Domain-specific Topic Model for Knowledge Discovery in Computational and Data-Intensive Scientific Communities

Yuanxun Zhang, Prasad Calyam, *Senior Member, IEEE,* Trupti Joshi, Satish Nair, Dong Xu

Abstract—Shortened time to knowledge discovery and adapting prior domain knowledge is a challenge for computational and data-intensive communities such as e.g., bioinformatics and neuroscience. The challenge for a domain scientist lies in the actions to obtain guidance through query of massive information from diverse text corpus comprising of a wide-ranging set of topics when: investigating new methods, developing new tools, or integrating datasets. In this paper, we propose a novel "domain-specific topic model" (DSTM) to discover latent knowledge patterns about relationships among research topics, tools and datasets from exemplary scientific domains. Our DSTM is a generative model that extends the Latent Dirichlet Allocation (LDA) model and uses the Markov chain Monte Carlo (MCMC) 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 more than 25,000 of papers over the last ten years, featuring hundreds of tools and datasets that are commonly used in relevant studies. Evaluation experiments based on generalization and information retrieval metrics show that our model has better performance than the state-of-the-art baseline models for discovering highly-specific latent topics within a domain. Lastly, we demonstrate applications that benefit from our DSTM to discover intra-domain, cross-domain and trend knowledge patterns.

Index Terms—Topic Model, Theoretical Model for Big Data, Latent Dirichlet Allocation, Multi-disciplinary Knowledge Discovery

1 Introduction

Scientific domains such as bioinformatics and neuroscience have the potential to benefit from Big Data analytics that uses underlying machine learning techniques for solving computational and data-intensive research problems. Bold innovations will increasingly emerge from processing a large number of datasets or recognizing complex knowledge patterns using text mining. Moreover, the bold innovations will occur from solving multi-disciplinary research problems that require prior knowledge discovery within disciplines and from cross-domain scientist collaborations. To enable the rapid pace of innovation, 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. As shown in Fig. 1, it involves both intra-domain and cross-domain knowledge discovery. An *intra-domain* scenario commonly occurs when scientists are looking to adapt a state-of-the-art solution in their domain.

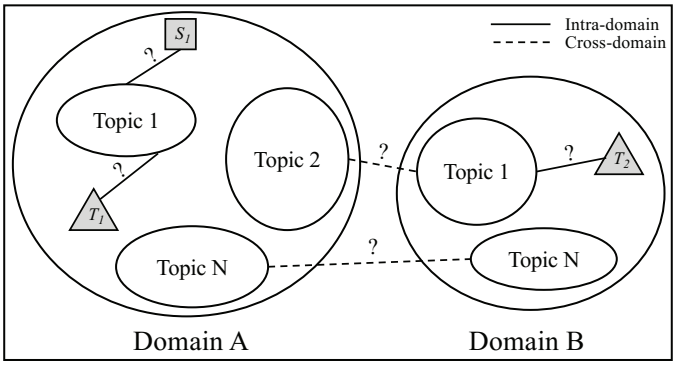


Fig. 1: Discovery of relationships between research topics, tools (with "T" notation) and datasets (with "S" notation) for scientific domains with intra-domain knowledge and cross-domain knowledge discovery.

For example, biologists wanting to know if there is a new tool developed for improving the performance of sequence alignment. Alternately, a *cross-domain* scenario can be seen when scientists are investigating new solutions by extending relevant methods from other domains. A few example scenarios are as follows: biologists applying relevant machine learning and statistical methods for protein structure predictions; machine learning studies may need to extend new algorithms/tools to solve unique problems in personalized medicine; data-intensive neuroscience efforts could adopt cyberinfrastructure integration best practices from bioinformatics [1] for building workflows across distributed computing resources.

In this paper, we propose a novel "domain-specific topic model" (DSTM) to enable the discovery of latent knowledge

• *Y. Zhang, P. Calyam, S. Nair, T. Joshi and D. Xu are with the Department of Electrical Engineering and Computer Science, University of Missouri–Columbia, Columbia, MO, 65211.*
E-mail: yzd3b@mail.missouri.edu, calyamp@missouri.edu, joshitr@health.missouri.edu, nairs@missouri.edu, xudong@missouri.edu,

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.

patterns in scientific domains that rely on prior knowledge discovery and cross-domain collaborations. DSTM is fundamentally a generative model to discover the relationships among research topics, tools, and datasets within intra-domain and cross-domain cases. Our DSTM extends the existing topic models (such as Latent Dirichlet Allocation (LDA) [2], PLSA [3]) that only consider topics or relationships between topics. It is the first to handle documents by modeling the relationship among research topics, tools, and datasets within documents that are relevant to particular scientific communities. Our main focus goes beyond discovering topics among scientific publications and reveals the relevant knowledge patterns among topics, tools, and datasets. Such an approach allows us to understand trends, as well as cross-domain patterns of tools/datasets, use within scientific communities. Such insights can in turn provide pertinent guidance for scientists to choose suitable tools/datasets or observe potential trends for future research purposes.

Our DSTM assumes each topic is represented as a distribution 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 the Markov chain Monte Carlo inference algorithm for a specific domain. In order to tangibly apply our DSTM for research activities in scientific communities, we propose three algorithms to apply our DSTM under different knowledge pattern perspectives: (a) *intra-domain knowledge representation* that guides scientists to find existing solutions from their respective domains; (b) *cross-domain knowledge representation* that guides investigation of new solutions by referring to prior solutions in other synergistic domains; (c) *trend representation* that tracks the change in trends or reveals emerging trends over time, in order to guide scientists to make intelligent decisions while choosing tools or datasets to solve a research problem at hand.

We evaluate the performance of DSTM and reveal the latent patterns in large collections of scientific publications from reputed journal archives belonging to two exemplar scientific domains: neuroscience and bioinformatics. Specifically, we collect $16,721$ papers from reputed neuroscience journals including the areas of theory and computation, $10,681$ papers from well-known bioinformatics journals including the fields of genome computation, over the last ten years. We extract 30% contents for each document/paper to generate the results in our evaluation experiments, which may contain all the contents of a paper's abstract and introduction for understanding the paper's topics. We generate a combined vocabulary size of $V = 19,488$ terms. We also collect the names of hundreds of tools and types of datasets that are commonly used in these domains. Our evaluation results feature a perplexity metric that can measure the DSTM performance in revealing the highly-specific latent topics in accordance with a user's exploration scope of a catalog of tools or datasets. We demonstrate our model in different aspects to provide insightful guidance for choosing pertinent tools and datasets for solving cutting-edge domain research problems. Given our design of DSTM, our model can be easily extended with satisfactory generalization performance to other domains (e.g., material science) by changing the datasets relating to the types of publications, tools, and datasets.

The remainder of the paper is organized as follows: Section 2 describes the problem motivation in greater detail. Section 3 discusses the related works. In Section 4, we describe our DSTM's generative process, inference algorithms for model parameter estimation, and three exemplar applications for use of DSTM to discover latent *intra-domain*, *cross-domain*, and *trend* knowledge patterns. In Section 5, we discuss the datasets that we used for DSTM evaluations and demonstrations. Section 4.2 discusses the model parameter selections for achieving optimal generalization performance based on the datasets. We also provide comparisons of generalization and information retrieval performance of DSTM with state-of-the-art models. Application demonstrations for insightful knowledge discovery cases are detailed in Section 7. Section 8 concludes the paper.

2 Problem Motivation

One of the major challenges in obtaining useful guidance through query of massive information is to discover knowledge patterns amongst diverse text corpus comprising of a wide-ranging set of topics. With the access to such knowledge pattern digests that feature a pertinent list of topics, tools and data sets, it is possible for domain scientists to more easily answer research 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?"; "Which deep learning tool is most popular for a specific bioinformatics research problem?"; Given that computational and data-intensive research problems are expensive and time-consuming to solve, relevant topic models are critical to provide useful guidance through text mining of massive open information within a scientific domain or across domains. They can significantly benefit domain scientists by drastically shortening the time to knowledge discovery and adaptation of prior domain knowledge for their innovations.

In our preliminary experiments that involved manually querying/surveying the publications from neuroscience and bioinformatics domains, we found that common knowledge patterns within a domain, as well as across the domains, can be useful to domain scientists. We found by observing novice/expert researchers that significant text corpus relating to popular tools (e.g., Pegasus [4] in bioinformatics and NEURON [5] in neuroscience) and datasets (e.g., RNA, Interneuron) are frequently used as guidance using a manual (slow/inefficient) approach. Also, the latest computational and data-intensive research problems in neuroscience tend to be influenced by efforts in prior bioinformatics literature that successfully accomplished investigations of 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.

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 knowledge base matures, its text corpus increases in size/variety, and new developments arise in tools or datasets for improving the state-of-the-art solutions. For instance, observing Fig. 1, uncertainty can occur when efficiently determining whether the tool T_2 is an appropriate tool for solving a problem related to Topic 1 in a scientific domain B. Alternately, uncertainty can also occur when scientists from domain A are looking for referring relevant solutions from another domain B by exploring similar topics from corresponding literature. Therefore, the design of an ideal topic model should be scalable and flexible to deal with daily/monthly/yearly changes in the Big Data "volume", "velocity", "variety" and "value" within scientific domains, and satisfy knowledge discovery related query needs on state-of-the-art problems for domain scientists.

In the context of the knowledge discovery problem, scientists can leverage our model to find suitable "tools/datasets" for different research topics. In addition, our DSTM can also be effectively applied to different scientific or engineering communities based on their research needs by replacing the "tools/datasets" with their specific interests. For example, in medical science, our model can help researchers/clinicians to find relevant research topics for certain drugs or genes; in material sciences, our model can help researchers/engineers to discover the topics based on certain material properties.

3 Related Work

Prior related works can be organized under three broad categories: (a) probabilistic topic models; (b) inference algorithms; (c) deep learning in topic models; and (d) cross-domain recommendations.

Probabilistic Topic Models. Topic model is a suite of algorithms that are used to extract useful information from text corpus in an unsupervised learning manner. LSA [6] and pLSA [3] are a few of the early topic models that decompose documents into latent vector representations using the SVM and the probabilistic model, respectively. In recent years, Latent Dirichlet Allocation (LDA) [2] is the most widely used topic model that was invented by Blei *et al.* in 2003. It discovers the latent topic structures from a collection of documents or text corpus automatically with a Bayesian hierarchical model. In LDA, each topic is modeled as a distribution over words, and each document is represented as a mixture over topic proportions. The LDA model has been widely applied to document classification, searching, and recommendation.

Based on the LDA model, many researchers have tried to extend it for discovering interesting patterns of documents. Rosen-Zvi *et al.* [7] proposed an Author-Topic model that extends the LDA by including authorship information to establish the relationships between topics and authors. Relationships between topics and authors are explored by representing each author with a mixture of weights for different topics.

Mimno *et al.* [8] also proposed a similar author topic model for matching papers with peer-reviewers. Blei *et al.* [9] proposed a dynamic topic model (DTM) in order to extract the evolution of topics within sequentially organized documents. Blei and Lafferty in [10] 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 latent based topic modeling to different areas. Li *et al.* [11] adapted the LDA model for image scene categorization without any human annotations, which achieved comparable performance. Flaherty *et at.* [12] developed a model that is able to cluster genes within experiments that do not require inputs of a gene or drug. Wang *et al.* [13] combined the collaborative filtering and probabilistic topic models (i.e., LDA variants) in a recommendation system to recommend scientific articles. Their model leverages collaborative filtering and a topic model to perform matrix factorization that decomposes a rating matrix into latent users and items structure. Luo *et al* [14] proposed a generative probabilistic model for optimizing workforce personnel allocation using employee-activity logs. They used latent variables to learn hidden patterns between employees and activities in terms of their job performance. Zhao *et al* [15] applied LDA to biological or medical datasets for clustering analysis.

The drawback in the existing topic models is that none of them utilize any kind of domain-specific knowledge to explore specific latent patterns that are meaningful for particular research problems involving tools and datasets. Consequently, they are not suitable for the domain-specific topic modeling problems that we address in our DSTM approach for the knowledge discovery problems in computational and data-intensive scientific communities or other domains based on users' interests. Moreover, in contrast with the LDA, our DSTM not only discovers what topics are expressed in a published document, but also provides insightful information about the pertinent tools or datasets that are associated with each topic. Our DSTM is the first topic model to discover the latent knowledge patterns among research topics, tools, and datasets for scientific communities.

