 1, the DSTM can be trained using large amounts of tools or datasets obtained from publications, and the pre-trained model can be compiled as a Numpy binary file. A chatbot can load this model file for learning the knowledge of relationships amongst topics, tools and data sets in order to have a user dialog.

To demonstrate the benefits of DSTM and reveal the latent patterns in large collections of scientific publications, we apply our model to large collections of data relating to reputed journal archives belonging to two scientific domains: neuroscience and bioinformatics. We collect 367 papers from *Frontiers in Computational Neuroscience* and *Journal of Computational Neuroscience*, 476 papers from *Journal of BMC Bioinformatics*. Given the fact that just the abstract may not describe much about tools and dataset, we use full documents/papers to generate the results in our evaluation experiments. This in turn, leads to a vocabulary size of $V = 10,718$ terms in the neuroscience domain, and $V = 9,699$ terms in the bioinformatics domain. We also collect the names of tools, types of datasets that are commonly used in these domains. The analysis results from our data mining experiments using a perplexity metric shows that the DSTM model can reveal the highly specific latent topics, and provide useful guidance for choosing tools and datasets for cutting-edge domain research problem areas. Given our design of DSTM, our model can be easily extended to be applied with satisfactory generalization performance to other domains by changing information relating to the types of publications, tools, and datasets.

The rest of the paper is organized as follows: Section II discusses the related works. In Section III, we describe our generative model. In Section IV, we describe our inference algorithms for parameter estimation. Details of experiments and findings are listed in Section V. We conclude the paper in Section VI.

II. Related Work

Topic models have been found to be a successful method to automatically extract useful information from text corpus in an unsupervised learning manner. Latent Dirichlet Allocation (LDA) [4] is one of most popular topic models that was invented by David M. Blei in 2003. It discovers the latent topic structures from a collection of documents or text corpus automatically. In LDA, each topic is modeled as a distribution over words, and each document is represented as a mixture over topics proportions. The LDA model has been widely applied to document classifications, searching, and recommendations.

Based on the LDA model, many researchers have tried to propose model variants for discovering different patterns of documents. Rosen-Zvi *et al.* [5] proposed an Author-Topic model that extends LDA by including authorship information to establish the relationships between topics and authors. For exploring relationships between documents and authors, it allows the mixture of weights for different topics to be determined by the authors of documents.

Mimno *et al.* [6] also proposed a similar author topic model for matching papers with peer-reviewers. Blei *et al.* [7] proposed a dynamic topic model in order to extract the evolution of topics within sequentially organized documents. Blei and Lafferty in [8] proposed a correlated topic model (CTM) to demonstrate the correlations between topics using a logistic normal distribution on the simplex to model dependence between two topics. This distribution represents the correlations between components.

Other researchers also successfully applied LDA to different areas. Li *et al.* [9] adapted the LDA model for image scene categorization without any human annotations, which achieved comparable performance. Flaherty *et at.* [10] developed a model that is able to cluster genes within experiments that do not require inputs of a gene or drug. Wang *et al.* [11] combined the traditional collaborative filtering and probabilistic topic models (i.e., LDA variants) in a recommendation system to recommend scientific articles. Their system provides a latent structure that can be interpreted for users and items. This structure also could be presented as recommendations pertaining to both existing as well as newly published documents. Sun *et al.* [12] proposed a probabilistic generative model to explore the expert behaviors in collaborative networks. Tang *et al.* [13] adapted the LDA and Author-Topic model to find potential cross-domain collaborations.

None of the existing topic models can benefit from little or any amount of domain-specific knowledge to explore specific latent patterns that are meaningful for particular research problems involving tools and datasets. Consequently, they are not applicable for the domain-specific topic model problem being addressed by our DSTM approach for effective and efficient knowledge discovery in computational and data intensive scientific communities. Moreover, in contrast with the LDA, our DSTM not only discovers what topics are expressed in a published document, but also considers information about the relevant tools or datasets that are associated with each topic.

Inference algorithms to infer the latent variables in a probabilistic model (such as the LDA) have also been an area of active research investigations. The original LDA work [4] used variational expectation maximization algorithm to estimate latent parameters. Hoffman *et al.* [14] designed an online stochastic optimization with a natural gradient step. Their optimization results showed the convergence to a local optimum of the variational Bayes objective function. Griffiths *et at.* [15] presented a collapsed Gibbs sampling algorithm (i.e., a Markov chain Monte Carlo method) to infer latent parameters of their model. In our work, we similarly use the Gibbs sampling method that is easy to implement without compromising the learning speed and generalization performance.

III. The Generative Model

Table I. Notations for the generative model.

Symbols	**Description**
D	a collection of documents $D = \{d_1, ..., d_n\}$
K	the number of topics
T	a set of tools
S	a set of datasets
V	a set of word in vocabulary
N_d	the number of word tokens in document d
$\boldsymbol{t}_d$	a set of tools in document d
$\boldsymbol{s}_d$	a set of dataset in document d
$w_{dn}, \boldsymbol{w}$	the n^{th} word token in document d
$x_{dn}, \boldsymbol{x}$	tool indicator chosen from $\boldsymbol{t}_d$ for word w_{dn}
$y_{dn}, \boldsymbol{y}$	dataset indicator chosen from $\boldsymbol{s}_d$ for word w_{dn}
$z_{dn}, \boldsymbol{z}$	the topic assignment for word w_{dn}
$L_{dn}, \boldsymbol{L}$	binary indicator to label which is responsible for z_{dn}
$\pi_d, \boldsymbol{\pi}$	Bernoulli parameter for generating label $\boldsymbol{L}$ for document d
$\boldsymbol{\eta}$	$(\eta_{\pi_0}, \eta_{\pi_1})$ parameters for Beta distribution prior
$\phi_z, \boldsymbol{\phi}$	multinomial distribution over words specific to topic z
$\theta_t, \boldsymbol{\theta}$	multinomial distribution over topics specific to tool t
$\lambda_s, \boldsymbol{\lambda}$	multinomial distribution over topics specific to dataset s
α, β, γ	parameters of symmetric Dirichlet priors

In this section, we present our DSTM generative model to discover the latent patterns underlying a collection of documents for a particular scientific domain in term of the relationship among research topics, tools and datasets. The idea of a generative model for text modeling is to generate each word in a document based on the distributions of words. As opposed to the LDA model that generates each word based on random topics, our model generates each word based on reference tools or dataset occurrence in a document. In the generative process, we do not assume that a tool and/or dataset are responsible for a certain word simultaneously. For simplifying the computational complexity, each word is generated by either a tool or a dataset.