Inference Algorithms. Inferring the latent variables in a probabilistic model (such as the LDA) has also been an area of active research investigations. The original LDA work [2] used a variational expectation maximization algorithm to estimate latent parameters. Hoffman *et al.* [16] designed an online stochastic optimization with a natural gradient step. Their optimization results showed the variational Bayes objective function convergence to a local optimum. Griffiths *et at.* [17] presented a collapsed Gibbs sampling algorithm (i.e., a Markov chain Monte Carlo method) to infer latent parameters of their model. In general, MCMC estimates the latent variables using sampling methods [18]. In contrast, variational inference uses optimization methods [19] to minimize the lower bound. Hence, MCMC may need more computation time to achieve similar performance as the variational inference. In comparison to the variational inference, Gibbs sampling has a low bias but with a high variance. In our work, we do not consider computation time as a factor, thus we use Gibbs sampling as our inference algorithm and mitigate the issues of high variance by run-

ning multiple Markov chains. Moreover, Gibbs sampling is slightly easier to implement without comprising much of the learning speed and the generalization performance.

Deep Learning in Topic Models. In recent years, the deep latent model is also a popular area that has been investigated in many prior works. In general, deep latent models utilize the deep neural network to infer latent variables. In 2013, Kingma *et al.* [20] proposed the VAE model that uses the Stochastic Gradient Variational Bayes (SGVB) with a reparameterization trick to approximate the posterior distribution. VAE model shows an efficient variational inference method using stochastic gradient descent. Prior works have started to use this method to infer latent variables in the topic model. Miao *et al.* [21] proposed a neural variational document model (NVDM) to present documents with the latent variables using variational autoencoder. Srivastava *et al.* [22] presented an autoencoding variational Bayes (AEVB) based model ProdLDA for efficient LDA inference, which demonstrated better performance in topic coherence, computational efficiency, and simplicity. Deep learning methods have shown their efficiency in inference and simplicity without requiring rigorous mathematical derivations. However, they have limitations in the aspects of interpretability and controllability of learning the causal relationships based on human knowledge.

Cross-domain Recommendations. Cross-domain recommendation or cross-domain collaboration is an important study area that helps to transfer knowledge from one domain to another domain effectively. Associated methods exploit knowledge for auxiliary (synergistic) domains containing pertinent information to improve knowledge utility in a target domain. For example, we can learn users' preferences using their purchase records in the "Movie" domain to recommend possible book items. The common approaches for cross-domain recommendation are collaborative filtering (including matrix factorization or factorization machines techniques) [23] when dealing with a structured dataset (e.g., movie or book ratings). Probabilistic modeling is often applied when dealing with an unstructured dataset (e.g., text, logs). Li *et al.* [24] proposed a Topic Correlation Analysis (TCA) model for cross-domain text classification, which extracts both the shared and the domain-specific latent features to transfer knowledge from a given source domain to a target domain. Gao *et al.* [25] proposed a supervised cross collection Latent Dirichlet Allocation (scLDA) model that extends the traditional LDA model for dealing with data from multiple data collections. Sun *et al.* [26] proposed a probabilistic generative model to explore the expert behaviors in collaborative networks by analyzing IBM ticket tracking logs. Tang *et al.* [27] adapted the LDA and Author-Topic model [7] to find potential cross-domain collaborations. They computed the chance of collaborations between two researchers based on their research topics and existing connections among these two domains. Sleeman in [28] applied a Dynamic Topic Model (DTM) to extract topics for cross-domain analysis. Zhao *et al.* [29] used embedding techniques for product review sentiment analysis. Shi *et al.* [30] embedded heterogeneous information to provide auxiliary data in a recommendation system using a meta-path based random walk strategy. Zheng *et al.* [31] surveyed the methods to fuse data from multiple domains. Min *et al.* [32] proposed a cross-platform multi-modal topic model (CM^{3}TM) for inter-platform recommendations that can differentiate the shared topics among different platforms and align multiple modalities.

However, scientists or researchers in most cases want to have an efficient way to explore synergies from other domains in an interactive manner. Hence, instead of only recommending and ranking possible solutions with scores, our work provides a visualization method with a topic embedding algorithm that enables domain scientists/researchers to self-explore the closest synergistic topics in a low dimension space. Grbovic *et al.* [33] proposed items and users embedding techniques for comparing the similarity between items. t-SNE [34] is a popular algorithm that can map high dimensional datasets to a lower dimensional space (such as 2D or 3D space) for better visualization. However, it is not suited to be directly applied for multivariate distribution (i.e., multinomial distribution in our case). Lee *et al.* [35] also developed a visualization solution to easily explore plots and tables in scientific articles. They used machine learning or deep learning methods to classify 8 million figures into five types (e.g., plots, tables, equations). However, these classification types do not reflect the relevant latent topics of knowledge patterns, and our approach makes it significantly easier to interpret topic associations in the processed information of cross-domain collaborations.

TABLE 1. 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 words in the vocabulary
N_d	the number of word tokens in a document d
$\mathbf{t}_d$	a set of tools in a document d
$\mathbf{s}_d$	a set of dataset in a document d
$w_{dn}, \mathbf{w}$	the n^{th} word token in a document d
$x_{dn}, \mathbf{x}$	tool indicator chosen from $\mathbf{t}_d$ for word w_{dn}
$y_{dn}, \mathbf{y}$	dataset indicator chosen from $\mathbf{s}_d$ for word w_{dn}
$z_{dn}, \mathbf{z}$	the topic assignment for word w_{dn}
$L_{dn}, \mathbf{L}$	binary indicator to label which is responsible for z_{dn}
$\pi_d, \boldsymbol{\pi}$	Bernoulli parameter for generating label $\mathbf{L}$ for a 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 a topic z
$\theta_t, \boldsymbol{\theta}$	multinomial distribution over topics specific to a tool t
$\lambda_s, \boldsymbol{\lambda}$	multinomial distribution over topics specific to a dataset s
α, β, γ	parameters of symmetric Dirichlet priors

4 Domain-specific Topic Model

In this section, we first detail our Domain-specific Topic model (DSTM) in terms of its generative process, inference algorithm, and model parameter estimation. Following this, we present three possible applications for the DSTM to be used for the discovery of knowledge patterns for guiding researchers in computational and data-intensive scientific communities.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3093350, IEEE Transactions on Knowledge and Data Engineering

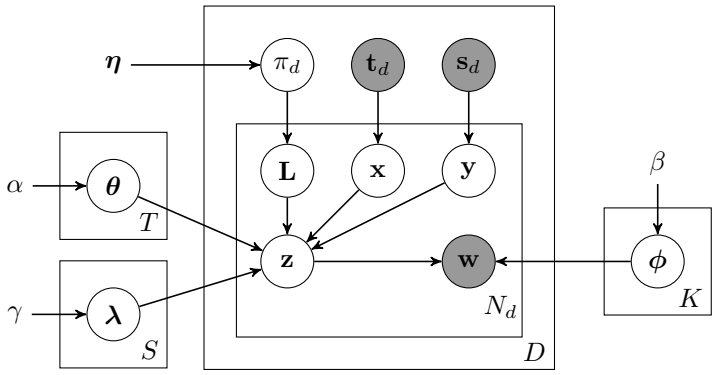


Fig. 2: Graphical representation of the generative model. The boxes are "plates" representing replicates; the "shaded" nodes are observed variables; the "unshaded" nodes are unobserved variables; the nodes without cycle are hyperparameters; See Table 1 for node notations.

4.1 The Generative Model

The purpose of our DSTM generative model is to discover the latent knowledge patterns underlying a collection of documents for a specific scientific domain in terms of the relationships among research topics, tools, and datasets. The generative model refers to our text modeling approach, where we 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 datasets occurrence in a document. In the generative process, we do not assume that a tool and/or dataset is responsible for a certain word simultaneously. For simplifying the computational complexity, each word is generated by either a tool or a dataset. Considering this sentence that is extracted from the BMC Bioinformatics journal: "We tested the performance of three widely used short-read alignment tools (*BWA*, *Bowtie* and *Bowtie2*) on simulated sequencing runs of varying coverage", we can intuitively infer that the tools *BWA*, *Bowtie* and *Bowtie2* contribute to generate the bioinformatics topic "short-read alignment tools".

The graphical representation of our generative model is illustrated in Fig. 2 using a plate notation with all of the various notations summarized in Table 1. During the pre-processing stage, we collect papers from particular collections of documents $D = \{d_1, ..., d_n\}$. Next, we label the corresponding 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. 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}. T denotes the total number of tools and S denotes the total number of dataset categories we created in a catalog. In each document d, the word is an observed variable with "shaded" color, and the other observed variables are a set of tools $\mathbf{t}_d$ and a set of dataset categories $\mathbf{s}_d$. In the model, we assume that there are K number of topics for collection documents D. $\boldsymbol{\phi}$ denotes the $K \times V$ matrix of topics distribution over vocabulary V. $\boldsymbol{\theta}$ 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 a binary indicator variable to label whether the topic assignment is from the tool distribution $\boldsymbol{\theta}$ or the dataset distribution $\boldsymbol{\lambda}$.

Algorithm 1 Generative process in the model

for each topic $k = 1, ..., K$ **do**
  Draw a multinomial over vocabulary $\phi_k \sim Dir(\beta)$;
end for
for each tool $t = 1, ..., T$ **do**
  Draw a multinomial over topics $\theta_t \sim Dir(\alpha)$;
end for
for each dataset $s = 1, ..., S$ **do**
  Draw a multinomial over topics $\theta_t \sim Dir(\gamma)$;
end for
for each doc $d = 1, ..., D$ **do**
  Sample binary indicator $L_{dn} \sim Bern(\pi_d)$;
  if $L_{dn} == 0$ **then**
    Select a tool $x_{dn} \sim Unif(\mathbf{t}_d)$;
    Sample a topic $z_{dn} \sim Multi(\theta_{x_{dn}})$;
  end if
  if $L_{dn} == 1$ **then**
    Select a dataset $y_{dn} \sim Unif(\mathbf{s}_d)$;
    Sample a topic $z_{dn} \sim Multi(\lambda_{y_{dn}})$;
  end if
  Choose a word $w_{dn} \sim Multi(\phi_{k=z_{dn}})$;
end for

Algorithm 1 describes the generative process of the model. First, each topic is associated with a multinomial distribution over V vocabulary drawn from symmetric $Dirchlet(\beta)$ prior. Each tool t draws a multinomial distribution over topics from $Dirchlet(\alpha)$ prior, represented by θ_t. And each dataset s draws a multinomial distribution over topics from $Dirchlet(\gamma)$ prior, denoted as λ_s. Second, for each word in document d, we draw a binary indicator $\mathbf{L}$ from $Bernoulli(\pi_d)$ distribution to decide whether this word is generated by a tool or a dataset. The $Bernoulli(\pi_d)$ distribution is applied when both $\mathbf{t}_d$ and $\mathbf{s}_d$ are not empty. If either of them is empty, the $\mathbf{L}$ is assigned to the non-empty one. Then, a tool or a dataset is chosen from either a set of tools ($\mathbf{t}_d$) or a set of datasets ($\mathbf{s}_d$) 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.

By estimating the latent variables $\{\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{z}, \mathbf{x}, \mathbf{y}, \boldsymbol{\pi}, \mathbf{L}\}$ of the model, we obtain information about topics of the collection of documents, and which tools or datasets are pertinent to be used for a particular research problem.

4.2 Inference and Parameter Estimation

In this subsection, we describe the algorithm to estimate the latent variables by using the Gibbs sampling method [17]. We choose the Gibbs sampling as our inference algorithm, mainly because of its simplicity. Given the high variance issues in Gibbs sampling, we run multiple Markov chains to estimate posterior distribution. More specifically, we explain below how we use the Gibbs sampling method [17] to infer and estimate latent variables $\{\boldsymbol{\phi}, \theta, \boldsymbol{\lambda}, \mathbf{z}, \mathbf{x}, \mathbf{y}, \boldsymbol{\pi}, \mathbf{L}\}$ of the model. To build the Gibbs sampling, we construct a posterior distribution of latent variables conditioned on all other variables, and repeatedly sample from the conditional probability distribution until it converges to a target or equilibrium 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 $\{\boldsymbol{\phi}, \theta, \boldsymbol{\lambda}, \boldsymbol{\phi}\}$ 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 $\mathbf{x}, \mathbf{y}, \mathbf{z}$, we can still estimate $\{\boldsymbol{\phi}, \theta, \boldsymbol{\lambda}\}$ through posterior distribution. To simplify equations, we define the set of hyperparameters as $\boldsymbol{\Omega} = \{\alpha, \beta, \gamma, \boldsymbol{\eta}, T, S\}$.