The graphical representation of our generative model is illustrated in Figure 3 using a plate notation. We collect papers from particular collections of documents $D = \{d_1, ..., d_n\}$. A document d is represented as a bag-of-words with N_d unique word tokens, and the n^{th} word in document d is denoted as w_{dn}. In each document d, word is observed variable with "shaded" color, and the other observed variables are a set of tools $\boldsymbol{t}_d$ and a set of dataset categories $\boldsymbol{s}_d$. During the pre-processing stage, we extract tools and dataset categories mentioned in each document based on our collections of tool names and dataset categories provided by a domain scientist as domain-specific knowledge. T denotes the total number of tools and S the total number of dataset categories we collected. In the model, we assume there are K number of topics for collection documents D. $\boldsymbol{\phi}$ denotes the $K \times V$ matrix of topics distribution over vocabulary. $\boldsymbol{\theta}$ denotes the $T \times K$ matrix of tools distribution over topics, and $\boldsymbol{\lambda}$ denotes the $S \times K$ matrix of datasets distribution over topics. L is binary indicator variable to label whether the topic

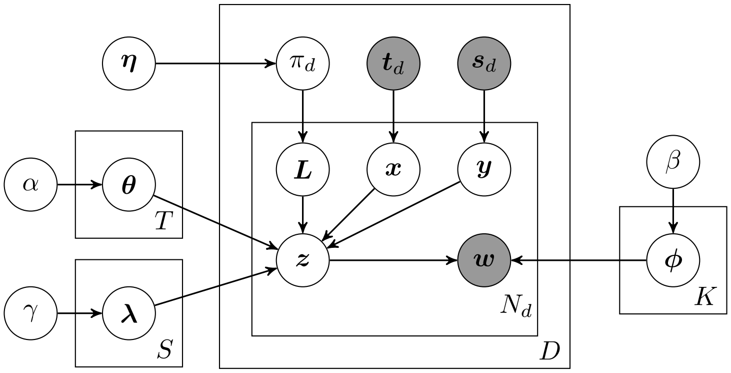


Figure 3: Graphical representation of the generative model. The boxes are "plates" representing replicates; the "shaded" nodes are observed variables; the "unshaded" nodes are unobserved variables. See Table I for node notations.

assignment from tool distribution $\boldsymbol{\theta}$ or dataset distribution $\boldsymbol{\lambda}$. Table I summarizes all of the various notations used in our generative model.

Algorithm 1: Generative process in the model

```
1. For each topic $k = 1, ..., K$:
   (a) Draw a multinomial over vocabulary $\phi_k \sim Dir(\beta)$;
2. For each tool $t = 1, ..., T$:
   (a) Draw a multinomial over topics $\theta_t \sim Dir(\alpha)$;
3. For each dataset $s = 1, ..., S$:
   (a) Draw a multinomial over topics $\lambda_s \sim Dir(\gamma)$;
4. For each doc in $d = 1, ..., D$:
   (a) Generate $\pi_d \sim Beta(\eta_{\pi_1}, \eta_{\pi_2})$;
   (b) For each word $n = 1, ...N_d$;
         i. Sample binary indicator $L_{dn} \sim Bern(\pi_d)$;
        ii. If $L_{dn} == 0$, then:
              • Select a tool $x_{dn} \sim Unif(\boldsymbol{t}_d)$;
              • Sample a topic $z_{dn} \sim Multi(\theta_{x_{dn}})$;
       iii. If $L_{dn} == 1$, then:
              • Select a dataset $y_{dn} \sim Unif(\boldsymbol{s}_d)$;
              • Sample a topic $z_{dn} \sim Multi(\lambda_{y_{dn}})$;
        iv. Choose a word $w_{dn} \sim Multi(\phi_{k=z_{dn}})$;
```

Algorithm 1 describes the generative process of the model. First, each topic is associated with a multinomial distribution over V vocabulary drawn from symmetric Dirchlet(β) prior. Each tool t samples a multinomial distribution over topics from Dirchlet(α) prior, represented by θ_t. And each dataset s samples a multinomial distribution over topics from Dirchlet(γ) prior, denoted as λ_s. Second, for each word in document d, we draw a binary indicator $\boldsymbol{L}$ from Bernoulli(π_d) distribution to decide whether this word is generated by a tool or a dataset. The Bernoulli(π_d) distribution is applied when both $\boldsymbol{t}_d$ and $\boldsymbol{s}_d$ are not empty. If either of them is empty, the $\boldsymbol{L}$ is assigned to the non-empty one. Then, a tool or a dataset is chosen from either set of tools ($\boldsymbol{t}_d$) or set of datasets ($\boldsymbol{s}_d$) randomly and uniformly. A topic assignment z_{dn} is selected based on the tools ($\boldsymbol{\theta}$) or datasets ($\boldsymbol{\lambda}$) distributions over topics. Finally, a word is generated according to topic distribution ($\boldsymbol{\phi}$) over words.

There is a special case with our model that not all papers mention tools, or datasets in their papers, and it is impossible for us to collect all types of tools, datasets to match with papers. For dealing with this case, we reserve last 20 indexes of tools (T) and datasets (S) for unknown tools and datasets. When both $\boldsymbol{t}_d$ and $\boldsymbol{s}_d$ are both empty, the last 20 indexes of a tool or a dataset will be selected in an uniformly random fashion. The following generative process is same as the normal case. In order to keep the generative algorithm concise, this special case is not described in Algorithm 1.

By estimating the latent variables $\{\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{z}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\pi}, \boldsymbol{L}\}$ of the model, we obtain information about topics of the collection of documents, and which tools or datasets are preferred to be used for a particular topic. In the following section, we describe the algorithm to estimate the latent variables by using the Gibbs sampling method [15].

IV. Inference and Parameter Estimation

In this section, we describe the Gibbs sampling method that we use to infer and estimate latent variables $\{\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{z}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\pi}, \boldsymbol{L}\}$ of the model. To build Gibbs sampling, we construct posterior distribution of latent variables conditioned on all other variables, and repeatedly sample from conditional probability until it converges to a target distribution. In practice, we do not need to construct Gibbs sampling equations for each latent variable. By taking advantage of conjugate prior, the latent variables $\{\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\pi}\}$ can be integrated out as follows: Dirichlet is the conjugate prior of multinomial, and Beta is the conjugate prior of Bernoulli. Using density estimation of $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$, we can still estimate $\{\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\lambda}\}$ through posterior distribution.