4.2.1 Inference Algorithm

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} jointly as a block $(L_{dn} = 0, z_{dn}, x_{dn})$ or $(L_{dn} = 1, z_{dn}, y_{dn})$ conditioned on all other variables. Then, the full conditional probability for considering the n^{th} word generated by tool $(L_{dn} = 0, z_{dn} = k, x_{dn} = t)$ is as follows:

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

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 a word in document d, $\mathbf{L}_{-dn}$ 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 $C_{tk,-dn}^{TK}$ 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 $C_{vk,-dn}^{VK}$ denotes excluding the current instance.

Similarly, the full conditional probability considering the n^{th} word generated by dataset $(L_{dn} = 1, z_{dn} = k, y_{dn} = s)$ is as follows:

$$\begin{aligned}\mathrm{P}(L_{dn} = 1, z_{dn} = k, y_{dn} = s | \mathbf{L}_{-dn}, \mathbf{y}_{-dn}, w_{dn}, \\ \mathbf{z}_{-dn}, \mathbf{w}_{-dn}, \mathbf{x}, \boldsymbol{\Omega}) \\ = \frac{C_s^L + \eta_{\pi_1} - 1}{C_t^L + C_s^L + \eta_{\pi_0} + \eta_{\pi_1} - 1} \\ \times \frac{C_{sk,-dn}^{SK} + \gamma}{\sum_k C_{sk,-dn}^{SK} + K\gamma} \\ \times \frac{C_{vk,-dn}^{VK} + \beta}{\sum_v C_{Vk,-dn}^{VK} + V\beta}\end{aligned} \tag{2}$$

where C_{sk}^{SK} represents the number of times dataset s is assigned to topic k, with the subscript $C_{sk,-dn}^{SK}$ denoting the exclusion of the current instance.

Having obtained the full conditional distributions from Equations 1 and 2, the whole Gibbs sampling algorithm is straightforward. First, we initialize the variables $\{\mathbf{L}, \mathbf{z}, \mathbf{x}, \mathbf{y}\}$ randomly. Then, in each iteration, we update $\{\mathbf{L}, \mathbf{z}, \mathbf{x}, \mathbf{y}\}$ in turn from the full conditional distributions with Equations 1 and 2, until it converges to a target distribution.

4.2.2 Parameter Estimation

Collecting sets of samples $\mathbf{L}, \mathbf{z}, \mathbf{x}, \mathbf{y}$ obtained from the 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,

$$\begin{aligned}\mathrm{P}(\phi_k | \mathbf{z}, \mathbf{w}, \beta) &\propto \mathrm{P}(\mathbf{w} | \mathbf{z}, \phi_k) \mathrm{P}(\phi_k | \beta) \\ &\propto \prod_{d=1}^{D} \prod_{n=1}^{N_d} \phi_{k, w_{dn}} \prod_{v=1}^{V} \phi_k^{\beta - 1} \\ &= \prod_{v=1}^{V} \phi_k^{C_{vk}^{VK} + \beta - 1} = \mathrm{Dir}(C_{vk}^{VK} + \beta)\end{aligned} \tag{3}$$

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

$$\phi_{vk} = \frac{C_{vk}^{VK} + \beta}{\sum_v C_{vk}^{VK} + V\beta} \tag{4}$$

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

$$\theta_{tk} = \frac{C_{tk}^{TK} + \alpha}{\sum_k C_{tk}^{TK} + K\alpha} \tag{5}$$

$$\lambda_{sk} = \frac{C_{sk}^{SK} + \gamma}{\sum_k C_{sk}^{SK} + K\gamma} \tag{6}$$

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

$$\begin{aligned}\mathrm{P}(\pi_d | \mathbf{L}, \boldsymbol{\eta}) &\propto P(\mathbf{L} | \pi_d) P(\pi_d | \boldsymbol{\eta}) \\ &\propto \pi_d^{C_t^L} (1 - \pi_d)^{C_s^L} \pi_d^{\eta_{\pi_0} - 1} (1 - \pi_d)^{\eta_{\pi_1} - 1} \\ &= \pi_d^{C_t^L + \eta_{\pi_0} - 1} (1 - \pi_d)^{C_s^L + \eta_{\pi_1} - 1} \\ &= \mathrm{Beta}(C_t^L + \eta_{\pi_0}, C_s^L + \eta_{\pi_1})\end{aligned} \tag{7}$$

Lastly, the expectation of the Beta distribution to estimate the probability of choosing tools or datasets for a document d is as follows:

$$\pi_d^{L_{dn}=0} = \frac{C_t^L + \eta_{\pi_0}}{C_t^L + C_s^L + \eta_{\pi_0} + \eta_{\pi_1}} \tag{8}$$

$$\pi_d^{L_{dn}=1} = \frac{C_s^L + \eta_{\pi_1}}{C_t^L + C_s^L + \eta_{\pi_0} + \eta_{\pi_1}} \tag{9}$$

4.3 Applications of Knowledge Pattern Discovery

We now present three applications for our DSTM to be used for the discovery of knowledge patterns for guiding researchers via user interfaces that feature conversational agents (or chatbots) as outlined in [36]. Through an interactive interface, DSTM can help the researchers in their common research activities involving: (i) *Intra-domain Knowledge Pattern Representation* that can be directly achieved by DSTM when used to explore potential tools or datasets in individual domains to solve a research problem with better performance; (ii) *Cross-domain Knowledge Pattern Representation* that involves a topic embedding solution for visualizing topics from multiple domains in a two-dimensional space in order to allow researchers to explore close topics and adapt successful solutions from other domains; and (iii) *Trend Knowledge Pattern Representation* that tracks the change in trends or reveals emerging trends over time, in order to guide researchers to make intelligent decisions on choosing the pertinent tools or datasets relevant to solving their research problem at hand.

4.3.1 Intra-domain Knowledge Pattern Representation

Researchers seek to discover intra-domain knowledge patterns pertaining to tools or datasets from existing solutions or prior successful experiences in a given scientific domain. To understand the application of intra-domain knowledge patterns, we can consider the following example: a research group has applied a deep learning method to achieve better performance in bioinformatics tasks such as sequence analysis and structure prediction. In this context, the knowledge pattern from their solution can be learned and re-purposed by other bioinformatics scientists for similar research tasks. We obtain intra-domain knowledge patterns by training a DSTM object using a dataset (documents D, tools T, datasets S) from an individual domain (e.g., bioinformatics or neuroscience that are an exemplar for the purposes of our work). Based on this training, we generate the topic distribution $\boldsymbol{\phi}$, tool distribution $\boldsymbol{\theta}$, and dataset distribution $\boldsymbol{\lambda}$.

4.3.2 Cross-domain Knowledge Pattern Representation

Besides exploring existing solutions in their respective domains, researchers in many scientific communities seek to investigate new solutions by referring to knowledge patterns (such as e.g., methods, tools) from other synergistic domains. In such scenarios for creation of cross-domain knowledge representations, our DSTM can be applied for user guidance. Common approaches for cross-domain recommendations involve using social information to build connections from other domains [37], [38], and also using transfer learning that makes use of auxiliary data [39]. Our idea for the DSTM application builds upon these existing approaches for finding similar topics from different synergistic domains to help with references to new methods, tools, and datasets from a cross-domain perspective.

The topics in our model are represented as multinomial distributions over a fixed vocabulary. The methods to measure difference or distance between two distributions could be Kullback-Leibler (KL) divergence or Maximum Mean Discrepancy (MMD) distance. The output value indicates the similarity between the two topics. In our work, instead of only providing a similarity score for users to compare and rank topics, we provide a novel visualization solution that embeds topics into a two-dimensional plane for easier exploration of similar topics in a lower dimensional space.

There have been prior algorithms implemented for topic embedding purposes. A few implementations directly treat multinomial distributions as a high dimensional dataset and then use PCA or t-SNE for mapping all topics onto a two-dimensional (2D) space. However, such an approach may not measure the real distance between any two distributions. Authors in [40] use Jensen-Shannon divergence to measure the distance between two distributions and then use multidimensional scaling (also known as Principal Coordinates Analysis) or t-SNE [34] to project the topics onto a 2D space. However, Jensen-Shannon divergence is a scale factor that outputs a value between 0 and 1. If the two distributions are similar, the value measured by JS divergence is very close, which makes it hard to differentiate in a 2D space. On the other hand, if the two distributions are not overlapped, the value will be always 1 no matter how different the two distributions are.

In our work, we have investigated an effective way of using t-SNE [34] for topic embedding via a novel MMD-t-SNE algorithm that maps topics onto a 2D space. The original t-SNE uses Euclidean distance to measure distance between two high-dimensional datasets, which gives equally weights for each data entry. This approach is suitable for those datasets (e.g., images, bag-of-words), where each entry is a real value. However, for multinomial distributions, each entry is a probability that cannot be equally weighted.

So, we use Maximum Mean Discrepancy (MMD) [41] to measure the distance between distributions in a high-dimensional space. More specifically, we replace the Euclidean distance that is used in the original t-SNE algorithm for measuring pairwise similarities in the high-dimensional space with the MMD kernel function. Hence, the symmetric conditional probability between topic i and j is defined as,

$$p_{ij} = \frac{\exp\left(-\mathrm{MMD(i, j)}^2/2\sigma^2\right)}{\sum_{k \neq l} \exp\left(-\mathrm{MMD(k, l)}^2/2\sigma^2\right)} \tag{10}$$

where l denotes i or j, because of its symmetry. The Maximum Mean Discrepancy (MMD) distance between topic distribution ϕ_i and ϕ_j for multinomial distribution is suggested using the Pearson's Chi-square statistic test [41]. In order to make Chi-square distance symmetric, the final $\mathrm{MMD}(i, j)$ is calculated by the average of the two Chi-square χ_{ij}^2 and χ_{ji}^2

values as given by,

$$\text{MMD}(i,j) = \frac{1}{2}\Big[\sum_{v}\frac{(\phi_{iv}-\phi_{jv})^2}{\phi_{jv}} + \sum_{v}\frac{(\phi_{jv}-\phi_{iv})^2}{\phi_{iv}}\Big] \tag{11}$$

With MMD-t-SNE algorithm, we can project any of the topics from any number of domains into the same 2D space in the form of a cross-domain knowledge pattern representation. For cross-domain recommendation, we simply need to explore topics from a 2D space to find closely related topics from other domains for referring research ideas, tools, as well as datasets. The procedure to obtain a cross-domain knowledge pattern fundamentally involves training multiple DSTM objects using datasets collected from multiple domains (for instance, using two domains a and b in our work). Subsequently, we take the topic distributions $\boldsymbol{\Phi}_a$, $\boldsymbol{\Phi}_b$ from the two pre-trained DSTM models. Finally, we apply our MMD-t-SNE algorithm to embed topics into a low-dimensional topic representation $\hat{\boldsymbol{\Phi}}$, which can help us understand the relationships among topics between two different domains.

4.3.3 Trend Knowledge Pattern Representation

As detailed above, the intra-domain and cross-domain knowledge pattern representations in our DSTM applications allow discovery of broadly popular tools and/or datasets featured in all publications over the last ten years being considered. However, their use in a visually interactive manner within an interactive user interface requires a new kind of DSTM application. Our trend knowledge pattern representation application not only addresses this issue, but also ensures that the interactive knowledge discovery uses the most current information i.e., the information presented to the user tracks the change in trends or reveals the latest emerging trends in the relevant text corpus. For example, bioinformatics scientists in the past used to use the MATLAB SVM toolbox for pattern recognition; however, these days most scientists choose a deep learning framework (e.g., Keras, TensorFlow, PyTorch) to accomplish similar tasks. Our DSTM application can discover such trend knowledge patterns with an unsupervised learning approach, and can be adapted within effective visualizations in user interfaces. Hence, the trend knowledge pattern representation can be considered to be complementary to the intra-domain knowledge pattern. This complementary nature is helpful for incorporating the time-series pattern that can guide users to select suitable tools/datasets. Otherwise, users will only consider the average information within the ten years of data, and miss deriving potential trends for future research. Moreover, the intra-domain knowledge pattern is also necessary to obtain overall information and help in choosing suitable tools/datasets that do not have a clear trend in change patterns.

Algorithm 2 details our novel algorithm to discover trend knowledge patterns for selecting suitable tools or datasets. Our trend pattern analysis does not involve simply ranking tools or datasets by counting the number of times they are mentioned in papers for each year. Our goal is to learn the trend knowledge patterns for tools or datasets in terms of the particular topics by year using our pre-trained model.

Algorithm 2 Trend Knowledge Pattern Representation

Input: D, S, T from particular domain
Output: $S_{ks}^{(i)}, T_{kt}^{(i)}$

function TREND(D, S, T)
  Get a pre-trained DSTM model's parameters: $\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\lambda}$
  Initialize $T_{kt}^{(i)} = 0$; $S_{ks}^{(i)} = 0$; and i denotes year;
  for each doc $d \in D$ **do**
    Set y to publication year of doc d;
    Initialize tool assignment $TA_t = 0$;
    Initialize dataset assignment $SA_s = 0$;
    Initialize topic assignment $KA_k = 0$;
    for each word $w_{dn} \in d$ **do**
      Sample a topic z_{dn} and tool x_{dn} with Eqn. 1;
      $TA_{x_{dn}} += 1$; $KA_{z_{dn}} += 1$;
      Sample a topic z_{dn} and dataset y_{dn} with Eqn. 2;
      $SA_{y_{dn}} += 1$; $KA_{z_{dn}} += 1$;
    end for
    Choose tool id $tid = argmax(TA_t)$;
    Choose dataset id $sid = argmax(SA_s)$;
    Choose topic id $kid = argmax(KA_k)$;
    $T_{kid,tid}^{(y)} += 1$;
    $S_{kid,sid}^{(y)} += 1$;
  end for
  return $S_{ks}^{(i)}, T_{kt}^{(i)}$
end function

The core idea of learning the trend knowledge pattern is to infer the tools/datasets' topics for each paper using our pre-trained model. For example, we use a pre-trained model from an individual scientific domain. Then, we group papers by year and run an inference algorithm using our pre-trained model as shown in Algorithm 2 to infer its topics for each paper. Following this, we can track the changes in trends or the evolution of the use of tools or datasets for a particular topic by year.

More specifically, in Algorithm 2, we firstly use a pre-trained model in an individual domain. Secondly, we initialize two 3D arrays $T_{kt}^{(i)}$ (with dimensions of the number of years, the number of topics, the number of tools) and $S_{ks}^{(i)}$ (with dimensions of the number of years, the number of topics, the number of tools) to count samples generated by our inference algorithm. Thirdly, for each paper, we infer its topic and relevant tools or datasets by assigning them to particular topics based on our pre-trained model and then we update relevant $T_{kt}^{(i)}$ or $S_{ks}^{(i)}$. We run these steps for several iterations to make sure that the model converges. After completion of the inference procedure, the $T_{kt}^{(i)}$ has information the times of tool t been assigned to topic k in year i; similarly, the $S_{ks}^{(i)}$ has information the times of dataset s been assigned to topic k in year i. Finally, we obtain sets of samples $T_{kt}^{(i)}, S_{ks}^{(i)}$ to estimate the trends within tools or datasets for a particular topic by year.

5 DATASETS

We use three categories of datasets (papers, tools, and datasets) from two exemplar scientific domains viz., *neuroscience* and *bioinformatics* for understanding the synergistic relationships among research topics, tools, and datasets. Table 2 provides a more detailed description of the collected data for analysis, and the related data pre-processing steps performed on this collected data are as follows:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3093350, IEEE Transactions on Knowledge and Data Engineering

TABLE 2. Description of collected data for analysis from neuroscience and bioinformatics domain communities.

Category	Neuroscience	Bioinformatics
Papers	We have collected $16,721$ latest neuroscience papers from four reputed journal archives: *Frontiers in Computational Neuroscience*, *Journal of Computational Neuroscience*, *Journal of Neuroscience*, and *Neuron* published from 2009 to 2019. This results in a vocabulary size of $V = 19,488$ unique words joint with bioinformatics text corpus and a total of $4,527,987$ word tokens.	We collected $10,681$ bioinformatics papers from *Journal of BMC Bioinformatics*, *Journal of BMC Genomics*, *Genome Biology*, *Nucleic Acids Research*, *PLOS Computational Biology* published in the recent 10 years between 2009 to 2019. We extracted abstracts for each of the papers. This resulted in a vocabulary size of $V = 19,488$ unique words with neuroscience text corpus, and a total of $4,527,987$ word tokens.
Tools	We have collected the commonly used tools in neuroscience research activities including computation, simulations, databases and visualization, such as MATLAB, NEURON [5], PyNN [42], ModelDB [43], and new machine learning frameworks (e.g., TensorFlow, Keras) may be applied in recent neuroscience research. This results in a total of 189 tools.	We have collected 219 types of 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), and new machine learning frameworks (e.g., TensorFlow, Keras).
Datasets	Datasets described in neuroscience literature are usually recognized by cell types [44] (i.e., pyramidal, interneuron) or brain regions (i.e., neocortex, retina). We collected the common dataset types in neuroscience experiments, which results in a total of 169 different types of 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). This results in a total of 32 type datasets.

- **Papers**: We collected full papers from well-known journals in neuroscience and bioinformatics domain communities. We removed any words that occurred in less than 10 papers that are supposed to be highly infrequent words, and belonged to the list of "stop words" that are significantly frequent (e.g., "the", "a") in papers. Each paper was represented as a "bag of words" in our model.
- **Tools**: We collected the most commonly used tools of neuroscience and bioinformatics domains separately in collaboration with domain experts. This list of tools covers a wide range of research efforts in computation, simulations, databases and visualization.
- **Datasets**: We collected common types of datasets used in experiments featured in the neuroscience and bioinformatics domains.

6 Model Performance Evaluation

In this section, we first perform model selection to choose optimal parameters (such as number of topics, number of iterations) based on the two collections of datasets mentioned in Section 5. Following this, we evaluate the generalization and information retrieval performance with state-of-the-art models such as Latent Dirichlet Allocation (LDA) [9] and Probabilistic Latent Semantic Analysis (PLSA) [45].

6.1 Model Selection

The model we described in Section 4.1 has four hyperparameters $\{\alpha, \beta, \gamma, \eta, T\}$. The $\{\alpha, \beta, \gamma\}$ are hyperparameters for symmetric Dirichlet prior. They may affect the number of topics: smaller values are supposed to find more topics from a text corpus, and larger values tend to collect a relatively smaller number of topics from a text corpus. Hence, for processing a large volume of text corpus with wide ranging research problems, we choose smaller values of these hyperparameters and vice versa. For simplifying the hyperparameters tuning in our model, we follow the suggestions from [17], keeping them constant: $\alpha = \gamma = 50/K, \beta = 0.1$, respectively. And the hyperparameter η adds prior probability for choosing tools or datasets for generating a single word. The hyperparameter is also kept fixed at $\eta = (3, 2)$. This because we empirically suppose that tools in each paper have a higher chance to be selected to generate words. Hence, keeping the hyperparameters $\{\alpha, \beta, \gamma, \eta\}$ fixed, we decide the optimal number for topics and iterations. In addition, these optimal numbers are decided in terms of datasets, and our results are obtained based on the collected bioinformatics and neuroscience datasets.

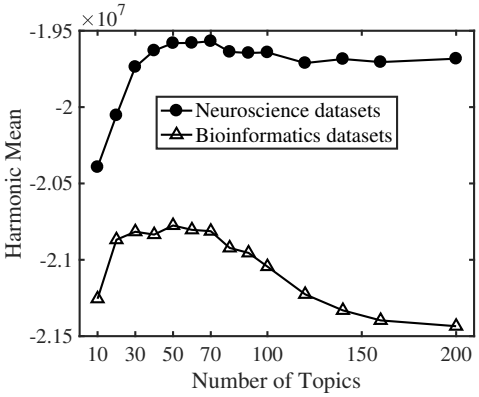
6.1.1 Number of Topics

We can find the optimal number of topics T for the model based on a particular text corpus. Specifically, we need to compute the likelihood of words give a particular number of topics $P(\mathbf{w}|T)$ for all documents. This involves a step to integrate out all possible topic assignments $\mathbf{z}$ for each word using $\sum_{\mathbf{z}} P(\mathbf{w}|\mathbf{z}, T)$. However, we can approximate the likelihood of words by using the harmonic mean of $P(\mathbf{w}|T)$ when $\mathbf{z}$ is sampled from a posterior distribution $P(\mathbf{z}|\mathbf{w})$ [17], [46]. Hence, we get,

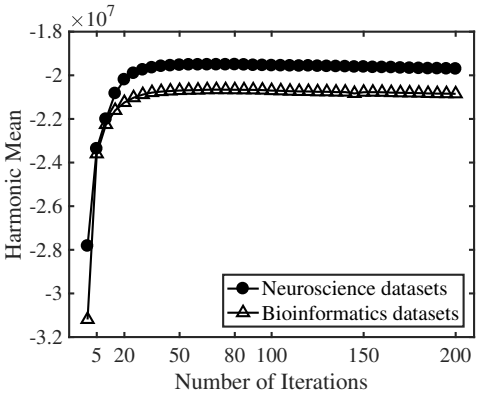
$$P(\mathbf{w}|T) \approx \Big\{ \frac{1}{m} \sum_{i=1}^{m} P(\mathbf{w}|z_i, T)^{-1} \Big\}^{-1} \tag{12}$$

As shown in Fig. 3a, the harmonic mean suggests that the optimal number of topics are near $K = 50$ for the bioinformatics dataset collection, and $K = 70$ for the neuroscience dataset collection. In most topic models, we are inclined to choose a smaller number of topics based on the perplexity metric performance for cases when there is a similar performance between the two models. Such a choice in turn helps us to avoid the handling of repeated or redundant topics. On the other hand, choosing a large number of topics will consume more computation resources to reach equilibrium.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3093350, IEEE Transactions on Knowledge and Data Engineering



(a) Model selection for optimal number of topics



(b) Model selection for optimal number of iterations

Fig. 3: DSTM model selection for choosing optimal number of topics and number of iterations using harmonic mean based on neuroscience and bioinformatics dataset collections.

6.1.2 Number of Iterations

In our DSTM, we use the Gibbs sampling as explained earlier in Section 4.2.1 for inferring latent variables in our model. The Gibbs sampling starts from a random initialization and runs over a few iterations to reach its equilibrium distribution. Hence, it is also important to estimate the optimal number of iterations for either achieving the best performance or saving computation time.

Using the optimal number of topics $K = 50$ and $K = 70$ for the bioinformatics and neuroscience datasets respectively, we apply Equation 12 to compute the harmonic mean in terms of the number of iterations. As shown in Fig. 3b, our DSTM reaches its equilibrium distribution at iteration 50 with the bioinformatics dataset, and at iteration 80 with the neuroscience dataset.

6.2 Model 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-the-art models: Latent Dirichlet Allocation (LDA) [9] and Probabilistic Latent Semantic Analysis (PLSA) [45]. Next, we evaluate information retrieval performance and discuss issues by comparing them with state-of-the-art models. Following this, we demonstrate the benefits of applying our model for choosing appropriate tools or datasets for the computational and data-intensive research problems discussed earlier in Section 4.3.