In our model, the Gibbs sampling has two procedures in each iteration: (i) sampling label $\boldsymbol{L}$ to decide which (tool or dataset) is responsible for generating a word; (ii) sampling topic assignment $\boldsymbol{z}$ for a word based on the label assignment $\boldsymbol{L}$ and related tool distribution θ or dataset distribution λ. To simplify equations, we define the set of hyperparameters as $\boldsymbol{\Omega} = \{\alpha, \beta, \gamma, \boldsymbol{\eta}, T, S\}$.

A. *Sampling the Label* $\mathbf{L}$

For each n^{th} word of document d, we construct Gibbs sampling equation for label L_{dn}, topic assignment z_{dn}, and tool assignment x_{dn} or dataset assignment y_{dn} as a block (L_{dn}, z_{dn}, x_{dn}) or (L_{dn}, z_{dn}, y_{dn}) conditioned on all other variables. Then, the full conditional probability for labeling tool $(L_{dn} = 0, z_{dn} = k, x_{dn} = t)$ is as follows:

$$\begin{aligned}\mathrm{P}(L_{dn} = 0, &z_{dn} = k, x_{dn} = t|\boldsymbol{L}_{-dn}, \boldsymbol{x}_{-dn}, w_{dn} \\ &\boldsymbol{z}_{-dn}, \boldsymbol{w}_{-dn}, \boldsymbol{y}, \boldsymbol{\Omega}) \\ =&\frac{C_t^L + \eta_{\pi_0} - 1}{C_t^L + C_s^L + \eta_{\pi_0} + \eta_{\pi_1} - 1} \\ &\times \frac{C^{TK}_{tk,-dn} + \alpha}{\sum_k C^{TK}_{tk,-dn} + K\alpha} \\ &\times \frac{C^{VK}_{vk,-dn} + \beta}{\sum_v C^{VK}_{Vk,-dn} + V\beta}\end{aligned} \tag{1}$$

where C_t^L is the number of times including current instance that a tool is selected for generating word in document d, C_s^L is the number of times including current instance that dataset is selected for generating word in document d, $\boldsymbol{L}_{-dn}$ denotes all the label assignments excluding the current instance. C^{TK}_{tk} is the number of times tool t is assigned to topic k, and the subscript $C^{TK}_{tk,-dn}$ denotes the exclusion of the current instance. C^{VK}_{vk} is the number of times word v in vocabulary V is assigned to topic k, and the subscript $C^{VK}_{vk,-dn}$ denotes excluding the current instance. Subsequently, we sum over all tools T, topics K, and vocabularies V to get $\mathrm{P}(L_{dn} = 0)$,

$$\begin{aligned}\mathrm{P}(L_{dn} = 0|&\boldsymbol{L}_{-dn}, \boldsymbol{x}_{-dn}, \boldsymbol{z}_{-dn}, w_{dn}, \boldsymbol{w}_{-dn}, \boldsymbol{y}, \boldsymbol{\Omega}) \\ =&\sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{v=1}^{V}\Big[\frac{C_t^L + \eta_{\pi_0} - 1}{C_t^L + C_s^L + \eta_{\pi_0} + \eta_{\pi_1} - 1} \\ &\times \frac{C^{TK}_{tk,-dn} + \alpha}{\sum_k C^{TK}_{tk,-dn} + K\alpha} \\ &\times \frac{C^{VK}_{vk,-dn} + \beta}{\sum_v C^{VK}_{Vk,-dn} + V\beta}\Big]\end{aligned} \tag{2}$$

Similarly, the full conditional probability for dataset being selected $(L_{dn} = 1, z_{dn} = k, y_{dn} = s)$ is as follows:

$$\begin{aligned}\mathrm{P}(L_{dn} = 1, &z_{dn} = k, y_{dn} = s|\boldsymbol{L}_{-dn}, \boldsymbol{y}_{-dn}, w_{dn}, \\ &\boldsymbol{z}_{-dn}, \boldsymbol{w}_{-dn}, \boldsymbol{x}, \boldsymbol{\Omega})\end{aligned} \tag{3}$$

$$\begin{aligned}=&\frac{C_s^L + \eta_{\pi_1} - 1}{C_t^L + C_s^L + \eta_{\pi_0} + \eta_{\pi_1} - 1} \\ &\times \frac{C^{SK}_{sk,-dn} + \gamma}{\sum_k C^{SK}_{sk,-dn} + K\gamma} \\ &\times \frac{C^{VK}_{vk,-dn} + \beta}{\sum_v C^{VK}_{Vk,-dn} + V\beta}\end{aligned} \tag{4}$$

where C^{SK}_{sk} represents the number of times dataset s is assigned to topic k, with the subscript $C^{SK}_{sk,-dn}$ denotes excluding the current instance. Next, we integrate out all possible datasets S, topics K, and vocabularies V to get $\mathrm{P}(L_{dn} = 1)$,

$$\begin{aligned}\mathrm{P}(L_{dn} = 1|&\boldsymbol{L}_{-dn}, \boldsymbol{y}_{-dn}, \boldsymbol{z}_{-dn}, w_{dn}, \boldsymbol{w}_{-dn}, \boldsymbol{x}, \boldsymbol{\Omega}) \\ =&\sum_{k=1}^{K}\sum_{s=1}^{S}\sum_{v=1}^{V}\Big[\frac{C_s^L + \eta_{\pi_1} - 1}{C_t^L + C_s^L + \eta_{\pi_0} + \eta_{\pi_1} - 1} \\ &\times \frac{C^{SK}_{sk,-dn} + \gamma}{\sum_k C^{SK}_{sk,-dn} + K\gamma} \\ &\times \frac{C^{VK}_{vk,-dn} + \beta}{\sum_v C^{VK}_{Vk,-dn} + V\beta}\Big]\end{aligned} \tag{5}$$