6.2.1 Generalization Performance Evaluation

The generalization performance is an important factor to evaluate how well a probabilistic model predicts a previously not observed sample based on the model parameters learned in the training stage. *Perplexity* is a standard metric that is 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 $\mathbf{w}_d$, and is conditioned on the known tools $\mathbf{t}_d$, datasets $\mathbf{s}_d$ of the document d and trained model, is defined as,

$$\begin{aligned}\text{perplexity}(\mathbf{w}_d|\mathbf{t}_d, \mathbf{s}_d, D_{train}) \\ = \exp\Big\{ - \frac{\log \mathrm{P}(\mathbf{w}_d|\mathbf{t}_d, \mathbf{s}_d, D_{train})}{N_d} \Big\}\end{aligned} \tag{13}$$

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

$$\begin{aligned}\text{perplexity}(D_{test}) \\ = \frac{\sum_{d=1}^{D_{test}} \text{perplexity}(\mathbf{w}_d|\mathbf{t}_d, \mathbf{s}_d, D_{train})}{D_{test}}\end{aligned} \tag{14}$$

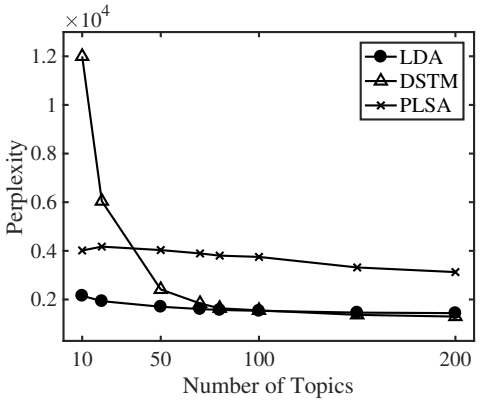
The probability of words $\mathbf{w}_d$ in the document d with known tools $\mathbf{t}_d$ or datasets $\mathbf{s}_d$ can be obtained by integrating all latent variables,

$$\begin{aligned}\mathrm{P}(\mathbf{w}_d|\mathbf{t}_d, \mathbf{s}_d) = \prod_{n=1}^{N_d} \sum_{k=1}^{K} \Big[\frac{1}{\mathbf{t}_d} \sum_{t=1}^{T} \pi_d^{L_n=0} \phi_{k,w_n} \theta_{kt} \\ + \frac{1}{\mathbf{s}_d} \sum_{s=1}^{S} \pi_d^{L_n=1} \phi_{k,w_n} \lambda_{ks} \Big]\end{aligned} \tag{15}$$

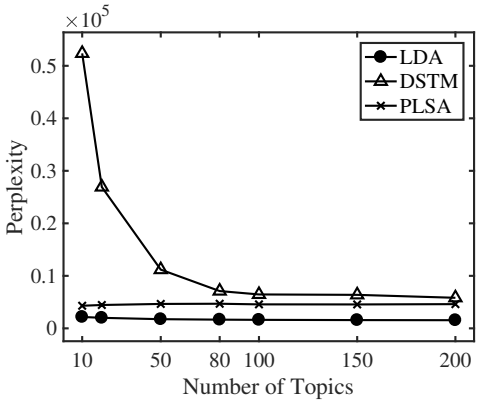
Where the $\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\lambda}$ can be estimated through model training stage using Equations 4, 5, 6, respectively; also, the π_d needs to be sampled based on the new test documents d. This equation is similar to Equation 12, but it is computed with new test documents. Practically, we run the Gibbs sampling algorithm in Equations 8, 9 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 and PLSA for both dataset collections (i.e., neuroscience and bioinformatics) shown in Table 2. In both cases and in both the models, we held out 20% of the same data for testing the generalization performance and used 80% of the same data for training.

Fig. 4 shows that the perplexity scores of DSTM are significantly higher than the LDA, PLSA perplexity scores in both cases initially. We note that this has been caused by overfitting issues when the number of topics are relatively small. However, after increasing the number of topics, the DSTM quickly achieves similar generalization performance with LDA and PLSA after topic $(K = 50)$. Especially, the neuroscience dataset achieves better performance after



(a) Perplexity comparison on the neuroscience dataset



(b) Perplexity comparison on the bioinformatics dataset

Fig. 4: Perplexity comparison with LDA, PLSA models on different dataset collections and for different number of topics.

topic $(K = 70)$ as shown in Fig. 4a. Additionally, in both cases, the LDA and PLSA models' perplexity scores change slightly with the different number of topics, and the difference between the maximum and minimum scores are indistinguishable.

The above evaluation results provide the insights of an interesting phenomenon where the LDA, PLSA models have overall better generalization performance for the most number of topics; whereas, our DSTM has better performance within a range of the particular number of topics (that in turn depends upon the diversity of the number of tools or datasets). The reason for this phenomenon is that the LDA, PLSA have a random generative process for producing each word. However, our DSTM generates words based on the occurrence of tools or datasets within each document to guide our model to find particular topics. From Fig. 4, we can also find that the neuroscience dataset achieves better performance than the bioinformatics dataset. The reason is that the number of tools and datasets in the neuroscience dataset are larger than the number in the bioinformatics dataset.

In order to analyze the impact of the size of datasets (e.g., tools, datasets) on the generalization performance in our DSTM, we re-construct our training data with bioinformatics publications using different sizes of tools/datasets. More specifically, we train multiple DSTM variants by randomly selecting subsets of tools/datasets from the full sets measured with different ratios (i.e., 20%, 40%, ..., 100%) while keeping other parameters the same as in the previous experiments. Subsequently, we use the same testing datasets to test each model's generalization performance using a perplexity score. As shown in Table 3, we can conclude that the perplexity score can be improved by increasing the size of tools/datasets in the training data. This evaluation result presents the evidence for why the neuroscience publication datasets can achieve better performance than the bioinformatics publication datasets, and also indicates how our DSTM performance can be improved depending on the characteristics of the datasets being considered.

In summary, our DSTM's performance is dependent on the availability of auxiliary datasets (e.g., tools/datasets in our work) for model training. In essence, our DSTM has good generalization performance in comparison to state-of-the-art models (i.e., LDA, PLSA) for finding highly specific topics within a domain. Hence, our DSTM is more suitable for domain scientists in finding particular resources (such as tools or datasets) as part of their knowledge discovery to solve research problems at hand.

TABLE 3. Perplexity comparison with our DSTM models on bioinformatics dataset. We have trained five DSTM models with the different ratios of tools/datasets, and tested them with same test datasets.

Ratio	20%	40%	60%	80%	100%
Perplexity	13977.93	9842.35	8483.45	8217.74	7801.58

6.2.2 Topic Coherence Performance Evaluation

The perplexity metric cannot directly reflect the quality of topics generated by the models. Hence, we also evaluate the topic's coherence scores by using a number of standard metrics such as *NPMI*, *UCI* and *UMass*. Both *NPMI* and *UCI* are measured by point-wise mutual information (PMI) of the top n words of the topics generated by topic models; whereas, the *UMass* is calculated by smoothed conditional probability between top words pairs [47]. Note that a higher coherence score implies better performance. Further, we have evaluated the coherence performance of our DSTM with state-of-the-art PLSA and LDA models using both bioinformatics and neuroscience datasets. In this context, we extract the top 5 and top 10 words from each topic for computing the coherence score. Table 4 shows that our DSTM achieves slightly better performance than the state-of-the-art models in almost all metrics and scenarios, which are benefited by incorporating extra information (i.e., tools/datasets) for generating consistent topics. Therefore, the performance of our coherence score further proves that our DSTM can generate good quality topics in comparison with the state-of-the-art models.

6.2.3 Information Retrieval Performance Evaluation

In this section, we compare the information retrieval performance of our DSTM with LDA and PLSA models. Information retrieval performance is used to evaluate - "how relevant is the retrieved document d that satisfies the user's

TABLE 4. Coherence score comparison (the higher the better) using bioinformatics and neuroscience datasets. We evaluate our DSTM coherence performance with the PLSA and LDA model, and the coherence scores are computed using top-K words.

Topic Model	Bioinformatics						Neuroscience					
	Top@5			Top@10			Top@5			Top@10		
	NPMI	UCI	UMass	NPMI	UCI	UMass	NPMI	UCI	UMass	NPMI	UCI	UMass
PLSA	-0.209	-5.186	-1.313	-0.215	-5.540	-1.393	-0.211	-5.533	-1.455	**-0.220**	-5.792	-1.642
LDA	-0.199	-5.051	-1.291	-0.214	-5.464	-1.313	-0.214	-5.423	-1.441	-0.228	-5.845	-1.583
DSTM	**-0.189**	**-4.030**	**-1.116**	**-0.207**	**-4.778**	**-1.233**	**-0.209**	**-4.779**	**-1.381**	-0.223	**-5.246**	**-1.582**

TABLE 5. Information retrieval performance comparison of Precision (P), Recall (R), and NDCG (N) for top-K retrievals in both Bioinformatics and Neuroscience domain.

Domain	Topic Model	Evaluation Metrics								
		P@3	P@5	P@10	R@3	R@5	R@10	N@3	N@5	N@10
Bioinformatics	PLSA	0.029	0.017	0.026	0.004	0.004	0.013	0.243	0.229	0.241
	LDA	0.043	0.026	0.017	0.007	0.007	0.009	0.222	0.226	0.231
	DSTM	**0.043**	**0.043**	**0.039**	**0.007**	**0.011**	**0.020**	**0.247**	**0.263**	**0.254**
Neuroscience	PLSA	0.029	0.017	0.026	0.004	0.004	0.013	0.243	0.229	0.241
	LDA	0.043	0.026	0.017	0.007	0.007	0.009	0.222	0.226	0.231
	DSTM	**0.043**	**0.043**	**0.039**	**0.007**	**0.011**	**0.020**	**0.247**	**0.263**	**0.254**

query q". Mathematically, we need to estimate the conditional probability $\mathrm{P}(q|d)$.

TABLE 6. Samples of query sentences collected from bioinformatics and neuroscience experts for information retrieval performance evaluation purposes.

Bioinformatics	Neuroscience
DNA sequence analysis	Computational neuroscience
Comparative genomics	Auditory perception
Analysis of gene expression	Neural mechanisms of attention and memory
Genome annotation	Neural mechanisms of decision making
Sequence alignment analysis	Neural circuits for vision

To construct the query datasets, we obtain guidance from the domain experts in bioinformatics and neuroscience on common research topics they use to search and query from literature archives as shown in Table 6, where we list a sample of 5 query sentences (out of the total of 23 and 32 query sentences we used) from each domain, respectively. To construct the retrieval datasets, we randomly select 1000 papers from our datasets mentioned in Table 2 for each domain.

Given that we do not have the ground truth labels for indicating the relevance score between each query set and each retrieval set, we leverage the standard information retrieval method Okapi BM25 [48] to compute the relevance scores as our ground truth. For each query set, the predicted relevance scores are computed by the probabilistic scores $\mathrm{P}(q|d)$, and then we can evaluate the performance for each query with the ground truth in terms of precision, recall, and normalized discounted cumulative gain (NDCG) metrics.

To estimate $\mathrm{P}(q|d)$, we approximately compute the probability of query words $\mathbf{w}_q$ given tools or datasets in retrieval documents $\mathrm{P}(\mathbf{w}_q|\mathbf{t}_d, \mathbf{s}_d)$, which can be computed using Equation 15. This can be understood by knowing whether the topics of tools or datasets are a fit for the query words or not. For example, the tool TensorFlow is not a good fit for a query sentence that the user inputs as "Bayesian statistical inference", however it is a good fit for a query sentence such as "gene expression inference using deep learning". On the other hand, in the LDA, PLSA models, the $\mathrm{P}(q|d)$ is estimated by directly considering topics of retrieval documents and query words $\mathbf{w}_q$ given the topics represented by $\sum_{k=1}^{K} P(\mathbf{w}_q|k)P(k|d)$.

Table 5 presents the information retrieval performance of DSTM in comparison with the PLSA and LDA models. We have evaluated the information retrieval performance in terms of Precision (P), Recall (R) and NDCG (N) at K in $\{3, 5, 10\}$. The results shows that our DSTM outperforms other models in all metrics for both bioinformatics and neuroscience domains, and thus has significantly better benefits in retrieving domain-specific topics.

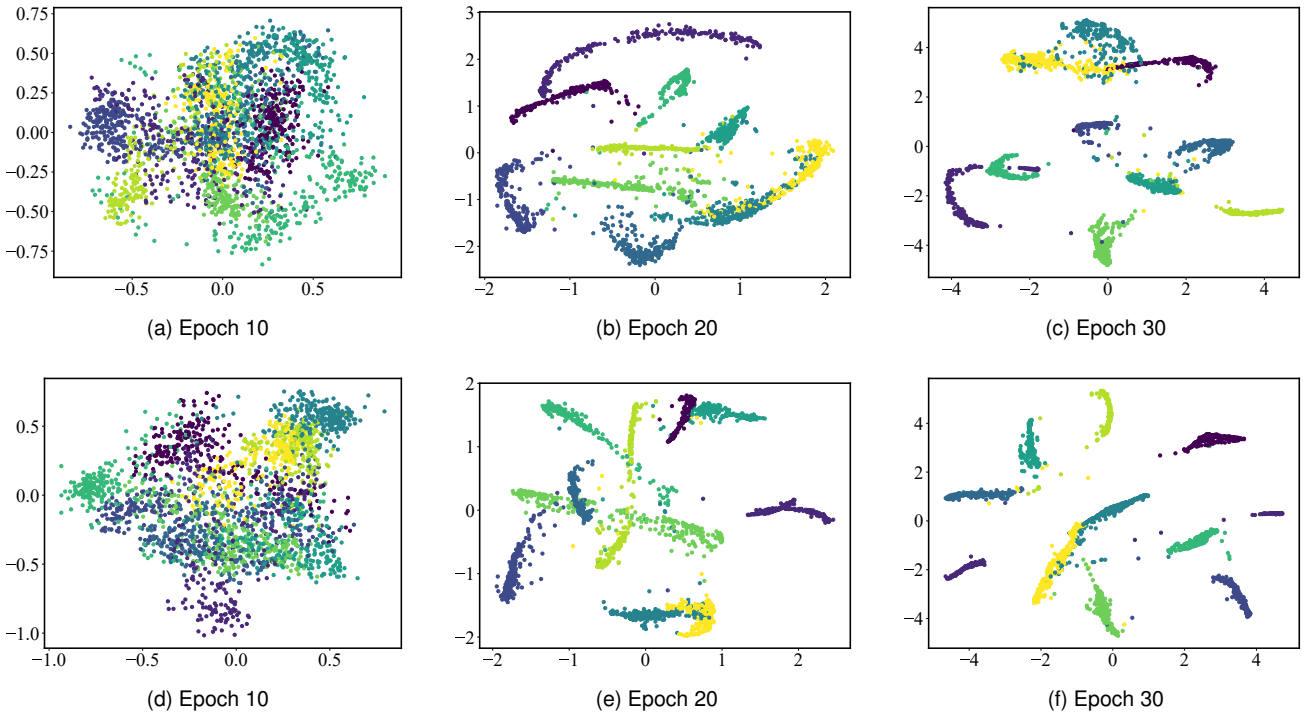
TABLE 7. Clustering performance evaluation using Calinski-Harabaz Index (CHI) and Silhouette Coefficient metrics; in both cases, a higher score indicates better performance and each score is calculated by averaging the scores from 10 tests.

Epochs	MMD-t-SNE		t-SNE	
	CHI	Silhouette	CHI	Silhouette
10	123.713	**-0.115**	**123.965**	-0.125
20	**594.904**	**0.115**	578.965	0.090
30	**1637.887**	**0.395**	1366.184	0.399
40	**2039.281**	**0.406**	1536.376	0.407
50	**2349.961**	**0.492**	1934.680	0.473

6.2.4 MMD-t-SNE Performance Evaluation

In this section, we evaluate our MMD-t-SNE algorithm detailed in Section 4.3.2. Our MMD-t-SNE extends the t-SNE [34] algorithm. Our design is suited for visualizing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3093350, IEEE Transactions on Knowledge and Data Engineering



(a) Epoch 10 (b) Epoch 20 (c) Epoch 30

(d) Epoch 10 (e) Epoch 20 (f) Epoch 30

Fig. 5: Performance evaluation of our MMD-t-SNE with t-SNE algorithm using the MNIST dataset. The Figures (a)-(c) are generated by the t-SNE algorithm; and Figures (d)-(e) are generated by our MMD-t-SNE algorithm.

complex distributions such as (categorical, multinomial distributions) in a low dimension space, and we replace the Euclidean distance kernel with the MMD kernel.

We use MNIST [49] as our baseline dataset to compare the performance between our MMD-t-SNE algorithm with the t-SNE algorithm. Given that the MNIST datasets are digit numbers with labels 0 to 9, we are able to observe the correctness of the algorithms in a straightforward manner. In order to simplify the MNIST, we replace the grayscale (0-255) representation with a binary (0, 1) representation. The reason we use a binary representation is that we only compute the probability of non-white area at a certain pixel. Hence, the dataset with/without grayscale will be the same for the purposes of our algorithm related data transformation. There are 784 pixels in each image, and the value of each pixel will be 1 if its grayscale is larger than 1, otherwise it is set to 0. Following this, we aggregate 8 pixels into 1 pixel by counting the values within 8 pixels for simulating the probability of having values in those areas. Finally, we can calculate the probability of having values for each pixel based on the image label. After these transformations, each image will be represented as a multinomial distribution with 98 entries.

We compare the t-SNE algorithm with our MMD-t-SNE algorithm using this transformed MNIST dataset. In Fig. 5, the upper figures Fig. 5(a), Fig. 5(b), and Fig. 5(c) are generated for the t-SNE algorithm with different epochs; and the lower figures Fig. 5(d), Fig. 5(e), and Fig. 5(f) are generated for the MMD-t-SNE algorithm with same epochs as the t-SNE algorithm. We can clearly observe that our MMD-t-SNE algorithm exhibits relatively better performance in differentiating these datasets with multinomial distribution representations.

In addition to the visualization evaluation, we also evaluate their performances with quantitative metrics. Specifically, we use two unsupervised learning based metrics to measure the quality of clustering performance: *Calinski-Harabaz Index* (CHI) [50] and *Silhouette Coefficient* [51]. The score computed by *Calinski-Harabaz Index* indicates a ratio of the sum of between-clusters dispersion and of within-cluster dispersion for all clusters. In this context, dispersion is defined as the sum of distances squared, and the higher score indicates that the clusters are dense and well separated. Similarly, the Silhouette Coefficient considers intra-cluster distance and nearest inter-cluster distance, which generates a score between -1 (the worst) and 1 (the best); the values near 0 indicate overlapping clusters and the negative values normally indicate that a sample has been assigned to the wrong cluster.

We compare the performance with different epochs for evaluation of the algorithms' effectiveness. Each epoch is run 10 times to get the average values of CHI score and Silhouette Coefficient score respectively, because both MMD-t-SNE and t-SNE algorithms have randomization process in the initialization. As shown in Table 7, in the earlier epochs, their performance results are roughly similar. However, our MMD-t-SNE algorithm achieves distinctly better performance than the t-SNE algorithm from 20 epochs and beyond.

6.2.5 Performance Results Discussion

Based on the performance evaluation results, we note that our DSTM achieves its optimal performance when it uses a suitable number of topics given a particular dataset. For

example, the bioinformatics dataset uses the number of topics $K = 50$; and the neuroscience dataset uses the number of topics $K = 70$. Following this, the number of iterations will gradually converge after 30 iterations. We can thereafter use the harmonic mean shown in Equation 12 to search the optimal parameters. In addition, the generalization performance can be improved by increasing the size of datasets (i.e., the number of tools/datasets) to train the DSTM. We have demonstrated that the neuroscience dataset achieves better performance than any of the state-of-the-art models, however, the bioinformatics dataset has slightly worse performance than the state-of-the-art models. This slightly worse performance is due to the fact that the bioinformatics dataset has fewer data points corresponding to the number of tools/datasets than the neuroscience dataset.

We have evaluated our above observation by training DSTM with different sizes of tools/datasets, and have performed testing with the same datasets. Our experimental results show that the generalization performance can be improved by adding more tools/datasets to train the DSTM. Our coherence score shows that our DSTM can generate good quality topics in comparison with state-of-the-art models. We remark that the generation of good quality topics benefits by the addition of tools/datasets categories into our DSTM and can lead to more specific topics. Finally, our DSTM did not achieve better information retrieval performance than the state-of-the-art models because our DSTM does not have document parameters to measure the probability that a document is comprised of particular words. Therefore, based on our experimental results we conclude that our DSTM is not suitable for information retrieval from documents.

7 Application Demonstration

In this section, we demonstrate the knowledge patterns discovery enabled by our model for exemplar scientific domains. Our demonstrations use the DSTM constructed with appropriate parameters based on the discussion provided in Section 6.1. Our demonstration experiments use full datasets for both neuroscience and bioinformatics cases and feature three types of patterns relating to the applications outlined in Section 4.3 for helping researchers to choose appropriate tools or datasets for a particular research topic: (i) *Intra-domain Knowledge Pattern Demonstration* presents the overall patterns for choosing tools or dataset within a specific scientific domain; (ii) *Cross-domain Knowledge Pattern Demonstration* shows the patterns for helping researchers to investigate new approaches, tools, or datasets by referring to solutions from other synergistic domains; (iii) *Trend Knowledge Pattern Demonstration* presents the evolution of tools/datasets for particular topics over time, which helps researchers to make guided decisions based on the past successes and the latest emerging trends.

7.1 Intra-domain Knowledge Pattern Application

To obtain intra-domain knowledge patterns, we can follow the method mentioned in Section 4.3.1 to train a neuroscience model and a bioinformatics model with neuroscience and bioinformatics dataset collections, respectively.

7.1.1 Exemplar Neuroscience Domain Dataset Discussion

Table 8 illustrates 4 samples of topics from 70 topics learned by DSTM for the neuroscience dataset. These samples are extracted from a single chain at the 80^{th} iteration of the Gibbs sampler. Each sub-table in Table 8 shows the top 10 words that are most likely to be generated conditioned on that topic; the top 4 most likely tools to be used for the topic; the top 4 most likely types of datasets that have come from the topic. Note that we have provided the topic heading annotation to each topic for demonstration in the sub-tables. The first sample Topic 13 "*Sensory & Anatomical Signals*" involves intra-/extra-cellular recordings, LFP, EEG, etc. that use popular tools such as MATLAB, EEGLAB, Klusters with commonly used datasets such as vertical and horizontal. The second sample Topic 18 is related to "*Circuit Analysis*" that determines circuit connections using datasets that include excitatory/inhibitory type with possible neuromodulation, and standard tools for topological analysis, including SPM. The third sample Topic 27 "*Structural Morphology*" focuses on the cellular and circuit morphology including 3-D orientation and coursing of axons along tracts. The fourth sample Topic 38 describes research about the *"Neuron Model"* that features single neuron models focusing on the role of channels in cellular excitability and the variety of response patterns such as tonic spiking and bursting, using several new tools popular in the past decade including NEURON, GENESIS, and XPPAut that are hosted in databases such as ModelDB. These knowledge patterns were confirmed to be valid and helpful by neuroscience experts, which demonstrates that our DSTM effectively captures the salient domain knowledge patterns.

7.1.2 Exemplar Bioinformatics Domain Dataset Discussion

Table 9 shows 4 samples topics out of 50 topics for the bioinformatics dataset that are extracted from a single chain at the 50^{th} iteration of the Gibbs sampler. Each sub-table in Table 9 shows the top 10 words that are most likely to be generated based on that topic; the top 4 most likely tools to be used for the topic; the top 4 most likely types of datasets that are commonly used in or related the topic. Here also, we have provided the topic heading annotation to each topic for demonstration in the sub-tables. The first sample Topic 9 falls under the research topic of *"Next-Generation Sequencing Analysis"*, for which the most popular tools being used are BWA, Bowtie, or BLAST, and the dataset types commonly used/related for this topic are *wgs* ("Whole Genome Shotgun"), *rnaseq* ("RNA sequencing"), etc. The second sample Topic 10 is related to *"Protein Sequence & Structure Modeling"*, for which popular tools such as *rfam*, *psipred* are accurately captured by our model, types of datasets such as *ENCODE*, *mirna* ("microRNA") are also commonly used for this topic. The third sample Topic 12 describes the research area of *"Metabolic Analysis"*, and tools such as COBRA, MATLAB, and datasets e.g., metabolomics are highly selected for this research. The fourth sample Topic 44 represents the *"Pattern Recognition"* research area in bioinformatics (such as gene expression, protein structure prediction). These knowledge patterns were also confirmed to be valid and helpful by bioinformatics experts, which demonstrates that our DSTM once again effectively captures the salient domain knowledge patterns.

TABLE 8. 4 sample topics (out of 70 topics in total) extracted for the neuroscience publications from 2009 to 2019. Each topic is associated with 10 most likely words, 4 most likely tools and datasets that have the highest probability conditioned on that topic.