B. Sampling the Topic Assignment $\boldsymbol{z}$

We also construct (z_{dn}, x_{dn}) or (z_{dn}, y_{dn}) as a block, conditioned on all other variables, where we sample $\boldsymbol{z}$ and $\boldsymbol{x}$ or $\boldsymbol{z}$ and $\boldsymbol{y}$ jointly. Given this, the full conditional probability of (z_{dn}, x_{dn}) at the situation of $L_{dn} = 0$ is,

$$\begin{aligned}\mathrm{P}(z_{dn} = k, &x_{dn} = t|L_{dn} = 0, \boldsymbol{L}_{-dn}, w_{dn}, \boldsymbol{w}_{-dn}, \\ &\boldsymbol{z}_{-dn}, \boldsymbol{x}_{-dn}, \boldsymbol{y}, \boldsymbol{\Omega}) \\ \propto&\frac{C^{TK}_{tk,-dn} + \alpha}{\sum_k C^{TK}_{tk,-dn} + K\alpha}\frac{C^{VK}_{vk,-dn} + \beta}{\sum_v C^{VK}_{vk,-dn} + V\beta}\end{aligned} \tag{6}$$

where $(z_{dn} = k,\ x_{dn} = t)$ represents the assignments of n^{th} word in document d to topic k and the tool t, respectively; $\boldsymbol{z}_{-dn}, \boldsymbol{x}_{-dn}$ denotes all topic and tool assignments not including current instance.

Similarly, we can get the full conditional probability of (z_{dn}, y_{dn}) under the condition of $L_{dn} = 1$ is,

$$\begin{aligned}\mathrm{P}(z_{dn} = k, &y_{dn} = s|L_{dn} = 1, \boldsymbol{L}_{-dn}, w_{dn}, \boldsymbol{w}_{-dn}, \\ &\boldsymbol{z}_{-dn}, \boldsymbol{x}, \boldsymbol{y}_{-dn}, \boldsymbol{\Omega}) \\ \propto&\frac{C^{SK}_{sk,-dn} + \gamma}{\sum_k C^{SK}_{sk,-dn} + K\gamma}\frac{C^{VK}_{vk,-dn} + \beta}{\sum_v C^{VK}_{vk,-dn} + V\beta}\end{aligned} \tag{7}$$

where $\boldsymbol{y}_{-dn}$ represents all datasets assignments not including the current instance.

Having obtained the full conditional distributions equations of $\boldsymbol{L}$, $\boldsymbol{z}$, the whole Gibbs sampling algorithm is straightforward. First, we initialize the variables $\{\boldsymbol{L}, \boldsymbol{z}, \boldsymbol{x}, \boldsymbol{y}\}$ randomly. Then, in each iteration, we update $\boldsymbol{L}$ and $\{\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{y}\}$ in turn from the full conditional distribution with Equations 2, 5, 6, 7 until it converges to a target distribution.

C. Parameter Estimation

Collecting sets of samples $\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{y}$ obtained from Gibbs Sampling algorithm, we can estimate variables $\{\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\pi}\}$ with expectation of posterior distribution. The posterior distribution of topics k over vocabularies ϕ_k is written as,

$$\begin{aligned}\mathrm{P}(\phi_k|\boldsymbol{z}, \boldsymbol{w}, \beta) &\propto \mathrm{P}(\boldsymbol{w}|\boldsymbol{z}, \phi_k)\mathrm{P}(\phi_k|\beta) \\ &\propto \prod_{d=1}^{D}\prod_{n=1}^{N_d}\phi_{k,w_{dn}}\prod_{v=1}^{V}\phi_k^{\beta-1} \\ &= \prod_{v=1}^{V}\phi_k^{C^{VK}_{vk}+\beta-1} = \mathrm{Dir}(C^{VK}_{vk} + \beta)\end{aligned} \tag{8}$$

Then, the expectation of the Dirichlet distribution to estimate parameter ϕ_{vk}, which is the probability of the vocabulary v assigned to topic k for any single sample,

$$\phi_{vk} = \frac{C_{vk}^{VK} + \beta}{\sum_v C_{vk}^{VK} + V\beta} \tag{9}$$

Similarly, the parameters estimation of θ_{tk} that is the probability of tool t assigned to topic k, and λ_{sk} that is the probability of dataset s assigned to topic k are as follows:

$$\theta_{tk} = \frac{C_{tk}^{TK} + \alpha}{\sum_k C_{tk}^{TK} + K\alpha} \tag{10}$$

$$\lambda_{sk} = \frac{C_{sk}^{SK} + \gamma}{\sum_k C_{sk}^{SK} + K\gamma} \tag{11}$$

And the posterior distribution of π_d that describes the probability of choosing a tool or dataset for generating a word, is written as,

$$\begin{aligned} P(\pi_d|\boldsymbol{L}, \boldsymbol{\eta}) &\propto P(\boldsymbol{L}|\pi_d)P(\pi_d|\boldsymbol{\eta}) \\ &\propto \pi_d^{C_t^L}(1-\pi_d)^{C_s^L}\pi_d^{\eta_{\pi_0}-1}(1-\pi_d)^{\eta_{\pi_1}-1} \\ &= \pi_d^{C_t^L+\eta_{\pi_0}-1}(1-\pi_d)^{C_s^L+\eta_{\pi_1}-1} \\ &= \mathrm{Beta}(C_t^L + \eta_{\pi_0}, C_s^L + \eta_{\pi_1}) \end{aligned} \tag{12}$$

The expectation of Beta distribution to estimate the probability of choosing tools or datasets for document d,

$$\pi_d^{L_{dn}=0} = \frac{C_t^L + \eta_{\pi_0}}{C_t^L + C_s^L + \eta_{\pi_0} + \eta_{\pi_1}} \tag{13}$$

$$\pi_d^{L_{dn}=1} = \frac{C_s^L + \eta_{\pi_1}}{C_t^L + C_s^L + \eta_{\pi_0} + \eta_{\pi_1}} \tag{14}$$

V. Performance Evaluation

In this section, we apply our DSTM on large collections of publications from two exemplar scientific domains: *neuroscience* and *bioinformatics*. Specifically, we evaluate the generalization performance of our model with the state-of-art LDA [7] model. Following this, we further demonstrate the benefits of using our model for choosing appropriate tools or datasets for particular computational and data intensive research problems.

A. Datasets

In our model, we used three categories of datasets (papers, tools and datasets) from two scientific domains as shown in Table II: *neuroscience* and *bioinformatics* for understanding the relationships among research topics, tools, and datasets.