Topic 13 *Sensory & Anatomical Signals*	
Word	**Prob.**
signals	.0320
stimulus	.0299
single	.0181
phase	.0174
movement	.0155
sensory	.0149
recording	.0147
response	.0140
timing	.0107
Tool	**Prob.**
MATLAB	.3550
EEGLAB	.1480
Klusters	.0693
opengl	.0476
Dataset	**Prob.**
vertical	.2555
horizontal	.2477
modulated	.1741
dorsal	.0637

Topic 18 *Circuit Analysis*	
Word	**Prob.**
circuit	.0543
circuits	.0362
sensory	.0214
changes	.0187
input	.0147
circuitry	.0112
connections	.0108
stimuli	.0106
results	.0104
Tool	**Prob.**
MATLAB	.1147
SPSS	.1077
Topological	.0411
SPM	.0270
Dataset	**Prob.**
circuit	.6181
excitatory	.1576
cholinergic	.0390
inhibitory	.0296

Topic 27 *Structural Morphology*	
Word	**Prob.**
axon	.0524
axons	.0464
axonal	.0407
cells	.0198
myelin	.0137
nerve	.0122
growth	.0109
cell	.0081
channels	.0068
Tool	**Prob.**
Neo	.2521
ansys	.0657
GENESIS	.0257
Caret	.0257
Dataset	**Prob.**
axon	.6568
myelinated	.1865
cone	.0350
ganglion	.0206

Topic 38 *Neuron Model*	
Word	**Prob.**
neurons	.0328
channels	.0323
neuron	.0255
somatic	.0220
bursting	.0162
network	.0162
bursts	.0155
model	.0154
channel	.0153
Tool	**Prob.**
NEURON	.2520
XPPAut	.1037
ModelDB	.0762
GENESIS	.0547
Dataset	**Prob.**
somatic	.3782
bursting	.3252
excitatory	.0365
pyloric	.0318

TABLE 9. 4 sample topics (out of 50 topics in total) extracted for the bioinformatics publications from 2009 to 2019. Each topic is associated with 10 most likely words, 4 most likely tools and datasets that have the highest probability conditioned on that topic.

Topic 4 *Next-Generation Sequencing Analysis*	
Word	**Prob.**
sequencing	.0484
reads	.0411
read	.0230
reference	.0174
sequence	.0141
mapping	.0141
assembly	.0141
short	.0141
high	.0141
Tool	**Prob.**
BWA	.2147
Bowtie	.1393
BLAST	.1382
Samtools	.1028
Dataset	**Prob.**
wgs	.3441
rnaseq	.1753
ENCODE	.0972
wes	.0646

Topic 10 *Protein Sequence & Structure Modeling*	
Word	**Prob.**
structure	.0610
sequence	.0368
structures	.0353
secondary	.0307
structural	.0248
sequences	.0212
prediction	.0160
method	.0132
alignment	.0126
Tool	**Prob.**
rfam	.1486
psipred	.1215
BLAST	.1196
pfam	.0915
Dataset	**Prob.**
ENCODE	.2940
mirna	.1468
Hi-C	.1269
mrna	.0980

Topic 12 *Metabolic Analysis*	
Word	**Prob.**
metabolic	.0818
flux	.0315
reactions	.0255
metabolites	.0204
network	.0152
analysis	.0146
different	.0109
fluxes	.0108
metabolomics	.0097
Tool	**Prob.**
COBRA	.3689
MATLAB	.3318
cplex	.1059
BLAST	.0625
Dataset	**Prob.**
metabolomics	.9074
mrna	.0274
wgs	.0208
mtdna	.0154

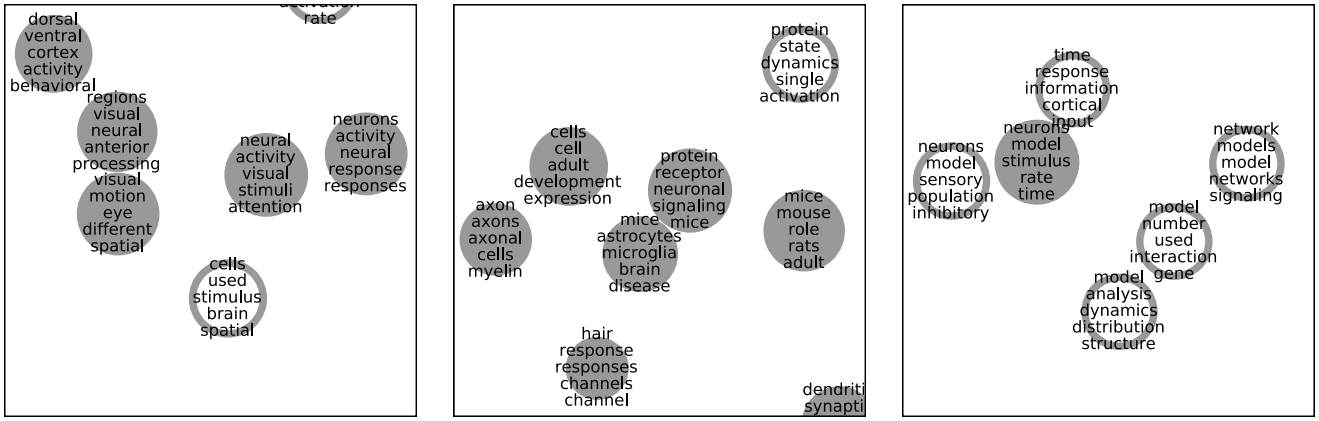
Topic 44 *Pattern Recognition*	
Word	**Prob.**
learning	.0349
neural	.0316
features	.0220
image	.0195
images	.0178
based	.0124
performance	.0113
predict	.0111
training	.0106
Tool	**Prob.**
MATLAB	.7481
Keras	.0668
TensorFlow	.0437
Theano	.0423
Dataset	**Prob.**
mrna	.2412
tcga	.1637
rnaseq	.1162
dnaseseq	.0810

7.2 Cross-domain Knowledge Pattern Application

To discover cross-domain knowledge patterns, we apply the MMD-t-SNE algorithm described in Section 4.3.2 to train a neuroscience model and a bioinformatics model with relevant dataset collections mentioned in Table 2, respectively. We embed the topics in a 2D space to generate a low dimensional topics representation vector $\hat{\boldsymbol{\phi}}$. Subsequently, we can easily plot this information into a 2D space and observe the cross-domain knowledge patterns. Due to the plot size limitations, we note that we had to crop some exemplars of cross-domain patterns from the original embedding plot for demonstration.

As shown in Fig. 6, we list three cross-domain patterns extracted from the original plot. In each sub-figure, the node denotes a topic learned by our model that is either from the neuroscience domain ("shaded" node) or from the bioinformatics domain ("unshaded" node). Fig. 6(a) illustrates that researchers from both neuroscience and bioinformatics domains aim to understand the relationship between vision and brain. We can see from the plot that these types of research efforts are very mature in the neuroscience domain but rare in the bioinformatics domain. Neuroscience researchers try to understand this problem from different aspects (e.g., responses, attention, motion, spatial, and different cell regions) compared with the single aspect perspective in the bioinformatics domain. Hence, bioinformatics researchers can learn notably from the neuroscience domain about synergistic research ideas, tools, and datasets.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3093350, IEEE Transactions on Knowledge and Data Engineering



(a) cross-domain topic for understanding the relationship between vision and brain

(b) cross-domain topic for understanding the role of protein in brain

(c) cross-domain topic for neuron modeling or neuron simulation

Fig. 6: Exemplars of cross-domain knowledge patterns recognized by our model. Each node denotes a topic learned by our model that is either from neuroscience domain ("shaded" node) or bioinformatics domain ("unshaded" node); and each topic is annotated by the top 5 most likely words.

TABLE 10. Description of the exemplars of cross-domain knowledge patterns shown in Fig. 6

Figure No.	Description	Tools For Sharing		Datasets For Sharing	
		Neuroscience	Bioinformatics	Neuroscience	Bioinformatics
Fig. 6(a)	This figure presents the cross-domain topics about understanding the relationship between vision and brain, which are a mature area in neuroscience, but a rare area in bioinformatics	MATLAB, Fieldtrip, EEGLAB, Talairach, FreeSurfer	MATLAB	horizontal, vertical, retina, modulated, excitatory, inhibitory, circuit	mrna, rnaseq
Fig. 6(b)	This figure shows the cross-domain topics about discovering the effects of protein structures on brain stimulation. Neuroscience researchers attempt to understand this problem from different approaches (such as signaling, gene expression, cell regions or types). Bioinformatics researchers also try to understand this problem from the signal activation aspect.	graphpad, MATLAB	MATLAB, Gromacs	dendritic, axon, tyrosine, gabaergic, microglia, astrocytes	mrna, mirna
Fig. 6(c)	This figure illustrates the cross-domain topics the cross-domain topics about neuron stimulation	Helmholtz, klustakwik	MATLAB, Taverna, DARIO, glasso, Mikado	vertical, excitatory, horizontal	mirna, mtdna, metabolomics, rnaseq, tcga, ENCODE

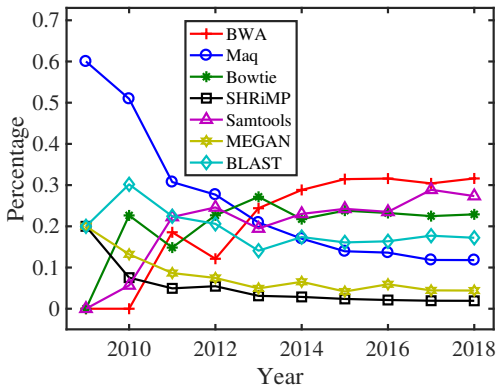
We also observe from Fig. 6(b) that there is an overlapping research topic in two domains for understanding the effects of protein structure on brain stimulation. Neuroscience researchers attempt to understand this problem using approaches such as signaling, gene expression, cell regions or types. Bioinformatics researchers also try to understand this problem from the aspect of signal activation. Hence, cross-domain knowledge recommendation can be fostered by mutually sharing resources and findings across the two synergistic domains. Specifically, cross-domain knowledge sharing can notably improve the understanding of the neuron simulation shown in Fig. 6(c).

7.3 Trend Knowledge Pattern Application

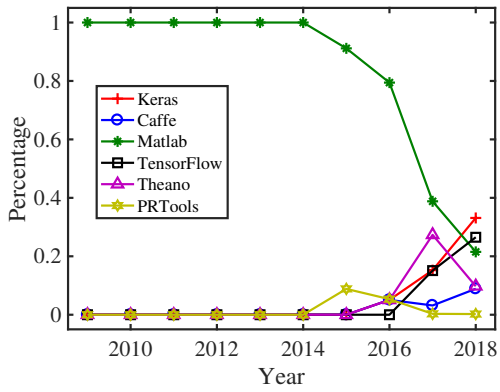
We apply Algorithm 2 to obtain trend the knowledge pattern for each topic. Fig. 7 illustrates two examples of trend knowledge patterns in the bioinformatics domain. In Fig. 7(a), we can clearly see that the tools trend pattern for Topic 4 (*"Next-Generation Sequencing Analysis"*): the Maq [52] used to be the most popular tool for sequence analysis. However, in recent years, tools such as BWA [53], Samtools [54], and Bowtie [55] have become increasingly popular. Fig. 7(b) clearly captures the history of deep learning research in bioinformatics for pattern recognition. As we can see from the plot, MATLAB almost dominated this area before 2015, for which SVM toolboxes in MATLAB were commonly used. From 2015 to 2016, scientists started to use deep learning methods for pattern recognition, for which tools such as Keras, Caffe [56], Theano were used for deep learning research. In 2016, we can see observe that TensorFlow started to be used for deep learning, which can be also be validated by the fact that Google released the first TensorFlow version on November 9, 2015. After 2017, Theano has become less popular, because MILA stopped supporting Theano. Consequently, TensorFlow and Keras have emerged as the most popular tools in bioinformatics for deep learning; and MATLAB lost its dominating position in pattern recognition.