- **Papers**: we have collected full papers from well-known journals in neuroscience and bioinformatics domain communities. We removed any words that occurred in less than 3 papers that are supposed to be very infrequent words, or belonged to the list of "stop words" that are supposed to be very frequent words (e.g., "the", "a") in papers. Each paper was represented as a "bag of words" in our model.
- **Tools**: We have collected the most commonly used tools of specific domains: neuroscience and bioinformatics communities separately in collaboration with domain scientists. This list of tools covers a wide range of research efforts in computation, simulation, database and visualization.
- **Datasets**: We have collected common types of datasets used in specific domains individually i.e., within neuroscience and bioinformatics experiments.

B. Generalization Performance Evaluation

The generalization performance is an important factor to evaluate how well a probabilistic model predicts an unobserved sample based on parameters estimation in the training stage. *Perplexity* is standard metric widely used in probabilistic or text modeling to measure the predictive power of a model. A lower perplexity score indicates better generalization performance of held-out test datasets. Formally, the perplexity score of a test document d that contains words $\boldsymbol{w}_d$, and is conditioned on the known tools $\boldsymbol{t}_d$, datasets $\boldsymbol{s}_d$ of the document d, is defined as,

$$\mathrm{perplexity}(\boldsymbol{w}_d|\boldsymbol{t}_d, \boldsymbol{s}_d) = \exp\Big\{-\frac{\mathrm{logP}(\boldsymbol{w}_d|\boldsymbol{t}_d, \boldsymbol{s}_d)}{N_d}\Big\} \tag{15}$$

where $\mathrm{P}(\boldsymbol{w}_d|\boldsymbol{t}_d, \boldsymbol{s}_d)$ is the probability of words $\boldsymbol{w}_d$ conditioned on known tools $\boldsymbol{t}_d$ or datasets $\boldsymbol{s}_d$ in document d, and where the N_d is the number of words in document d. To compute the overall perplexity score of all test documents D_{test}, we simply average the perplexity over test documents:

$$\mathrm{perplexity}(D_{test}) = \frac{\sum_{d=1}^{D_{test}} \mathrm{perplexity}(\boldsymbol{w}_d|\boldsymbol{t}_d, \boldsymbol{s}_d)}{D_{test}} \tag{16}$$

The probability of words $\boldsymbol{w}_d$ in document d with known tools $\boldsymbol{t}_d$ or datasets $\boldsymbol{s}_d$ can be obtained by integrating all latent variables,

$$\begin{aligned} \mathrm{P}(\boldsymbol{w}_d|\boldsymbol{t}_d, \boldsymbol{s}_d) = \prod_{n=1}^{N_d}\sum_{k=1}^{K}\Big[\frac{1}{\boldsymbol{t}_d}\sum_{t=1}^{T}\pi_d^{L_n=0}\phi_{k,w_n}\theta_{kt} \\ + \frac{1}{\boldsymbol{s}_d}\sum_{s=1}^{S}\pi_d^{L_n=1}\phi_{k,w_n}\lambda_{ks}\Big] \end{aligned} \tag{17}$$

Where the $\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\lambda}$ can be estimated through model training stage using Equations 9, 10, 11, respectively. And the π_d needs to be sampled based on the new test documents d. Practically, we run Gibbs sampling Equation 13, 14 with a few iterations to get a stable estimation for each test document.

In our experiments, we compared the generalization performance of our DSTM with the LDA for both dataset collections (i.e., neuroscience and bioinformatics) shown in Table II. In both cases and in both the models, we held out 10% of of same data for testing the generalization performance, and used 90% of same data for training.

Figure 4 shows that the perplexity scores of DSTM are significantly higher than the LDA perplexity scores in both cases initially. This might have been caused by overfitting issues when the number of topics are relatively small.

Table II. Description of collected data for analysis from neuroscience and bioinformatics domain communities.

Category	Neuroscience	Bioinformatics
Papers	We have collected 367 latest computational neuroscience papers from two reputed journal archives: *Frontiers in Computational Neuroscience* and *Journal of Computational Neuroscience* published from 2016 to 2018. This leads to a vocabulary size of $V = 10,719$ unique words and a total of $1,153,047$ word tokens.	We collected 476 latest bioinformatics papers from *Journal of BMC Bioinformatics* published between 2016 to 2018. This leads to a vocabulary size of $V = 9693$ unique words and a total of $1,389,599$ word tokens.
Tools	We have collected the commonly used tools in neuroscience research activities including computation, simulation, database and visualization, such as Matlab, Python, NEURON [3], PyNN [16], ModelDB [17], and new machine learning framework (e.g., TensorFlow, Keras) may be applied in recent neuroscience research. This leads to a total of 46 tools.	We have collected 73 types common used tools which cover a, variety of bioinformatics research works, including sequencing alignment tools (e.g., FASTA, BLAST), genome analysis tools (e.g., GATK, GenomeTools), quality control tools (e.g., FastQC, RSeQc), workflow management tools (e.g., Pegasus), new machine learning framework (e.g., TensorFlow, Keras) and popular programming languages in bioinformatics (e.g., Matlab, Python, R)
Datasets	Datasets described in neuroscience literature are usually recognized by cell types (i.e., pyramidal, interneuron) or brain regions (i.e., neocortex, retina). We collected the common datasets types in neuroscience experiments, which leads to a total of 173 different types datasets.	We have collected types of datasets in bioinformatics, including types of Ribonucleic acid (e.g., rRNA, tRNA, miRNA), types of sequencing (e.g., Chip-seq, Dap-seq, RNA-seq). We also collected some evaluation benchmark (e.g., CASP11, CASP12), and public biology database (e.g., TCGA, CCLE). This leads to a total of 45 types datasets.

However, after increasing the number of topics, the DSTM quickly achieves similar or sightly better generalization performance at $(K = 50)$. As we continually increase the number of topics, the DSTM exhibits significantly better performance than the LDA model at $(K = 100)$ for both dataset collections. Additionally, in both cases, the LDA model's perplexity scores slightly change with the increase in the number of topics, and the different between the maximum and minimum scores are not quite obvious.

The above evaluation results provide insights of an interesting phenomenon where the LDA model has overall better generalization performance for most number of topics; whereas, our DSTM has better performance within a range of particular number of topics. The reason for this phenomenon could be that the LDA has a completely random generative process for producing each word, and our DSTM generates words based on the occurrence of tools or dataset within each document in order to guide our model to reach a target number of topics. We remark that this happens completely in an unsupervised manner. In essence, our DSTM has better performance for finding highly specific topics within a domain, which is more suitable for domain scientists in finding particular resources (such as tools or datasets) to solve computational and data intensive research problems.