Fig. 8 illustrates two exemplars of trends in knowledge patterns in neuroscience. In Fig. 8(a), we can observe trends

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3093350, IEEE Transactions on Knowledge and Data Engineering

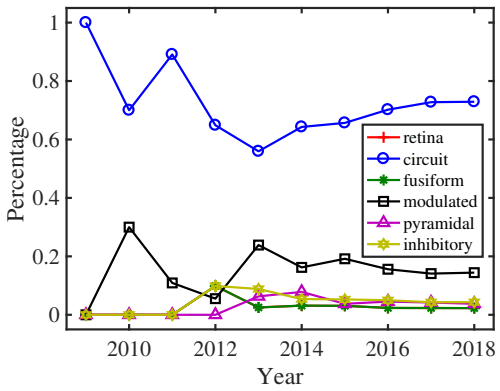


(a) Tools trend pattern on Topic 4 (*Next-Generation Sequencing Analysis*)

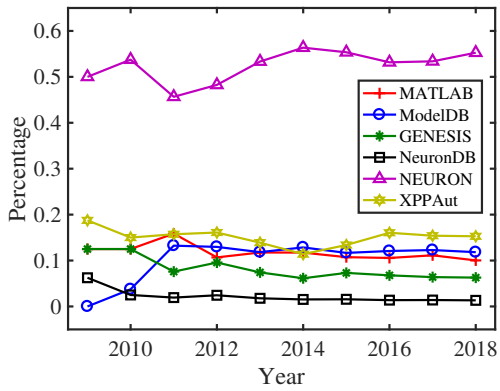


(b) Tools trend pattern on Topic 44 (*Pattern Recognition*)

Fig. 7: Exemplars of trend knowledge patterns captured by DSTM for the bioinformatics domain.



(a) Datasets trend pattern on Topic 18 (*Circuit Analysis*)



(b) Tools trend pattern on Topic 38 (*Neuron Model*)

Fig. 8: Exemplars of trend knowledge patterns captured by DSTM for the neuroscience domain.

as captured by our model in the reports of usage of datasets related to circuit analysis research i.e., the circuit is the most popular in usage of datasets in the last ten years. Other datasets such as those related to modulation, inhibition, and pyramidal cells are also used by researchers with a slightly increasing trend that has now saturated in recent times. The second neuroscience example shows the trend patterns in the usage of neuron models in Fig. 8(b). Among the numerous neuronal modeling packages, NEURON seems to be the most popular in the last ten years. Other popular packages are XPPAut and GENESIS as captured by our model. The same plot also shows that the model database repositories for code using these tools are the ModelDB and NeuronDB.

Another major benefit in using trend knowledge patterns can be seen in cases where we can combine it with the intra-domain knowledge pattern discussed in Section 7.1.2 pertaining to the results in Table 9. We can see that MATLAB is the traditional tool that is highly used in this area over the last ten years as captured by our model. Our model also captures trends in the use of new deep learning tools such as Keras, TensorFlow that are increasing at a dramatic pace in recent years, but with lower probability in the past years comparison with MATLAB. This intra-domain knowledge pattern result might mislead researchers to select MATLAB for pattern recognition research. However, by combining the trend knowledge pattern with the intra-domain knowledge pattern, researchers can have better guidance to select the latest trending tools to suit their research problems.

8 Conclusion and Future Work

In this paper, we presented a novel "domain-specific topic model" (DSTM) for discovering latent knowledge patterns among research topics, tools, and datasets for computational and data-intensive scientific communities. Our DSTM is a generative model whose design incorporates as little, or any amount of domain knowledge while exploring highly-specific topic patterns within a given domain. DSTM can shorten the time to knowledge discovery and help scientists/researchers to adapt prior domain knowledge when pursuing multi-disciplinary investigations. Similar to the popular LDA method in prior works, DSTM uses a completely randomly generative process to generate words based on reference tools or datasets. Using large collections of two different text corpus from neuroscience and bioinformatics domains (includes more than 25,000 papers over the ten years from reputed journal archives), our evaluation experiments with quantitative perplexity scores

and qualitative domain expert feedback show that our DSTM has better generalization performance for revealing highly specific latent topics within a domain. We have also shown that the information retrieval performance results for DSTM outperform other state-of-the-art methods such as LDA and PLSA in retrieving domain-specific topics. We proposed three exemplar applications of DSTM with concrete algorithms for *Intra-domain*, *Cross-domain*, and *Trend* knowledge patterns discovery from large datasets obtained from scientific publication archives. We showed how DSTM can be relevant for researchers seeking to make intelligent decisions using knowledge discovery for developing solutions to multi-disciplinary research problems using state-of-the-art best practices for tools and datasets.

Possible future directions for this work include building visualization/drill-down interfaces to browse the knowledge patterns among research topics, tools and datasets. Such interfaces can further foster an efficient query to obtain appropriate resources (e.g., tools, and datasets) for cutting-edge research investigations in many other disciplines (e.g., material science, business analytics). Our DSTM extensions can be developed and integrated within a recommendation system with an online learning feature to recommend appropriately distributed computing resource configurations to domain scientists based on their real-time workflow requirements. One can also compare our inference algorithm and other inference algorithms such as variational EM to improve the inference performance in terms of computational issues. Further, DSTM can be extended to be used within conversational agents (i.e., chatbots) to help users of science gateways to discover latent knowledge patterns among research topics, tools, and datasets in an interactive manner [36]. Moreover, future work could explore deep learning methods such as e.g., doc2vec and BERT for documents vectorization and latent topic representation to potentially improve the performance of DSTM. Particularly, deep learning based methods are suitable if one can solve issues of inference and interpretability with neural networks towards building a deep learning based domain-specific topic model for knowledge discovery in scientific communities.

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

[3] T. Hofmann, "Probabilistic latent semantic analysis," *arXiv preprint arXiv:1301.6705*, 2013.

[4] 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.

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

[6] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[7] 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.

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

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Z. Luo, L. Liu, J. Yin, Y. Li, and Z. Wu, "Latent ability model: A generative probabilistic learning framework for workforce analytics," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 5, pp. 923–937, 2018.

[15] W. Zhao, W. Zou, and J. J. Chen, "Topic modeling for cluster analysis of large biological and medical datasets," in *BMC bioinformatics*, vol. 15, no. S11. Springer, 2014, p. S11.

[16] 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.

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

[18] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, "An introduction to mcmc for machine learning," *Machine learning*, vol. 50, no. 1-2, pp. 5–43, 2003.

[19] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[20] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[21] Y. Miao, L. Yu, and P. Blunsom, "Neural variational inference for text processing," in *International conference on machine learning*, 2016, pp. 1727–1736.

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

[23] B. Loni, Y. Shi, M. Larson, and A. Hanjalic, "Cross-domain collaborative filtering with factorization machines," in *European conference on information retrieval*. Springer, 2014, pp. 656–661.

[24] L. Li, X. Jin, and M. Long, "Topic correlation analysis for cross-domain text classification," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[25] H. Gao, S. Tang, Y. Zhang, D. Jiang, F. Wu, and Y. Zhuang, "Supervised cross-collection topic modeling," in *Proceedings of the 20th ACM international conference on Multimedia*, 2012, pp. 957–960.

[26] 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.

[27] 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.

[28] J. Sleeman, M. Halem, T. Finin, and M. Cane, "Discovering scientific influence using cross-domain dynamic topic modeling," in *IEEE International Conference on Big Data*, Dec 2017, pp. 1325–1332.

[29] W. Zhao, Z. Guan, L. Chen, X. He, D. Cai, B. Wang, and Q. Wang, "Weakly-supervised deep embedding for product review sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 1, pp. 185–197, Jan 2018.

[30] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 2, pp. 357–370, Feb 2019.

[31] Y. Zheng, "Methodologies for cross-domain data fusion: An overview," *IEEE transactions on big data*, vol. 1, no. 1, pp. 16–34, 2015.

[32] W. Min, B.-K. Bao, C. Xu, and M. S. Hossain, "Cross-platform multi-modal topic modeling for personalized inter-platform recommendation," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1787–1801, 2015.

[33] M. Grbovic and H. Cheng, "Real-time personalization using embeddings for search ranking at airbnb," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 311–320.

[34] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[35] P. Lee, J. D. West, and B. Howe, "Viziometrics: Analyzing visual information in the scientific literature," *IEEE Transactions on Big Data*, vol. 4, no. 1, pp. 117–129, March 2018.

[36] Y. Zhang, P. Calyam, T. Joshi, S. Nair, and D. Xu, "Domain-specific topic model for knowledge discovery through conversational agents in data intensive scientific communities," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 4886–4895.

[37] A. Tiroshi, S. Berkovsky, M. A. Kaafar, T. Chen, and T. Kuflik, "Cross social networks interests predictions based ongraph features," in *Proceedings of the 7th ACM Conference on Recommender Systems*. ACM, 2013, pp. 319–322.

[38] Y. Shi, M. Larson, and A. Hanjalic, "Tags as bridges between domains: Improving recommendation with tag-induced cross-domain collaborative filtering," in *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 2011, pp. 305–316.

[39] W. Pan, E. W. Xiang, and Q. Yang, "Transfer learning in collaborative filtering with uncertain ratings," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[40] C. Sievert and K. Shirley, "Ldavis: A method for visualizing and interpreting topics," in *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, 2014, pp. 63–70.

[41] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. Mar, pp. 723–773, 2012.

[42] 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.

[43] 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.

[44] F. Feng, D. B. Headley, A. Amir, V. Kanta, Z. Chen, D. Paré, and S. S. Nair, "Gamma oscillations in the basolateral amygdala: biophysical mechanisms and computational consequences," *Eneuro*, vol. 6, no. 1, 2019.

[45] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 1999, pp. 50–57.

[46] M. A. Newton and A. E. Raftery, "Approximate bayesian inference with the weighted likelihood bootstrap," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 56, no. 1, pp. 3–26, 1994.

[47] M. Röder, A. Both, and A. Hinneburg, "Exploring the space of topic coherence measures," in *Proceedings of the eighth ACM international conference on Web search and data mining*, 2015, pp. 399–408.

[48] K. S. Jones, S. Walker, and S. E. Robertson, "A probabilistic model of information retrieval: development and comparative experiments," *Information processing & management*, vol. 36, no. 6, pp. 809–840, 2000.

[49] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[50] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[51] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[52] H. Li, J. Ruan, and R. Durbin, "Maq: Mapping and assembly with qualities," *Version 0.6*, vol. 3, 2008.

[53] H. Li and R. Durbin, "Fast and accurate short read alignment with burrows–wheeler transform," *bioinformatics*, vol. 25, no. 14, pp. 1754–1760, 2009.

[54] A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin *et al.*, "The sequence alignment/map (sam) format and samtools," *Bioinformatics*, vol. 25, pp. 2078–2079, 2009.

[55] B. Langmead, C. Trapnell, M. Pop, and S. L. Salzberg, "Ultrafast and memory-efficient alignment of short dna sequences to the human genome," *Genome Biology*, vol. 10, no. 3, p. R25, Mar 2009.

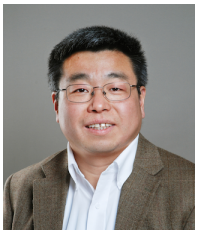
[56] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

Yuanxun Zhang received his BE degree from Southwest Jiaotong University, China, in 2006. He is currently pursuing his PhD degree in the Department of Electrical Engineering and Computer Science at the University of Missouri-Columbia. He is interested in the theory and practice of understanding and modeling complex big data for making better decisions, for solving these problems involving information retrieval, recommendation system, and machine learning.

Prasad Calyam received his MS and PhD degrees from the Department of Electrical and Computer Engineering at The Ohio State University in 2002 and 2007, respectively. He is currently an Associate Professor in the Department of Electrical Engineering and Computer Science at the University of Missouri-Columbia. His current research interests include distributed and cloud computing, computer networking, and cyber security. He is a Senior Member of IEEE.

Trupti Joshi received her PhD from the University of Missouri-Columbia in 2013. She received her MS from University of Tennessee-Knoxville in 2003. She is currently a Director, Translational Bioinformatics and Assitant Professor in Health Management and Informatics at the University of Missouri-Columbia. Her current research interests include bioinformatics, computational systems biology and genomics.

Satish Nair obtained his BS degree from Indian Institute of Technology, Kanpur in 1983, and his MS and PhD degrees from the Ohio State University in 1984 and 1988, respectively, all in mechanical engineering. He is presently Professor in the Department of Electrical Engineering and Computer Science at the University of Missouri-Columbia. His research focuses on dynamical foundations of neuroscience, computational neuroscience and nonlinear/adaptive systems. He is a Fellow of ASME.



Dong Xu received his PhD from University of Illinois at Urbana-Champaign in 1995. He received his MS and BS from Peking University. He is currently the James C. Dowell Professor in the Department of Electrical Engineering and Computer Science at the University of Missouri-Columbia. His current research interests include bioinformatics, protein structure prediction and computational systems biology. He is a Fellow of AAAS.