C. Model Selection

For hyperparameters (α, β, γ), we followed the suggestions from [15], keeping them constant: $\alpha = \gamma = 50/K, \beta = 0.1$, respectively. Smaller values of β leads the model to generate more topics that address a particular scientific research problem. And the hyperparameter η is simply kept fixed at $\eta = (2, 3)$. This is because, in each paper, the times for mentioning the experimental datasets are usually more than the same for mentioning the tools.

To find the optimal number of topics, we also evaluated the generalization performance *perplexity* with different number of topics for each dataset collections. For all runs of our algorithm, we kept other hyperparameters fixed at $\alpha = \gamma = 50/K, \beta = 0.1, \eta = (2, 3)$, and used the 90% dataset for training and 10% dataset for testing the generalization performance.

As shown in Figure 5, the perplexity scores suggest that the optimal number of topics are $K = 100$ for both neuroscience and bioinformatics dataset collections. The observed scores are extremely high initially, however they quickly reach the optimal values around 100 with the increase in the number of topics, and slightly increase thereafter. The explanation of this phenomenon is same as the one described in the previous Section V-B.

D. Analysis Results obtained from the Experimental Data

We constructed our DSTM with appropriate parameters based on the discussion provided in Section V-C. In order to better illustrate the latent patterns within the data, we use full data for both neuroscience and bioinformatics cases.

For the neuroscience dataset, 400 iterations of the Gibbs sampling algorithm took about 48 hours on a 2.30 GHz CPU Server with 370 seconds per iteration. In comparison, the bioinformatics dataset analysis took around 40 hours for 400 iterations with 355 seconds per iteration on the same server.

Table III illustrates 4 samples of topics from 100 topics learned by DSTM for the neuroscience dataset. These samples are extracted from a single sample at 400^{th} iteration of the Gibbs sampler; and Table IV shows 4 sample topics out of 100 topics for the bioinformatics dataset that are extracted from a single sample at 400^{th} iteration of the Gibbs sampler. Each sub-table in Tables III and IV show the top 10 words that are most likely to be generated conditioned on the topic in the first column; the top 10 most likely tools to be used for the topic in the second column; and the top 10 most likely types of datasets that have come from the topic in the third column.

Table III. 4 sample topics (out of 100 topics in total) extracted for the neuroscience publications from 2016 to 2018. Each topic is associated with 10 most likely words, tools and datasets that have the highest probability conditioned on that topic.

Topic 8					
Word	**Prob.**	**Tool**	**Prob.**	**Dataset**	**Prob.**
burst	.0579	nest	.1275	bursting	.7510
neurons	.0485	matplotlib	.1014	subiculum	.1414
firing	.0440	cplusplus	.0737	pyramidal	.0345
spiking	.0330	matlab	.0734	dopaminergic	.0180
potential	.0311	genesis	.0278	inhibitory	.0061
membrane	.0215	modeldb	.0186	myelinated	.0023
bursts	.0175	neuron	.0069	perisomatic	.0023
bursting	.0175	octave	.0010	excitatory	.0021
correlation	.0174	brian	.0010	ganglion	.0017
voltage	.0172	freesurfer	.0005	dendritic	.0017

Topic 46					
Word	**Prob.**	**Tool**	**Prob.**	**Dataset**	**Prob.**
excitatory	.0852	brian	.1800	inhibitory	.5817
inhibitory	.0758	cplusplus	.1736	excitatory	.2530
neurons	.0397	matlab	.1439	gabaergic	.0669
rate	.0255	nest	.0391	pyramidal	.0502
firing	.0251	neuron	.0226	circuit	.0182
connection	.0213	genesis	.0060	somatic	.0041
population	.0200	matplotlib	.0034	neocortex	.0035
inhibition	.0193	octave	.0007	thalamocortic	.0028
cortical	.0184	modeldb	.0004	parvalbumin	.0021
activity	.0167	pynn	.0001	dopaminergic	.0018

Topic 92					
Word	**Prob.**	**Tool**	**Prob.**	**Dataset**	**Prob.**
stimulation	.0470	neuron	.3352	myelinated	.6243
firing	.0274	matlab	.1385	axon	.2572
neurons	.0244	cplusplus	.0011	astrocyte	.0131
channels	.0241	nest	.0006	vertical	.0112
thresholds	.0209	fmrib	.0006	bursting	.0109
type	.0197	brian	.0006	modulated	.0069
nerve	.0185	octave	.0004	glutamatergic	.0053
electrode	.0177	pymoose	.0001	shepherd	.0047
fibers	.0174	ligplot	.0001	ganglia	.0045
current	.0149	genesis	.0001	pyramidal	.0039

Topic 97					
Word	**Prob.**	**Tool**	**Prob.**	**Dataset**	**Prob.**
brain	.0251	fmrib	.4137	hipp	.3098
regions	.0172	freesurfer	.2994	dorsal	.3054
cortex	.0154	matlab	.0208	callosum	.2993
left	.0139	neuron	.0030	anterior	.0151
region	.0132	nest	.0027	bag	.0066
right	.0132	matplotlib	.0023	amygdala	.0029
hemisphere	.0125	cplusplus	.0016	circuit	.0029
differences	.0113	octave	.0016	hippocampus	.0022
abrupt	.0106	modeldb	.0002	olfactory	.0022
fiber	.0105	genesis	.0002	neuropil	.0019

Table IV. 4 sample topics (out of 100 topics in total) extracted for the bioinformatics publications from 2016 to 2018. Each topic is associated with 10 most likely words, tools and datasets that have the highest probability conditioned on that topic.

Topic 1					
Word	**Prob.**	**Tool**	**Prob.**	**Dataset**	**Prob.**
layer	.0253	keras	.4007	dnaseseq	.0590
cnn	.0244	tensorflow	.3917	casp1	.0553
learning	.0229	python	.1320	tcga	.0144
data	.0221	umls	.0346	mrna	.0114
deep	.0190	blast	.0170	rrna	.0052
drug	.0181	matlab	.0067	wgs	.0029
dataset	.0170	rnastar	.0023	methylation	.0012
lstm	.0133	glmnet	.0014	proteomics	.0012
classification	.0129	ucsc	.0013	cosmic	.0010
training	.0128	sklearn	.0008	trna	.0003

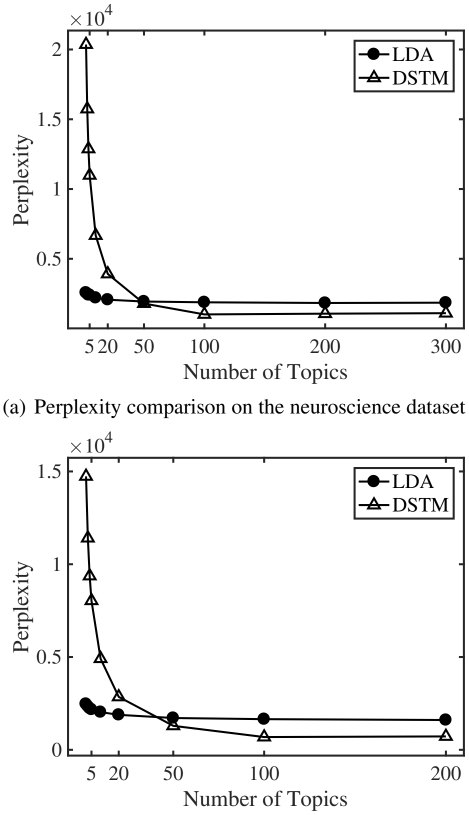
Topic 36					
Word	**Prob.**	**Tool**	**Prob.**	**Dataset**	**Prob.**
sequence	.0356	clustalw	.7042	mrna	.4577
sequences	.0340	fasta	.2042	rrna	.1154
alignment	.0180	blast	.0649	rnaseq	.0614
based	.0150	samtools	.0043	sirna	.0038
dataset	.0145	cplusplus	.0027	trna	.0034
binding	.0131	sklearn	.0025	chipseq	.0022
distance	.0115	tensorflow	.0012	wgs	.0007
algorithm	.0110	edger	.0012	tcga	.0005
methods	.0104	umls	.0012	cosmic	.0003
family	.0101	emboss	.0010	methylation	.0003

Topic 43					
Word	**Prob.**	**Tool**	**Prob.**	**Dataset**	**Prob.**
reads	.1190	bwa	.5247	wgs	.1848
tools	.0272	bowtie	.2429	rnaseq	.1566
error	.0251	fasta	.0877	chipseq	.0848
reference	.0248	samtools	.0855	methylation	.0826
genome	.0247	blast	.0340	rrna	.0626
alignment	.0235	edger	.0084	mrna	.0474
sequencing	.0161	gatk	.0045	dnaseseq	.0092
quality	.0152	htseq	.0019	cosmic	.0059
low	.0142	pfam	.0014	tcga	.0009
end	.0136	tophat	.0011	proteomics	.0004

Topic 51					
Word	**Prob.**	**Tool**	**Prob.**	**Dataset**	**Prob.**
disease	.0296	umls	.9660	rrna	.0040
question	.0246	keras	.0123	mirna	.0025
mentions	.0207	cufflinks	.0023	mrna	.0017
query	.0165	rpackage	.0018	rnaseq	.0012
types	.0164	python	.0014	cosmic	.0012
disorder	.0156	samtools	.0012	sirna	.0008
information	.0147	cplusplus	.0010	casp1	.0008
semantic	.0147	ucsc	.0010	tcga	.0008
questions	.0144	fasta	.0010	wgs	.0008
set	.0142	htseq	.0008	methylation	.0006

1) Neuroscience Domain Dataset Example Discussion: The topics within the neuroscience dataset collection in Table III fall into recognizable areas as perceived by a domain scientist collaborator. Topic 8 seems to capture single cell models of the integrate and fire type that can be easily modeled using packages such as NEST, Bursting, including differences with spiking can be studied using a variety of datasets as listed. Topic 46 is somewhat related to Topic 8, but may be more on the area of network models since the words ‘connection’ and ‘population’ are unique to that topic. Note that the tools for both topics are somewhat related, and so the difference may be in the type of single or network models being studied. Topic 92 is somewhat distinct from the previous two in that – it may represent single cell models with more biological details such as those using the Hodkin-Huxley formulation modeled with packages such as NEURON. Words in this topic include channels, thresholds, electrode, fiber, currents, etc. which do not show up in the previous topics. Finally, the proposed approach captures the distinct, important as well as a growing topic of higher level modeling that uses brain imaging data from varied regions (hippocampus, amygdala, etc.) and tools such as fmrib, freesurfer, etc.

2) Bioinformatics Domain Dataset Example Discussion: The results from the bioinformatics dataset collection in Table IV shows Topic 1 representing the deep learning research



(a) Perplexity comparison on the neuroscience dataset

(b) Perplexity comparison on the bioinformatics dataset

Figure 4: Perplexity comparison with LDA model on different datasets collections, for different number of topics.

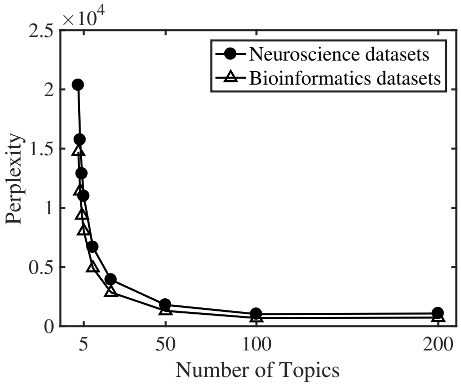


Figure 5: DSTM selection on neuroscience and bioinformatics dataset collections, for an increasing number of topics.

area in bioinformatics, as perceived by a domain scientist collaborator. In this area, the most commonly selected deep learning frameworks tools are Keras and TensorFlow. From Topic 1 results, we can also conclude that this topic is more specific to tools and not the dataset as shown by its very low probability. This is because, in the model, each word is contributed by either a tool or a dataset. Topic 36 describes the sequence alignment research area, for which the most popular tools of sequence alignments are ClustaW or Fasta, and the dataset types selected for sequence alignment are mRNA, rRNA etc. Topic 44 is related to alignment for reference genomes and reads, for which tools like BWA, Bowtie and datasets like Whole genome sequencing (WGS), RNA-seq or Chip-seq are most commonly used. Topic 51 describes the topic about the biomedical information semantic query research area, where Unified Medical Language System is a highly recommended tool used for this kind of study.

Table V. Sample tools case study results in bioinformatics.

Topics	**Tool = TensorFlow**
Topic 1	layer, cnn, learning, data, deep, drug, dataset, lstm, classification, training
Topic 90	entities, sentence, biomedical, corpus, words, entity, word, relations, extraction, use
Topic 86	model, results, features, performance, set, methods, binding, prediction, table, different

Topics	**Tool = ClustalW**
Topic 36	sequence, sequences, alignment, based, dataset, binding, distance, algorithm, methods, family
Topic 40	size, approach, trees, use, tree, method, respectively, shows, shown, maximum
Topic 47	data, performance, study, dataset, additional, procedure, case, defined, file, specific

3) Choosing Suitable Tools for a Research Problem: The DSTM can be beneficial in choosing suitable tools for a research problem in a specific domain as shown in the results of Table V. For this case study of DSTM, we choose two sample tools from the bioinformatics domain. The result of TensorFlow (a deep learning framework) shows that scientists often in bioinformatics use the CNN model for classification task in Topic 1; whereas, scientists in some cases also use TensorFlow for semantic extraction or mining in Topic 90; Topic 86 is somewhat using TensorFlow for achieving better performance. Results for the ClustalW tool show that it is a common tool used for genome sequence alignment in Topic 36. Moreover, scientists also tend to use it for analysis of phylogenetic trees, a rare fact that is captured by our DSTM analysis.

Based on the above discussions, we can conclude that our experiments results from the neuroscience and bioinformatics datasets (quantitively, and as qualitatively perceived by domain science collaborators) show that our DSTM effectively extracts meaningful and useful guidance from large collections of datasets to help a domain scientist in choosing pertinent tools or datasets for a particular research problem at hand. With our DSTM, domain scientists can also efficiently digest the whole results in a few minutes to obtain relevant key knowledge patterns, instead of manually surveying (slow approach) large literature archives for obtaining similar information.

VI. Conclusion

In this paper, we proposed a novel "domain-specific topic model" (DSTM) that can be used within conversational agents to help users to discover latent knowledge patterns among research topics, tools and datasets for computational and data intensive scientific communities. The DSTM provides an efficient and effective method because of its design to incorporate little or any amount of domain knowledge while exploring highly specific topic patterns within a given domain. Although our DSTM approach extends the popular LDA model, it is uniquely suited for topic digests with little or any amount of domain knowledge. Further, it uses a completely randomly generative process (in contrast to the LDA model) in order to generate words based on reference tools or datasets. Using large collections of two types of text corpus from neuroscience and bioinformatics domains, our evaluation experiments with quantitative perplexity scores and qualitative domain scientist feedback showed that our model has better generalization performance for revealing highly specific latent topics within a domain. Our experiment findings also demonstrated that chatbots within science gateway portals can use the DSTM in user dialogs to provide helpful knowledge patterns among research topics, tools and datasets for solving multi-disciplinary research problems within computational and data intensive scientific communities.

Our work in this paper on topic-based recommenders that are domain-specific can aid the relevant knowledge discovery during the adaptive response generation for the conversational agents dialog with users within science gateways. Moreover, our domain-specific topic model is trained using unsupervised machine learning and can be extended to easily query additional information from diverse text corpus comprising of a wide-ranging set of topics.

Possible future directions for this work include building visualization interfaces involving science gateway chatbots to browse the knowledge patterns among research topics, tools and datasets. Such dialog interfaces can foster the efficient query to obtain appropriate resources (e.g., tools, and datasets) for cutting-edge research investigations. Our DSTM can also be integrated within a conversational agent to recommend proper resources to domain scientists based on particular topics of interest. Lastly, future work could be pursued to extend the DSTM to address cross-domain knowledge pattern discovery.

Acknowledgment

This material is based upon work supported by the National Science Foundation under award number OAC-1730655. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Y. Liu, S. M. Khan, J. Wang, M. Rynge, Y. Zhang, S. Zeng, S. Chen, J. V. Maldonado dos Santos, B. Valliyodan, P. P. Calyam, N. Merchant, H. T. Nguyen, D. Xu, and T. Joshi, "Pgen: large-scale genomic variations analysis workflow and browser in soykb," *BMC Bioinformatics*, vol. 17, no. 13, p. 337, Oct 2016.

[2] E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz, "Pegasus: a framework for mapping complex scientific workflows onto distributed systems," *Scientific Programming Journal*, vol. 13, no. 3, pp. 219–237, 2005.

[3] N. T. Carnevale and M. L. Hines, *The NEURON book*. Cambridge University Press, 2006.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[5] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *UAI'04*. Arlington, Virginia, United States: AUAI Press, 2004, pp. 487–494.

[6] D. Mimno and A. McCallum, "Expertise modeling for matching papers with reviewers," in *KDD'07*. New York, NY, USA: ACM, 2007, pp. 500–509.

[7] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *ICML'06*. New York, NY, USA: ACM, 2006, pp. 113–120.

[8] J. D. Lafferty and D. M. Blei, "Correlated topic models," in *NIPS'06*, Y. Weiss, B. Schölkopf, and J. C. Platt, Eds. MIT Press, 2006, pp. 147–154.

[9] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *CVPR'05*, vol. 2, June 2005, pp. 524–531 vol. 2.

[10] P. Flaherty, G. Giaever, J. Kumm, M. I. Jordan, and A. P. Arkin, "A latent variable model for chemogenomic profiling," *Bioinformatics*, vol. 21, no. 15, pp. 3286–3293, 2005.

[11] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *KDD'11*. New York, NY, USA: ACM, 2011, pp. 448–456.

[12] H. Sun, M. Srivatsa, S. Tan, Y. Li, L. M. Kaplan, S. Tao, and X. Yan, "Analyzing expert behaviors in collaborative networks," in *KDD'14*. New York, NY, USA: ACM, 2014, pp. 1486–1495.

[13] J. Tang, S. Wu, J. Sun, and H. Su, "Cross-domain collaboration recommendation," in *KDD'12*. New York, NY, USA: ACM, 2012, pp. 1285–1293.

[14] M. Hoffman, F. R. Bach, and D. M. Blei, "Online learning for latent dirichlet allocation," in *NIPS'10*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 856–864.

[15] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *PNAS'04*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.

[16] A. P. Davison, D. Brüderle, J. M. Eppler, J. Kremkow, E. Muller, D. Pecevski, L. Perrinet, and P. Yger, "Pynn: a common interface for neuronal network simulators," *Frontiers in neuroinformatics*, vol. 2, p. 11, 2009.

[17] M. Migliore, T. M. Morse, A. P. Davison, L. Marenco, G. M. Shepherd, and M. L. Hines, "Modeldb," *Neuroinformatics*, vol. 1, no. 1, pp. 135–139, Mar 2003.